

WEB TASARIM REHBERİ

KODLAB
www.kodlab.com

WEB TASARIM REHBERİ

MUSTAFA
AYDEMİR

KODLAB
Yayın Dağıtım Yazılım ve Eğitim
Hizmetleri San. ve Tic. Ltd. Şti.

KODLAB 13

WEB TASARIM REHBERİ

MUSTAFA AYDEMİR

ISBN 978-605-4205-13-4
Yayıncılık Sertifika No: 13206

1. Baskı: Ekim 2009

Yayın Yönetmeni: Suat Özdemirci
Kapak Tasarım: Nebi Yıkaroğlu
Sayfa Düzeni: Fikret Eldem
Satış: Hüseyin Üstünel
Baskı: Şefik Matbaası, Tel: (212) 671 59 81

Bu kitabın bütün yayın hakları Kodlab Yayın Dağıtım Yazılım ve Eğitim Hizmetleri San. ve Tic. Ltd. Şti.'ne aittir. Yayınevimizin yazılı izni olmaksızın kısmen veya tamamen alıntı yapılamaz, kopya çekilemez, çoğaltılamaz ve yayınlanamaz.

KODLAB Yayın Dağıtım Yazılım ve Eğitim Hizmetleri San ve Tic Ltd.Şti
Yerebatan Caddesi Salkım Söğüt Sokak Keskinler İş Hanı No: 8
D: 401 Sultanahmet / İSTANBUL

tel: (212) 514 55 66 | **fax:** (212) 514 66 61
e-posta: bilgi@kodlab.com | **web:** www.kodlab.com

Mustafa Aydemir

1971 İstanbul doğumludur. İktisat fakültesi mezunu olan yazar, halen bir kamu kurumda görev yapmaktadır. İçerik yönetim sistemleri ile başladığı web tasarım çalışmalarını, zamanla edindiği tecrübeler ışığında, hazırladığı projelerle sürdürmektedir. Öğrenmeyi ve öğretmeyi çok seven Mustafa AYDEMİR'in, Mehmet Yusuf (7) ve Ahmet Salih (1) adında iki oğlu ve artık bir kitabı var. Yazara görüşlerinizi ve sorularınızı *www.webtasarimrehberi.net* adresinden iletebilirsiniz.

TEŞEKKÜR

Varlık sebebim olan sevgili anne ve babama, evliliğimiz boyunca verdiği inanılmaz desteği, bu kitabın yazılma aşamasında daha da artırarak sürdüren sevgili eşime, yürek dolusu sevgiler ve teşekkürler. İyi ki varsınız.

Ayrıca bu kitabın yazımında bana inanan, güvenen ve yazım aşamasında da desteğini esirgemeyen değerli yayıncım Suat ÖZDEMİRCİ'ye ve kitaba emeği geçen herkese de ayrı ayrı teşekkürler.

ÖNSÖZ

Değerli okurum; bu kalınlıkta bir kitabı satın alma cesaretini gösterdiğiniz için öncelikle sizi tebrik ederim. Bu hareketiniz, bu işi ne kadar ciddiye aldığınızın bir kanıtıdır. Ayrıca kitabın hacmi sizi etkilemesin. Okudukça okuyasınız gelecek, bundan şüpheniz olmasın. Şaka bir yana, bizim mesleğimiz gerçekten keyiflidir. Bu keyfi duyumsayabilen ve bu keyfin verdiği coşku ile çalışıp üretmekten zevk alabilen herkesin yapabileceği bir iştir.

İnternet, insanlara birbirleriyle iletişim kurabilme imkânı sağlamak için var. Tabi bir de ihtiyaç duydukları bilgilere ulaşabilmelerini temin için. Peki günümüzde teknoloji bize –web siteleri sayesinde– sadece metin olarak değil, görsel zenginliği ve derinliği muazzam olan aktarımlar yapma imkanı veriyor iken, tüm bu metin, ses, video... vb iletişim öğelerini insanlara etkin bir yapı oluşturarak kim sunacak? Tabii ki web tasarımcılar ve programcılar. Yani siz!

Elinizde tutuğunuz bu kitap, hem kişisel hem de mesleki çalışmalarınıza katkı sağlamak amacıyla yazıldı ve aşağıdaki konuları içermekte...

KİTAPTA NELER VAR?

Kitapta yer alan konuları, öğrenmeyi kolaylaştıracak bir sıralama ve düzenleme ile vermeye çalıştık. Ayrıca kitabın hacmini gereğinden fazla büyütmemeye ve sıkıcı tekrarlardan kaçınmaya azami gayret gösterdik. Şimdi birlikte kitabın muhtevasına kısaca bir göz atalım.

INTERNETE GİRİŞ

Kitabımızın bu ilk bölümde: **İnternetin Tanımı**, **İnternetin Tarihçesi**, **İnternetin Altyapısı** ve **Tarayıcılar** (*Browsers*) konularına değinerek, hem kitaba hem de web tasarım&programlama maceramıza başlangıç yapmayı amaçladık.

XHTML

HTML (**H***ypertext* **M***arkup* **L***anguage*) worl wide web'in (*www*) temel işaretleme dilidir. Web sayfalarını tarayıcılarda görüntüleyebilmek için kullanılan dil olması sebebiyle, bizlerin de ilk öğrenmemiz gereken dildir. Bu amaçla kitabımıza, tasarımcılar ve programcılar için işin temeli kabul edilen HTML ile başladık.

XHTML belge yapısı, XHTML belge şablonu, metin biçimlendirme, listeler, köprüler –web bağlantıları, resimlerle çalışma, tablolar, formlar ve çoklu ortam (Ses&Video)– çalışmaları, bölümün temel konu başlıklarını oluşturmakta...

CSS

CSS (**C***ascading* **S***tyle* **S***heets-Basamaklı Stil Sayfaları*), web sayfalarını biçimlendirmemizi sağlayan en harika teknolojidir.

Bu harika teknolojinin büyülü dünyasında gezinirken değineceğimiz temel başlıklardan bazılarını ise; seçiciler, artalan (background) biçimlendirme, Yazıtipi (font) ve Metin (Text) biçimlendirme, kutu modeli (Box Model), tablo (table) özellikleri, boyut (dimension) özellikleri, sınıflandırma (classification) ve konumlama (positioning) özellikleri şeklinde sıralayabiliriz.

XML

XML (*Extensible Markup Language*), kullanıcılar tarafından yapılandırılan ve iletilecek bilginin tanımlanmasına yarayan platform bağımsız bir işaretleme dilidir.

Mutlaka bilgi sahibi olmanız gereken bu dil ile ilgili, kitabımızda değindiğimiz bazı konu başlıklarını da; XML belge yapısı, iyi oluşumlu XML belgeleri (well-formed), geçerli XML belgesi (valid document), belge tipi tanımlaması (document type definition-DTD), öznitelikler, varlıklar, XML şemaları (XML schema), XSL kullanımı olarak sıralayabiliriz.

DREAMWEAVER CS4

Dreamweaver CS4, öğrenilmesi kolay ve kullanması çok keyifli olan bir program. Bu bölümü ilgiyle okuduğunuzda kısa sürede programa adapte olacağınızdan şüpheniz olmasın.

İşte konu başlıklarından bazıları: Yeni bir site oluşturmak, image map (resim haritası) oluşturmak, rollover image eklemek, tablolar, formlar, spry kullanımı, dreamweaver ile ziyaretçi defteri, dreamweaver ile üyelik sistemi.

PHP

Bildiğiniz gibi günümüz web sitelerinde; ziyaretçi defteri, abonelik formları, üyelik sistemleri, arama özellikleri, yorum özelliği gibi pek çok özellikler bulunmaktadır. Bu özellikler, site ziyaretçilerinin sadece pasif bir biçimde kendilerine sunulan içeriği okumakla yetinmeyip, çeşitli şekillerde site içeriğine katkıda bulunmalarını ve birbirleriyle de iletişim halinde olmalarını sağlamaktadır. Tüm bunları bir HTML site ile yapamayız. Bahsettiğimiz işlemleri gerçekleştirmek için bir web programlama diline ihtiyaç duyarız.

İşte o programlama dili (PHP) ve konu başlıkları: Değişkenler, PHP'de program denetimi, fonksiyonlar, diziler, formlar, veritabanı işlemleri, oturum yönetimi (session), PHP ve XML.

JAVASCRIPT

JavaScript, web sitenizin görünüm olarak zenginleştirilmesi, dinamik yapısının güçlendirilmesi ve işlevselliğinin artırılması için kullanılan istemci taraflı (client-side script) bir dildir.

JavaScript bölümünün bazı başlıklarının da, JavaScript'in yapısı, kontrol yapıları, döngüler, fonksiyonlar, nesneler, olay yöneticileri (event handlers) olduğunu belirtelim.

FIREWORKS CS4

Fireworks, çok başarılı bir grafik ve web tasarım editörüdür. Web tasarımcılar için sağladığı pratik çözümler ile ön plana çıkan programlardan birisidir.

Ve işte fireworks bölümünün bazı konuları: Bitmap grafikler, vektörel nesneler, renkler, katmanlar, maskeler, dilimler, rollover'lar, animasyonlar, slideshow.

PHOTOSHOP CS4

Photoshop, bir fotoğraf ve grafik düzenleme editörüdür. Görsellerinizi renk, kontrast, ebat... vb birçok açıdan en ince detaylarına kadar işleyebileceğiniz üstün yeteneklere sahip bir programdır. Bir webmaster olarak hazırlayacağınız sitelerde, banner, logo... vb çalışmalarınızda ve yapacağınız daha pek çok görsel uygulamada en önemli yardımcılarınızdan biri olacaktır.

Photoshop bölümünde ise; renkler, temel rötuş işlemleri, fırçalar, filtreler, web sayfaları oluşturma değineceğimiz konulardan sadece birkaçı…

FLASH CS4

Flash programı ile, web siteleriniz için gereken animasyonları, ayrıca farklı türde banner, buton..vb öğeleri kolaylıkla üretebilirsiniz.

Flash bölümünde sizlerle, time line, grafikler, semboller, animasyonlar, 3D Rotation&Translation... vb konuları paylaşacağız.

İÇERİK YÖNETİM SİSTEMLERİ

CMS (**C***ontent* **M***anagament* **S***ystems-İçerik Yönetim Sistemleri*) dinamik içerik yönetim yapılanması sunan web uygulamalarıdır. PHP, ASP.NET... vb web programlama dilleri ile yazılırlar. Sistemlerin yönetimi Admin (yönetim) panelleri vasıtası ile yapılır. Bu sistemlerde içerik oluşturmak için sistemin yapısına müdahale etmek gerekmez. Adeta Word ya da benzeri bir kelime işlem programında yazı yazar gibi kolay ve hızlı içerik üretme imkanı verirler.

Kitabımızın bu bölümünde ise, e-ticaret sistemleri, forum sistemleri, blog sistemleri, portal sistemleri ve joomla konuları yer almaktadır.

Keyifli okumalar ve çalışmalarınızda başarılar...

MUSTAFA AYDEMİR

x

Eğer ben bir genç adama nasihat verecek olsaydım şunu söylerdim:
"Git. Ömrün oldukça zorluklarla ve mesuliyetlerle dolu bir hayat yaşa.
Zira bunlarsız hayat, taşımaya değmez bir yüktür."

Canımın iki parçası,
sevgili oğullarım Mehmet Yusuf ve Ahmet Salih'e...

İÇİNDEKİLER

1 İNTERNETE MERHABA **1**

İnternetin Tarihçesi 2

İnternetin Altyapısı 3

Ağ (Network) 3

TCP/IP Aktarma Denetim Protokolü/İnternet Protokolü (Transmission Control Protocol/Internet Protocol) 4

HTTP Hiper Metin Transferi Protokolü (Hyper Text Transfer Protocol) 4

FTP Dosya Aktarma Protokolü (File Transfer Protocol) 5

SMTP Basit Mesaj Aktarma Protokolü (Simple Message Protocol) 5

Web Sunucuları (Web Servers) 5

Alan Adları (Domain) 5

Tarayıcılar (Browsers) 6

Neden Web Tasarım&Programlama? 7

2 XHTML **9**

XHTML Nedir? 11

Neden XHTML? 11

XHTML Belge Yapısı 12

Metin Biçimlendirme 16

Başlık Etiketleri 16

Kalın ve İtalik Biçimlendirme 18

Listeler 20

Sıralı ve Sıralı Olmayan Listeler Oluşturma 20

İç İçe Listeler 21

Köprüler: Web Bağlantıları 22

Resimler 24

Resim Ekleme 24

Resimlere Link Vermek 25

Resimlere Açıklama Ekleme 25

Tablolar 26

Tablo Oluşturmak 26

Tablo Boyutunu Ayarlamak 27
Sütun Genişliklerini Ayarlamak 28
Hücreleri Birleştirmek 29
Tablolarla Sayfa Tasarımı 31
Bölümleme Yöntemi (div) ile Sayfa Tasarımı 34
Formlar 35
Form Oluşturma 35
Metin Kutuları 36
Butonlar 37
Onay Kutuları 39
Radyo Düğmeleri 39
Açılır Seçenek Kutuları 40
Çoklu Ortam (Ses &Video) 42

3 CSS 45

CSS'i Belgelere Ekleme 48
CSS'in Yapısı 50
Açıklama Satırı 51
Seçiciler 51
Sınıf Seçicisi (Class Selector) 51
Tekil Seçiciler (Id Selector) 53
Seçici Uygulama Hiyerarşisi 54
Çocuk Seçicileri (Direct Child Selectors) 55
Torun Seçicileri (Descendant Selectors) 56
Bitişik Kardeş Seçiciler (Adjacent Sibling Selector) 58
Evrensel Seçiciler (Universal Selector) 58
Öznitelik Seçicileri (Attribute Selectors) 61
Basit Öznitelik Seçicisi 61
Öznitelik Değeri Seçicisi 62
Kısmi Öznitelik Değeri Seçicisi 63
Ayrımlı Öznitelik Değeri Seçicisi 65
Artalan (Background) Biçimlendirme 65
Artalan Rengi (background-color) 65

Artalan Resmi (background-image) 66
Artalan Resim Yenileme (background-repeat) 67
Artalan Resim Sabitleme (background-attachment) 69
Artalan Resmini Konumlandırma (background-position) 69
Artalan (background) 71
Yazıtipi (Font) Biçimlendirme 72
Yazıtipi Ailesi (font-family) 72
Yazıtipi Boyutu (font-size) 73
Yazıtipi Kalınlığı (font-weight) 76
Yazıtipi Stili (font-style) 76
Yazıtipi (font) 77
Metin (Text) Biçimlendirme 78
Renkler (colors) 78
Metin Hizalama (text-alignment) 79
Metin İşaretleme (text-decoration) 81
Metin Girintileme (text-Indentation) 82
Satırlar Arası Yükseklik (line-height) 83
Sözcükler Arası Mesafe (word-spacing) 84
Harfler Arası Mesafe (letter-spacing) 85
Harf Boyutları (Text Transformation) 86
Kutu Modeli (Box Model) 88
Kenarlık Çizgisi (Border) 89
Kenar Çizgisi Stili (border-style) 89
Kenar Çizgisi Genişliği (border-width) 92
Kenar Çizgisi Rengi (border-color) 95
Kenar Çizgisi (border) 96
Dış-Çizgi (outline) Özellikleri 96
Dış-Çizgi stil (outline-style) Özelliği 96
Dış-Çizgi Renk (outline-color) Özelliği 97
Dış-Çizgi Eni (outline-width) Özelliği 98
Dış-Çizgi (outline) Özelliği 100
Border Dış (margin) Alanı 100

İçerik Dış (padding) Alanı 102
Liste (List) Biçimlendirme 103
Liste İmleme Tipi (list-style-type) 103
Liste İmleri Konum (list-style-position) 106
Liste İmleme Resmi (list-style-image) 107
Liste Biçimleme (list-style) 108
Tablo (table) Özellikleri 109
Sınır İndirgeme (border-collapse) Özelliği 109
Sınır Aralığı (border-spacing) Özelliği 110
Başlık Yanı (caption-side) Özelliği 112
Boş Hücreler (empty-cells) Özelliği 114
Tablo Yapısı (table-layout) Özelliği 115
Boyut (dimension) Özellikleri 117
Yükseklik (height) ve Genişlik (width) Özelliği 117
Azami Yükseklik (max-height) ve Azami Genişlik (max-width) 118
Asgari Yükseklik (min-height) ve Asgari Genişlik (min-width) 119
Sınıflandırma (classification) ve Konumlama (positioning) Özellikleri 120
Görüntüleme (display) Özelliği 120
Gezinme/Kayma (float) Özelliği 121
Silme (clear) Özelliği 123
Konum (position) Özelliği 123
Z-dizin (z-index) Özelliği 127
Görsel Sonuçlar 129
Taşma (overflow) Özelliği 129
Kırpma (clip) Özelliği 132
Görünülebilme (visibility) Özelliği 134
İmleç (cursor) Özelliği 135
Sözde Sınıflar (pseudo-classes) 137
Linkli Sözde-sınıflar (Anchor Pseudo-classes) 137
İlk-çocuk (fırst-child) Sözde Sınıfı 138
Dil (lang) Sözde Sınıfı 140
Sözde-Öğeler (pseudo-elements) 142

İlk Satır (first-line) Sözde Öğesi 143
İlk Harf (first-letter) Sözde Öğesi 145
Before Sözde Öğesi 146
After Sözde Öğesi 146
Stillerde Öncelik Hakkı 146
Nokta (.) Özelliği 147

4 XML 149
Neden XML 149
Tarihçesi 150
HTML 151
XML Belge Yapısı 152
XML Belgesinin Bölümleri 154
XML Bildirimi 154
İşlem Yönergeleri 154
Öğeler (Elements) 154
CDATA Bölümleri 155
Yorumlar 156
Öznitelikler (Attribute) 156
İyi Oluşumlu XML Belgeleri (well-formed) 157
Geçerli XML Belgesi (Valid Document) 159
Belge Tipi Tanımlaması (Document Type Definition-DTD) 159
DTD Genel Yapısı 160
Öğe (element) Bildirimi 161
Öznitelikler 165
Öznitelik Türleri 166
CDATA 166
ENTITY-ENTITIES 166
Enumarated (Sıralı) Öznitelikler 167
ID Öznitelikleri 167
IDREF-IDREFS Öznitelikleri 167
NMTOKEN-NMTOKENS Öznitelikleri 168
NOTATION Özniteliği 168

Varsayılan Değerler 169
#REQUIRED 169
#IMPLIED 169
#FIXED 169
Varlıklar 169
XML Şemaları (XML SCHEMA) 171
Basit Tipteki Şemalar 172
Şema Yapısı 172
Basit Tip Tanımlama 173
Sabit Tanımlı Tipler 173
Özel Tip Oluşturma 174
Karmaşık Tipteki Şemalar 176
XML Belgelerini Biçimlendirme (CSS-XSL) 178
CSS ile Biçimlendirme 179
XSL Kullanımı 183

5 DREAMWEAVER 187

Dreamweaver Çalışma Alanı 188
Menüler 191
File/Dosya 191
Edit/Düzen 192
View/Görünüm 192
Insert/Ekle 193
Modify/Değiştir 193
Format/Biçimle 193
Command/Komutlar 193
Sıte/Site 194
Window/Pencere 194
Dreamweaver Çalışma Alanını Özelleştirme 194
Dreamweaver Tercihleri 195
General (Genel Dreamweaver Tercihleri) Kategorisi 195
Code Coloring (Kod Renklendirme) Kategorisi 198
Code Format (Kod Biçimlendirme) Kategorisi 199

Code Hints (Kod Tamamlama) Kategorisi 201
Code Rewriting (Kod Yeniden Düzenleme) Kategorisi 202
Copy/Paste (Kopyala/Yapıştır) Kategorisi 203
Fıle types/Editors Kategorisi 204
New Document (Yeni Belge) Kategorisi 205
Site Kategorisi 206
Yeni Bir Site Oluşturmak 207
Yeni Bir Sayfa Oluşturmak 212
Varlıklar 214
Kitaplık 215
Belge Oluşturmak 215
Madde İmleri Eklemek (Listeler) 224
Önizleme 225
Link (Bağlantı) Eklemek 226
Dahili Link Eklemek 226
E-posta Linki Eklemek 227
Harici Sayfalara Bağlantı Vermek (Hyperlink) 227
Resim Eklemek 228
Sayfaya Arkaplan Eklemek 229
Image Map (Resim Haritası) Oluşturmak 229
Rollover Image Eklemek 230
Tablolar 231
Tablo İçeriklerini Sıralamak 234
Çoklu Ortam Dosyaları 235
Formlar 236
Şablonlar 237
Spry Kullanımı 241
Efektler 242
Menüler 243
Dreamweaver ile Ziyaretçi Defteri 245
Test Sunucusu Oluşturma 245
MySQL Bağlantısı 247

Formlar ile Veritabanına Veri Girişi 250
Veritabanındaki Bilgileri Yazdırma 253
Dreamweaver ile Üyelik Sistemi 257
Üyelik Formu 258
Üyelik Kontrolü 259
Üye Giriş (Login) 260
Çıkış (Logout) 261

6 PHP 263
PHP'nin Yapısı 268
PHP Nasıl Çalışır? 268
Temel Sözdizimi 269
Değişkenler 270
Veri Tipleri 271
Değişken Tipleri 272
Tip Değiştirmek 273
Değişken Değerlerinden Boşluk Kaldırma 273
Süper Global Değişkenler 274
Sabitler 275
Operatörler 276
String Operatör 276
Atama Operatörleri 276
Karşılaştırma Operatörleri 277
Mantıksal Operatörler 277
Aritmetik Operatörler 277
PHP'de Program Denetimi 279
Kontrol Yapıları 280
if Deyimi 280
Switch Deyimi 281
Döngüler 282
While Döngüsü 282
While Döngüsü 283
For Döngüsü 283

continue 284
Fonksiyonlar 285
Fonksiyon Tanımlama 285
Geriye Değer Döndüren Fonksiyonlar 286
Global Değişkenli Fonksiyonlar 288
Statik Değişkenli Fonksiyonlar 288
mail Fonksiyonu 290
Tarih-Zaman Fonksiyonları 292
getdate() Fonksiyonu 292
date() Fonksiyonu 293
time() Fonksiyonu 295
Diziler 295
Foreach() Fonksiyonu 297
sort()-rsort() Dizileri Sıralama 298
asort()-ksort() İlişkili Dizileri Sıralama 299
array_push() Dizilere Değişken Ekleme 300
array_merge() Dizileri Birleştirme 300
array_slice Dizinin Bir Kısmını Alma 301
array_sum() Toplama 301
Formlar 302
İçerik Kontrolü 304
If ( ) Komutu 305
Isset ( ) Komutu 305
Empty ( ) Komutu 305
Strlen( ) Komutu 306
Ereg( )-eregi( ) Komutu 307
ereg_replace()-eregi_replace() Komutu 307
split() Komutu 308
substr() Komutu 309
chr() Komutu 309
ord() Komutu 310
printf()-sprintf() Komutu 310

Düzenli İfadeler Regular Expressions) 311
preg_match() 313
Dosya ve Dizin İşlemleri 313
include-require Komutu 314
file_exists 315
is_readable 315
is_writable 316
filesize 316
touch 316
Unlink 317
fopen() 317
fclose() 318
fwrite() 318
fgets()-feof() 318
fread() 319
fgetc() 319
flock() 320
opendir()-readdir() 320
mkdir()-rmdir() 321
Yönlendirme (Location ve Refresh) 321
Dosya ve Dizin İzinleri (CHMOD) 322
Veritabanı İşlemleri 324
mysql_connect() 325
mysql_select_db() 325
phpMyAdmin ile Yedek Alma ve Yedeği Geri Yükleme 336
Oturum Yönetimi (Session) 338
Oturumu Açmak 339
PHP ve XML 342

7 JAVASCRIPT 345

JavaScript'in Yapısı 346
Belirleyiciler (Identifier) 348
Anahtar Kelimeler (Keyword) 349

Değerler (Literals) 349
Tamsayı Değerler (Integer Lıterals) 349
Kesirli Değerler (Floating-Point Literals) 349
Mantıksal Literaller (Boolean Literals) 349
Karakter Dizisi (String Literals) 350
Değişkenler (Variable) 350
Dizi-Değişkenler (Array) 351
Uygulama: Bilgi Giriş Penceresi 353
Kontrol Yapıları 354
Operatörler 354
Atama (Assignment) Operatörü 354
Arttırma ve Azaltma Operatörleri 355
Mantıksal Operatörler 356
String Operatörleri 356
Karşılaştırma Operatörleri 356
ıf (eğer) Deyimi 357
ıf ... Else (eğer ... diğer) Deyimi 357
ıf else ıf 358
Koşullu Operatör 358
Switch Deyimi 359
Döngüler 360
Whıle Deyimi 360
do...while Deyimi 360
for Deyimi 361
Yuvalanmış Döngü 362
Continue 363
Fonksiyonlar 364
Nesneler 365
JavaScript Hazır Nesneleri 367
Nesnelere Ait Özellikler 368
Document Nesnesine Ait Özellikler 368
Form Nesnesine Ait Özellikler 369

Nesnelere Ait Metotlar 370
Document Nesnesine Ait Metotlar 370
Form Nesnesine Ait Metotlar 370
Date Nesnesine Ait Metotlar 370
Window Nesnesine Ait Metotlar 371
String Nesnesine Ait Metotlar 371
Math Nesnesine Ait Metotlar 372
Olay Yöneticileri (Event Handlers) 372
onLoad-onUnLoad 372
onSubmit Olayı 373
onMouseOver Olayı 374
onClick Olayı 374
onFocus-onBlur 376
onChange Olayı 376

8 FIREWORKS 379
Fireworks 379
Neler Yapılır? 379
Bitmap ve Vektörel Grafikler 380
Fireworks Çalışma Alanı 381
Başlangıç Sayfası 381
Fireworks Arayüzü 382
Tools (Araçlar) Paneli 382
Properties (Özellikler) Paneli 383
Panel Grupları 384
Optimize Paneli 385
History Paneli 386
Align Paneli 386
Pages Paneli 387
States Paneli 387
Layers Paneli 388
Styles Paneli 388
Color Palette Paneli 389

Swatches Paneli 390
Path Paneli 390
Image Editing Paneli 391
Special Characters Paneli 391
Document Library Paneli 392
Common Library 392
Symbol Properties Paneli 393
Tool Bar (Araç Çubuğu) 393
Belgeler ve Belge İşlemleri 393
Yeni Belge Açma ve Kaydetme 393
Dosya İthal Etme 395
Büyütme ve Kaydırma İşlemleri 395
Tuval (Canvas) Değişiklikleri 396
Cetvel, Kılavuz ve Izgara Kullanımı 396
Klavye Kısayolları 398
Nesnelerle Çalışmak 400
Nesneleri Seçme 400
Nesneleri Kesme-Kopyalama 405
Nesneleri Döndürme 406
Nesneleri Ölçeklendirme 407
Nesneleri Gruplama 408
Nesneleri İstifleme 408
Nesneleri Hizalama 409
Bitmap Grafikler 409
Bitmap Nesnesi Oluşturma 409
Bitmap Nesnesi Çizme 409
Boyama 410
Gradient 410
Renk Örnekleme 410
Klonlama 411
Bulanıklık ve Keskinlik Verme 411
Renkleri Katıştırma 411

Aydınlatma ve Karartma 411
Renk Kaplama 412
Kırpma İşlemi 412
Filtreler 412
Levels Özelliği ile Renk Ayarları 412
Curves Özelliği ile Renk Ayarları 413
Brightness (Parlaklık) ve Contrast (Karşıtlık) Ayarları 414
Renk Değiştirme 415
Renkleri Tersine Çevirme 415
Renkleri Bulanıklaştırma 415
Renkleri Keskinleştirme 415
Vektörel Nesneler 415
Vektörel Nesne Çizme 416
Çizgiler Oluşturma 416
Serbest Çizimler 417
Rectangle ve Ellipse Araçları ile Geometrik Şekiller 417
Otomatik Şekiller 418
Metinlerle Çalışmak 422
Text Aracının Özellikleri 422
Özel Karakterler Kullanma 424
Metni Farklı Hatlara Yerleştirme 424
Metinleri Vektörel Yollara İliştirme 424
Renkler 426
Color Palette Paneli 427
Sayfalar 428
Katmanlar 429
Maskeler 431
Dilimler 432
Rollover'lar 433
Resimlerle Toplu İşlemler 435
Açılır Menü 440
Animasyonlar 443

Tween Animasyon 446
Slideshow 447
Optimizasyon 454
Fireworks Tercihleri 456
General Kategorisi 457
Edit Kategorisi 458
Guides And Grids Kategorisi 459
Type Kategorisi 460
Photoshop Import/Open Kategorisi 461
Lunch And Edit Kategorisi 462
Plug-ins Kategorisi 463
Uygulamalar 463
Yazıya Yansıma Efekti Verme 464
Küre Yapımı 465
Türk Bayrağı Yapımı 466

9 PHOTOSHOP 469
Photoshop Çalışma Alanı 470
Photoshop Araçları (Tools) 470
Hazır Ayarlar 471
Görüntüyü Sürükleme ve Büyütme 472
Durum Çubuğu Kullanımı 472
Cetvel, Izgara ve Kılavuz Kullanımı 473
Yapılan İşlemleri Geri Almak 474
Yeni Belge Açma ve Kaydetme 474
Zevkler ve Renkler 476
Renk Seçme 480
Tools Panelinden Renk Seçme 480
Damlalık Aracı ile Renk Seçme 481
Renk Seçim Penceresi 481
Seçim İşlemleri 482
Marquee Araçları ile Seçim 482
Seçimi Taşımak 483

Seçimi Kaydetmek 484
Maskeler 485
Kanallar 486
Temel Rötuş İşlemleri 486
Kırpma İşlemi 489
Perspektif Dönüştürme 489
Döndürme İşlemleri 490
Kırmızı Göz Giderme 491
Renk Değiştirme 491
Lekeleme İşlemi 491
Bulanıklaştırma İşlemi 491
Keskinleştirme İşlemi 492
Soldurma, Yakma ve Doygunluk İşlemleri 492
Katmanlar 492
Arkaplan ve Katman Dönüştürme 493
Yeni Grup-Katman Oluşturma ve Silme 494
Katmanları Çoğaltmak 494
Katmanları Bağlama 495
Katman Stilleri 495
Yeni Bir Katman Stili Oluşturmak 496
Katman Maskeleri 497
Maske Oluşturma 497
Maske Düzenleme 498
Fırçalar 498
Boyama Araç Seçenekleri 499
Desenlerle Boyama 500
Silgiler 500
Eraser Tool 500
Background Eraser Tool 500
Magic Eraser Tool 500
Brushes Paneli 501
Brush Tip Shape 502

Blend Mode 504
Gradient İşlemleri 505
Boya Kovası 508
Desenler 509
Filtreler 509
Filtre Galerisi 510
Artistic Filtre Grubu 510
Blur Filtre Grubu 511
Brush Strokes Filtre Grubu 512
Distort Filtre Grubu 513
Noise Filtre Grubu 513
Pixelate Filtre Grubu 514
Render Filtre Grubu 514
Sharpen Filtre Grubu 515
Sketch Filtre Grubu 515
Stylize Filtre Grubu 515
Texture Filtre Grubu 516
Çizim 516
Çizim Modları 516
Kalem Aracı 518
Metinler 519
Web Sayfaları 524
Sayfaları Dilimleme 524
Görüntüleri En İyileştirme 526
Görüntüyü Kaydetme 527
Photoshop CS4 Automate 528
Automate>Batch 528
Automate>Creat Droplet 533
Uygulamalar 534
Gölge Efekti 534
Resmi Karakaleme Çevirmek 537
Metni Resimle Doldurmak 538

Animasyon Uygulaması 539
Metin Düzenleme 541
Web 2.0 Badge Yapımı 545
Web Site Tasarımı 548

10 FLASH CS4 557
Çalışma Alanı 557
Çalışma Sahnesi 560
Cetvelleri Kullanma 561
Kılavuzları Kullanma 562
Izgara Kullanma 563
Dosya İthal Etme 563
Yayınlama Ayarları 564
Paneller 565
Properties Paneli 565
Library Paneli 566
Actions Paneli 567
Color Paneli 568
Align Paneli 568
Info Paneli 569
Transform Paneli 569
History Paneli 570
Scene Paneli 570
Klavye Kısayolları 571
Tıme Line 571
Grafikler 576
Pencil Tool 576
Rectangle Tool & Oval Tool 577
Free Transform 579
Pen Tool 581
Seçim İşlemleri 582
Semboller 584
Yeni Sembol Oluşturma 584

Şekilleri Sembole Dönüştürme 585
Sembol Özellikleri 586
Semboller ve Time Line 588
Animasyonlar 588
Classic Tween 589
Test ve Ön İzleme 590
Maske Uygulama 590
Parlaklık Uygulaması 590
Şekil Animasyonları 593
Motion Tween 594
Metinler 595
Metinlere Maske Uygulama 596
Ses & Video 597
Ses Dosyaları ile Çalışmak 597
Nesnelere Ses Ekleme 598
Videolar ile Çalışmak 599
CS4 Yenilikler 600
Deco Tool 600
3D Rotation Tool 603
3D Translation Tool 603
Motion Editor Paneli 603
Motion Presets Paneli 605
Spray Brush Tool 607
Bone Tool 607

11 İÇERİK YÖNETİM SİSTEMLERİ 611

CMS Nedir? 611
CMS Türleri ve Örnek Bazı CMS Uygulamaları 612
E-Ticaret Sistemleri 612
Oscommerce 612
Zen-cart 613
Prestashop 614
Forum Sistemleri 614

PhpBB3 615
MyBB 616
SMF 616
Blog Sistemleri 618
b2evolution 618
Nucleus CMS 618
Serendipity 618
WordPress 619
Portal Sistemleri 620
Elxis 620
Drupal 621
Örnek CMS İncelemesi: Joomla 623
Tarihçe 623
Genel Terimler 624
Bileşen (Component) 624
Modül (Module) 624
Uyumlu Ek (Plug-in) 624
Tema-Şablon (Template) 624
Pozisyon-Yerleşim 624
Cpanelde Veritabanı Oluşturma 625
Uzak Sunucuda (Cpanel) Joomla Kurulumu 630
Joomla Genel Ayarlar 639
Joomla 1.5.x Genel Yapılandırma Ayarları (Site Sekmesi) 640
Site Ayarları 641
Meta Veri Ayarları 641
SEO Ayarları 641
Joomla 1.5.x Genel Yapılandırma Ayarları (Sistem Sekmesi) 642
Sistem Ayarları 643
Üye Ayarları 643
Medya Ayarları 644
Hata Ayıklama Ayarları 644
Önbellek Ayarları 645

Oturum Ayarları 645
Joomla 1.5.x Genel Yapılandırma Ayarları (Sunucu sekmesi) 646
Sunucu Ayarları 646
Yerel Ayarlar 647
FTP Ayarları 647
Veritabanı Ayarları 647
Posta Ayarları 648
Joomla'da İçerik Planlaması (Bölüm Kategori İçerik Mantığı) 648
Bölüm Oluşturma 649
Kategori Oluşturma 650
İçerik Oluşturma 650
Genel İçerik Ayarları 651
İçerik Ekleme 654
Sayfa Bilgileri 654
Resim, Ek Sayfa, Devamını Oku... Tuşları ve İşlevleri 655
Ek Sayfa 656
Devamını Oku... 656
TinyMCE editör 657
Menüler ve Menü Yönetimi 658
Menüler 658
Menü Öğeleri 659
Menü Öğe Parametreleri 662
Bileşen Ayarlamaları 663
Menü Öğe Çeşitleri 664
İçerikler 665
Üye Yönetimi 666
Üye Yönetimi ve Üyelik Ayarlamaları 666
Üye Detayları 667
Ayarlamalar 668
Eklenti ve Tema Yöneticisi 669
Kur/Kaldır İşlemleri 669
Tema Yönetimi 671

Joomla 1.5.x Temalarındaki Ana Dosyalar 673
templateDetails.xml 673
index.php 673
template_thumbnaıil.png 673
css/template.css 673
index.php 673
SONSÖZ **678**

1

İNTERNETE MERHABA

İnternetin Tarihçesi 2
İnternetin Altyapısı 3
Ağ (Network) 3
TCP/IP Aktarma Denetim Protokolü/İnternet Protokolü (Transmission Control Protocol/ Internet Protocol) 4
HTTP Hiper Metin Transferi Protokolü (Hyper Text Transfer Protocol) 4
FTP Dosya Aktarma Protokolü (File Transfer Protocol) 5
SMTP Basit Mesaj Aktarma Protokolü (Simple Message Protocol) 5
Web Sunucuları (Web Servers) 5
Alan Adları (Domain) 5
Tarayıcılar (Browsers) 6
Neden Web Tasarım&Programlama? 7

Merhaba; kitabımızın bu ilk bölümde; **İnternetin Tanımı**, **İnternetin Tarihçesi**, **İnternetin Altyapısı** ve **Tarayıcılar** (*Browsers*) konularına değinerek, hem kitaba hem de web tasarım&programlama maceramıza başlangıç yapmış olacağız. Evet macera başlıyor. Lütfen kemerlerinizi bağlayın.

İnternet; çok sayıda bilgisayarın birbirleriyle, belli kurallar (protokoller) çerçevesinde, elektronik olarak veri aktarımında bulundukları bir iletişim ağıdır (*network*).

Bilgisayar ve internet teknolojilerinde meydana gelen hızlı gelişme trendi sayesinde, elektronik ağlar üzerinden yapılan veri aktarımı her geçen gün daha da çeşitlenmekte ve daha da hız kazanmaktadır. Günümüzde; metin, resim, ses, video... vb. değişik formatlardaki veriler çok daha hızlı, çok daha kolay ve çok daha az maliyetle iletilebilmektedir.

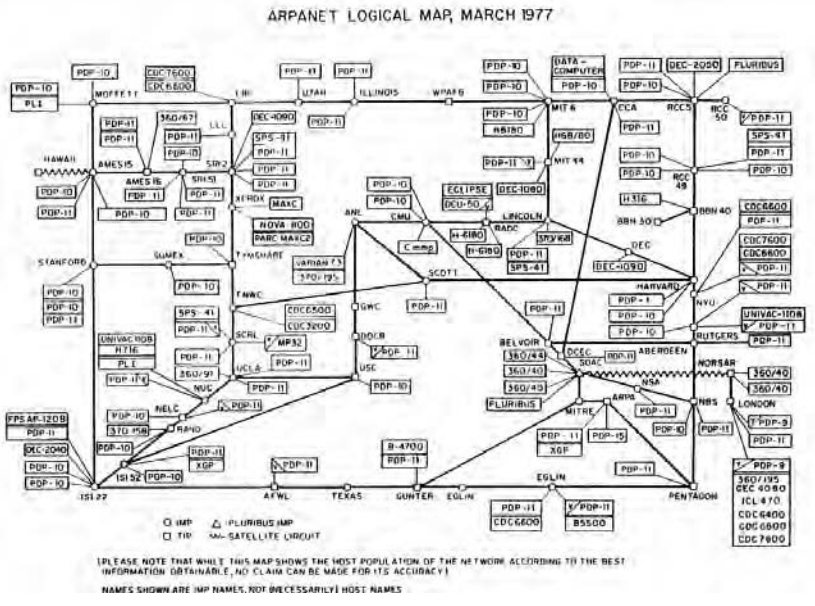
Ülkemizdeki internet kullanımı ile ilgili rakamlar ve rakamların yıllara göre gösterdiği artışlar incelendiğinde (internet kullanan kişi sayısı takriben 20 milyon civarındadır) web tasarım&programlama konusundaki

iş potansiyelinin çok büyük olduğu daha iyi anlaşılacaktır. Ayrıca internetin sunduğu imkanların sadece ülkemizle de sınırlı olmadığını hatırlatmak isterim.

Tüm bu açıklamaların ışığında, önümüzdeki dönemde daha çok yetişmiş web tasarımcıya&programcıya ihtiyaç duyulacağı açıktır. İnancım o ki, elinizde tuttuğunuz bu kitap, bu işe çok sağlam bir başlangıç yapmanıza vesile olacaktır.

İNTERNETİN TARİHÇESİ

Amerikan Genelkurmay Başkanlığına bağlı, Savunma Araştırmaları Proje Dairesi (DARPA) tarafından, askeri ihtiyaçlar için ve üniversitelerle birlikte yapılan çalışmalar neticesinde 1969 yılında **ARPANET** ortaya çıktı. Bu sistem o kadar başarılı oldu ki, kısa bir süre sonra sadece askeri amaçlı olmaktan çıkıp öncelikle üniversiteler, daha sonra da özel şirketler tarafından kullanılır oldu.



ARPANET mantıksal haritası

ARPANET'te ilk kez 1972 senesinde **elektronik posta** (*e-mail*); 1983 tarihinde ise **İletişim Kontrol Protokolu** (*Transmission Control Protokol/ internet protokol-TCP/IP*) kullanılmaya başladı. 1984 yılında **DNS** (**D***omain* **N***ame* **S***ystem*) başlatıldı.

NOT

TIM BERNERS-LEE



İnternetin babası. 1989 yılında **Tim Berners-Lee**, İsviçre'deki Avrupa Nükleer Araştırma Merkezi'nde (*CERN*) başladığı çalışmaları neticesinde **World Wide Web**'in temel unsuru olan **adres sistemini** (*Uniform Resource Locator-URL*), **ağ protokolünü** (*HTTP*), **hiper-metin işaretleme dilini** (*Hyper-Text Markup Language-HTML*) ve ilk web tarayıcısını geliştirdi. 1994'te ise ağla ilgili standartları dünya çapında belirleyecek bir açık kaynak olan **World Wide Web Consortium**'u (*W3C*) kurdu.

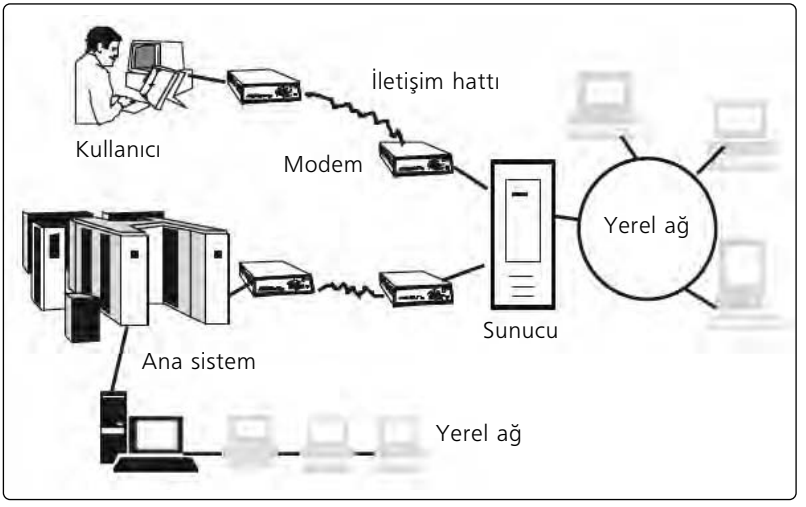
Yıl 1993'ü gösterdiğinde tüm dünyada İnternet kullanıcılarının sayısı `900.000`'i bulmuştu bile. 2000 yılında bu sayı `300` milyona, 2002'de `500` milyona, 2004'de `900` milyona ulaştı. 2009 yılı itibarı ile İnternet kullanıcılarının sayısı ise `1.5` milyar civarındadır.

İNTERNETİN ALTYAPISI

İsterseniz biraz da buraya kadarki açıklamalarımızda geçen terimlerden bahsedelim.

AĞ (NETWORK)

Ağ (*Network*), çok sayıda bilgisayarın birbirleriyle veri alışverişi yapacak şekilde bağlanması ile oluşan yapıdır. Tahmin edeceğiniz gibi en büyük ağ, milyarlarca insanın istifade ettiği, milyonlarca bilgisayara aynı network omurgaları üzerinden iletişim imkanı sunan **World Wide Web** yani **internet**'tir.



İnterneti tarif ederken bilgisayarlar arasındaki iletişimin belli kurallar (protokoller) dahilinde yapıldığını söylemiştik. Şimdi bu protokolleri genel hatlarıyla inceleyelim.

TCP/IP Aktarma Denetim Protokolü/ İnternet Protokolü (Transmission Control Protocol/Internet Protocol)

İnternette bilgisayarlar arası iletişim protokollerinin temelini **İnternet Protokolü** (*Intenet Protocol-IP*) teşkil etmektedir. Temel ağ işlemleri bu ana protokol katmanına tabidir. Kullanıcı uygulamalarını ise **TCP/IP Aktarma Denetim Protokolü** (*Transmission Control Protocol*) sağlar.

HTTP Hiper Metin Transferi Protokolü (Hyper Text Transfer Protocol)

HTTP, internette bilgisayarlar arasındaki veri iletiminin kurallarını düzenleyen sistemin adıdır. Tarayıcınızın adres satırına bağlanmak istediğiniz sitenin adresini yazdığınızda, tarayıcınızın bu ifadeyi otomatik olarak adresin başına eklediğini fark etmişsinizdir. Bunun sebebi, tarayıcınızın, gitmek istediğiniz sitenin barındığı sunucuya, oradan alına-

cak bilgilerin `http` protokolü çerçevesinde iletilmesi (birçok değişik amaçlarla tasarlanmış protokoller olduğundan) talimatını verme zorunluluğudur.

FTP Dosya Aktarma Protokolü (File Transfer Protocol)

FTP, internet ortamında iki bilgisayar arasındaki metin, resim, ses, video, program...vb. her çeşit dosya transferi işlemleri için kullanılır. Eskiden bu iş doğrudan birtakım ftp komutları yazmayı gerektiren zahmetli bir iş iken sonraları **FTP** programları yardımıyla çok kolay bir işlem halini aldı. Günümüzde çok sayıda **ücretli** ve **ücretsiz** (*freeware*) **FTP** yazılımı bulabilirsiniz.

SMTP Basit Mesaj Aktarma Protokolü (Simple Message Protocol)

Bilgisayarlar arasında **e-posta** (*e-mail*) gönderme işlemleri (istemci ile sunucu arasındaki işlemler) ile ilgili teknik kuralları belirleyen protokoldür.

Web Sunucuları (Web Servers)

İnternette kullanıma sokulacak verilerin barındırılması hizmetinde kullanılan bilgisayarlara **sunucu** (*server*) denir. **HTTP** başta olmak üzere **HTTPS**, **FTP**, **SMTP**, **POP3**, **DNS** hizmeti verebilirler.

Alan Adları (Domain)

DNS Alan Adı Sistemi (**D***omain* **N***ame* **S***ystem*), devasa internet ağ sistemini planlayan, sınıflandıran ve bu sınıflandırmaya göre de **ağ** üzerinden yapılan tüm iletişimi organize eden sistemdir.

Bu sisteme göre ağ üzerinde yer alan her bir birim kendine ait eşsiz bir **IP adresine** sahiptir. Rakamsal bir yapıda olan bu sistem (IP numaraları), kullanım kolaylığı sağlamak için **www.siteismi.com** tarzında adreslere dönüştürülür. Bu adreslerin IP numaraları DNS sunucuları tarafından saklanır.

Son olarak da alan adlarına verilen üst seviye uzantıların anlamlarına bakalım:

com Ticari kuruluşlar (**com***mercial*) ve ticari faaliyetler içindir.

net İnternet ve ağ ile ilişkili (**net***work*) faaliyetler içindir.

org Genelde ticari olmayan kuruluşlar ve organizasyonlar (**org***anization*) içindir.

gov Kamu kurumları (**gov***ernment*) için tahsis edilmiştir.

mil Askeri (**mil***itary*) kurumlara aittir.

edu Eğitim kurumları (**edu***cation*), tarafından kullanılırlar.

TARAYICILAR (BROWSERS)

İnternet tarayıcıları (*Web Browsers*), web sunucularındaki metin, resim, ses, video...vb dosyaların **istemci** (*client*) bilgisayarlarda açılması, görüntülenmesi, kaydedilmesi...vb birçok işlemin http protokolleri kullanılarak gerçekleştirilebilmesi için gereken yazılımlardır. İşin tarihine inecek olursak karşımıza, ilk tarayıcı olmamasına rağmen, günümüz tarayıcılarının başlangıcı diyebileceğimiz **mosaic** çıkar. Resim dosyalarının kullanımı da ilk bu tarayıcıyladır.



Günümüzde ise bir çok üst düzey tarayıcı geliştirilmiştir. Tarayıcılar arasında **HTML**, **CSS**, **JavaScript** gibi yazılım teknolojilerine uyumluluk; güvenlik, hız, görünüm ...vb birçok özellik konusunda kıyasıya bir yarış yaşanmaktadır.



Çok sayıda tarayıcı bulunmakla beraber; **Internet Explorer**, **Mozilla Firefox**, **Opera**, **Google Chrome** ve **Safari** en çok öne çıkanlardır diyebiliriz.

Neden Web Tasarım&Programlama?

Efendim, izlediğimiz mantıksal kronoloji neticesinde şu an işin en heyecanlı kısmına gelmiş bulunmaktayız.

Bahsettiğimiz tüm bu teknik altyapı, insanlara birbirleriyle iletişim kurabilme imkanı sağlamak için var. Tabi bir de ihtiyaç duydukları bilgilere ulaşabilmelerini temin için.

Peki günümüzde teknoloji bize –web siteleri sayesinde– sadece metin olarak değil görsel zenginliği ve derinliği muazzam olan aktarımlar

yapma imkanı veriyor iken, tüm bu metin, ses, video... vb iletişim öğelerini insanlara etkin bir konsept oluşturarak kim sunacak? Tabii ki web tasarımcılar ve programcılar. Yani siz!

Ama tabi bunu yapabilmemiz için önce gerekli bilgilere sahip olmamız lazım değil mi? Hadi bakalım, kolları sıvayalım ve hepsini teker teker öğrenmeye başlayalım. Sizi bilmem ama ben şahsen çok heyecanlandım. Harika bir yolculuk başlıyor çünkü... Size kolaylıklar ve başarılar dilerim.

2

XHTML

XHTML Nedir? 11
Neden XHTML? 11
XHTML Belge Yapısı 12
Metin Biçimlendirme 16
Başlık Etiketleri 16
Kalın ve İtalik Biçimlendirme 18
Listeler 20
- Sıralı ve Sıralı Olmayan Listeler Oluşturma 20
- İç İçe Listeler 21

Köprüler: Web Bağlantıları 22
Resimler 24
- Resim Ekleme 24
- Resimlere Link Vermek 25
- Resimlere Açıklama Ekleme 25

Tablolar 26
- Tablo Oluşturmak 26
- Tablo Boyutunu Ayarlamak 27
- Sütun Genişliklerini Ayarlamak 28
- Tablolarla Sayfa Tasarımı 31
- Bölümleme Yöntemi (div) ile Sayfa Tasarımı 34

Formlar 35
- Form Oluşturma 35
- Metin Kutuları 36
- Butonlar 37
- Onay Kutuları 39
- Radyo Düğmeleri 39
- Açılır Seçenek Kutuları 40

Çoklu Ortam (Ses &Video) 42

HTML (**H***yper***T***ext* **M***arkup* **L***anguage*) **worl wide web**'in (*www*) temel işaretleme dilidir. Web sayfalarını tarayıcılarda görüntüleyebilmek için kullanılan dil olması sebebiyle, bizlerin de ilk öğrenmemiz gereken dildir. Bu amaçla kitabımıza, tasarımcılar ve programcılar için işin temeli kabul edilen **HTML** ile başladık.

İlk cümlemizde, **HTML** için bir işaretleme dilidir dedik. Bu açıdan ilk olarak işaretleme dili kavramını tarif etmek gerekirse; işaretleme dili, metinlerin nasıl yapılandırılacağını ve biçimlendirileceğini birtakım işaretlerle (etiket) belirleyen dildir, diyebiliriz.

İlk işaretleme dili **GML** (**G***eneralized* **M***arkup* **L***anguage*) 1960 yılında IBM firması tarafından geliştirilmiştir. 1978 yılında ise **GML** bir grup araştırmacı tarafından daha da geliştirilerek **SGML** (**S***tandardized Generalized* **M***arkup Language*) işaretleme dili oluşturuldu. **SGML** 1986 yılında **ISO** (*the* **I***nternational* **O***rganization for* **S***tandardization*) kurumu tarafından uluslararası bir standart olarak kabul edildi. Çok güçlü bir dil olan **SGML**; otomotiv, basın...vb bir çok endüstri kolunda kullanılmasına karşın, karmaşık bir yapıya sahip olması ve bu dille uygulama geliş-

tirme maliyetinin yüksek olması sebebiyle kullanımı çok da fazla yaygınlaşamamıştır.

HTML dili de, yukarıda bahsettiğimiz **SGML** dilinin alt yapısı kullanılarak 1989 yılında Tim Berners-Lee ve Anders Berlung tarafından geliştirilen bir işaretleme dilidir. Bir başka ifade ile, **HTML** (**H***yper***T***ext* **M***arkup* **L***anguage*), programlama dili değildir. Programlama dillerinin bile kolaylıkla öğrenilebilecek bir yapıya kavuştuğunu belirterek, **HTML** işaretleme dilini çok kolaylıkla öğreneceğinizden şüpheniz olmasın diyorum.

Yeri gelmişken kitabımızın **PHP** bölümünde çok zevkli ve çok yaygın bir programlama dilini öğreneceğimizi hatırlatayım. Programlama dili öğrenmenin zor ve sıkıcı olduğu konusundaki tüm tabuları yıkan bir bölüm sizi bekliyor! Tabi sabırsızlanarak önce o bölümden başlamaya kalkmayın. Yoksa gerçekten zor olabilir. Şaka bir yana önsözde de belirttiğim gibi kitabımızdaki bölümler rasgele dizilmeyip, öğrenmeyi kolaylaştıran bir mantığa göre sıralanmıştır. Dolayısıyla konuları sırayla çalışmanızı öneririm.

Şimdi de gelelim bölüm başlığındaki **X** harfinin hikmetine. Evet neden başlığı **HTML** değil de **XHTML** şeklinde yazdık? Artık tarayıcıların ve çoğu web geliştiricisinin kullandığı işaretleme dili standardının XHTML olması sebebiyle ben de bu bölümü XHTML standartlarını gözeterek yazmaya karar verdim. Aksi takdirde önce HTML'nin yapısına göre konuları anlatmam sonra da birkaç sayfa ötede XHTML bölümünde, artık şu özellikleri şu şekilde kullanamazsınız, özellikle farklı tarayıcılarda sitenizin aynı kalitede aynı şekilde görünmesini istiyorsanız şu kurallara uymalısınız demem gerekirdi. Aynı şeyleri iki kez tekrar edip üstelik de kafa karışıklığına sebep olmak istemedim.

Biraz sonra detaylı olarak açıkladığımda göreceğiniz gibi XHTML bilen zaten HTML de biliyor demektir. Çünkü XHTML tamamen HTML altyapısına birtakım ek özellikler ile bazı standartlar eklenmesinden ibarettir ve zaten HTML'nin yerini almak üzere geliştirmiştir. Biz yapacağımız web sitelerini gelecek için yapacağımıza göre, zaman makinesi denen alet de hala sadece filmlerde olduğuna göre bir ondan bir bundan bahsedip karışıklık oluşturmaya gerek olmadığı düşüncesindeyim.

Bu girizgahtan sonra gelelim XHTML'nin faydalarına...

XHTML Nedir?

XHTML, **Genişleyebilir Köprü Metni İşaretleme Dili** (*Ex*tensible **H***yper* **T***ext* **M***arkup* **L***anguage*) web sayfalarının yapısal ve görsel olarak tarayıcılarda nasıl görüneceğini belirlemek için kullanılan bir işaretleme dilidir ve HTML dilinin yerini almak üzere geliştirmiştir.

XHTML, HTML dilinin 4. sürümünü temel aldığından sadece tüm yeni tarayıcılar ve eski tarayıcıların (örn: *Internet Explorer*) yeni sürümleri tarafından desteklenmektedir. Yani tarayıcınız güncel ise XHTML dilinin tüm nimetlerinden yararlanabilirsiniz.

HTML dilinin aksine, sıkı kurallara sahiptir. HTML'nin çıktığı dönemde bir standardının olmaması ve isteğe göre uyarlama şeklinde gelişen tavırlar –özellikle tarayıcı geliştiricileri– hep birçok soruna yol açmıştı. Ortalıkta standartları belirleyecek bir kurumun olmamasının doğurduğu sorunlar, web standartlarını belirleyen kurum olan **World Wide Web Consortium W3C** adlı organizasyonun kurulması ile çözüme kavuşmaya başladı.

XHTML, 2001 yılında çıkan 1.0 sürümüyle; XML dilinin genişleyebilir (*extensible*) yapısını bünyesinde barındırması, daha standart bir yapıya sahip olması, daha temiz bir HTML yazımı gerektirmesi, bilgisayarlar haricinde mobil telefonlar ve XML uyumlu birçok cihaz tarafından da desteklenmesi...vb sebeplerden dolayı, **World Wide Web Consortium-W3C** tarafından kabul edilen standart oldu.

Neden XHTML?

Yapacağınız web sitesinin, günümüzün gelişmiş özelliklere sahip tarayıcılarında doğru bir şekilde ve hepsinde de aynı şekilde görüntülenmesi için XHTML standartlarına uymanız gerekmektedir. Bunu; XML dilinin, etiket bütünlüğünü sağlayarak, tarayıcılarda en doğru sonucu almanıza imkan veren yapısına borçlu olduğumuzu söyleyebiliriz. (XML dilini detaylı olarak kitabımızın 4. bölümünde inceleyeceğiz.)

Klasik HTML'nin gevşek kuralları ile yazacağınız siteler, farklı tarayıcılarda farklı –bazen çok kötü– görüntülenebilir. Ayrıca tarayıcılar tarafından da daha geç algılanacaktır. Bunun anlamı, bazı tarayıcılarda kö-

tü görünen ve daha geç erişilen bir siteniz olacağıdır. Bu noktada XHTML dilinin kurallarının katı olmasına rağmen öğrenilmesinin kolay olduğunu da ifade etmek isterim. Az sonra bunun böyle olduğunu siz de göreceksiniz.

XHTML Belge Yapısı

Belli bir düzenin olmasını istediğimiz her yerde belli kuralların da olması gerektiği gerçeğini siz de takdir edersiniz. Dünyanın çok farklı bölgelerinden milyonlarca insanın, değişik tip ve modeldeki cihazlarla (bilgisayar, mobil telefonlar...vb) görüntülediği web sayfalarının da belli standartlara oturtulması kaçınılmazdır. (Bu amacla kurulan W3C'den bahsetmiştim.)

İşte bu gerçek, haliyle XHTML dili söz konusu olduğunda da geçerlidir. XHTML dili için oluşturulan standartların belgeye uygulanabilmesi için öncelikle elimizdeki belgenin bir XHTML belgesi olduğunun tarayıcılar tarafından algılanması gerekmektedir. Bu sebeple de biz, oluşturacağımız belgelerin başına belge türü bildirimini –`DOCTYPE`– muhakkak koyacağız.

XHTML belgemizde kullanacağımız belge türü bildirimi:

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Bu bildirimin anlamı; belgemizin bir XHTML belgesi olduğu ve belgemizde **http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd** adresindeki dtd dosyasında belirtilen standartların geçerli olacağıdır. DTD ve kullanımı konusunda kitabımızın XML bölümünde gereken bilgiler mevcuttur.

Üç tip belge türü kullanabiliriz. Birinci tür yukarıdaki örnekte gördüğünüz **Geçişli** (Transitional), diğeri ise **Kesin**'dir. (*Strict*). Geçişli türünde eski HTML standartları da geçerli olmaktadır. Kesin türü ise onaylanmamış etiketleri kabul etmez. Ayrıca XHTML dilinin bir takım üst düzey özelliklerini kullanmak için de bu türü seçmemiz gerekmektedir. **Çerçeve** (*frameset*) türü ise çerçeve kullanımı içindir.

Biz kitabımızda –DVD içindeki örnek uygulamalarda da– XHTML'nin üst düzey, gelecekte yaygınlaşacak özelliklerine değinmeyeceğimiz

için ve daha genel olması sebebiyle **Geçişli** (*Transitional*) türünü kullanacağız.

XHTML Belge Şablonu

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xlmns="http://www.w3org/1999/xhtml">
<head>
</head>
<body>
   TARAYICILAR TARAFINDAN GÖRÜNTÜLENECEK <br /> TÜM BELGE
İÇERİĞİ BURADA YER ALIR.
</body>
</html>
```

Yukarıda gördüğünüz kalıp, XHTML belgelerinin yapısını göstermektedir. En üstte belge türü tanımlaması gelmektedir. Ardından belgenin bir html belgesi olduğunu gösteren `<html>` etiketi (*tag*) gelmektedir. Gördüğünüz gibi belgemiz başladığı gibi yine `<html>` etiketi ile bitmektedir. Tek farkı / işareti ile başlamasıdır. "html" etiketi tüm belgeyi içine alan kapsayıcı etikettir. Ek olarak html etiketine `xlmns="http://www.w3org/1999/xhtml"` şeklinde bir XML ad alanı (*namespace*) ekliyoruz.

XHTML'de sadece html etiketi değil tüm etiketler (`<head> </head>` ; `<body> </body>` etiketlerinde olduğu gibi) "/" işareti ile başlayan kapanış etiketi ile mutlaka kapatılır. Tek taraflı etiketler ise (örn: satırbaşı etiketi olan `br`) bir boşluk bırakılarak ve / işareti eklenerek `<br />` şeklinde yazılır.



Açma ve kapama etiketleri, içerikler ve bu arada yer alan tüm unsurlar **öğe** (*element*) diye adlandırılır. Örn: `<h1 align="center">METİN</h1>` satırı komple bir XHTML öğesidir.

Örneğimizdeki diğer etiket olan `<head></head>` etiketi, belgemizin web tarayıcılarında görüntülenecek sayfa başlığını ve meta bilgilerini ifade eder. Ayrıca bazı tip CSS ve JavaScript dosyalarını çalıştıracak satırlar da burada yer alabilir.

<body> </body> etiketleri belgemizin tarayıcılarda görüntülenecek kısmını ifade eder. Yani tarayıcılarda gösterilmesi gereken her şeyi bu etiketlerin arasına yazacağız.

Bu belge şablonunu DVD içinden –uygulamalar kısmından– alarak **NOTPAD** programına yapıştırıp tüm çalışmalarınızda kullanabilirsiniz. Size, NOTPAD programına yapıştırdığınız belgenizi **ilk_belgem.html** adıyla kaydedip hatıra olarak saklamanızı öneririm. Ne de olsa ilk olan şeyler önemlidir değil mi?

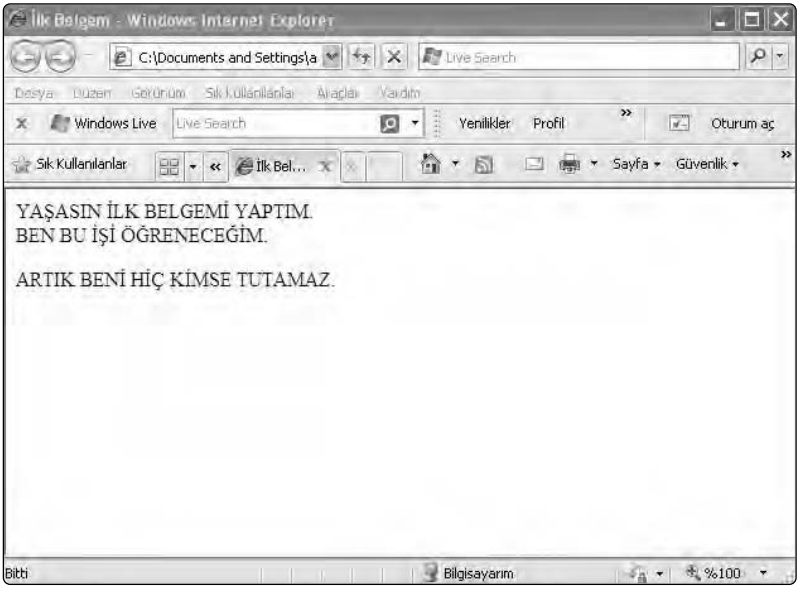
Hatta bu anı daha da unutulmaz yapmak için <body></body> etiketleri arasındaki ifadeyi istediğiniz bir cümle ile değiştirelim.

Mesela:

```
YAŞASIN İLK BELGEMİ YAPTIM. <br /> BEN BU İŞİ ÖĞRENECEĞİM.
<p>ARTIK BENİ HİÇ KİMSE TUTAMAZ. </p>)
```

yazıp belgenizi tekrar kaydedin ve bu ilk HTML belgenizi monitörünüzden doya doya seyredin.

Cümle aralarına da <br /> etiketini koyalım ki cümleler alt alta gelsin. Hatta yeri gelmişken paragraf etiketini de görelim. Paragraf etiketi , <br> etiketinden farklı olarak sadece satırbaşı değil aynı zamanda da yeni bir paragraf başlatır. Bir diğer farkı da şeklinde kullanılan çift taraflı bir etiket olmasıdır.



Bu arada size, çalışmalarınızı daha verimli olarak sürdürebilmeniz için **NOTPAD++** programını öneririm. Programı **DVD**'de **Programlar** bölümünde bulabilirsiniz.

Belgemizi şekillendirmek ve geliştirmek için heyecan duymaktayız (siz ve ben elbette). Ama öncelikle `<head> </head>` etiketleri arasında görmemiz gereken iki husus var.

Bunlardan birincisi tarayıcılarda görüntülenecek sayfa başlığını oluşturma meselesi. Bunun için `<title> </title>` etiketlerini kullanıyoruz. Mesela `<title> İlk Belgem </title>` şeklinde bir örnek yapmış olalım. Bunu belgenize ekleyip tarayıcıdan sayfanıza tekrar bakarsanız, sayfa başlığınızda **`İlk Belgem`** ifadesinin yer aldığını görürsünüz.

`<head> </head>` kısmındaki değinmemiz gereken ikinci husus ise **meta** etiketleridir. Bu etiket sayfanıza, sayfanız ile ilgili ve sayfa içeriğinize uygun anahtar sözcükler eklemenize olanak tanır. Bu anahtar sözcükler arama motorlarında (örn: MSN) arama yapan kişiler tarafından sitenizin daha kolay bulunabilmesi içindir. Bazı arama motorları arama sonuçlarını belirlemede buraya eklenecek anahtar sözcüklere önem verirler.

Genel Yapısı

```
<meta name="keywords" content="kelime1, kelime2, kelime3" />
```

name ve content özniteliklerini açıklayacak olursak; name, meta etiketimizde kullanılacak içeriğin türünün ne olacağını, (örn: keyword, yani anahtar kelime) content ise bizzat içeriklerin kendilerini ifade eder. Etiketimizin tek taraflı bir etiket (br gibi) olduğundan bir boşluk ve / işareti ile kapadığımıza dikkat edin.

Cümlemizde geçen ve bundan sonra da sık sık geçecek olan **Öznitelik** (*Attributes*) kavramına değinelim. **Öznitelikler**, birlikte kullanıldıkları etiketlerin nasıl kullanılacaklarını belirleyen, her biri farklı işlevlere sahip öğelerdir. Örneğin yukarıdaki meta etiketi, anlattığımız anahtar kelime belirleme işlemindeki görevini birlikte kullanıldığı name ve content öznitelikleri sayesinde yapmaktadır.

Bir başka öznitelik kullanarak meta etiketinin farklı bir işlevi (örn: başka bir sayfaya yönlendirme işlemi) ne şekilde yapacağını belirleyebiliriz. Bir başka ifade ile, etiketlerin nasıl davranacaklarını öznitelikler ile belirliyoruz diyebiliriz.

NOT

<meta> etiketi, sayfanızı ziyaret edecek kişileri bir başka sayfaya yönlendirmek için de –mesela başka adrese taşındığınızı ifade etmek için– kullanılabilir.

Örneğin:

```
<meta http-equiv="refresh" content="5";
url=http://siteadresi.com/gidilecek_sayfa.html" />
```

Bu örnek 5 saniye sonra ziyaretçileri verdiğiniz adrese yönlendirecektir.

METİN BİÇİMLENDİRME

BAŞLIK ETİKETLERİ

<h1>'den <h6>'ya kadar büyükten küçüğe doğru giden başlık tipi söz konusudur. Mevcut şablonumuzu kullanarak <body> etiketinin hemen altına cümlemizi ekleyelim ve konuyu deneyelim.

```
<body>
<h1>YAŞASIN</h1> <h2>İLK BELGEMİ YAPTIM</h2>
<br />
<h3>BEN BU İŞİ ÖĞRENECEĞİM.</h3>
 <p><h4>ARTIK BENİ HİÇ KİMSE TUTAMAZ. </h4></p>
</body>
```



Başlık etiketlerini hizalamak istersek `align` özniteliğini kullanabiliriz. Bu öznitelik ile kullanılan değerler: `left` (*sola*), `right` (*sağa*), `center` (*ortalı*), `justify`'dır (*iki yana yaslı*).

Kullanımı

```
<h1 align="center">METİN</h1>
```

şeklindedir.

KALIN VE İTALİK BİÇİMLENDİRME

Kalın yazı için   etiketleri, italik yazı içinse   etiketleri kullanılır. Dilerseniz b etiketi yerine strong etiketini ve i etiketi yerine de em etiketini kullanabilirsiniz.

Aynı örneğimizden devam edelim:

```
<body>
<h1><b> YAŞASIN</b> </h1> <h2><i> İLK BELGEMİ YAPTIM </i> </h2>
<br />
<h3>BEN BU İŞİ ÖĞRENECEĞİM.</h3>
 <p><h4><b><i>  ARTIK BENİ HİÇ KİMSE TUTAMAZ. </i> </b>
</h4></p>
</body>
```

Dikkat ettiyseniz bir konu öğrendiğimizde o konuyla ilgili olarak örnek yaparken bir önceki konuda yaptığımız örnek üzerinden gidiyoruz. Burada amaç, konuları tekrar etmek ve pekiştirmektir. Ayrıca birçok etiketi bir arada nasıl kullanabileceğimizi göstermektir. Dikkat edeceğimiz husus, etiketleri ilk açılan son kapanır kuralına göre kullanmaktır.

Doğru Kullanım

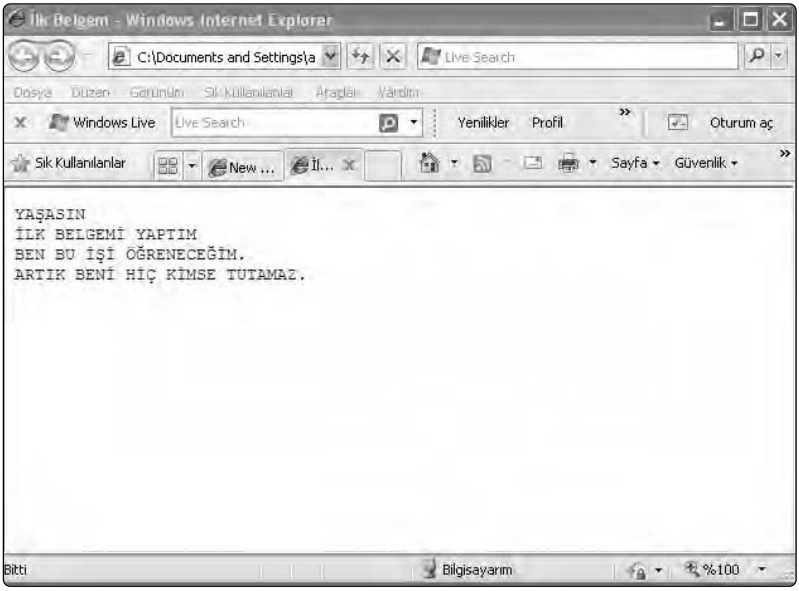
```
<b><i>ARTIK BENİ HİÇ KİMSE TUTAMAZ.</i></b>
```

Yanlış Kullanım

```
<b><i>ARTIK BENİ HİÇ KİMSE TUTAMAZ.</b></i>
```

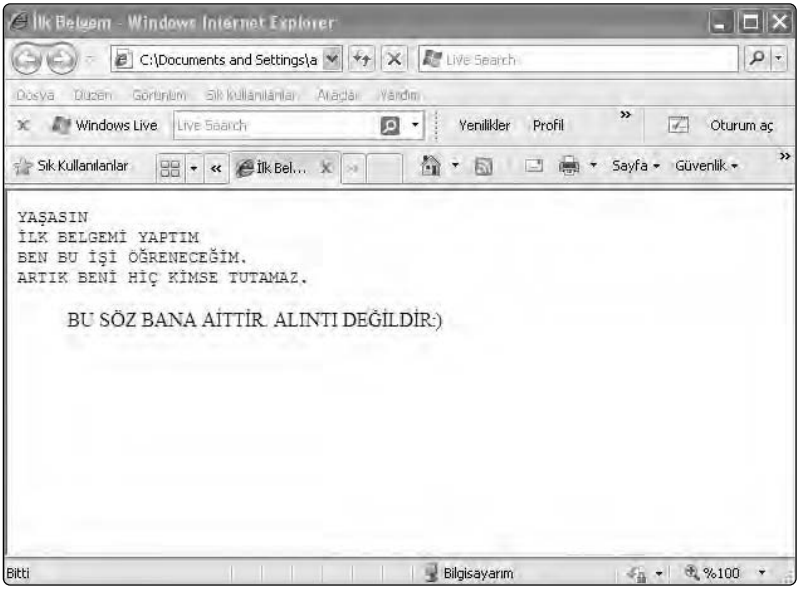
Bir diğer etiketimiz ise <pre></pre> etiketidir. Bu etiket white space denilen karakterlerin dikkate alınmasını sağlar. Yani araya <br /> etiketi koymamış olsak da boşlukları dikkate aldığından metin tek bir satır şeklinde değil de bizin eklediğimiz gibi görünür.

```
<body>
<pre>
YAŞASIN
İLK BELGEMİ YAPTIM
BEN BU İŞİ ÖĞRENECEĞİM.
ARTIK BENİ HİÇ KİMSE TUTAMAZ.
</pre>
</body>
```



Son olarak <blockquote></blockquote> etiketlerinden söz edelim. Bunun marifeti, metinlerdeki alıntıları metinden ayrı olarak vurgulamak içindir. Bu etiketi sadece alıntılarda değil girintili olarak gösterip vurgulamak istediğimiz her metin için de kullanabiliriz.

```
<body>
<pre>
YAŞASIN
İLK BELGEMİ YAPTIM
BEN BU İŞİ ÖĞRENECEĞİM.
ARTIK BENİ HİÇ KİMSE TUTAMAZ.
</pre>
<blockquote>
   BU SÖZ BANA AİTTİR. ALINTI DEĞİLDİR_
</blockquote>
</body>
```



Listeler

Verilerinizi web'de sunarken, onları daha anlaşılır ve düzenli yapabilmenin yöntemlerinden biri de listeli sunumdur. Bu bölümde verilerimizi listeler halinde nasıl sunabileceğimiz göreceğiz.

Sıralı ve Sıralı Olmayan Listeler Oluşturma

Sıralanmış liste (*ordered list*), numaralı listeleri ifade eder. Etiketi `<ol>`'dir. **Sıralanmamış liste** (*unordered list*) ise, madde imleri ile yapılan listeler için kullanılır. Etiketi `<ul>`'dir.

Listelerin içerisindeki her bir öğe içinse **liste öğesi** (*list item*) anlamına gelen `<li>` etiketi kullanılır. `<li>` etiketi `<ol>` veya `<ul>` etiketinin içinde kullanılır.

İki liste tipini de örneklendirelim.

Sıralanmış Liste	Web Sayfasındaki Görünümü
ÖĞRENMEM GEREKENLER ** ** - XHTML - CSS - DREAMWEAVER - FLASH - PHOTOSHOP ** **	**ÖĞRENMEM GEREKENLER** * XHTML * CSS * DREAMWEAVER * FLASH * PHOTOSHOP
ÖĞRENMEM GEREKENLER ** ** - XHTML - CSS - DREAMWEAVER - FLASH - PHOTOSHOP ** **	**ÖĞRENMEM GEREKENLER** 1. XHTML 2. CSS 3. DREAMWEAVER 4. FLASH 5. PHOTOSHOP

İç İçe Listeler

Listeler iç içe de kullanılabilirler. Bu sayede detaylı listeleri sergilemek mümkün olmaktadır. Burada önemli olan, içe koyduğumuz listenin etiketlerini doğru bir sıralamaya göre yerleştirmektir. Arka sayfadaki tabloda da göreceğiniz gibi alt listenin etiketleri <ul> , altında yer aldıkları liste öğesinin içine, XHTML: <ul> şeklinde yuvalanmıştır.

Liste	Web Sayfasındaki Görünümü

`<h3>ÖĞRENMEM GEREKENLER</h3>`
`<ol>`
`  <li>XHTML:`
`  <ul>`
`  <li>Belge Bildirimi</li>`
`  <li>Metin Etiketleri</li>`
`  <li>Listeler</li>`
`  <li>Tablolar</li>`
`  <li>Çerçeveler</li>`
`  </ul>`
`  </li>`
`  <li>CSS</li>`
`  <li>DREAMWEAVER</li>`
`  <li>FLASH</li>`
`  <li>PHOTOSHOP</li>`
`</ol>`
	ÖĞRENMEM GEREKENLER
	1. XHTML:
	o Belge Bildirimi
	o Metin Etiketleri
	o Listeler
	o Tablolar
	o Çerçeveler
	2. CSS
	3. DREAMWEAVER
	4. FLASH
	5. PHOTOSHOP

Numaralı bir listeyi 1'den değil de başka bir sayıdan başlatmak istiyorsanız `<ol>` etiketini `<ol start="n">` şeklinde `start` özniteliği ile kullanmalısınız. Buradaki `"n"` değeri, başlanacak sayıyı ifade eder. Örneğin; `<ol start="5">` dediğimizde numaralandırma 1'den değil 5'ten başlayacaktır.

KÖPRÜLER: WEB BAĞLANTILARI

Web bağlantıları (*hyperlink*) web sayfalarının birbirine bağlanmasıdır. Söz konusu linkler aynı sitenin farklı sayfalarına ya da bir siteden başka bir sitenin herhangi bir sayfasına verilmiş olabilir.

Bağlantı bir metin olabileceği gibi, resim de olabilir. Bir linki tıklamak bugüne kadar internete gezinirken defalarca yaptığınız bir işlem. Bu işlemin nasıl gerçekleştiğini şimdi öğreneceğiz.

Bu işlem için `<a>` etiketini ve `href` özniteliğini kullanacağız.

Örnek

```
<a href="http://www.siteadresi.com"> Tıklanacak Metin </a>
```

Tarayıcıda; **Tıklanacak Metin** şeklinde görünecektir. Bağlantı metinleri varsayılan olarak altı çizgili olarak görüntülenir.

Gidilecek adres aynı klasördeki bir dosya ise, site adresini uzun uzun yazmak yerine,

```
<a href="dosya_ismi.html"> Tıklanacak Metin </a>
```

örneğinde olduğu gibi sadece dosya adını yazmanız yeterli olacaktır. Alt klasörde yer alan bir dosyaya link vermek istediğimizde ise

```
<a href="klasör/dosya_ismi.html"> Tıklanacak Metin </a>
```

örneğindeki gibi klasörün adını da yazmamız gerekir. Üst klasöre link vermek için ise, link verilecek dosyanın bir üst klasörde olduğunu belirtmek üzere

```
<a href="../dosya_ismi.html"> Tıklanacak Metin </a>
```

örneğindek gibi `../` şeklindeki bir kullanım yeterli olacaktır.

Eğer linkimizi aynı sayfanın belli bir yerine vermek istiyorsak orada bir **çapa** (*anchor*) kullanmamız gerekir. Bu sayede bir web sayfasında mesela alt kısma kaydırma çubuğu kullanmadan bir anda inmek mümkün olacaktır. Yapılacak iş; önce çapanın yerini belirlemek daha sonra da köprü oluşturmaktır. Çapanın yerini belirlemek için `<a>` etiketi ile `name` özniteliğini kullanacağız.

Kullanım

Çapanın yeri: `<a name="çapa_adı"> Metin </a>`

Link: `<a href="#çapa_adı">Tıklanacak Metin</a>`

Örnek olarak üstte **Listeler** konusunu izah ederken kullandığımız örneği ele alalım. Sayfanın baş tarafındaki listede yer alan XHTML maddesi ile ilgili izahatın sayfamızın sonunda olduğunu varsayalım. Biz, aradaki diğer konulara bakmadan XHTML yazısına tıklayıp hemen sayfanın en altındaki açıklama paragrafına inmek istiyoruz.

Bunun için önce sayfanın en altındaki açıklama paragrafımızın başlığına, inilecek hedef yerin burası olduğunu gösteren

```
<a name="XHTML"> XHTML Açıklama </a>
```

etiketi yazıyoruz. Ardından sayfanın başındaki listeden tıklanacak liste öğemiz olan XHTML öğesine

```
<a href="#XHTML">XHTML</a>
```

şeklinde link veriyoruz. Artık linke tıkladığımızda doğrundan sayfanın en altındaki paragrafa ineceğiz.

Bu başlık altında son olarak bir e-posta adresine bağlantı oluşturmayı göreceğiz. Bunun için `href` özniteliği ile beraber `mailto:` ve e-posta adresini yazacağız.

Kullanımı

```
<a href="mailto:deneme@siteismi.com">Bana ulaşmak için tıklayın</a>
```

RESİMLER

Web sayfalarımız teknik olarak sadece metinlerden oluşabilir. Fakat resim içeren sayfaların daha ilgi çekici ve daha etkileyici olacağı yadsınamaz bir gerçektir. Geliştirilmiş bir çok resim formatı olmasına karşın tüm bilindik tarayıcılar tarafından desteklenen resim formatları **GIF**, **JPG** ve **PNG** formatlarıdır.

RESİM EKLEME

Sayfamızda, resmin görüntülenmesini istediğimiz yere `img` etiketini kullanarak

```
<img src="resim.gif" />
```

şeklinde resmi yerleştiriyoruz.

Eğer resmimiz XHTML belgemizle aynı yerde ise:

```
<img src="resim.gif" />
```

şeklindeki kullanım yetirlidir.

Ancak resmimiz örneğin "Resimler" adındaki bir alt klasörde ise

```
<img src="Resimler/resim.gif" />
```

şeklinde resim klasörünün adını da yazmamız gerekir.

Eğer resim farklı bir ortamda ise, resmin konumuna ait adresin tamamını yazmamız gerekir.

```
<img src="http://www.siteismi.com/Resimler/resim.gif" />
```

Resimlere Link Vermek

Metinlere link vermek ile resimlere link vermek arasında, işin mantığını bakımından hiçbir fark yoktur.

Metinlere;

```
<a href="http://www.siteadresi.com"> Tıklanacak Metin </a>
```

şeklinde link vermiştik. Resimlerde ise tıklanacak metnin yerine resmin konumunu yazacağız.

Kullanımı

```
<a href="http://www.siteadresi.com"> <img src="Resimler ↓
/resim.gif" /></a>
```

Sayfamızda yer alan link verilmiş bir resim, varsayılan olarak resmin üzerine gelindiğinde beliren bir kenarlığa sahiptir.

Resme tıklandığında açılacak olan hedef sayfanın, farklı bir tarayıcı penceresinde açılmasını istersek `target="_blank"` özniteliğini kullanmamız gerekir.

O takdirde kullanılacak yapı:

```
<a href="http://www.siteadresi.com"  target="_blank"  >
<img src="Resimler /resim.gif" /></a>
```

şeklinde olacaktır.

Resimlere Açıklama Ekleme

Resimlerin üzerine fare ile gelindiğinde görünen bir açıklamayı eklemek için `alt` özniteliğini kullanacağız. Bu metin, aynı zamanda resmin

herhangi bir nedenden dolayı görüntülenemediği durumlarda daha önemli olacak ve haliyle açıklayıcı bir rol üstlenmiş olacaktır. Bu değişkenin bir faydası da arama motorlarının resimleri sağlıklı ve anlamlı bir biçimde indeksleyebilmeleri içindir.

Örnek Kullanım

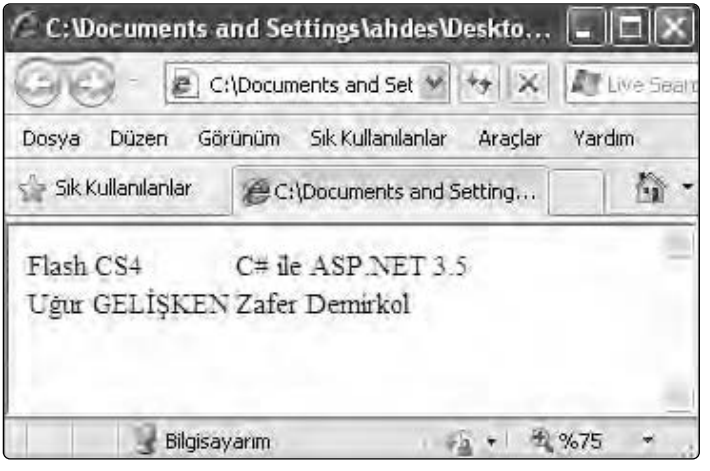
```
<img src="Resimler /resim.gif"  alt="site logosu" />
```

TABLOLAR

Çok sayıdaki veriyi daha net ve anlaşılır bir biçimde sunmayı kolaylaştıran tablolar, satırlar ve sütunlardan oluşurlar. Metinler, resimler ya da başka tablolar içerebilirler. Tablo oluşturmak için `<table>` etiketi kullanılır. Satırları tanımlamak için `<tr>` sütunları tanımlamak içinse `<td>` etiketleri kullanılır.

TABLO OLUŞTURMAK

Genel Kullanım Şekli




```
<table>
   <tr>
          <td>Flash CS4</td>
          <td>C# ile ASP.NET 3.5</td>
   </tr>
   <tr>
```

```
            <td>Uğur GELİŞKEN</td>
            <td>Zafer Demirkol</td>
    </tr>
</table>
```



Tablomuzu daha belirginleştirmek için çerçeve ekleyebiliriz. Bunun için `table` etiketi ile beraber `border` özniteliğini `<table border="1">` şeklinde kullanacağız. Değişkene atanacak `1`, `2`, `3` gibi değerler tablo kenarlığının kalınlığını belirleyecektir. Değer büyüdükçe kenarlıklar daha da kalınlaşacaktır.

Tablo Boyutunu Ayarlamak

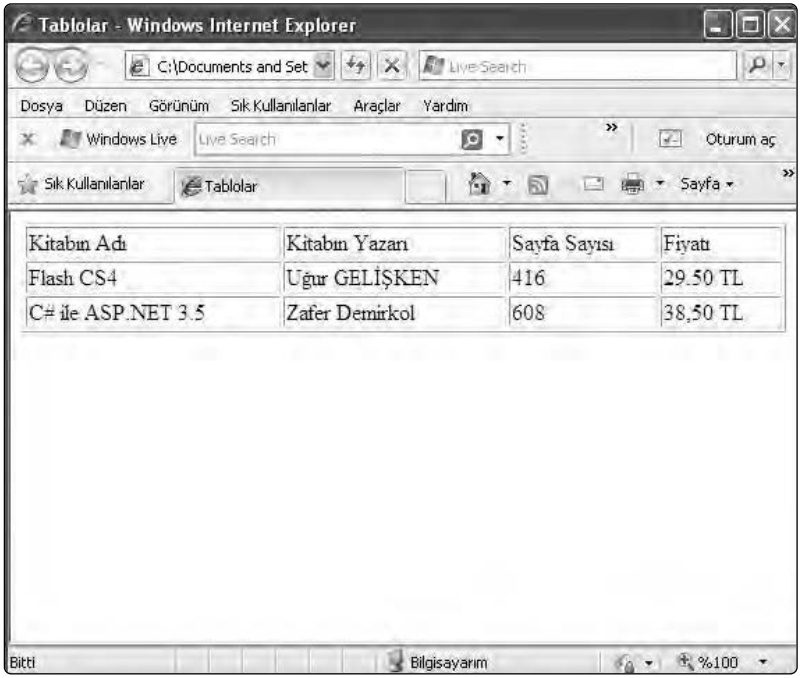
Varsayılan olarak tablonun ebatları, tablonun hücrelerindeki en uzun girdilere göre ayarlanmaktadır. Tablodaki girdiler değiştiğinde, satır veya sütun eklendiğinde tablo büyür ya da küçülür.

Örnek tablomuz resimdeki haliyle biraz tıkış tıkış görünüyor. Bu durumda tabloyu genişletip görsel olarak rahatlatmanın yolu tablonun genel ebatlarını %100 yapmaktır. Bu sayede tablomuz tarayıcı penceresi kadar genişleyecektir. Yapmamız gereken sadece `table` etiketine `width` özniteliğini eklemektir.

Kullanım

```
<table width="100%">
```

şeklindedir.



Yüzde 100 kullanmak yerine dilerseniz `50`, `75`, `90`... vb başka bir yüzde değeri de seçebilirsiniz. Bu yüzdeli kullanımın tek sakıncası değişik tarayıcı ebatlarında (her halükarda tarayıcı penceresini tamamen kaplaması gerektiğinden) tablo genişliğinin değişecek olmasıdır. Bu noktadaki alternatifimiz, yüzde değeri yerine sabit bir piksel değeri girmektir. Böyle yaptığımızda genişlik tarayıcının ebatlarından etkilenmeksizin her daim sabit kalacaktır. Mesela genişliği `<table width="800px">` şeklinde `800` piksel olarak ayarlarsak, `800` piksellik tablo genişliği hiç değişmeyecektir. Daha küçük tarayıcı ebatları söz konusu olduğunda sayfa sağa taşacağından yatay kaydırma çubuğu devreye girecektir.

Sütun Genişliklerini Ayarlamak

Tabloların tamamının ebatlarını ayarlamak mümkün olduğu gibi dilersek sadece sütun genişliklerini de ayarlayabiliriz. Bunun için `td` etiketini `<td width="150px">` şeklinde kullanmamız yeterli olacaktır. Tablonun genelinde olduğu gibi sütun genişliklerini belirlerken de sabit piksel değeri ya da yüzdelik değeri kullanabiliriz.

NOT Sütun genişliklerini ayarlarken daha önce tablo için `<table width="800px ">` şeklinde sabit bir piksel değeri belirlediyseniz sütun genişliği değişmez.

Hücreleri Birleştirmek

Yan yana olan hücrelerden istediklerimizi birleştirerek tek bir hücre haline getirebiliriz. Bir hücreyi sağındaki hücreyle birleştirmek istediğimizde `colspan` özniteliğini ve birleştirilecek hücre sütun adedini

```
<td colspan="4">
```

şeklinde kullanırız. Hücreyi altındaki hücrelerle birleştirmek içinse `rowspan` özniteliğini kullanırız.

Etiketi,

```
<td rowspan="2">
```

şeklinde birleştirilecek satır sayısı ile beraber kullanırız.

Öğrendiklerimizi, yukarıdaki resimde gördüğünüz tabloyu baz alıp deneyerek pekiştirelim.

```
<table width="600px" border="1">
   <tr>
          <td colspan="4">Kodlab Kitapları</td>
          <td rowspan="4"><img src="flashcsx.jpg" ↓
alt="Kodlab Kitapları"/></td>
   </tr>
   <tr>
          <td>Kitabın Adı</td>
          <td>Kitabın Yazarı</td>
          <td>Sayfa Sayısı</td>
          <td>Fiyatı</td>
   </tr>

   <tr>
          <td>Flash CS4</td>
          <td>Uğur GELİŞKEN</td>
          <td>416</td>
          <td>29.50 TL</td>
   </tr>
   <tr>
```

```
            <td>C# ile ASP.NET 3.5</td>
            <td>Zafer Demirkol</td>
            <td>608</td>
            <td>38,50 TL</td>
    </tr>
</table>
```

Gördüğünüz gibi tablomuzun en üstüne 4 sütunu birleştirerek bir tablo başlığı ekledik.

Bunun için etiketimizi:

```
<td colspan="4">Kodlab Kitapları</td>
```

şeklinde kullandık. Daha sonra da kitaplardan birinin resmini ekleyebilmek için mevcut 4 satırı birleştirerek elde ettiğimiz tek hücreye kitaplardan birinin resmini koyduk.

Bunun için de etiketi

```
<td rowspan="4"><img src="flashcsx.jpg" alt="Kodlab Kitapları"/></td>
```

şeklinde kullandık. Ve ortaya aşağıdaki resimde gördüğünüz tablo çıktı.



Siz de aynı mantıkla ihtiyacınız olan tabloları verdiğimiz bilgiler ışığında rahatlıkla oluşturabilirsiniz.

Tablolarla Sayfa Tasarimi

Şu ana kadar tabloları, verilerin daha düzenli sunumunu kolaylaştırmak için kullandık. Tabloları sayfamızın genel yapısını düzenlemek için de kullanabiliriz. Bu yöntem artık pek fazla tercih edilmemekte, genel sayfa düzenlemeleri az sonra değineceğimiz **bölümleme yöntemi** (*division-div*) ile yapılmaktadır.

Fakat yapacağınız bazı sayfaların özellikleri gereği tabloları kullanmak daha pratik hatta gerekli olabilecektir. Bu sebeple tablo kullanarak sayfa biçimleme konusunu da bilmeniz gerekmektedir.

Esasen sayfamızı tablo kullanarak şekillendirirken, teknik olarak şimdiye kadar gördüklerimizden farklı bir kullanım söz konusu değildir. Şu ana kadar, sunacağımız verileri tablo içine alıyorduk şimdi ise sayfamızı tablo içine alacağız.

Sayfamızı tek bir tablo içine alabileceğimiz gibi birkaç tabloya da bölebiliriz. Hatta sayfamızı oluşturan tabloları bir başka tablonun içine alabiliriz.

Şimdi bu söylediklerimizi pratiğe dökelim. Öncelikle aşağıdaki sayfamızı inceleyelim...

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xlmns="http://www.w3org/1999/xhtml">
<head>
<title> Tablo ile sayfa düzeni </title>
<meta name="keywords" content="kelime1, kelime2, kelime3"
/>
</head>
<body>
<table width="650px" border="4" align="center">
<tr>
<td>
   <h1 align="center">Tablo ile Sayfa Düzeni Denemesi</h1>
          <table width="600px" border="1" align="center">
                 <tr>
                        <td colspan="4" align="center">Kodlab ↓
Kitapları</td>
```

```
                    </tr>
                    <tr>
                            <td>Kitabın Adı</td>
                            <td>Kitabın Yazarı</td>
                            <td>Sayfa Sayısı</td>
                            <td>Fiyatı</td>
                    </tr>
                    <tr>
                            <td>Flash CS4</td>
                            <td>Uğur GELİŞKEN</td>
                            <td>416</td>
                            <td>29.50 TL</td>
                    </tr>
                    <tr>
                            <td>C# ile ASP.NET 3.5</td>
                            <td>Zafer Demirkol</td>
                            <td>608</td>
                            <td>38,50 TL</td>
                    </tr>
            </table>
   <br />
            <table width="600px" border="1" align="center">
                    <tr>
                            <td><p>

Esasen sayfamızı tablo kullanarak şekillendirirken teknik
olarak şimdiye kadar gördüklerimizden farklı bir kullanım
söz konusu değildir. Şu ana kadar sunacağımız verileri tablo
içine alıyorduk şimdi ise sayfamızı tablo içine alacağız.</p>

<p>Sayfamızı tek bir tablo içine alabileceğimiz gibi birkaç
tabloya da bölebiliriz. Hatta sayfamızı oluşturan tabloları
bir başka tablonun içine alabiliriz.</p>
                            </td>
                    </tr>
            </table>
   <h4 align="center">Alt Kısım</h4>
</td>
</tr>
</table>
```

```
</body>
</html>
```

Gördüğünüz gibi tüm sayfayı:

```
<table width="650px" border="4" align="center">
<tr>
<td>
TÜM SAYFA BU ARADA YER ALIYOR
</td>
</tr>
</table>
```

şeklinde tablo içine aldık.

Sayfadaki diğer içerikleri ve tabloları ise alt alta bu ana tablonun içine yerleştirdik. Ve aşağıda resmini gördüğünüz sayfa ortaya çıkmış oldu.



Sayfanın içindeki her şeyi bir ana tablonun içine almamız daha düzenli bir sayfa elde etmemizi sağladı.

Bölümleme Yöntemi (div) ile Sayfa Tasarımı

Yukarıda bahsettiğimiz gibi günümüzde en çok kullanılan sayfa biçimleme yöntemi, **bölümleme yöntemi** (*division-div*) olmaktadır.

Tablo etiketi olan `table` yerine `div` etiketi kullanılır. Sayfayı bu şekilde düzenlemenin en önemli avantajı, sayfanın tüm görsel düzenlemelerinin harici stil sayfaları ile yapılabilmesidir.

Bu sayede oluşturulan stil bir çok sayfada kullanılabilmektedir. Bu da, daha az kod kalabalığı ve çok büyük zaman tasarrufu anlamına gelmektedir. Stil sayfaları ile düzenleme yapma konusunu kitabımızın bir sonraki bölümü olan CSS bölümünde detaylı olarak inceleyeceğiz.

Etiketin Genel Kullanımı

```
<div id="eşsiz_isim">sayfa içeriği</id>
```

şeklindedir. Her `div` etiketi eşsiz bir `id` değeri alır. Bu sayede her `div` farklı tasarım özelliklerine sahip olabilecektir. Toparlamak gerekirse; XHTML sayfamızı `div` etiketi ile planlayıp, kullandığımız `id` değerlerini de harici bir stil dosyasında tanımlayarak sayfamızı şekle şemale sokmuş oluyoruz.

Yukarıdaki örneğimize dönecek olursak;

```
<table width="650px" border="4" align="center">
<tr>
<td>
TÜM SAYFA BU ARADA YER ALIYOR
</td>
</tr>
</table>
```

sayfamızın ana çerçevesini oluşturan bu ana `table` etiketleri yerine `div` etiketlerini koyalım.

```
<div id="ana_kapsayıcı">
TÜM SAYFA BU ARADA YER ALIYOR
</div>
```

En dışta yer aldığı için `id` değeri olarak `ana_kapsayıcı` ifadesini kullandım. Bu etiketi kullanırken biz tablo ifadesini değil de `kapsayıcı` ifadesini kullanacağız. Çünkü bu etiket, tablosuz tasarımı –CSS ile birlikte– ifade etmektedir. Bu değeri, diğer `id` değerlerinden farklı olmak şartı ile istediğiniz herhangi bir ifade olarak seçebilirsiniz. Fakat kapsamlı bir sayfa tasarlarken bu değeri anlamlı ve kullanıldığı görevi çağrıştıran bir biçimde seçerseniz işinizi kolaylaştırmış olursunuz.

Bu değişikliği yaptığımızda üstteki resimde gördüğünüz dış tablonun kaybolduğunu ve yerine de herhangi bir düzenlemenin gelmediğini göreceksiniz. Bunun sebebi `div` etiketimize atadığımız `ana_kapsayıcı` adlı id değerini içeren bir tasarımlama yapmadığımız içindir. Konu bütünlüğünü bozmamak için bu konuyu birkaç sayfa ötenizdeki CSS bölümüne havale ediyorum. Burada amacımız `div` etiketinin genel kullanım şeklini göstermekti. Onu da sanıyorum başardık.

FORMLAR

Formlar, site ziyaretçilerinden geri bildirim almak için kullanılırlar. Sitenizin kullanıcıları hakkında detaylı bilgi sahibi olurken aynı zamanda da onlarla daha etkin bir iletişim içerisinde olmuş olursunuz. Ayrıca formlar aracılığıyla onlara bir takım hizmetler de sunabilirsiniz.

FORM OLUŞTURMA

Oluşturacağımız form, `form` etiketleri arasında yer alır ve `method` özniteliği ile beraber kullanılır.

```
<form method="post">
.......
</form>
```

Buradaki `method` özniteliğinin görevi, form onaylandığında yapılacak işlemin ne olacağını belirlemektir. `post` ve `get` olmak üzere iki metod söz konusudur. **`get`**, bilgilerin site URL'sinin arkasına eklenerek gönderileceği anlamına gelir. **`post`** ise daha güvenli ve yaygın bir yöntemdir. HTTP sorgusu ile gönderilir. Fakat burada bu konunun detaylarına girmeyeceğiz. Çünkü şu anda bizi, formun tasarımı ilgilendiriyor. Form gönderme olayına PHP bölümünde detaylı olarak gireceğiz.

METİN KUTULARI

Metin kutuları formların içerik girilen ana unsurlarıdır.

Standart bir metin kutusu:

```
<input type="text" />
```

şeklindedir.

Genel amaçlı `input` etiketinin tek taraflı bir etiket olduğuna dikkat ediniz. Her zaman olduğu gibi bunu da bir boşluk ve `/` işareti ile bitiriyoruz. `type="text"` ifadesi girilecek veri tipinin bir metin olacağı anlamını taşır. `input` etiketi ile `text` haricinde ihtiyacınıza göre **password**, **checkbox**, **radio**, **submit**, **reset**, **file**, **hidden**, **image**, **button** değerlerinden herhangi biri de kullanılabilir.

Ayrıca her seferinde mutlaka, name özniteliği ile beraber bir özel değer de atanmalıdır. Bu alana girilecek veriler bu değişken ismi altında değerlendirilecektir.

Kullanımı

```
<input type="text"  name="ismi" />
```

Metin kutusunun varsayılan genişlik değeri `20` pikseldir. Bu değeri `size` özniteliği ile değiştirebilirsiniz.

Kullanımı

```
<input type="text"  name="ismi" size="30" />
```

Ayrıca girilecek metinin karakter sayısını sınırlamak istersek maxlength özniteliğini kullanırız.

Kullanımı

```
<input type="text"  name="ismi" size="30"  maxlength="100"  />
```

Formumuza bir de çok satırlı bir metin alanı (örn: adres bilgisi için) ekleyelim. Bunun için `textarea` özniteliğini kullanacağız. Ayrıca değişken ile birlikte `rows` değerini yani satır sayısını, `cols` ile de uzunluğun kaç karakter eninde olacağını belirleyeceğiz.

```
<textarea name="adresi" rows="5" cols="30"></textarea>
```

Örnek

```
<body>
<h2>Başvuru Formu</h2>
<table>
<form method="post">
<tr>
      <td>Adı : </td>
      <td><input type="text" name="ismi" /></td>
</tr>
<tr>
      <td>Soyadı:</td>
      <td><input type="text" name="soyismi" /></td>
</tr>
<tr>
      <td>Adresi:</td>
      <td><textarea name="adresi" rows="5" cols="30"></tex-
tarea></td>
</tr>
</form>
</table>
</body>
```

Örneğimize dikkat ederseniz form elemanlarımızın bir tablo yapısı içinde sunulduğu göreceksiniz. Bunu bu şekilde ayarlamamın sebebi formun düzgün, alt alta ve hizalı bir biçimde görünmesini temin etmek içindir. Bundan sonraki örnekleri de tablo içerisinde göstereceğimizi belirteyim. Böylece tablo konusunu da pekiştirmiş oluyoruz. Bir taşla iki kuş.

BUTONLAR

Forumların, sayfaya gelen ziyaretçiler tarafından doldurulduktan sonra size gönderilebilmeleri için bir **onaylama** (*submit*) düğmesine ihtiyaç vardır. Bu sefer `type` özniteliği ile `text` değerini değil `submit` değerini kullanacağız. Ayrıca düğmenin üzerinde görünecek ifadeyi de value özniteliği ile belirleyeceğiz.

Kullanım

```
<input type="submit" value="Gönder" />
```

Ayrıca, kişinin dilediğinde formu baştan doldurabilmesi için form içeriğini temizleyen bir düğme eklemek de kullanıcıların işini kolaylaştıran bir özelliktir.

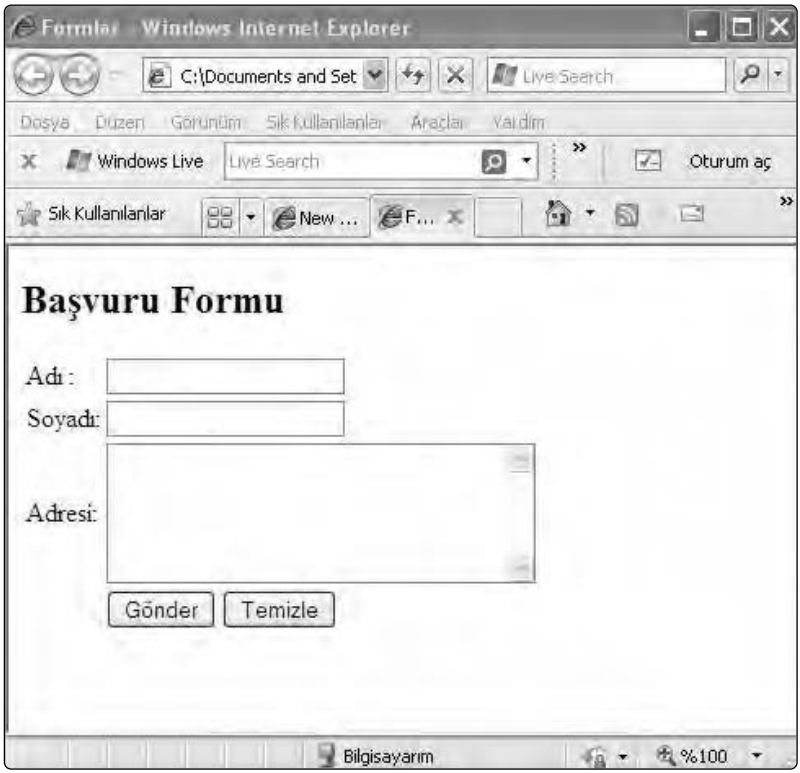
Kullanımı

```
<input type="reset" value="Temizle" />
```

Örnek

```
<tr>
      <td></td>
   <td><input type="submit" value="Gönder" />
   <input type="reset" value="Temizle" /> </td>
</tr>
```

Buton örneğimizden sonra formumuzun geldiği hâl, aşağıdaki resimdeki gibidir.



Onay Kutuları

Cevabın birkaç seçenekle sınırlı olduğu durumlarda onay kutuları kullanmak akıllıca olacaktır.

Onay Kutusu için

```
<input type="checkbox" name="id" value="Evet"  />
```

şeklinde, type değişkeni için checkbox değerini kullanıyoruz. Onay kutucuklarından birinin (mesela evet ve hayır olarak iki kutucuğumuz varsa evet kutucuğunun seçili olmasını isteyebiliriz.) varsayılan olarak seçili gelmesini istiyorsanız checked=" checked" değişkenini de eklemelisiniz.

```
<input type="checkbox" name="id" value="Evet" checked=" checked"  />
```

Örnek

```
<tr>
<td>Medeni Durumu:</td>
      <td><input type="checkbox" name="medeni" value="evli" ↓
/>Evli
      <input type="checkbox" name="medeni" value="bekar" ↓
/>Bekar</td>
</tr>
```

Radyo Düğmeleri

Bir grup seçenek arasından tercih yapma mantığına dayanan bu form elemanı da çok işlevseldir.

Kullanımı

```
<input type="radio" name="id" value="Flash" /> Flash <br />
<input type="radio" name="id" value="Photoshop" /> Photoshop <br />
<input type="radio" name="id" value="Fireworks" /> Fireworks <br />
```

Her düğmenin yanına düğmenin metnini ve düğmelerin alt alta gelerek düzgün görünmeleri için de <br /> etiketini ilave ettik.

Örnek

```
<tr>
      <td>Uzmanlık Alanı</td>
      <td><input type="radio" name="uzman" value="Flash" />
```

```
Flash<br />
      <input type="radio" name="uzman" value="Photoshop" />
Photoshop<br />
      <input type="radio" name="uzman" value="Fireworks" />
Fireworks<br /></td>
</tr>
```

Açılır Seçenek Kutuları

Çok fazla seçeneğin olduğu durumlarda onay kutuları ve radyo düğmeleri çok uzun bir liste halinde sayfayı kaplayacağından çok kötü ve kullanışsız bir durum ortaya çıkmış olacaktır. Bu durumda en uygun tercih açılır kutulardır. Bu sayede çok sayıda seçeneği hiç yer kaplamadan formumuza ekleyebiliriz.

Genel Yapısı

```
<select name="id" size="1">
.........
</select>
```

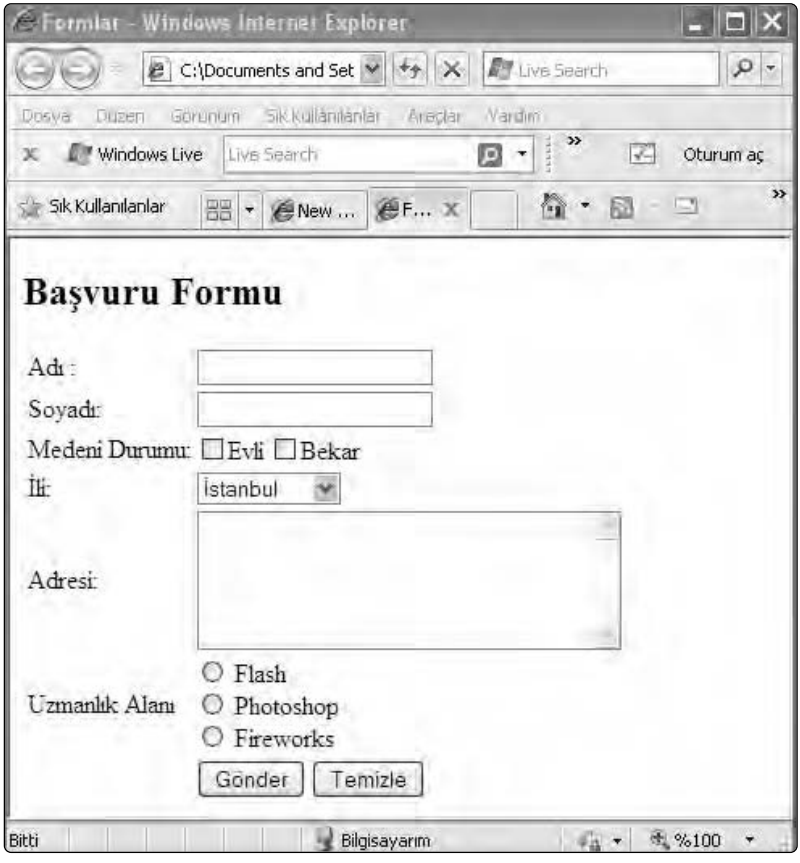
şeklindedir. `size` değerini artırmanız demek bir liste kutusu oluşturmanız demektir. Örneğin bu değeri "5" olarak ayarlarsanız 5 seçeneğin göründüğü bir kutucuk oluşturmuş olursunuz. Diğer seçenekler ise kutucuğu açtığımızda görünecektir. Kutucukta yer alacak her bir seçenek içinse; `<option>seçenek</option>` etiketini kullanacağız. Seçeneklerimiz çok fazla ise ve biz onları tasnif etmek istiyorsak bu sefer de `<optgroup label="grup_ismi"> </optgroup>` etiketini kullanacağız.

Örnek

```
<tr>
      <td>İli:</td>
      <td><select name="iller" size="1">
      <optgroup label="marmara">
            <option>İstanbul</option>
            <option>Bursa</option>
            <option>Kocaeli</option>
      </optgroup>
      <optgroup label="ege">
            <option>İzmir</option>
            <option>Aydın</option>
```

```
            <option>Manisa</option>
        </optgroup>
            <optgroup label="akdeniz">
            <option>Antalya</option>
            <option>Mersin</option>
            <option>Adana</option>
        </optgroup>
        </select>
        </td>
</tr>
```

Buraya kadar yaptığımız tüm işlemlerin bir arada görünen hali acaba nasıldır? Bu sorunu cevabı için resme bakın.



Çoklu Ortam (Ses &Video)

Web sayfalarımıza metin ve resim haricinde ses ve video dosyaları da ekleyebiliriz. Bunu iki yöntem ile yapacağız. İlk yöntem bir çoklu ortam dosyasına link vermek.

Bunun için:

```
<a href="dosya_ismi.mov">Tıklanacak Metin</a>
```

yapısını kullanacağız. Bu yöntemde ziyaretçi sitenize gelip bu bağlantıyı tıkladığında, bilgisayarında dosyayı çalıştırabilecek hangi program varsa o program devreye girip dosyayı çalmaya başlayacaktır.

Kullanacağımız diğer yöntem ise çoklu ortam dosyasını sayfamıza eklemektir. Bu durumda dosya, site ziyaretçilerinin bilgisayarlarındaki harici programlarla (Örn: *Media Player*) değil, ziyaretçilerin tarayıcılarındaki eklentiler aracılığıyla oynatılır.

Bu yöntem için `embed` etiketini kullanacağız.

Kullanımı

```
<embed src="dosya_ismi.wav"  autostart="false" />
```

`autostart` özniteliği false değeriyle beraber kullanıldığında dosya bittiğinde tekrar başlayacak ve otomatik olarak sürekli çalacaktır. Aynı işlem için loop özniteliğini `infinite` değeri ile de kullanabiliriz. Ayrıca `loop` değişkeniyle bir tam sayı kullanarak çoklu ortam dosyamızın kaç kere tekrar yürütüleceğini belirleyebiliriz.

Kullanımı

```
<embed src="dosya_ismi.wav"  loop="false" />
```

Örnek

```
<table width="400px" border="1" align="center">
      <tr>
           <td colspan="2" align="center">Video Embed</td>
      </tr>
```

```
        <tr>
            <td><embed src="çocuklar.avi" width="400" ↓
height="250" loop="false" /></td>
        </tr>
</table>
```



Evet, bu konuyla birlikte XHTML bölümünü bitirmiş bulunmaktayız. XHTML dilini işlerken bazı eski standart dışı olan ve artık kullanılmayan etiketlere değinmedik. Ayrıca sayfalarımızın görsel tasarımı konusuna da değinmedik. Bunun sebebi az sonra XHTML belgemize stil özelliklerini nasıl kazandıracağımızı CSS bölümünde detaylı olarak anlatacak olmamız. Hem böylece konu bütünlüğü de bozmamış olduk. Belki bölümü okurken, bu XHTML dili de ne sade bir şeymiş dediniz. Şimdi sayfayı çevirin ve okumaya devam edin. Yavaş yavaş ortalığın şenlendiğini göreceksiniz.

3

CSS

CSS'i Belgelere Ekleme 48
CSS'in Yapısı 50
Açıklama Satırı 51
Seçiciler 51
Artalan (Background) Biçimlendirme 65
Yazıtipi (Font) Biçimlendirme 72
Metin (Text) Biçimlendirme 78
Kutu Modeli (Box Model) 88
Liste (List) Biçimlendirme 103
Tablo (table) Özellikleri 109
Sınıflandırma (classification) ve Konumlama (positioning) Özellikleri 120
Görsel Sonuçlar 129
Sözde Sınıflar (pseudo-classes) 137
Sözde-Öğeler (pseudo-elements) 142

CSS (**C***ascading* **S***tyle* **S***heets-Basamaklı Stil Sayfaları*), web sayfalarını biçimlendirmemizi sağlayan en harika teknolojidir.

Harikadır çünkü sayfa yapısı ile sayfa tasarımını birbirinden ayırmamızı sağlar. Yani sayfamızın yapısını XHTML ile, tasarımını ise CSS ile oluştururuz. Bunun en önemli faydası, her sayfa için ya da sayfadaki her öğe için tekrar tekrar tasarım yapmak yerine, tasarımı bir kez CSS ile yapıp istediğimiz tüm sayfalara uygulamaktır. Onlarca sayfadan oluşan bir siteniz olduğunda her sayfa için tasarım yapmanın ne kadar zahmetli olacağını tahmin edersiniz. İşte CSS bize bu noktada çok büyük kolaylık sağlayarak zamandan ve emekten çok ciddi oranda tasarruf yapmamıza imkan vermektedir.

Bu noktaya gelmişken size, buraya kadar anlattıklarımı göstermek istiyorum. Aşağıdaki ilk resimde yayınevimizin sitesinin normal, yani tasarımlı halini; ikinci resimde ise CSS dosyasının devre dışı olduğu durumu görmektesiniz.



Sayfayı tüm tasarım CSS unsurları ile BERABER görmektesiniz.



Sayfayı tüm tasarım CSS unsurlarından ARINDIRILMIŞ olarak görmektesiniz.

Evet, oldukça ilginç gerçekten değil mi? Bu resimlerden CSS'in işin ne kadar önemli bir kısmını üstüne aldığını görmektesiniz. CSS işin önemli bir kısmını üstlenince haliyle çok daha sade ve hızlı XHTML sayfaları elde etmiş oluyoruz.

Bu iki resmi gördükten sonra, sanırım biraz önce okuduğumuz XHTML bölümündeki sade suya tirit durumlarının sebebi hikmetini daha iyi kavramış olduk. Bu arada unutmadan CSS bölümümüzde yapacağımız uygulamaların da DVD'de olduğunu hatırlatayım.

CSS'i Belgelere Ekleme

CSS'i XHTML belgelerimize eklemek için kullanabileceğimiz 4 yöntem mevcuttur. Bu aşamada, detaya girmeden sadece bu dört yöntemi tanıyalım.

İlk yöntem, doğrudan doğruya XHTML etiketi ile beraber (*In-line*) kullanmaktır.

Kullanımı

```
<div style="color:blue">CSS seni de öğreneceğim</div>
```

şeklindedir.

Bu yöntem çok kullanışlı değildir. Çünkü tüm etiketlere bu şekilde stil uygulamaya kalkmak çok zahmetli olacağından, CSS kullanmamızın esprisini de yarı yarıya azaltmış olacaktır. Fakat bir stili sadece bir kez ve bir tek etikete uygulamak gibi durumlarda, hızlıca uygulanıp sonuç alma bakımından kullanılması gerekebilir. Bu yöntemin kullanılmasını gerektiren başka durumlar da olacaktır. Fakat benim ifade etmek istediğim aslında bu yöntemi yaygın olarak kullanmayacağımızdır.

İkinci yöntemimiz, uygulanacak stili, belgemizin `<head> </head>` kısmında bir kez tanımlayarak belgemizin istediğimiz her yerinde kullanmaktır.

Kullanımı

```
<head>
<style type="text/css">
div{color:blue;}
```

```
</style>
</head>
```

Bu yöntem ile hem CSS'i XHTML etiketlerinden ayırmış olduk, hem de aynı stili kullanmamız gereken tüm etiketler için ayrı ayrı yazmaktan da kurtulmuş olduk.

Üçüncü yöntem, harici bir CSS dosyası eklemektir. Bu yöntemde CSS komutlarını ayrı bir *.css* uzantılı dosyaya kaydederiz. Sonra da sayfamıza bu harici CSS dosyasının yolunu gösteririz. Bu yol gösterme işini gene belgemizin `<head> </head>` etiketleri arasında yaparız.

Kullanımı

Harici CSS dosyası. Mesela **ilk_deneme.css**

```
p {color: red;}
a {color: blue;}
```

XHTML dosyamız

```
<head>
   <link rel="stylesheet" type="text/css" href=" ilk_deneme.css" />
</head>
```

Bu yöntem CSS'in farkını esas fark ettiren yöntemdir. Çünkü stiller harici bir dosyadadır ve bu dosyayı bir çok sayfada kullanabiliriz. Dolayısıyla CSS dosyası bir kez yüklendikten sonra sitemizdeki diğer sayfalarda da kullanılacağından bu durum sitemize hız da kazandıracaktır. Üstelik bir değişiklik gerektiğinde belgemize hiç dokunmadan CSS dosyasından stil özelliklerini değiştirebiliriz.

Dördüncü ve son yöntemimiz `@import` ile eklemektir.

Kullanımı

```
<head>
<style type="text/css">
   @import "ornek.css";
</style>
</head>
```

Kullanım mantığı üçüncü yöntemle aynıdır. `@import` ile ekleme yaptığınızda, eski web tarayıcıları harici CSS dosyasını görüntüleyemezler.

Yöntemleri tek tek saydıktan sonra, en avantajlısı ve en kapsamlısı olması sebebiyle üçüncü yöntemin kullanılması tavsiye edilir.

CSS'İN YAPISI

CSS'in detaylı kullanımına geçmeden, öncelikle yapısal olarak iyice bir tanıyalım. Bu sayede CSS kodlarını yazarken hata yapma ihtimalimiz büyük oranda azalmış olacaktır.

CSS'in yapısı, **Seçiciler** (*Selector*) ve **Bildirim Bloğu** (*Declaration Block*) olmak üzere iki ana kısımdan oluşmaktadır. Bildirim Bloğu ise **Özellik** (*Property*) ve **Değer** (*Value*) şeklinde iki parçadan meydana gelir.

```
seçici { Özellik:Değer; }
h1     { font-family:Arial; }
```

`p`, `h1`, `body`... vb bütün XHTML etiketleri CSS için birer seçicidirler. Bir başka ifade ile CSS bu seçicilere özellik ve değer atayarak tasarım işini ifa eder.

Bildirim bloğu süslü parantezler **`{ }`** arasına yazılır. **Özellik** ile **değer** arasına "**`:`**" iki nokta konulur. Bir bildirim bloğu **`;`** (noktalı virgül) ile bitirilir.

Örnek

Bir bildirim bloğu ile seçicileri tek tek kullanmak bir zorunluluk değildir. Gerekli olduğunda aralarına virgül "," konularak istenildiği kadar seçici atanabilir. Bu kullanım şekli **gruplama** olarak adlandırılır.

Örnek

```
h1,h2,h3,h4,h5,h6 { color:green }
```

Örneğimizde tüm XHTML başlık elemanlarını tek bir bildirim bloğuna atayarak hepsinin yeşil renkte olmasını sağladık.

Açıklama Satırı

İstersek CSS kodlamamıza açıklama satırları ekleyebiliriz. Açıklama satırları bize yol göstermesi açısından önemlidir. Sadece tasarımı yapan kişi için değil dosyayı kullanacak herkese (hele de uzun bir CSS dosyası ise) neyin nerede olduğu ve hangi amaçla kullanıldığı konusunda yol göstererek insanların işini kolaylaştıracak ve böylece epey zaman tasarrufu sağlayacaktır.

Kullanımı

```
/* Açıklama satırı */
```

şeklindedir.

Örnek

```
/* Açıklama satırı */
p{
text-align:center;
/*bir başka açıklama satırı*/
color:black;
font-family:arial
}
```

Seçiciler

Seçiciler; CSS'in kalbi ve ruhudur. XHTML etiketlerine stil tanımlamak için kullandığımız bu yapıda, daha önce de değindiğimiz gibi `p`, `h1`, `body`... vb bütün XHTML etiketleri CSS için birer seçicidirler. Şimdi gelin seçicilerin kullanım şekillerini tek tek inceleyelim.

Sınıf Seçicisi (Class Selector)

Bir XHTML etiketine birden fazla özellik kazandırmak istediğimizde ya da bir özelliği birçok XHTML etiketine uygulamak istediğimizde kullanırız. Bir XHTML etiketine birden fazla özellik kazandırmak istediğimiz durumda kullanım:

```
seçici.sınıf_adı { Özellik:Değer; }
```

şeklindedir.

```
<head>
<style type="text/css">
      p.sol {tetx-align:left;}
      p.sag {tetx-align:right;}
</style>
</head>
<body>
      <p class="sag"> Sağa hizalanacak metin.</p>
      <p class="sol"> Sola hizalanacak metin.</p>
</body>
```

Burada p etiketine biz iki özellik atadık. İstersek bu yöntemle daha fazla özellik de atayabiliriz.

Bir özelliği birçok XHTML etiketine uygulamak istediğimizde kullanım:

```
.sol {tetx-align:left;}
```

şeklinde olur.

Bu kullanıma da bir örnek verelim.

```
<head>
<style type="text/css">
      .orta {tetx-align:center;}
</style>
</head>

<body>
      <h1 class="orta">Başlık Metni</h1>
      <p class="orta">Ortalanacak Metin</p>
</body>
```

Örneğimizde özelliğimizi yani head bölümünde tanımladığımız metinleri ortalama özelliğini h1 ve p etiketlerine uyguladık. Dilersek daha fazla etikete de uygulayabiliriz.



Tekil Seçiciler (Id Selector)

Tekil seçicilerin yukarıda gördüğümüz **sınıf** (*class*) seçicilerinden farkı sadece tek bir etikete atanabilmeleridir. Baş tarafına **#** (*diyez*) işareti konularak atanırlar.

Kullanımı

```
#seçici {Özellik:Değer;}
```

şeklindedir. Aşağıdaki örneği inceleyelim.

```
<head>
<style type="text/css">
      #cizgi {text-decoration:underline;}
</style>
</head>
```

```
<body>
      <p id="cizgi">Altı çizilecek önemli bir metin</p>
</body>
```

Seçici Uygulama Hiyerarşisi

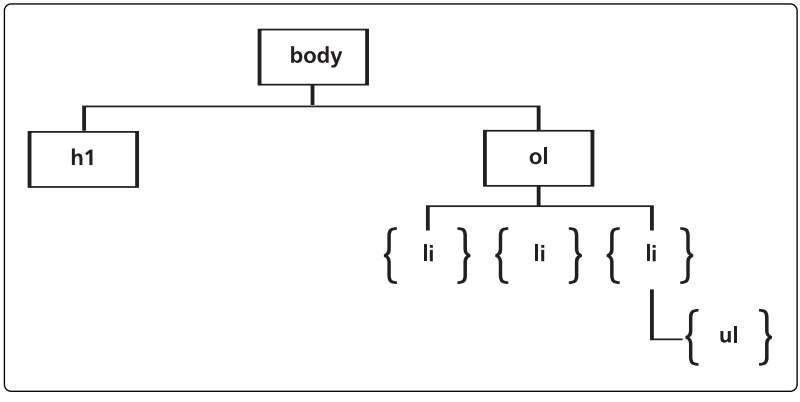


XHTML belge mimarisinin hiyerarşik yapısı, seçicilerimizi oluştururken de işimize çok yarayacak bir yapıdır. Aşağıdaki örnekte XHTML dilinin hiyerarşik yapısını şema üzerinde daha da net olarak görmektesiniz.

```
<body>
<h3>ÖĞRENMEM GEREKENLER</h3>
<ol>
      <li>XHTML:
         <ul>
```

```
            <li>Belge Bildirimi</li>
            <li>Metin Etiketleri</li>
            <li>Listeler</li>
            <li>Tablolar</li>
            <li>Çerçeveler</li>
          </ul>
        </li>
        <li>CSS</li>
        <li>DREAMWEAVER</li>
</ol>

</body>
</html>
```



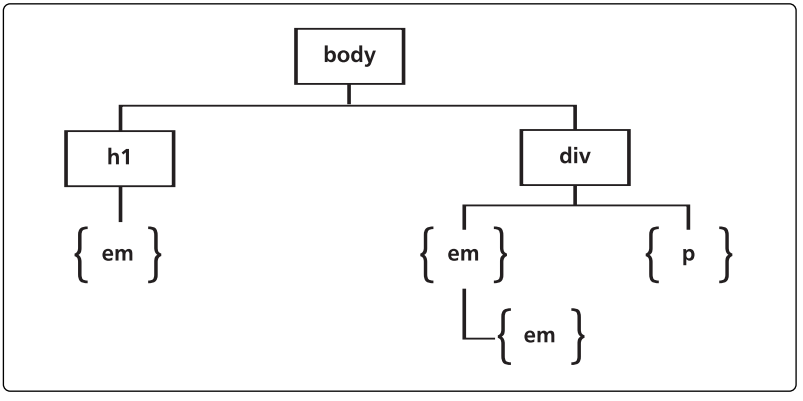
Bu hiyerarşi için gereken tanımlamalar aile soyağacı mantığından yola çıkılarak yapılmıştır. Buna göre yukarıdaki şemadaki `ol` etiketi **ata** (*ancestor*), `li` etiketi **çocuk** (child), onun altındaki `ul` ise **torun** (*descendant*) seçici olmaktadır. Ayrıca yan yana olan `li` etiketleri de **Bitişik Kardeş Seçiciler** (*Next Sibling Selector*) olmaktadırlar.

Çocuk Seçicileri (Direct Child Selectors)

Bir etiket (örn: `em`) bir belge ağacında bir çok kez kullanılmış olabilir. İşte bu etiketlerin bazılarına farklı özellikler atamak istediğimizde hepsiyle tek tek uğraşmak çok zahmetli olacaktır. Bu durum hiyerarşide

aralarında bir basamak fark olan seçiciler için söz konusu olduğunda **Çocuk Seçiciler** imdadımıza yetişmektedir.

```
<body>
    <h1>Başlık <em>metni</em></h1>
    <div>
        Açılama <em>metni</em>
        <p>Paragraf <em>metni</em></p>
    </div>
</body>
```



Şemadan da anlaşılacağı gibi biz stilimizi sadece `div` etiketinin bir altındaki etikete, yani `div` etiketine göre **çocuk** olan `em` etiketine uygulayacağız. İki alt seviyede olan etikete (*torun*) uygulamayacağız.

Bu durumda kodumuz:

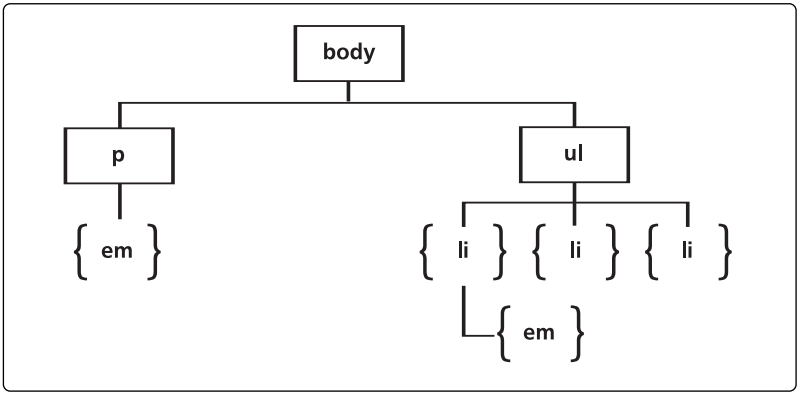
```
div>em {color: blue;}
```

şeklinde olacaktır. Buradaki **>** (büyüktür) işareti bu seçiciye has bir kullanımdır.

Torun Seçicileri (Descendant Selectors)

Çocuk seçicileri anlatırken söylediğimiz olay, hiyerarşide aralarında iki basamak fark olan seçiciler için söz konusu olduğunda **Torun Seçiciler** diye tabir edilen yapıyı kullanıyoruz.

```
<body>
<p>Lorem ipsum dolor <em>sit</em> amet.</p>
<ul>
<li>öge 1</li>
<li>öge 2</li>
<li><em>öge 3</em></li>
</ul>
</body>
```



Yukarıdaki örnek XHTML belgemize stil verirken eğer kodumuzu:

```
em {color: blue;}
```

şeklinde yazarsak atayacağımız değer, tüm `em` etiketlerine uygulanır. Fakat biz değerimizi sadece şemada büyük puntolarla gösterdiğimiz `em` etiketine uygulamak istiyorsak ne yapacağız? Tahmin ettiğiniz gibi **torun seçici** olayını devreye sokacağız. Yani belgenin hiyerarşik yapısından yararlanarak bu isteğimizi kolayca gerçekleştireceğiz.

Bu takdirde kodumuz:

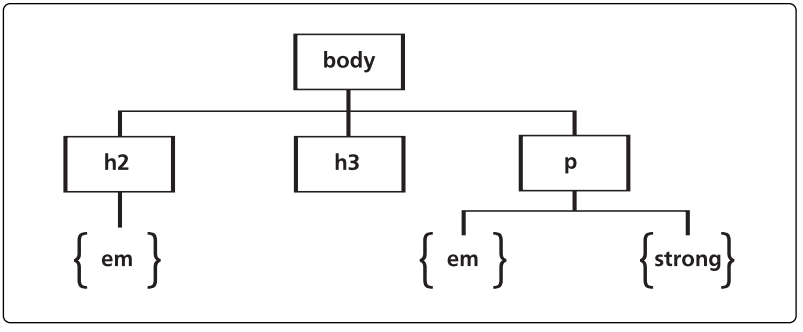
```
ul em {color: blue;}
```

şeklinde olacaktır. Bu kullanımın anlamı; stil atamamızın sadece `ul` etiketinin altındaki, yani sadece `ul` etiketiyle akraba olan, bir başka ifade ile `ul` etiketinin torunu olan `em` etiketine uygulanacağı, diğerine ise uygulanmayacağıdır.

Bitişik Kardeş Seçiciler (Adjacent Sibling Selector)

Belge hiyerarşisinde aynı katmandaki seçiciler arasındaki ilişkiyi ifade eden terimdir.

```
<body>
<h2>Başlık2 <em>Metni</em></h2>
<h3>Başlık3 Metni</h3>
<p>Okuduğunuz bu <em>metin</em> örnekleme <strong>amaçlı-
dır</strong></p>
</body>
```



Şemadan kolaylıkla anlaşıldığı gibi `h2` ile `h3` ve `em` ile `strong` etiketleri aynı seviyede olduklarından kardeş etikettirler. Buna göre örnek bir kullanımı inceleyelim.

```
h2 + h3 {margin: 2em;}
```

şeklindeki kullanımda `em` ve `strong` **kardeş seçici** durumunda olduklarından, aralarına + (artı) işareti konularak kullanılır.

Evrensel Seçiciler (Universal Selector)

Tanımlanacak stilin, belgenin tamamına ya da bir seçicinin içindeki tüm öğelere uygulanmak istendiği durumlarda kullanılır.

```
* {Özellik:Değer;}
```

şeklinde kullanılır.

Başındaki * (yıldız) işareti özelliğe uygulanan değerin tüm belgede geçerli olacağı anlamına gelmektedir.

```
<body>
<h2>Başlık Metni</h2>
<h3>Altbaşlık Metni</h3>
<p>Lorem ipsum dolor sit amet, <i>consectetuer adipiscing</i> elit.
 <strong> Vestibulum convallis </strong> sapien eget dolor. Duis
eget      massa et nisi blandit sagittis. </p>
</body>
```

Yukarıdaki gibi bir XHTML belgede örneğin;

```
* {text-decoration:underline;}
```

şeklindeki bir kullanımda, aşağıdaki resimde de gördüğünüz gibi, uyguladığımız özellik belgenin tamamında geçerli olmaktadır.



Özelliği tüm belgeye değil de, bir seçiciye uygulayacağımızda ise kullanımı:

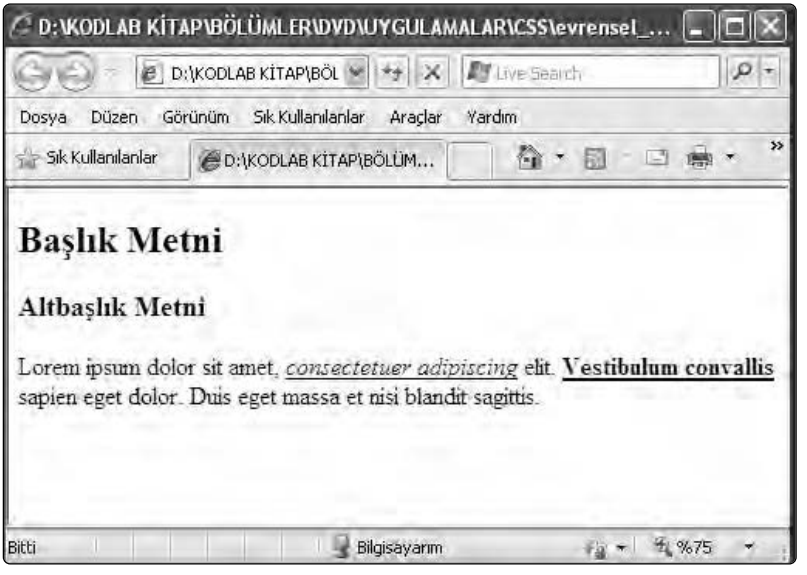
```
seçici * {Özellik:Değer;}
```

seklinde olmaktadır.

Buna göre, bir örnek yapalım:

```
p*{text-decoration:underline;}
```

şeklindeki kod, sadece belgedeki p seçicisinin içindeki tüm öğelere uygulanacaktır.



Öznitelik Seçicileri (Attribute Selectors)

XHTML etiketleri ile birlikte kullanılan özniteliklere göre stil ataması yapmamızı sağlayan yöntemdir. **Basit Öznitelik Seçicisi**, **Öznitelik Değeri Seçicisi**, **Kısmi Öznitelik Değeri Seçicisi**, **Ayrımlı Öznitelik Değeri Seçicisi** olmak üzere dört tür öznitelik seçicisi mevcuttur.

Basit Öznitelik Seçicisi

Basit Öznitelik Seçicisi, bir etikete ait özniteliğe göre stil ataması yapmamızı sağlar.

Kullanımı

```
seçici[öznitelik]{Özellik:Değer;}
```

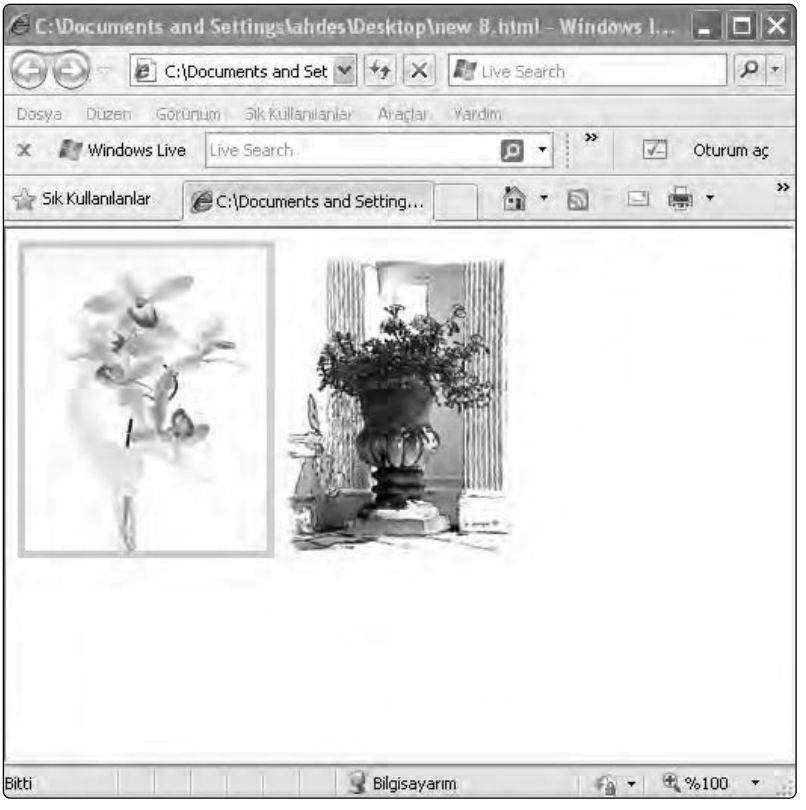
şeklindedir.

Konuyu aşağıdaki örnek ile pekiştirelim.

```
<head>
<style type="text/css">
img[alt]{border:5px;}
</style>
</head>

<body>
<img src="resim1.jpg" alt="ilk resim" width="160" height="200" />
<img src="resim2.jpg" width="160" height="200" />
</body>
```

şeklindeki bir uygulamada `border` değeri sadece `alt` özniteliğine sahip resme uygulanacaktır.



Öznitelik Değeri Seçicisi

Bir etikete ait öznitelik değerine stil ataması yapmamızı sağlar.

Kullanımı

```
seçici[öznitelik="öznitelik değeri"]{Özellik:Değer;}
```

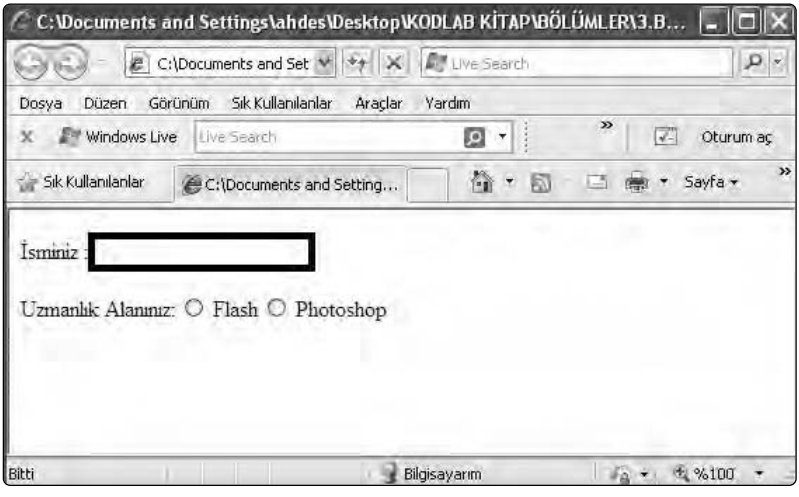
şeklindedir.

Değişik öznitelik değerlerine sahip bir uygulamada (aşağıdaki örneğimizde bir form) biz stilimizi sadece bir öznitelik değerine uygulamak isteyebiliriz. Örneğimizde formumuzun elemanlarının her biri farklı öznitelik değerine (örn: `type`, `radio`) sahip olduğundan ve radyo düğmelerine çerçeve değeri uygulamak istemediğimizden bir önceki tür olan **basit öznitelik seçicisi**'ni burada kullanamazdık. Bu tip durumlar

için **Öznitelik Değeri Seçicisi**'ni kullanarak öznitelik değerleri arasında ayrım yapabiliyoruz.

```
<head>
<style type="text/css">
input[type="text"] {border: 5px solid #000; width: 150px;}
</style>
</head>

<body>
<p>İsminiz :<input type="text" name="ismi" /></p>
<p>Uzmanlık Alanınız: <input type="radio" name="uzman" ↓
value="Flash" /> Flash <input type="radio" name="uzman" ↓
value="Photoshop" /> Photoshop </p>
</body>
```



Kısmi Öznitelik Değeri Seçicisi

Şimdi işi biraz daha abartalım! Stilimizi bu sefer de öznitelik değerinin sadece bir parçasına uygulayalım. Biraz dıdısının dıdısı gibi oldu! İşte CSS'in gücü. Bize çok detaylı düzenlemeler yapma imkanı veriyor. Bu türün kullanım şekli;

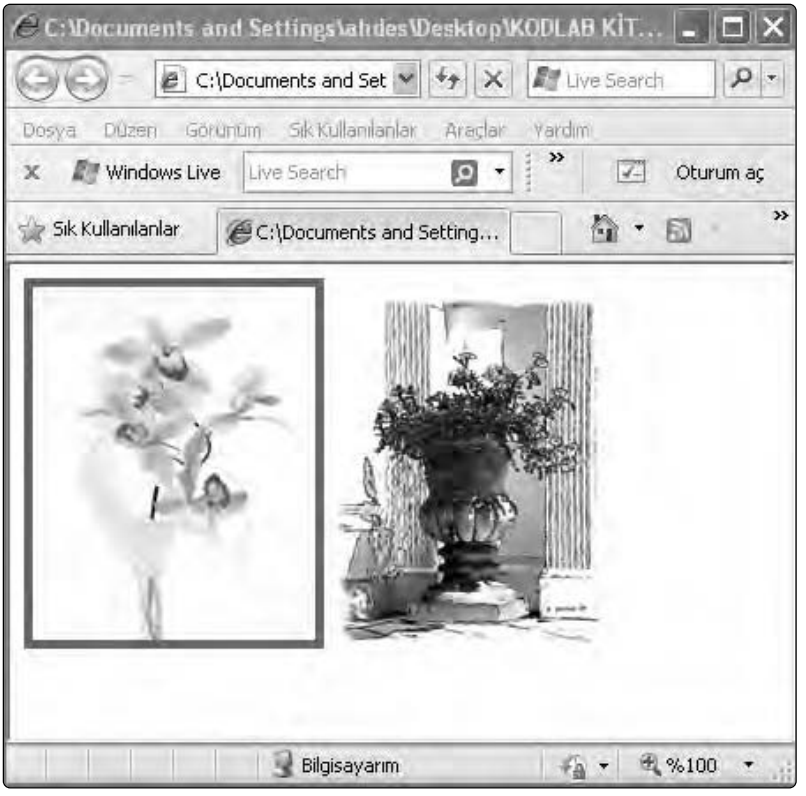
```
seçici[öznitelik ~="öznitelik değerinin bir bölümü"]{Özellik:Değer;}
```

şeklindedir.

Bir örnek de bu tür için yapalım:

```
<head>
<style type="text/css">
img[alt~="ilk"] {border: 5px;}
</style>
</head>

<body>
<img src="resim1.jpg" alt="ilk resim" width="160" height="200" />
<img src="resim2.jpg" alt="son resim" width="160" height="200" />
</body>
```



Ayrımlı Öznitelik Değeri Seçicisi

Bu yöntem - (tire) işareti ile ayrılmış kısımları bulunan öznitelik değerlerinin bir kısmını kullanmamıza olanak tanır.

Kullanımı:

```
seçici[öznitelik | ="öznitelik değerinin bir bölümü"]{Özellik:Değer;}
```

şeklindedir.

Artalan (Background) Biçimlendirme

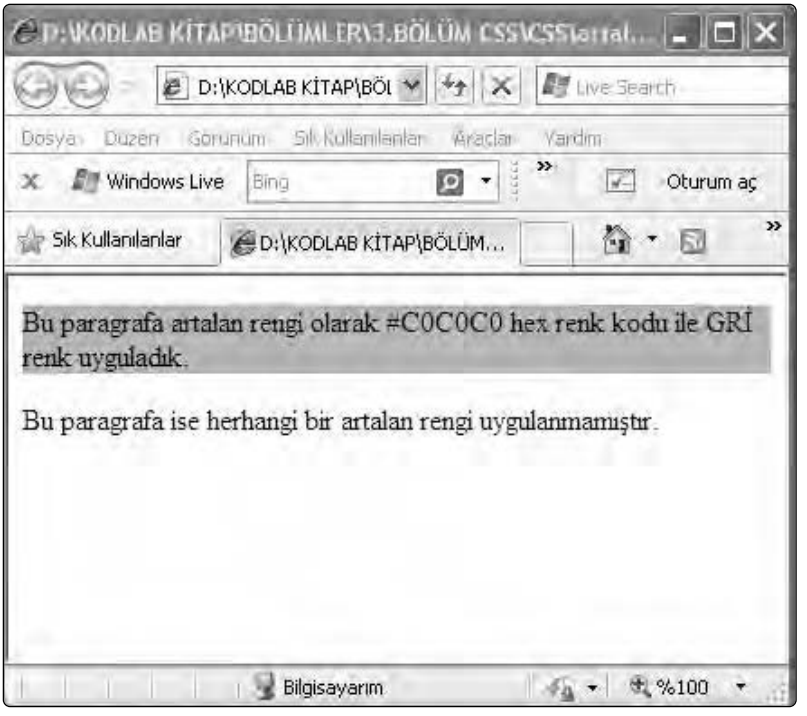
Artalan özelliklerini (artalan rengi, artalan resmi... vd) biçimlendirmek için kullanılır.

Artalan Rengi (background-color)

Öğelerin artalanlarına renk atamak için kullanılır. Değer olarak istediğiniz rengi, ayrıca da `transparent` değerini atayabilirsiniz.

```
<head>
<style type="text/css">
   p.artalan{background-color: #C0C0C0;}
</style>
</head>

<body>
<p class="artalan">Bu paragrafa artalan rengi olarak ↓
#C0C0C0 hex renk kodu ile GRİ renk uyguladık.</p>
<p>Bu paragrafa ise herhangi bir artalan rengi uygulanmamıştır. ↓
</p>
</body>
```

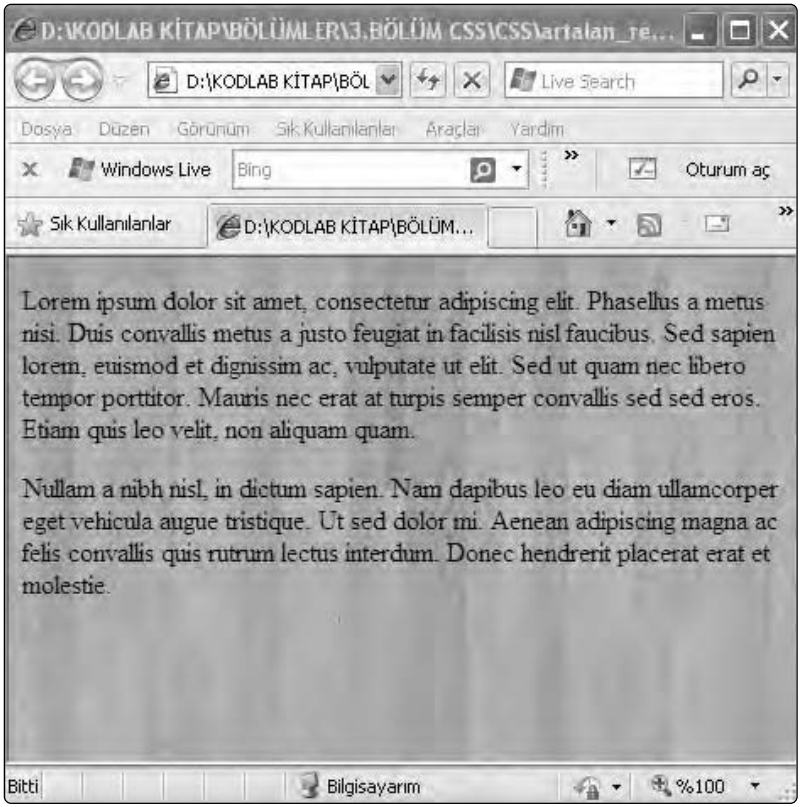


Artalan Resmi (background-image)

Artalana resim atamak için kullanılır.

```
<head>
<style type="text/css">
body {background-image:url('artalan.jpg')}
</style>
</head>

<body>
<p>Artalan olarak bir resim kullandık.</p>
</body>
```



Artalan Resim Yenileme (background-repeat)

Artalan resminin yineleme ayarlamaları için kullanılır. Bu özelliği kullanmadığımızda artalan resmi hem yatay hem de dikey olarak tekrarlar ve tüm artalanı kaplar. İşte bu özellikle bu standart kullanımdan farklı biçimlendirmeler yapılabilmektedir.

Aldığı değerler

`repeat`: Resim hem yatay hem de dikey olarak yinelenir.

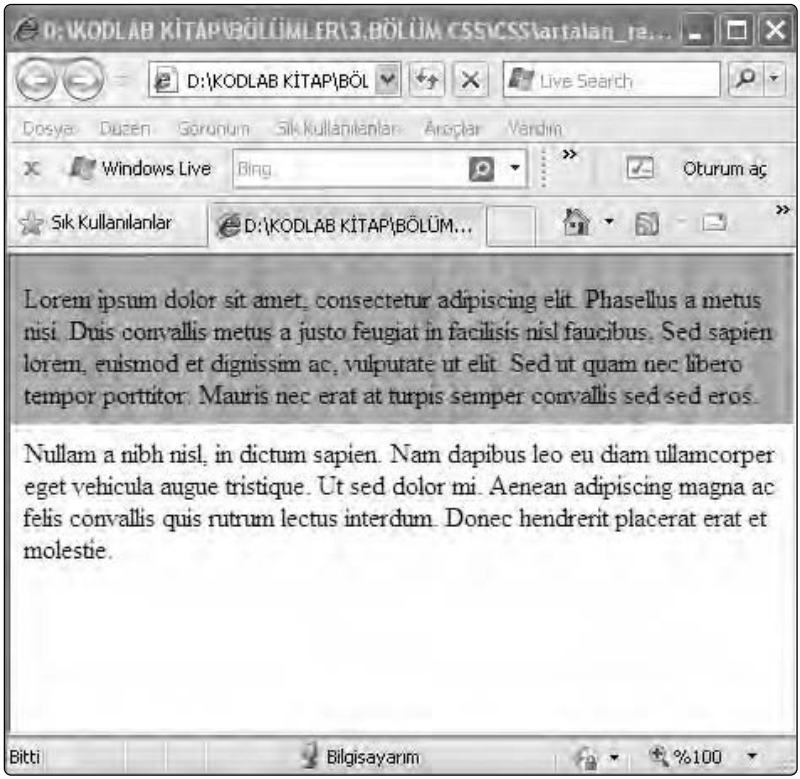
`repeat-x`: Resim yalnız yatay düzlemde yinelenir.

`repeat-y`: Resim yalnız dikey düzlemde yinelenir.

`no-repeat`: Resim yinelenmez.

```
<head>
<style type="text/css">
{
background-image:url('artalan.jpg');
background-repeat:repeat-x;
}
</style>
</head>

<body>
<p>Artalan resmini sadece yatay düzlemde tekrar edecek şekilde
kullandık.</p>
</body>
```



Artalan Resim Sabitleme (background-attachment)

Bu özellik; `scroll` değeriyle kullanıldığında artalanın sayfa ile beraber kaymasını; `fixed` değeri ile kullanıldığında ise artalanın sabit kalmasını sağlar.

```
<head>
<style type="text/css">
body
{
background-image:url('artalan.jpg');
background-repeat:no-repeat;
background-attachment:fixed
}
</style>
</head>

<body>
<p>Artalan resmini sabit kalacak şekilde kullandık.</p>
</body>
```

Artalan Resmini Konumlandırma (background-position)

Artalan resminin konumunu belirlemek için kullanılır.

Aldığı değerler

Yatay hizalama için: `left`, `center`, `right`

Dikey hizalama için: `top`, `center`, `bottom`

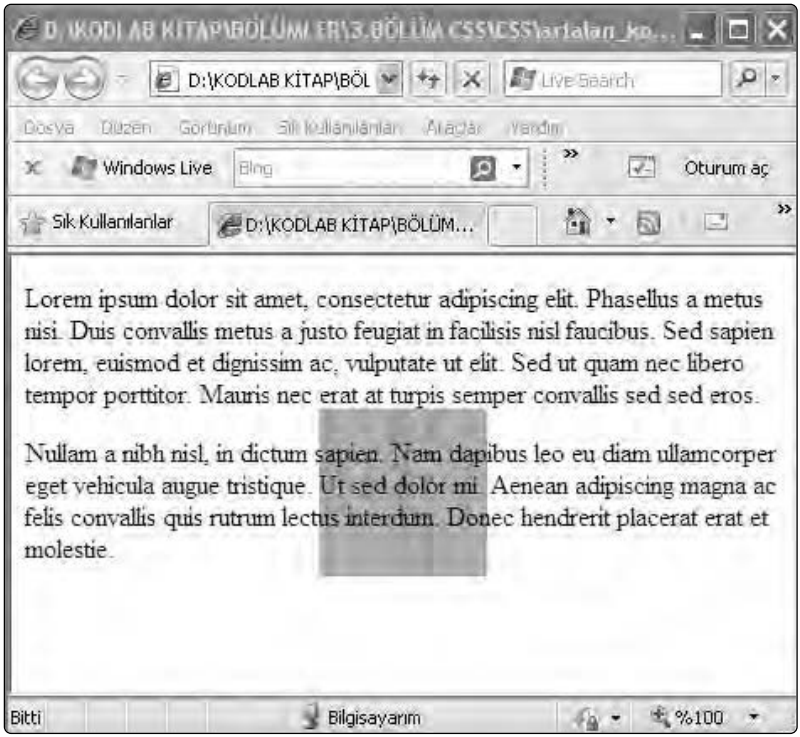
Bu değerler `top left`, `top center`, `top right`, `center left`, `center center`, `center right`, `bottom left`, `bottom center`, `bottom right` şeklinde birlikte de kullanılabilir.

Bu değerlerin yerine yüzde (`%`) olarak da değer belirtilebilir. Varsayılan değer `0% 0%` olmaktadır. Bu kullanım `top left` şeklindeki kullanıma karşılık gelmektedir. `100% 100%` değeri ise `bottom right` değerine karşılık gelir. Hem yatay hem dikey olarak ortalamak için kullanacağınız değer `50% 50%` değeridir. Bu da `center center` kullanımına karşılık gelecektir.

Gerek `top left` seklindeki kullanımda olsun gerekse `0% 0%` şeklindeki kullanımda olsun ilk değer yatay hizalama, ikinci değer ise dikey hizalama içindir. Eğer atamanızda sadece tek bir değer kullanırsanız ikinci değer varsayılan olarak `center` değeri olacaktır.

```
<head>
<style type="text/css">
body
{
background-image:url('artalan.jpg');
background-repeat:no-repeat;
background-attachment:fixed;
background-position:center;
}
</style>
</head>

<body>
<p>Artalan resmini ortalı şekilde kullandık.</p>
</body>
```



ARTALAN (BACKGROUND)

Buraya kadar anlattığımız artalan özelliklerini bir arada kullanabilmemize olanak tanır. Bu özellikle beraber; `background-color`, `background-image`, `background-repeat`, `backgroundattachment`, `background-position` değerleri kullanılır.

```
<head>
<style type="text/css">
    body { background: #00ff00 url('artalan.jpg') no-repeat
fixed center; }
</style>
</head>

<body>
<p>Tüm artalan özelliklerini bir arada kullandık.</p>
</body>
```

Yazıtipi (Font) Biçimlendirme

Bu konunun kapsamına; metinlerimizde kullanacağımız yazı-tiplerinin (*font*) büyüklük, kalınlık, biçem... vb temel font işlemleri girmektedir.

Yazıtipi Ailesi (font-family)

Oluşturacağımız metinde kullanacağımız yazı tipi grubunu belirlemek için kullanılır. Birden fazla yazı-tipi atanacaksa aralarına **,** (virgül) konulur. Bu kullanım tarzı yaygındır, çünkü sizin belirleyeceğiniz yazı-tipi kullanıcının bilgisayarında olmayabilir. Dolayısıyla mantıklı olan da birkaç tane font tanımlamak olacaktır. İlk sıradaki ilk tercihimizdir. Bilgisayar ilk önce bu fontu dener. Eğer onu çalıştıramazsa sırayla diğerlerini dener. İstemcinin bilgisayarında hiçbirini bulamazsa kullanılan tarayıcının varsayılan fontunu kullanır.



Kullanacağımız yazı-tipinin adı birden çok kelimeden oluşuyorsa (örn: Times New Roman) atama yaparken **""** (çift tırnak) kullanılır.

```
<html>
<head>
<style type="text/css">
  p.serif{font-family:"Times New Roman",Georgia,Serif}
  p.sansserif{font-family:Arial,Verdana,Sans-serif}
</style>
</head>

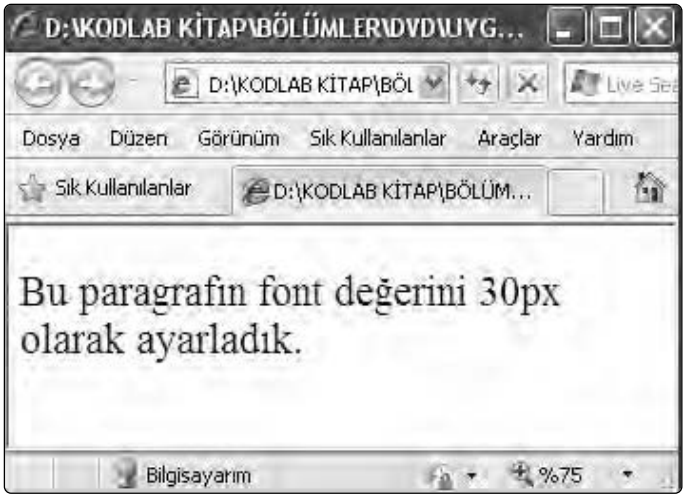
<body>
<h1>CSS font-family kullanımı.</h1>
<p class="serif">Bu paragraf Times New Roman fontu ile
yazılmıştır.</p>
<p class="sansserif">Bu paragraf Arial fontu ile yazılmıştır.</p>
</body>
</html>
```



Yazıtipi Boyutu (font-size)

Yazı-tipi boyutunu belirlemek için kullanılır.

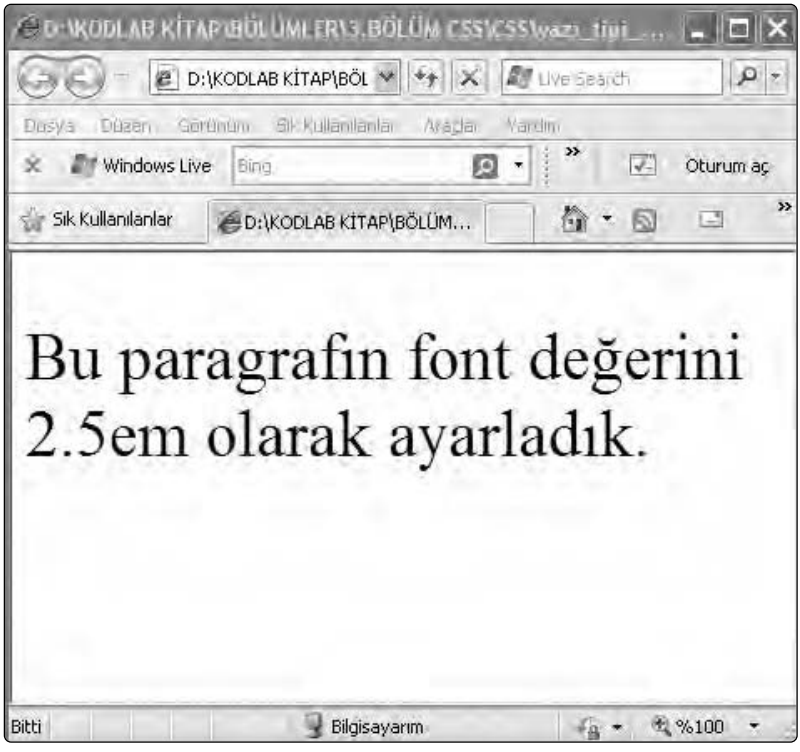
```
<html>
<head>
<style type="text/css">
p {font-size:30px}
</style>
</head>
<body>
<p>Bu paragrafın font değerini 30px olarak ayarladık.</p>
</body>
</html>
```



Yukarıdaki kullanım (px kullanımı) **Firefox**, **Chrome** ve **Safari** tarayıcılarında metni yeniden boyutlandırmanıza izin verse de (sitenize koyacağınız yazı boyutunu değiştirmeye yarayan butonlarla) **Internet Explorer** tarayıcısında bu mümkün olmayacaktır. Bu sakıncalı durumdan kurtulmak için **W3C** tarafından px yerine em kullanmanız önerilmektedir.

1 em, tarayıcıların varsayılan ölçüsüne eşittir. Tarayıcılar genellikle varsayılan olarak 16px kullandıklarından 1em=16px olmaktadır.

```
<html>
<head>
<style>
    p {font-size:2.5em} /* 40px/16=2.5em */
</style>
</head>
<body>
<p>Bu paragrafın font değerini 2.5em olarak ayarladık.</p>
</body>
</html>
```



Yukarıdaki gibi bir kullanımda ise; bir metni yeniden boyutlandırmak istediğinizde font, **IE**'de ayarlamak istediğinizden daha büyük, küçültmek istediğinizde de istediğinizden daha küçük olacaktır. Bu durumu aşmak için yapılması gereken, `body` etiketine yüzde olarak bir varsayılan font değeri atamak olacaktır.

```
<head>
<style type="text/css">
body {font-size:100%}
p {font-size:2.5em} </style>
</head>

<body>
<p>body etiketine yüzde olarak bir varsayılan font değeri
atadık</p>
```

```
</body>
</html>
```

Yukarıdaki kullanım en optimum kullanım şekli olmaktadır. Bütün tarayıcılar font büyüklüğünü aynı gösterirler. Ayrıca tüm tarayıcılar bu kullanım tarzı sayesinde bize, metni yakınlaştırma veya metnin font büyüklüğünü yeniden ayarlama izin vereceklerdir.

YAZITİPİ KALINLIĞI (FONT-WEIGHT)

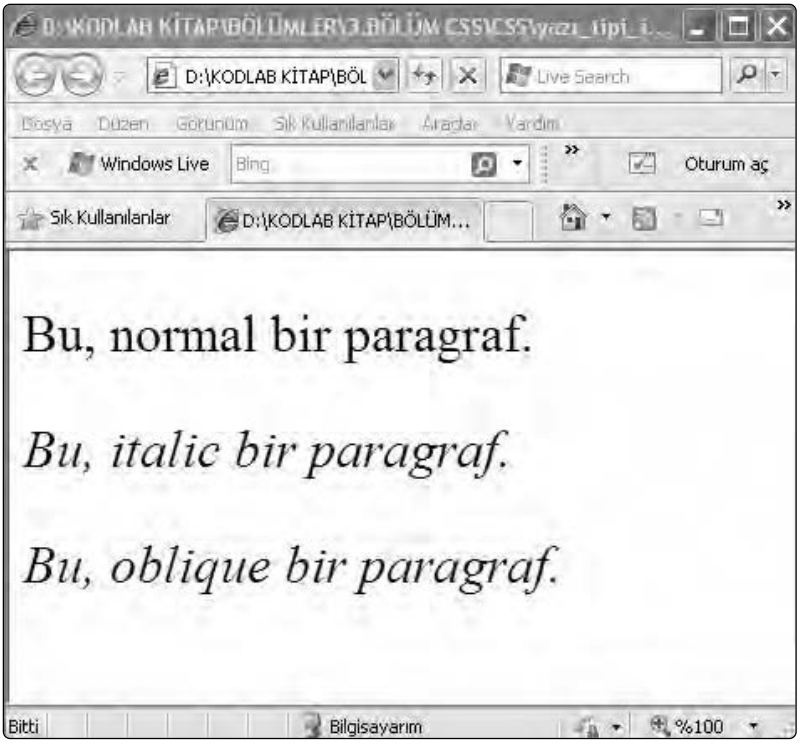
Yazı-tipi kalınlığını belirlemek için kullanılır. Değer olarak, `normal`, `bold` ve `bolder` kullanılır. Ayrıca inceden kalına doğru `100` ile `900` arasında bir değer de alabilir.

```
<style type="text/css">
p { font-weight: bold;}
</style>
<body>
<p>Bu paragrafa bold değeri atanmıştır.</p>
</body>
```

YAZITİPİ STİLİ (FONT-STYLE)

Fontun **yatık** (*italik*) olup olmayacağını belirlemek için kullanılır. Değer olarak `normal`, `italic` ve `oblique` kullanılır.

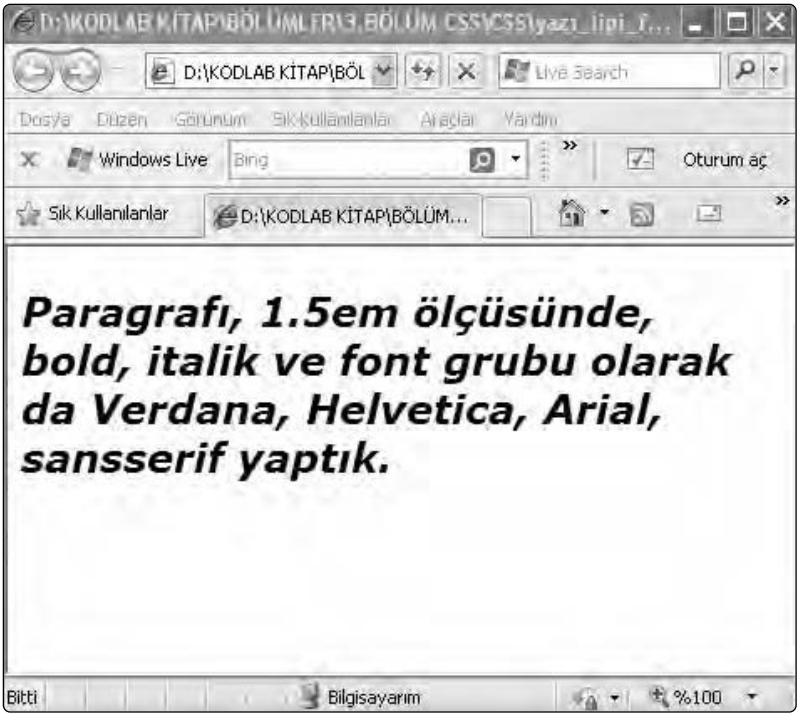
```
<html>
<head>
<style type="text/css">
  p {font-size:2em}
  p.normal {font-style:normal}
  p.italic {font-style:italic}
  p.oblique {font-style:oblique}
</style>
</head>
<body>
<p class="normal">Bu, normal bir paragraf.</p>
<p class="italic">Bu, italic bir paragraf.</p>
<p class="oblique">Bu, oblique bir paragraf.</p>
</body>
</html>
```



Yazıtipi (font)

Tüm yazı-tipi özelliklerini bir arada tanımlamak için kullanılır.

```
<head>
<style type="text/css">
   p {font:bold italic 1.5em Verdana, Helvetica, Arial,
sansserif;}
</style>
</head>
<body>
<p>Paragrafı, 1.5em ölçüsünde, bold, italik ve font grubu
olarak da Verdana, Helvetica, Arial, sansserif yaptık.</p>
</body>
```



METİN (TEXT) BİÇİMLENDİRME

Metinlerin genel düzenleri ve biçimlendirilmeleri ile ilgili işlemleri kapsamaktadır.

RENKLER (COLORS)

Metinlerin rengini belirlemek için kullanırız. Rengin **adı** (örn: *red*), **RGB** değeri –örn: rgb(255,0,0)– ve **hex** (*hexadecimal*) değeri (örn: `#ff0000`) olmak üzere üç değişik değer türü kullanabiliriz.

Sözünü ettiğimiz renk sistemlerini de tanıtalım. **RGB** renk sistemi, İngilizce'deki 'Red' 'Green' 'Blue' ('Kırmızı' 'Yeşil' 'Mavi') kelimelerinin baş harflerinden oluşturulmuş, web'de kullanılan renk standardıdır. Bu standardı genellikle **hexadecimal** (*hex*) sistemde yazarız. Verilebilecek en küçük değer `0` (hex `00`), en büyük değer ise `f`'dir (hex `FF`). Bu şekilde **hexadecimal** (*hex*) sistemde `16 milyon` renk üretmek mümkündür.

Her ne kadar teorik olarak 16 milyon renk üretmek mümkünse de, günümüz modern monitörleri ortalama 16 bin farklı renk gösterme kapasitesine sahiptirler.

Bir de, 16 adet rengi adları ile kullanabileceğinizi belirtelim. Bunlar; aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, ve yellow renkleridir.

RENKHex Değeri	RGB Değeri
SİYAH#000000	rgb(0,0,0)
KIRMIZI#FF0000	rgb(255,0,0)
YEŞİL#00FF00	rgb(0,255,0)
MAVİ#0000FF	rgb(0,0,255)
SARI#FFFF00	rgb(255,255,0)
GRİ#C0C0C0	rgb(192,192,192)
BEYAZ#FFFFFF	rgb(255,255,255)

Örnek bir renk tablosu.

```
<style type="text/css">
h1 { color:red /*rengimizin adı */}
h2 { color: rgb(255,0,0) /*rengimizin rgb değeri */}
p { color: #F00000 /*rengimizin kodu */}
</style>

<body>
<h1>Ana Başlık</h1>
<h2>Alt Başlık</h2>
<p>Renklendirilecek Paragraf</p>
</body>
```

Metin Hizalama (text-alignment)

Metinleri hizalamak için kullanılır.

left: Metni **sola dayalı** olacak şekilde hizalar.

right: Metni **sağa dayalı** olacak şekilde hizalar.

center: Metni **ortalı** olacak şekilde hizalar.

justify: Metni **iki yana yaslı** olacak şekilde hizalar.

```
<head>
<style type="text/css">
  h1 {text-align:center}
```

```
    p.tarih {text-align:right}
    p.paragraf{text-align:justify}
</style>
</head>

<body>
<h1>Metin Hizalama</h1>
<p class="tarih">Eylül, 2009</p>
<p class="paragraf">left: Metni sola dayalı olacak şekilde
hizalar. right: Metni sağa dayalı olacak şekilde hizalar.
center: Metni ortalı olacak şekilde hizalar. justify: Metni
iki yana yaslı olacak şekilde hizalar.</p>
</body>
```



Metin İşaretleme (text-decoration)

Metinleri çeşitli sebeplerle vurgulamak için işaretlemede kullanılır.

overline: Metnin üst kısmını çizer.

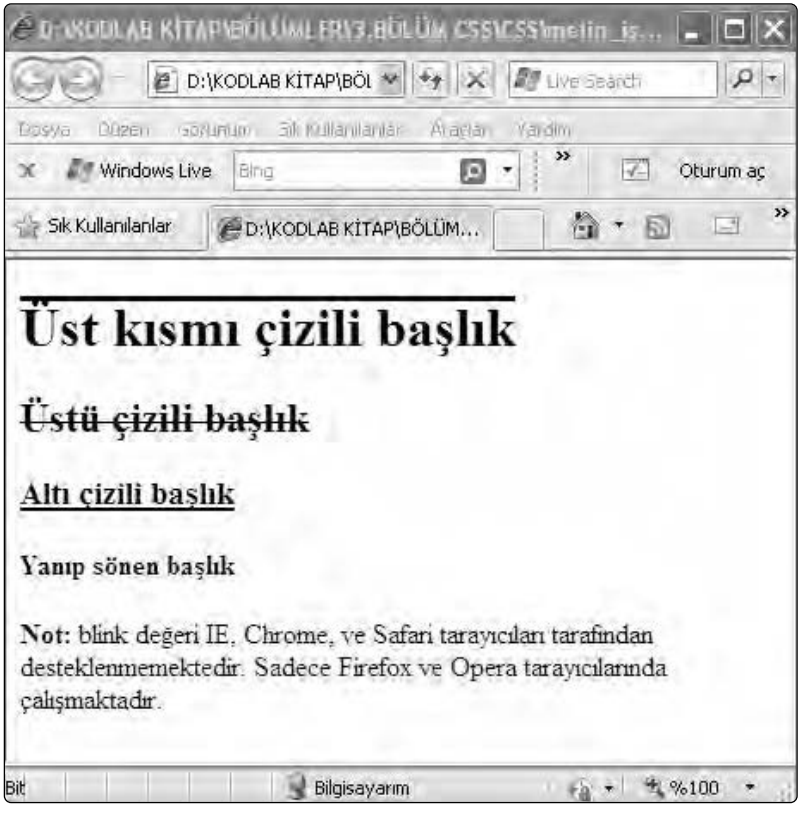
line-through: Metnin üzerini çizer.

underline: Metnin altını çizer.

blink: Metnin yanıp sönmesini sağlar.

```
<head>
<style type="text/css">
h1 {text-decoration:overline}
h2 {text-decoration:line-through}
h3 {text-decoration:underline}
h4 {text-decoration:blink}
</style>
</head>

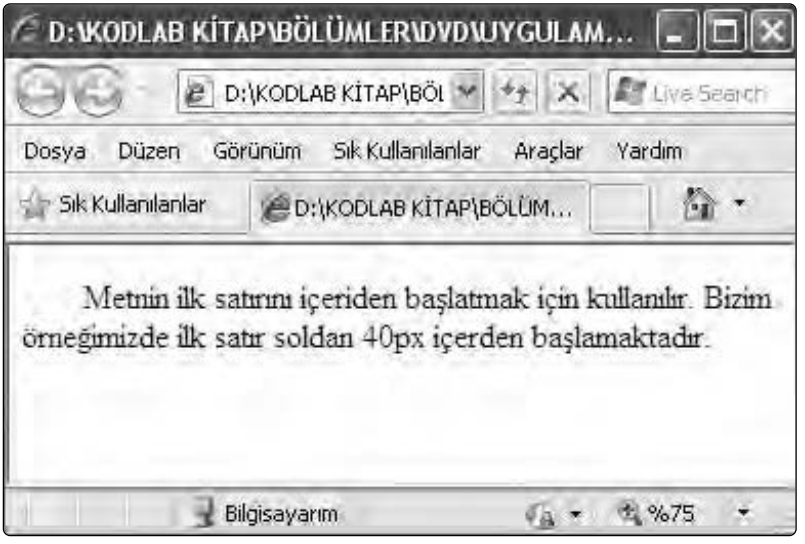
<body>
<h1>Üst kısmı çizili başlık</h1>
<h2>Üstü çizili başlık</h2>
<h3>Altı çizili başlık</h3>
<h4>Yanıp sönen başlık</h4>
<p><b>Not:</b> blink değeri IE, Chrome, ve Safari tarayıcıları
tarafından desteklenmemektedir. Sadece Firefox ve Opera
tarayıcılarında çalışmaktadır.</p>
</body>
```



METİN GİRİNTİLEME (TEXT-INDENTATION)

Metnin ilk satırını içerden başlatmak amacıyla kullanılır.

```
<head>
<style type="text/css">
  p {text-indent:40px}
</style>
</head>
<body>
<p>Metnin ilk satırını içeriden başlatmak için kullanılır. Bizim
örneğimizde ilk satır soldan 40px içerden başlamaktadır.</p>
</body>
```



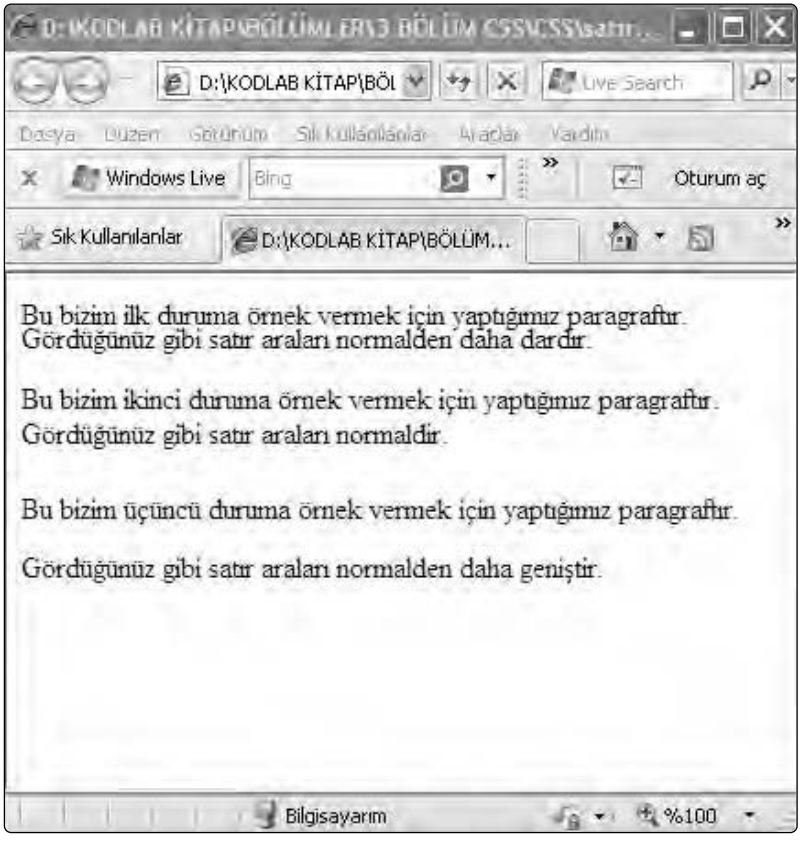
Satırlar Arası Yükseklik (line-height)

Metnin satırları arasındaki mesafeyi belirlemek için kullanılır. **Normal**, **piksel** (px) veya **yüzdelik** (%) değer alabilir.

```
<head>
<style type="text/css">
   p.dar {line-height: 80%}
   p.normal {line-height:normal}
   p.genis {line-height:200%}
</style>
</head>

<body>
<p class="dar">
Bu bizim ilk duruma örnek vermek için yaptığımız paragraftır.
Gördüğünüz gibi satır araları normalden daha dardır.
</p>
<p class="normal">
Bu bizim ikinci duruma örnek vermek için yaptığımız paragraftır.
Gördüğünüz gibi satır araları normaldir.
</p>
<p class="genis">
```

```
Bu bizim üçüncü duruma örnek vermek için yaptığımız paragraftır.
Gördüğünüz gibi satır araları normalden daha geniştir.
</p>
</body>
```



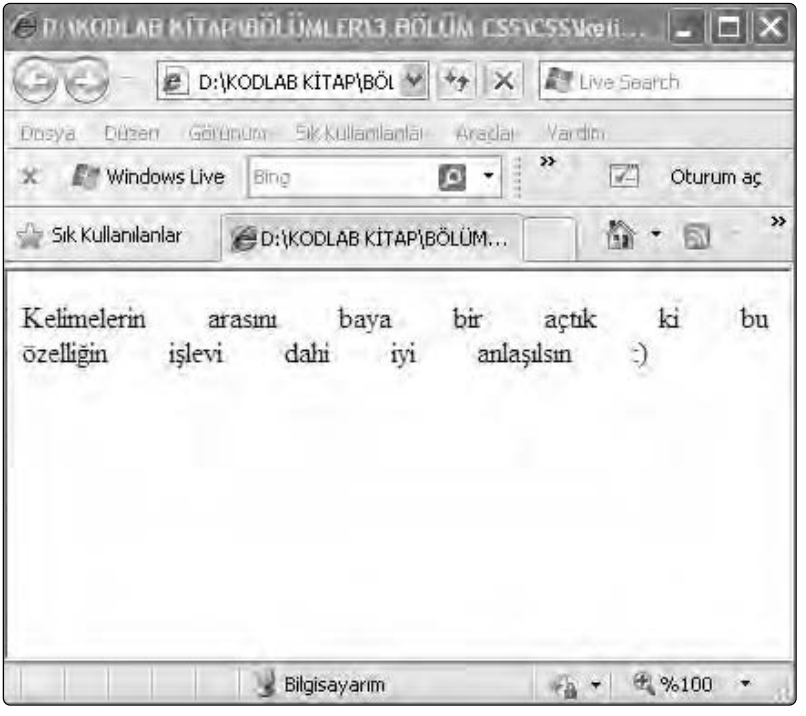
Sözcükler Arası Mesafe (word-spacing)

Metindeki sözcüklerin birbirleri arasındaki mesafeyi ayarlamak istediğimizde kullanırız.

```
<head>
<style type="text/css">
   p {word-spacing:30px;}
```

```
</style>
</head>

<body>
<p> Kelimelerin arasını baya bir açtık ki bu özelliğin işlevi
dahi iyi anlaşılsın :)
</p>
</body>
```



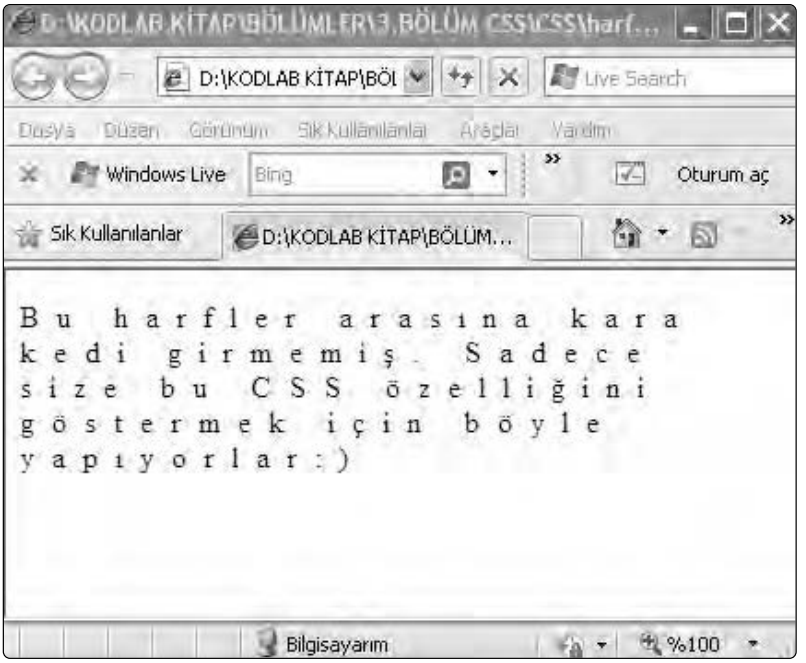
Harfler Arası Mesafe (letter-spacing)

Evet bu kadar da olmaz demeyin lütfen! CSS'in bize çok detaylı biçimlendirme imkanları sunduğunu söylemiştim. Ayrıca hiç ummadığınız bir anda bu özellik de işinize kesinlikle yarayacaktır.

```
<html>
<head>
<style type="text/css">
```

```
        p{letter-spacing: 10px;}
</style>
</head>

<body>
<p>
Bu harfler arasına kara kedi girmemiş. Sadece size bu CSS
özelliğini göstermek için böyle yapıyorlar:)
</p>
</body>
</html>
```



HARF BOYUTLARI (TEXT TRANSFORMATION)

Metindeki büyük ve küçük harfleri birbirlerine çevirmek için kullanılır.

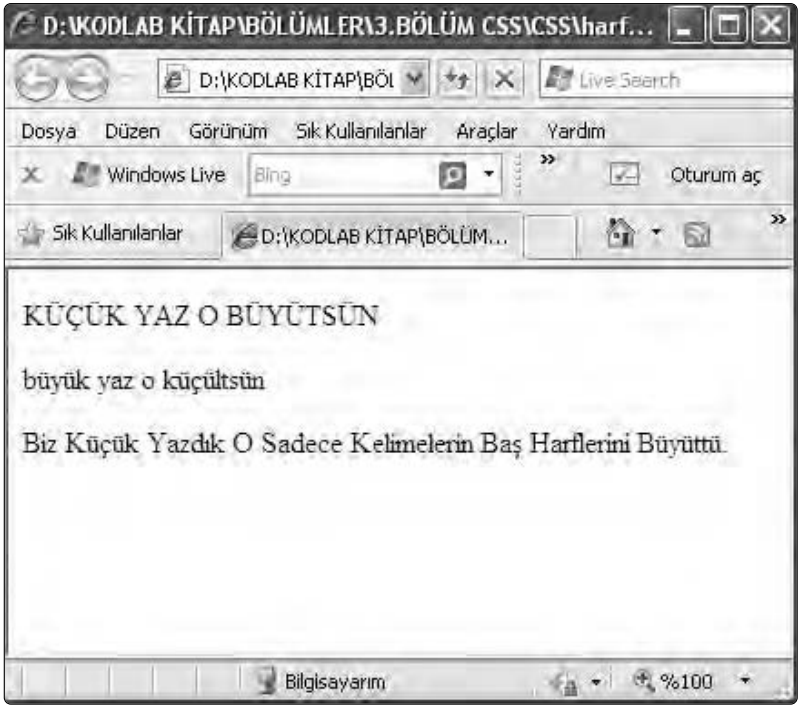
uppercase: Küçük harfleri büyük harfe çevirir.

lowercase: Büyük harfleri küçük harfe çevirir.

capitalize: Sadece kelimelerin baş harflerini büyütür.

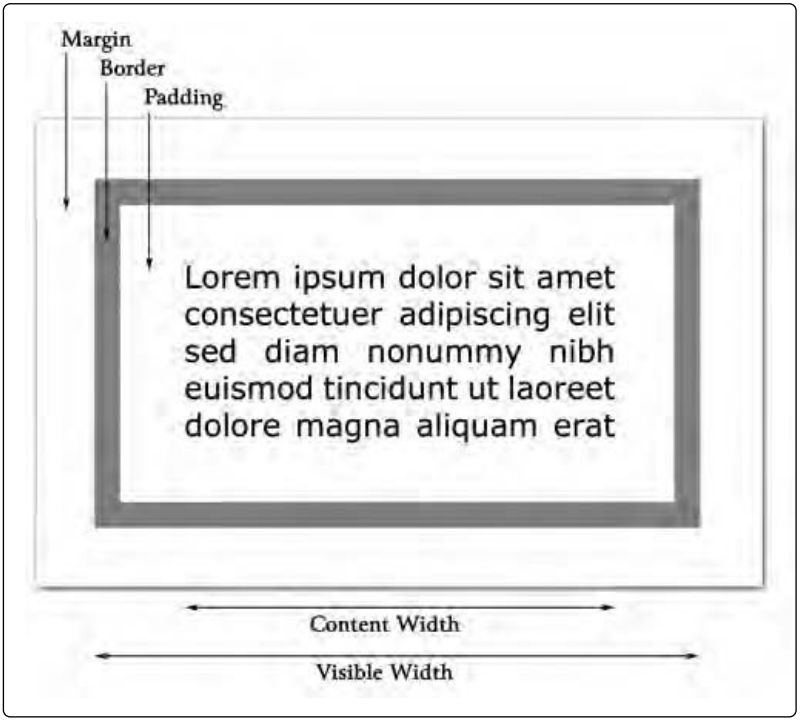
```
<html>
<head>
<style type="text/css">
    p.buyuk {text-transform:uppercase}
    p.kucuk {text-transform:lowercase}
    p.basharf {text-transform:capitalize}
</style>
</head>

<body>
<p class="buyuk">küçük yaz o büyütsün</p>
<p class="kucuk">BÜYÜK YAZ O KÜÇÜLTSÜN</p>
<p class="basharf">biz küçük yazdık o sadece kelimelerin
baş harflerini büyüttü.</p>
</body>
</html>
```



Kutu Modeli (Box Model)

CSS için en temel yapılandırma mantığı kutu modelidir. Bu mantığın, öğeleri biçimlendirirken, sayfa üzerinde neyi nereye ve nasıl konumlandıracağımızı belirlerken işimizi ne kadar kolaylaştırdığını ilerleyen satırlarda birlikte göreceğiz. Bu yaklaşımı, sayfamızda istediğimiz düzeni temin edebilmek için gereken bir formül gibi de düşünebiliriz.



İç içe katmanlardan oluşan bu yapı, adeta bir yerin haritadaki konumunu bulmamızı sağlayan koordinatlar gibidir ve biçimlendirmeye yardımcı olmaktadır.

İçten dışa doğru, modeli inceleyecek olursak; en içte içerikler yer alır ve bu kısmın konumlandırması `width` özelliği ile ifade edilir. Onun dışındaki `padding` katmanı, içeriğinizi `border` değerine göre konumlandırmak içindir. Yani içerik ile border arasındaki boşluğu ifade eder. Transparan bir yapıdadır.

İçeriğinize çerçeve vermek için kullanacağınız `border` değeri ise siz öyle ayarlamadıkça transparan değildir. Daha dıştaki katman olan `margin` (*kenar boşluğu*) değeri ise öğenizi komşu diğer öğelere göre hizalamak yani konumlandırmak için kullanılır. **Kenar boşluğu** (*margin*) değeri de transparandır.

`padding`: İçerik ile border arasındaki, içeriği kuşatan alandır.

`border`: İçeriğinize çerçeve vermek için kullanılır.

`margin`: Border çizgisinin etrafını kuşatan alandır.

NOT

Bir öğenin asıl uzunluk değeri, öğe içeriğinin **genişlik** (*width*) değeri ile beraber soldan sağa doğru sayacak olursak `sol-margin`, `sol-border`, `sol-padding`, `sağ-padding`, `sağ-border` ve `sağ-margin` değerlerinin toplamına eşit olacaktır.

Blok seviyesi öğesi (Block-level element): Blok seviyesi öğesinin formatını bir kutu gibi düşünebiliriz. İçinde bir başka kutu yada satır içi öğesi bulunabilir. Bir başlık (örn: `h6`), bir paragraf (`p`), ya da bir liste (`li`) ve benzeri öğeler, yani adı üstünde yeni bir blok oluşuran öğeler bu gruba girerler.

Satıriçi seviyesi öğesi (Inline element): Bir satıriçi seviyesi öğesini, bir blok seviyesi öğesinin aksine blok şeklinde değil de bir içerik gibi düşünebiliriz. (örn: `a`, `strong`, `Span`... vb) Aslında işin mantığı gayet basittir. İsterseniz bu noktada, daha fazla tanımlamalar yapmak yerine örneklerle konuyu geliştirelim. İşte başlıyoruz!

Kenarlık Çizgisi (Border)

Kenar çizgisini (*border*), içerik alanını çerçeveleyen çizgi olarak tanımlayabiliriz. Border ile ilgili olarak çizgi genişliği, çizgi rengi, çizgi stili...vb birçok özellik vardır. İşte onlar:

Kenar Çizgisi Stili (border-style)

Border çizgisini hiç göstermeyebileceğimiz gibi (`none`), noktalı (`dotted`), kesik çizgili (`dashed`) normal (`solid`) çit çizgili (`double`) olarak da gösterebiliriz.

```
<head>
<style type="text/css">
```

```
p.normal {border-style:solid;}
p.nokta {border-style:dotted;}
p.cizgi {border-style:dashed;}
p.cift {border-style:double;}
</style>
</head>

<body>
<p class="normal">Bu çerçeve genel çerçeve yapısını ifade eder.</p>
<p class="nokta">Bu çerçeve noktalarla çizilmiştir.</p>
<p class="cizgi">Bu çerçeve kesik çizgilerle çizilmiştir.</p>
<p class="cift">Bu metnin etrafına çift çizgili border
atanmıştır.</p>
</body>
```

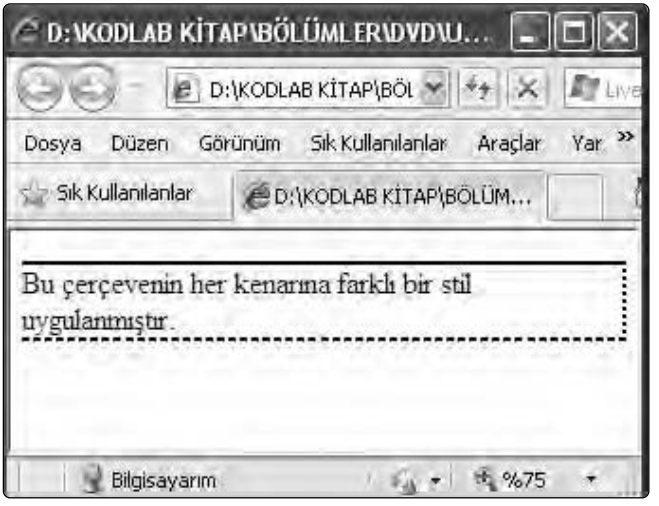


Örnekte de gördüğünüz gibi değişik border stilleri kullandık. Fakat border hep çerçeve şeklinde idi. Yani içeriğimizin tam olarak çerçeveliyordu. CSS ile istersek her kenara ayrı stil de tanımlayabiliriz.

Bunun için; `border-top-style`, `border-right-style`, `border-bottom-style`, `borderleft-style` özelliklerini kullanacağız.

```
<head>
<style type="text/css">
p {border-top-style:solid;
border-right-style:dotted;
border-bottom-style:dashed;
borderleft-style:none;
}
</style>
</head>

<body>
<p>Bu çerçevenin her kenarına farklı bir stil uygulanmıştır.</p>
</body>
```



KENAR ÇİZGİSİ GENİŞLİĞİ (BORDER-WIDTH)

Çizgi genişliğini belirlemek için kullanılır. Tek başına değil stil tanımlaması ile birlikte kullanılır.

thin: İnce kenar çizgisi içindir.

medium: Orta kalınlıkta kenar çizgisi için.

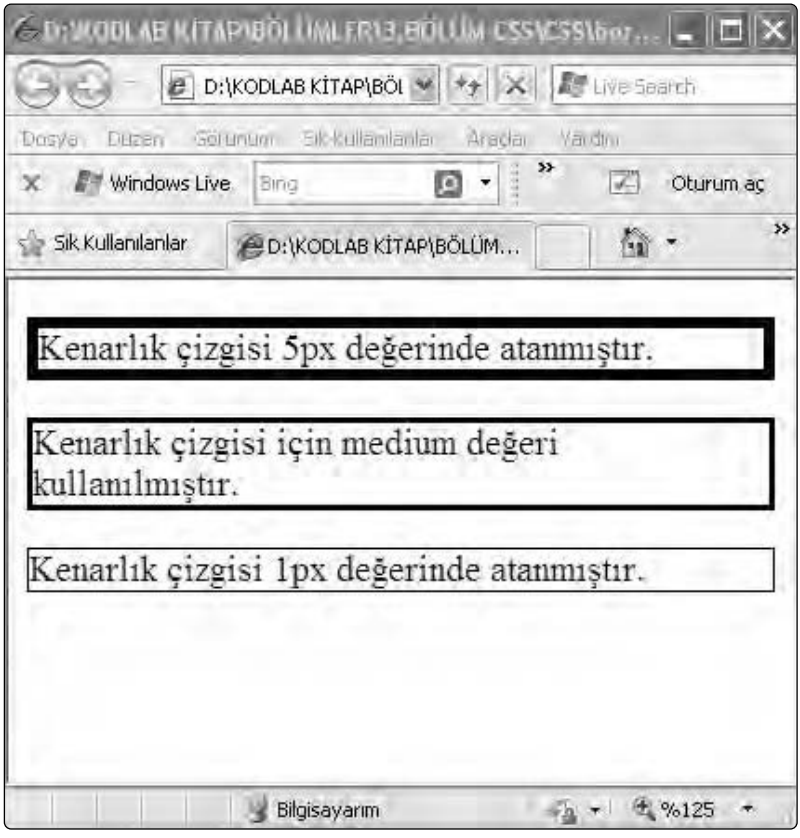
thick: Kalın kenar çizgisi için.

Ayrıca kenarlık kalınlığı için **piksel** (px) olarak da değer atanabilir.

```
<head>
<style type="text/css">
p.kalin
{
border-style:solid;
border-width:5px;
}
p.orta
{
border-style:solid;
border-width:medium;
}
p.ince
```

```
{
border-style:solid;
border-width:1px;
}
</style>
</head>

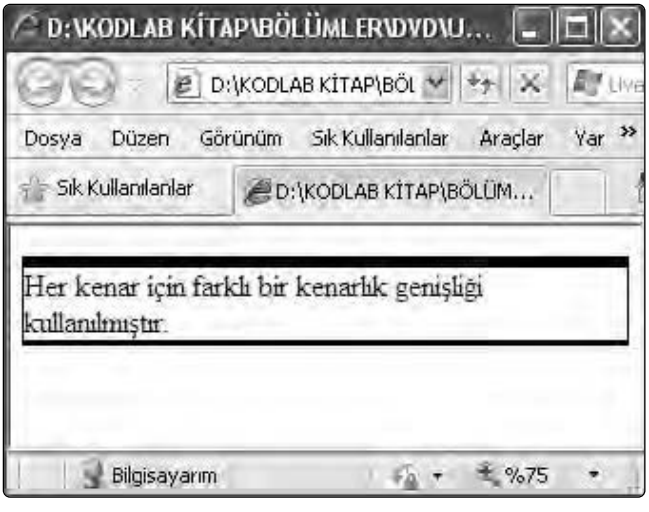
<body>
<p class="kalin">Kenarlık çizgisi 5px değerinde atanmıştır.</p>
<p class="orta">Kenarlık çizgisi için medium değeri
kullanılmıştır.</p>
<p class="ince">Kenarlık çizgisi 1px değerinde atanmıştır.</p>
</body>
```



Border stillerinde olduğu gibi burada da her kenara farklı genişlik değerleri atayabiliriz. Bunun için `border-top-width`, `border-right-width`, `border-bottom-width`, `border-left-width` özelliklerini kullanacağız.

```
<head>
<style type="text/css">
p
{
border-style:solid;
border-top-width:9px;
border-right-width: 1px;
border-bottom-width: thick;
border-left-width: 2px;
}
</style>
</head>

<body>
<p class="kalin">Her kenar için farklı bir kenarlık genişliği
kullanılmıştır.</p>
</body>
```



Kenar Çizgisi Rengi (border-color)

Kenarlık rengini belirlemek için kullanılır. Bu da tıpkı border genişliği gibi tek başına değil border stili ile birlikte kullanılır.

```
<head>
<style type="text/css">
p.red
{
border-style:solid;
border-color:red;
}
p.green
{
border-style:solid;
border-color:#98bf21;
}
</style>
</head>

<body>
<p class="red">Kenarlık rengi kırmızı olarak atanmıştır.</p>
<p class="green">Kenarlık rengi yeşil olarak atanmıştır.</p>
</body>
```

Her kenara farklı genişlikte, farklı stilde çizgi atanır da farklı renkte atanmaz mı. Bunun için de `border-top-color`, `border-right-color`, `border-bottom-color`, `border-left-color` özelliklerini kullanacağız.

```
<head>
<style type="text/css">
p
{
border-style:solid;
border-top-color:red;
border-right-color:#98bf21;
border-bottom-color:blue;
border-left-color:gray
}
</style>
</head>
```

```
<body>
<p class="kalin">Her kenar için farklı bir kenarlık rengi
kullanılmıştır.</p>
</body>
```

Kenar Çizgisi (border)

Evet gene beklediğiniz bir cümle geliyor. Tüm border özelliklerini bir arada tabi ki kullanabilirsiniz.

```
<head>
<style type="text/css">
p{border:5px solid red;}
</style>
</head>

<body>
<p>Burada 5px kalınlığında kırmızı renkli bir kenarlık
kullanılmıştır.</p>
</body>
```

Dış-Çizgi (outline) Özellikleri

Dış-çizgi özelliği, öğelerin etrafını kenar çizgileri dışından çizmek için kullanılır. Böylece öğe vurgulanarak ön plana çıkarılmış olur.

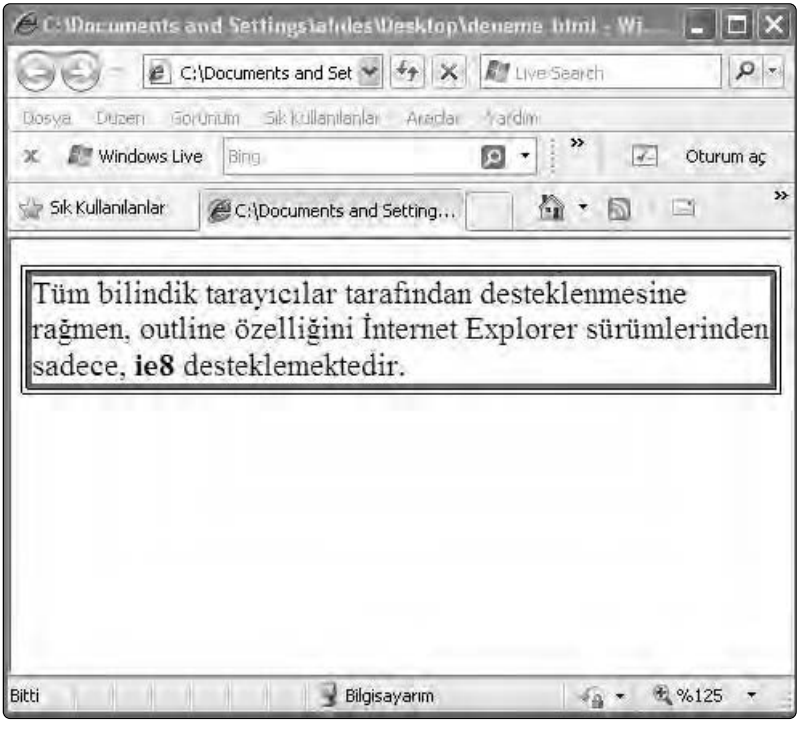
Dış-Çizgi stil (outline-style) Özelliği

Bu özellik dış-çizginin stilini belirlemek için kullanılır. Border için kullandığımız gibi bu özellik için de `dotted`, `dashed`, `solid`, `double` değerlerini kullanacağız.

```
<head>
<style type="text/css">
p
{
border:gray solid;
outline-style:double;
}
</style>
</head>

<body>
```

```
 <p>Tüm bilindik tarayıcılar tarafından desteklenmesine
rağmen, outline özelliğini İnternet Explorer sürümlerinden
sadece, <b>ie8</b> desteklemektedir.</p>
 </body>
```



Dış-Çizgi Renk (outline-color) Özelliği

Öğelerin etrafına çizilen dış çizginin rengini ayarlamak için kullanılır. Mutlaka `outline-style` özelliğini `outline-color` özelliğinden önce tanımlamalıyız. Olmayan bir çizginin rengini belirleyemeyiz değil mi?

Kullanılan değerler

`invert`: Renkleri ters çevirir. Art alan rengi ne olursa olsun çizginin görünmesini sağlar. Bu değer aynı zamanda da varsayılan değerdir.

renk: Özelliğe verilebilecek ikinci değer renktir. **Renk adı** (örn: `red`), **renk hex değeri** (`#ff0000`) ya da **renk rgb değeri** (örn: `255,0,0`) kullanılabilir.

Tüm bilindik tarayıcılar tarafından desteklenmesine rağmen, İnternet Explorer, `outline` özelliğini desteklememektedir.

```
<head>
<style type="text/css">
p
{
border:red solid thin;
outline-style:dotted;
outline-color:blue;
}
</style>
</head>

<body>
<p><b>Not: </b>Tüm bilindik tarayıcılar tarafından desteklenmesine
rağmen, outline özelliğini İnternet Explorer sürümlerinden sadece,
ie8 desteklemektedir.</p>
</body>
```

Dış-Çizgi Eni (outline-width) Özelliği

Dış-çizgi enini belirlemek için kullanılır. Bu özellik de tıpkı `outline-color` özelliğinde olduğu gibi `outline-style` özelliğinden sonra tanımlanmalıdır.

Kullanılan değerler

`thin`: İnci bir dış-çizgi kullanımını ifade eder.

`medium`: Normal bir dış-çizgi için kullanılır. Bu değer aynı zamanda varsayılan değerdir.

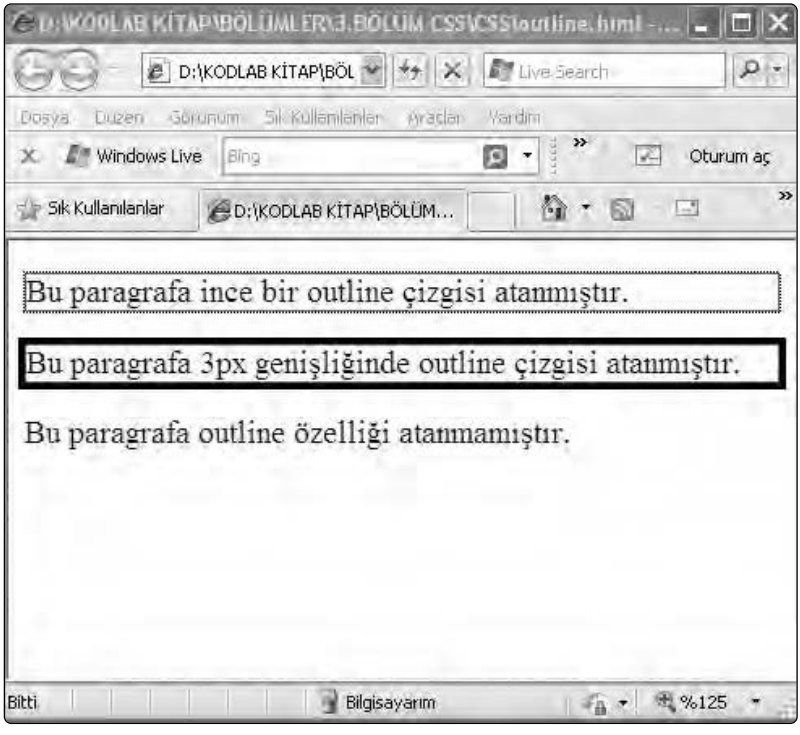
`thick`: Kalın bir dış-çizgi kullanımı içindir.

`uzunluk`: Dış-çizgi kalınlığı için rakamsal bir değer atamak için kullanılır.

```
<head>
<style type="text/css">
p.bir
{
border:red solid thin;
outline-style:dotted;
outline-width:thin;
```

```
}
p.iki
{
border:red solid thin;
outline-style:solid;
outline-width:3px;
}
</style>
</head>

<body>
<p class="bir">Bu paragrafa ince bir outline çizgisi atanmıştır.</p>
<p class="iki">Bu paragrafa 3px genişliğinde outline çizgisi
atanmıştır.</p>
<p>Bu paragrafa outline özelliği atanmamıştır.</p>
</body>
```



Dış-Çizgi (outline) Özelliği

Yukarda kullandığımız tüm özellikleri bir arada kullanmamızı sağlayan özelliktir.

```
<head>
<style type="text/css">
p {
border:red solid thin;
outline:#00ff00 dotted thick;
}
</style>
</head>

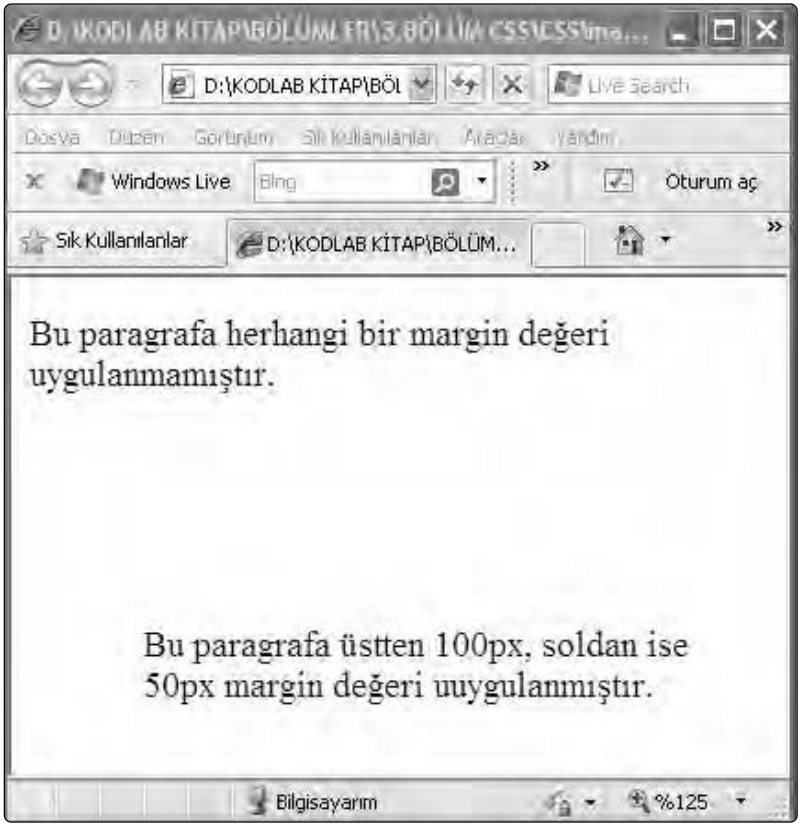
<body>
<p><b>Note: </b>Internet Explorer does not support the outline
properties.</p>
</body>
```

Border Dış (margin) Alanı

En başta da belirttiğimiz gibi margin öğenizi çevreleyen, bir başka ifade ile border çizgisini çevreleyen alandır. Uzunluk değeri (`em`, `px`... vd) yüzdelik (`%`) değeri ve auto değerlerini alabilir. `margin-top`, `margin-bottom`, `margin-right`, `margin-left` özellikleri ile kullanılır.

```
<head>
<style type="text/css">
p.margin {
margin-top:100px;
margin-left:50px;
}
</style>
</head>

<body>
<p>Bu paragrafa herhangi bir margin değeri uygulanmamıştır.</p>
<p class="margin">Bu paragrafa üstten 100px, soldan ise
50px margin değeri uuygulanmıştır.</p>
</body>
```



Bir de bu dört özelliği bir arada kullanmamızı sağlayan margin özelliği bulunmaktadır.

Kullanımı

```
margin: 25px 50px 75px 100px;
```

şeklindedir.

Dört değer yan yana kullanıldığında ilk değer üst margin'i (`top`), ikincisi sağ margin'i (`right`), üçüncüsü alt margin'i (`bottom`), dördüncüsü de sol margin'i (`left`) ifade eder. Bir başka ifade ile değerler saat yönünde sıralanırlar. (`üst`, `sağ`, `alt`, `sol`)

Üç değer yan yana kullanılırsa; (örn: margin: 25px 50px 75px) ilk değer üst margin'i (top), ikincisi sağ ve sol margin'i (right-left), üçüncüsü ise alt margin'i (bottom) ifade eder.

Sadece iki değer kullanılırsa (örn: margin: 25px 50px) ilk değer üst ve alt margin'i (top-bottom), ikincisi ise sağ ve sol margin'i (right-left) ifade eder.

Tek değer kullanıldığında ise kullanılan değer dört yön için de geçerli olacaktır.

```
<head>
<style type="text/css">
   p{ margin:100px }
</style>
</head>

<body>
<p>100px değeri tüm yönler için de geçerlidir.</p>
</body>
```

İçerik Dış (padding) Alanı

Padding alanı border çizgisi ile içeriğiniz arasında kalan ve içeriğinizi kuşatan alandır. Aynen margin değerinde olduğu gibi dört yönde de ayrı ayrı tanımlanabildiği gibi hepsini padding özelliği ile bir arada da tanımlayabilirsiniz. Uzunluk değeri (em, px... vd) ya da yüzdelik (%) değeri ile atanabilir.

```
<head>
<style type="text/css">
p {
padding-top:25px;
padding-left:50px;
}
</style>
</head>

<body>
<p>Bu paragrafa üstten 25px, soldan ise 50px padding değeri
uuygulanmıştır.</p>
</body>
```

Yukarıdaki örnek, farklı yönlerde padding uygulamasına, aşağıdaki örnek ise tüm yönlere tek seferde padding uygulamasına örnektir. Burada da margin değerini anlatırken söylediğimiz dörtlü, üçlü, ikili ve tekli kullanım ve değerlerin saat yönü esas alınarak dizilmesi kuralı aynen geçerlidir.

```
<head>
<style type="text/css">
   p{ padding:100px }
</style>
</head>

<body>
<p>100px değeri tüm yönler için de geçerlidir.</p>
</body>
```

Liste (List) Biçimlendirme

Liste özelliklerinin biçimlendirilmesi amacıyla kullanılır.

Liste İmleme Tipi (list-style-type)

XHTML bölümümüzden hatırlayacağınız gibi **unordered list** (`ul`) ve **ordered list** (`ol`) olmak üzere iki tip listemiz vardı. İşte bu özellikle liste tiplerinin imleme stillerini belirleyeceğiz. Bunun için, **unordered list** (`ul`) ile kullanacağımız değerler; `none`, `disc`, `circle` ve `square` olmaktadır.

`none`: İmleme kullanılmayacak anlamına gelir.

`disc`: İmleme olarak içi dolu daire kullanmak içindir.

`circle`: Bu ise içi boş daire kullanacağımız anlamına gelir.

`square`: Kare şeklinde imleme kullanmak için ise bu değeri seçeceğiz.

ordered list (`ol`) için ise `none`, `disc`, `circle`, `square`, `decimal`, `decimal-leading-zero`, `lower-latin`, `lower-roman`, `upper-latin` ve `upper-roman` kullanılmaktadır.

`none`: İmleme kullanılmayacak anlamına gelir.

`disc`: İmleme olarak içi dolu daire kullanmak içindir.

`circle`: Bu ise içi boş daire kullanacağımız anlamına gelir.

`square`: Kare şeklinde imleme kullanmak için ise bu değeri seçeceğiz.

decimal: Onluk sayı sistemi kullanmak içindir.

decimal-leading-zero: Her biri `0` ile başlayan (örn: `01`, `02`, `03`... vd.) onluk sayı sistemi.

lower-latin: Küçük latin harfleri (örn: a,b,c...vd) için.

lower-roman: Küçük roman rakamları (örn: i, ii, iii, iv...vd) için.

upper-latin: Büyük latin harfleri (örn: A, B, C, D...vd) için.

upper-roman: Büyük roman rakamları (örn: I, II, III, IV...vd) için.

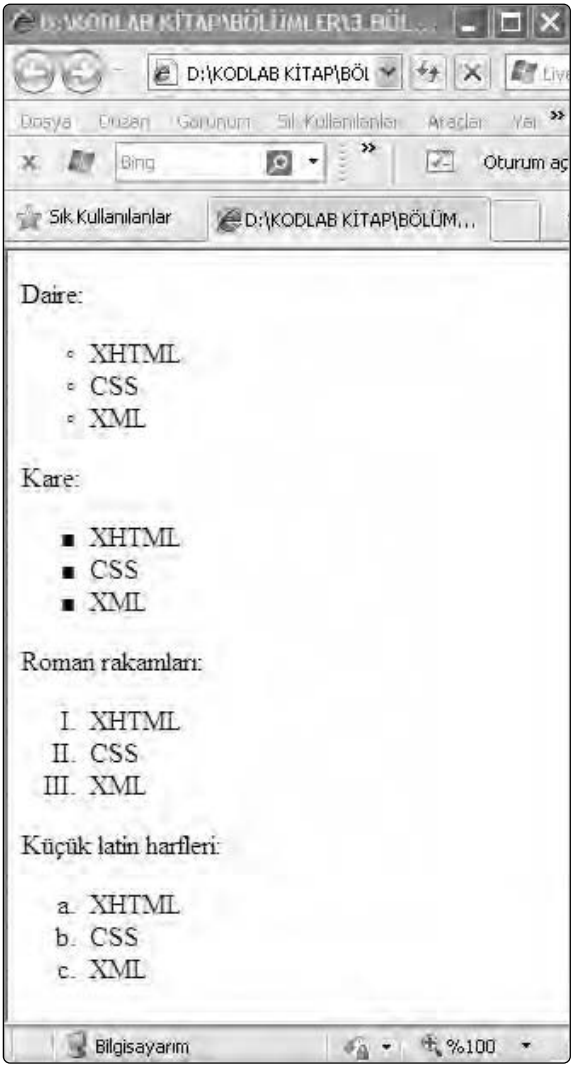
```
<head>
<style type="text/css">
   ul.circle {list-style-type:circle}
   ul.square {list-style-type:square}
   ol.upper-roman {list-style-type:upper-roman}
   ol.lower-latin {list-style-type:lower-latin}
</style>
</head>

<body>
<p>Daire:</p>
<ul class="circle">
<li>XHTML</li>
<li>CSS</li>
<li>XML</li>
</ul>

<p>Kare:</p>
<ul class="square">
<li>XHTML</li>
<li>CSS</li>
<li>XML</li>
</ul>

<p>Roman rakamları:</p>
<ol class="upper-roman">
<li>XHTML</li>
<li>CSS</li>
<li>XML</li>
</ol>
```

```
<p>Küçük latin harfleri:</p>
<ol class=”lower-latin”>
<li>XHTML</li>
<li>CSS</li>
<li>XML</li>
</ol>
</body>
```



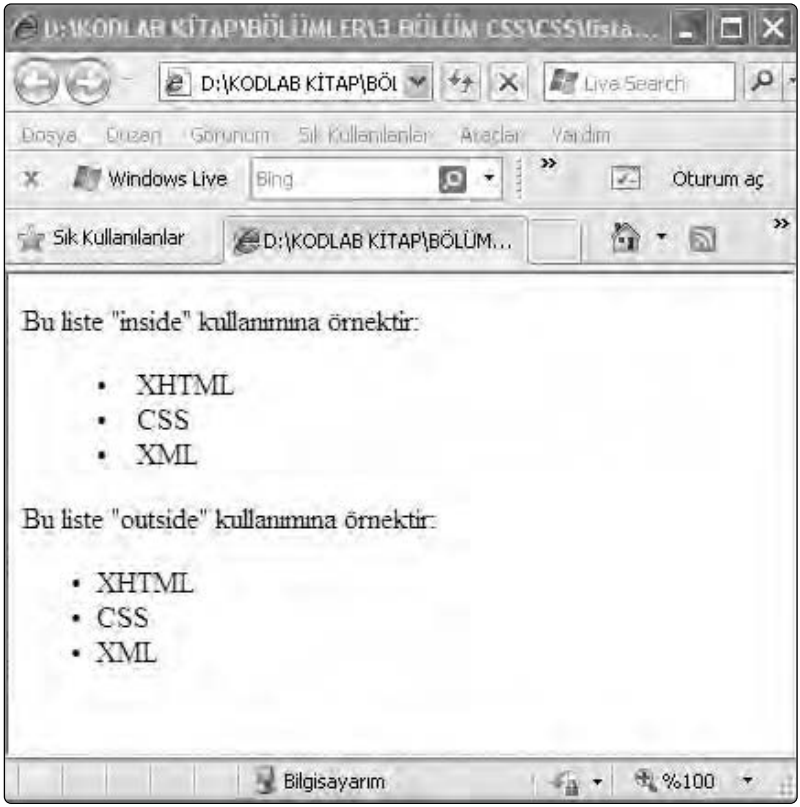
Liste İmleri Konum (list-style-position)

Bu özellik imlerin pozisyonlarını belirlemek içindir. `inside` imlerin daha içerden başlamasını sağlar. Varsayılan değer `outside` değeridir.

```
<head>
<style type="text/css">
  ul.inside {list-style-position:inside}
  ul.outside {list-style-position:outside}
</style>
</head>

<body>
<p>Bu liste "inside" kullanımına örnektir:</p>
<ul class="inside">
<li>XHTML</li>
<li>CSS</li>
<li>XML</li>
</ul>

<p>Bu liste "outside" kullanımına örnektir:</p>
<ul class="outside">
<li>XHTML</li>
<li>CSS</li>
<li>XML</li>
</ul>
</body>
```



Liste İmleme Resmi (list-style-image)

İmlerin yerine resim koymak için kullanılır.

```
<head>
<style type="text/css">
  ul {list-style-image:url('ok.gif');}
</style>
</head>
<body>
<ul>
<li>XHTML</li>
<li>CSS</li>
<li>XML</li>
</ul>
</body>
```



Liste Biçimleme (list-style)

Yukarıda anlattığım özelliklerin bir arada kullanılmalarını sağlar.

```
<head>
<style type="text/css">
   ul {list-style:inside url("ok.gif");}
</style>
</head>

<body>
<ul>
<li>XHTML</li>
<li>CSS</li>
<li>XML</li>
</ul>
</body>
```

Tablo (table) Özellikleri

Tabloların konum ve hizalama ayarları için kullanılır. Kullanılan özellikleri; `border-collapse`, `border-spacing`, `caption-side`, `empty-cells`, `table-layout` olarak sıralayabiliriz.

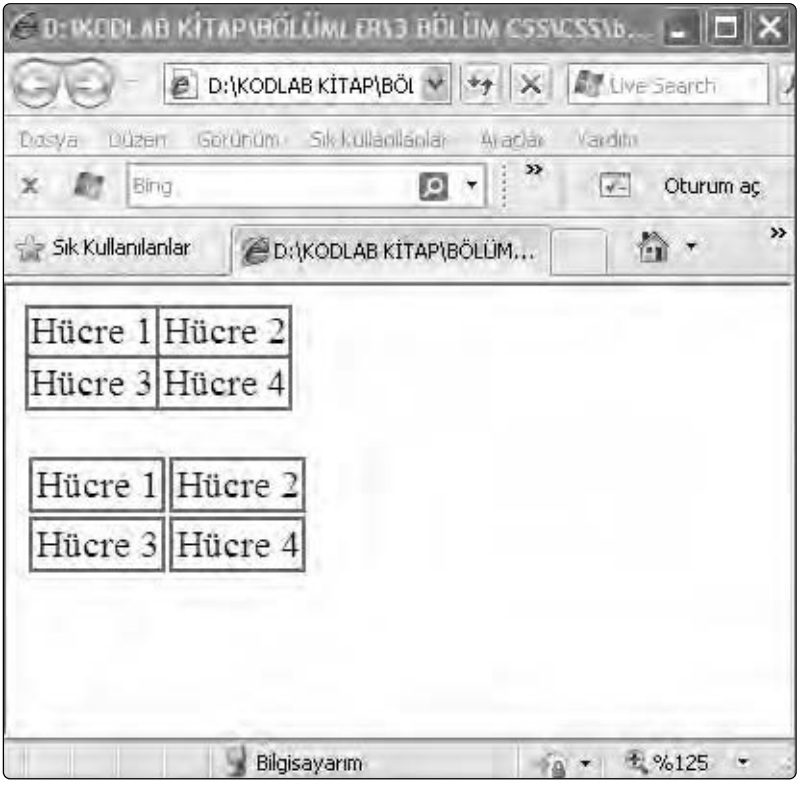
Sınır İndirgeme (border-collapse) Özelliği

Tablodaki satırların ve hücrelerin teke indirgenip indirgenmeyeceğini belirlemede kullanılır. Kullanılan değerlerden `separate`; satırların ve hücrelerin birbirlerinden **ayrı kenarlıklarının** olmasını sağlarken, `collapse`; satırların ve hücrelerin birbirlerine bitişerek, kenarlığın **teke indirgenmesini** sağlar.

```
<head>
<style type="text/css">
.tablo1 {border-collapse:collapse;}
.tablo2 {border-collapse:separate;}
.tablo1 td, .tablo2 td {border:solid 2px red;}
</style>
</head>

<body>
<table class="tablo1">
<tr>
    <td>Hücre 1</td>
    <td>Hücre 2</td>
</tr>
<tr>
    <td>Hücre 3</td>
    <td>Hücre 4</td>
</tr>
</table>
<br/>
<table class="tablo2">
<tr>
    <td>Hücre 1</td>
    <td>Hücre 2</td>
```

```
</tr>
<tr>
    <td>Hücre 3</td>
    <td>Hücre 4</td>
</tr>
</table>
</body>
```



Sınır Aralığı (border-spacing) Özelliği

Hücreler arasındaki aralığın ölüsünü belirlemek için kullanılır. Değer olarak; px, em, cm... vd uzunluk değerlerini alır. Tek uzunluk değeri atandığı durumlarda, söz konusu değer hem yatay hem de dikey aralıklar için geçerli olur.

İki değer atandığında ilki yatay aralığı ikincisi ise dikey aralığı ölçülendirmek için kullanılır.

```
<head>
<style type="text/css">
table.ilk
{
border-collapse:separate;
border-spacing:10px;
}
table.iki
{
border-collapse:separate;
border-spacing:10px 50px;
}
</style>
</head>

<body>
<table class="ilk" border="1">
<tr>
<td>XHTML</td>
<td>CSS</td>
</tr>
<tr>
<td>XML</td>
<td>Javascript</td>
</tr>
</table>
<br />

<table class="iki" border="1">
<tr>
<td>XHTML</td>
<td>CSS</td>
</tr>
<tr>
<td>XML</td>
<td>Javascript</td>
</tr>
</table>
</body>
```



Başlık Yanı (caption-side) Özelliği

Tablo başlığının konumunu ayarlamak için kullanılır. Kullanılan değerlerden; **`top`**; tablo başlığını konum olarak tablonun üst kısmına, **`bottom`**; ise alt kısmına konumlandırır. **`left`**; tablo başlığını tablonun soluna, **`right`** da tablonun sağına konumlandırır.

```
<head>
<style type="text/css">
caption {caption-side:bottom}
</style>
</head>

<body>
<table border="1">
<caption>Alt Başlık</caption>
<tr>
<td>XHTML</td>
<td>CSS</td>
</tr>
<tr>
<td>XML</td>
<td>JAVASCRIPT</td>
</tr>
</table>
</body>
```



Boş Hücreler (empty-cells) Özelliği

Bu özelliği tablomuzdaki boş hücrelerin gösterilip gösterilmeyeceğini belirlemek için kullanacağız. Özelliğe `hide` değeri atanırsa boş hücrelerin çizgileri gösterilmez. Değer olarak `show` değeri atandığında ise çizgiler gösterilir.

```
<head>
<style type="text/css">
table
{
border-collapse:separate;
empty-cells:hide;
}
</style>
</head>

<body>
<table border="1">
<tr>
<td>XHTML</td>
<td>Javascript</td>
</tr>
<tr>
<td>CSS</td>
<td></td>
</tr>
</table>
</body>
```



Tablo Yapısı (table-layout) Özelliği

XHTML tablo yapısı varsayılan olarak, içerdiği metin, resim... vd öğelerin genişliğine göre otomatik olarak genişliklerini ayarlayacak şekilde tasarlanmıştır. Yani içeriğin eni arttıkça tablonun eni de artacaktır.

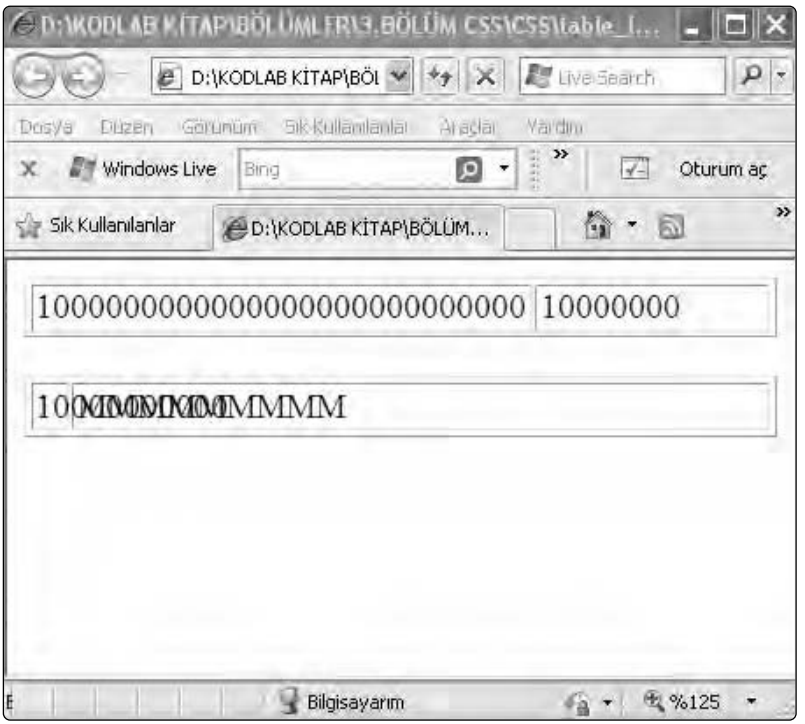
Bu esnek ve kullanışlı yapı genelde tercihimiz olmakla beraber tablo genişliğinin sabit kalmasını isteyeceğimiz durumlar da olacaktır. Böyle durumlarda bu özellik ile beraber `fixed` değerini kullandığımızda tablomuzun genişliği sabit kalacaktır.

```
<head>
<style type="text/css">
table.ex1 {table-layout:auto}
table.ex2 {table-layout:fixed}
```

```
</style>
</head>

<body>
<table class="ex1" border="1" width="100%">
<tr>
<td width="5%">10000000000000000000000000000</td>
<td width="95%">10000000</td>
</tr>
</table>
<br />

<table class="ex2" border="1" width="100%">
<tr>
<td width="5%">10000000000000000000000000000</td>
<td width="95%">10000000</td>
</tr>
</table>
</body>
```



Boyut (dimension) Özellikleri

Öğelerin yükseklik ve genişlik değerlerini ayarlamakta kullanılır. `height`, `width`, `max-height`, `min-height`, `max-width`, `min-width` özelliklerini kapsamaktadır.

Yükseklik (height) ve Genişlik (width) Özelliği

Öğelerin yüksekliklerini ve genişliklerini belirlemek için kullanılırlar.

Yükseklik (`height`) ve **genişlik** (`width`) özellikleri için; `auto`, **yüzdelik** (`%`) ya da uzunluk için (`px`, `em`... vd) değerlerinden herhangi biri kullanılabilir.

```
<head>
<style type="text/css">
img.normal {height:auto}
img.buyuk {height:200px}
p.ebat
{
height:100px;
width:200px;
}
</style>
</head>
<body>
<img class="normal" src="logocss.gif" width="95" height="84" />
<br />
<img class="buyuk" src="logocss.gif" width="95" height="84" />
<p class="ebat">Bu paragrafa 50px yükseklik değeri ile
200px genişlik değeri atanmıştır.</p>
   <p>Bu paragrafa herhangi bir boyut değeri atanmamıştır.
   Dolayısıyla paragrafın boyutu tarayıcı tarafından otomatik
   olarak belirlenecektir.</p>
 </body>
```



AZAMİ YÜKSEKLİK (MAX-HEIGHT) VE AZAMİ GENİŞLİK (MAX-WIDTH)

Bir öğenin alabileceği azami yükseklik (`max-height`) ve azami genişlik (`max-width`) değerlerini ifade ederler. Bu özellikler için de `auto`, yüzdelik (`%`) ya da uzunluk (`px`, `em`... vd) değerlerinden herhangi birini kullanabilirsiniz.

```
<head>
<style type="text/css">
   p { max-height:100px;}
</style>
</head>

<body>
```

```
<p>Bu paragrafın azami yüksekliği 100px olarak atanmıştır.</p>
</body>
<head>
<style type="text/css">
p {max-width:100px;}
</style>
</head>

<body>
<p>Bu paragrafın azami genişliği 100px olarak atanmıştır.</p>
</body>
```

Asgari Yükseklik (min-height) ve Asgari Genişlik (min-width)

Bir öğenin alabileceği asgari yükseklik (`min-height`) ve asgari genişlik (`min-width`) değerlerini ifade ederler. Bu özellikler için de azami değerlerde olduğu gibi auto, yüzdelik (`%`) ya da uzunluk (`px`, `em`... vd) değerlerinden herhangi birini kullanabilirsiniz.

```
<head>
<style type="text/css">
p{min-height:100px;}
</style>
</head>

<body>
<p>Bu paragrafın asgari yüksekliği 100px olarak atanmıştır</p>
</body>

<head>
<style type="text/css">
p {min-width:50px;}
</style>
</head>

<body>
<p>Bu paragrafın asgari genişliği 50px olarak atanmıştır</p>
</body>
```

Sınıflandırma (classification) ve Konumlama (positioning) Özellikleri

Bu özellikler; öğelerin nasıl görüntüleneceklerini ve nasıl konumlandırılacaklarını belirlemek için kullanılır.

Görüntüleme (display) Özelliği

Kullanılan değerler ve örnekler

none: Özellik (*display*) ile beraber bu değer kullanıldığında öğe görüntülenmez.

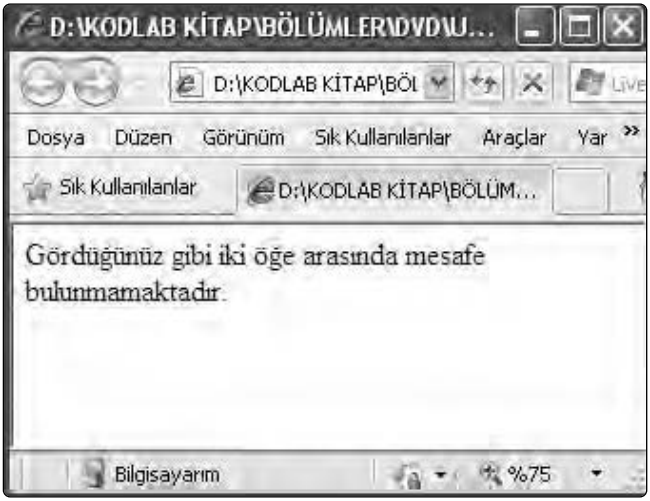
inline: Varsayılan normal görünüm budur.

```
<head>
<style type="text/css">
p {display: inline}
div {display: none}
</style>
</head>

<body>
<p>Gördüğünüz gibi iki öğe arasında</p>

<p>mesafe bulunmamaktadır.</p>

<div>Bu div bölümü görüntülenmeyecektir.</div>
</body>
```



block: değeri ile öğe, önündeki ve arkasındaki satır boşlukları ile beraber blok düzeyi bir öğe olarak görüntülenecektir.

list-item: değeri kullanıldığında öğe liste biçimde görüntülenir.

table, **inline-table**, **table-row-group**, **table-column**, **table-columngroup**, **table-header-group**, **table-footer-group**, **table-row**, **tablecell**, **table-caption:** Bu değerler öğenin bir tablo öğesi gibi görüntülenmesini sağlar.

GEZİNME/KAYMA (FLOAT) ÖZELLİĞİ

Bir resim ya da metnin bir üst öğe içerisindeki konumunu belirlemek için kullanılır. Kullanımı; `img{float: left}` şeklindedir.

Kullandığı değerler

left: Resim ya da metni üst öğenin soluna kaydırır.

right: Resim ya da metni üst öğenin sağına kaydırır.

none: Resim ya da metne herhangi bir kaydırma uygulanmaz. Aynı zamanda varsayılan değerdir.

```
<head>
<style type="text/css">
img {float:right;}
```

```
</style>
</head>

<body>
<p>
<img src="boyut.jpg" width="80" height="60" />
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas
eget mauris in neque condimentum laoreet. Curabitur iaculis lectus
sed dui fermentum imperdiet. Quisque elementum tempor lectus, non
vestibulum nibh vestibulum id. Proin scelerisque ipsum at lectus
faucibus vitae viverra mauris eleifend. Nullam et augue enim. Nunc
ut arcu nibh, nec tempor nisi. Proin molestie sem fringilla orci
venenatis posuere. Quisque fermentum nibh a orci pretium euismod.
</p>
</body>
```



Silme (clear) Özelliği

Float özelliğinin etkisini sonlandırmak için kullanılır.

left: Float özelliğinin etkisini sol taraf için kaldırır.

right: Float özelliğinin etkisini sağ taraf için kaldırır.

both: Float özelliğinin etkisini hem sol hem de sağ taraf için kaldırır.

none: Float özelliğinin etkisinin sürmesini sağlar. Aynı zamanda varsayılan değerdir.

```
<style type="text/css">
div.float {
  border:1px solid #ff9900;
  width:120px;
  float: right;
  }

div.clear {
  border:1px solid #cccccc;
  width:120px;
  clear: right;
  }
</style>
<div class="float">Bu kutuya float değeri verildi.</div>
<div class="clear">Bu kutuya "CSS clear" özelliği uygulandı.</div>
```

Konum (position) Özelliği

Position özelliğini, öğemizi konumlandırmak için kullanacağız. Kullanılan değerler:

static; değeri, postion özelliğinin varsayılan değeridir. Bir öğeye bu değeri atamanız demek, yerinde sabit kalacağı; position statik bir özellik olduğundan top, bottom, left, right... vb değerleri atadığınızda çalışmayacakları anlamına gelir. Konumlandırma normal akışa göre olacaktır.



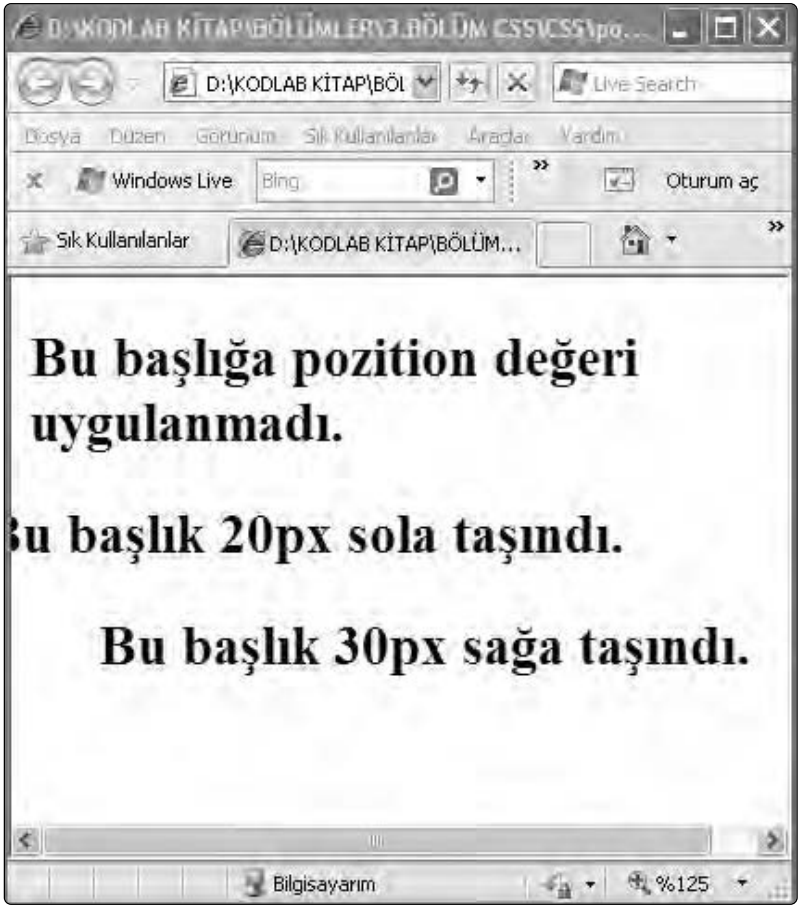
Bir sayfadaki öğelere konum (position) özelliği ya da float uygulanmamışsa sayfadaki öğeler ilk öğeden son öğeye doğru alt alta sıralanacaktır. Bu durum **Normal Akış** olarak ifade edilir.

relative; değeri kullanılarak konumu belirlenen bir öğenin ilk konumundaki yeri ve diğer öğelerle arasındaki mesafe korunur. Bir başka ifade ile, öğenin konumu ilk olarak **Normal Akış**'a göre belirlenir. Daha sonra da öğe, **göreli** (*relative*) konuma yerleştirilir.

```
<head>
<style type="text/css">
h2.sol
{
position:relative;
left:-20px;
}

h2.sag
{
position:relative;
left:30px;
}
</style>
</head>

<body>
<h2>Bu başlığa pozition değeri uygulanmadı.</h2>
<h2 class="sol">Bu başlık normal konumuna göre 20px sola
taşındı.</h2>
<h2 class="sag">Bu başlık normal konumuna göre 30px sağa
taşındı.</h2>
</body>
```



absolute; değeri ile beraber `top`, `bottom`, `left` ve `right` özelliklerini kullanarak bir öğeyi sayfamızın istediğimiz bir yerine konumlandırabiliriz.

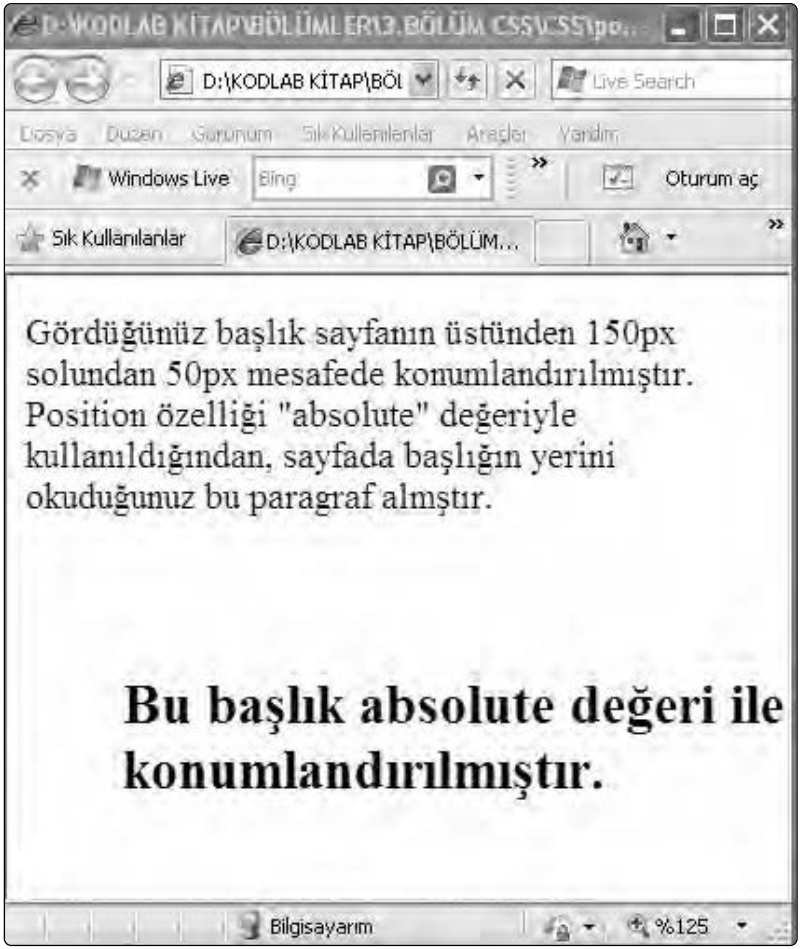
Yukarıdaki `relative` değerinin aksine `absolute` değeri ile konumlandırma yapıldığında, öğe eski yerini muhafaza etmez. Dolayısı ile diğer öğeler bu öğeyi görmezden gelerek akıştaki yerlerini alacaklardır.

```
<head>
<style type="text/css">
h2
```

```
{
position:absolute;
left:50px;
top:150px;

}
</style>
</head>

<body>
<h2>Bu başlık absolute değeri ile konumlandırılmış-
tır.</h2>
<p>Gördüğünüz başlık sayfanın üstünden 150px solundan 50px
mesafede konumlandırılmıştır. Position özelliği "absolute"
değeriyle kullanıldığından, sayfada başlığın yerini okuduğunuz
bu paragraf almştır.</p>
</body>
```



`fixed;` değeri uygulanan öğe, sayfadaki diğer öğeler kaydırma çubuğu ile aşağı indirilirken yerine sabit kalır ve diğer öğelerle beraber hareket etmez.

Z-DİZİN (Z-INDEX) ÖZELLİĞİ

Bu özellik, `position:absolute`, `position:relative` ya da `position:fixed` uygulanmış katmanların nasıl gösterileceğini ayarlamak için kullanılır.

```
<head>
<style type="text/css">
#div1 {
      width:200px;
      height:200px;
      background:blue;
      position:absolute;
      top:50px;
      left:50px;
      z-index:1;
}

#div2 {
      width:200px;
      height:200px;
      background:yellow;
      position:absolute;
      top:80px;
      left:80px;
      z-index:2;
}
</style>
</head>

<body>
      <p id="div1">İLK KATMAN</p>
      <p id="div2">İKİNCİ KATMAN</p>
</body>
```



Örneğimizde de gördüğünüz gibi z-index değeri büyük olan katman üste görünecektir.

Görsel Sonuçlar

Taşma (overflow) Özelliği

Bu özellik, öğe içeriğinin taşması durumda içeriğin nasıl görüntüleneceğini belirlemek için kullanılır.

`visible`: değeri ile içeriğin taşan kısımlarını gösteririz.

`hidden`: değeri ile içeriğin taşan kısımlarını gizleriz.

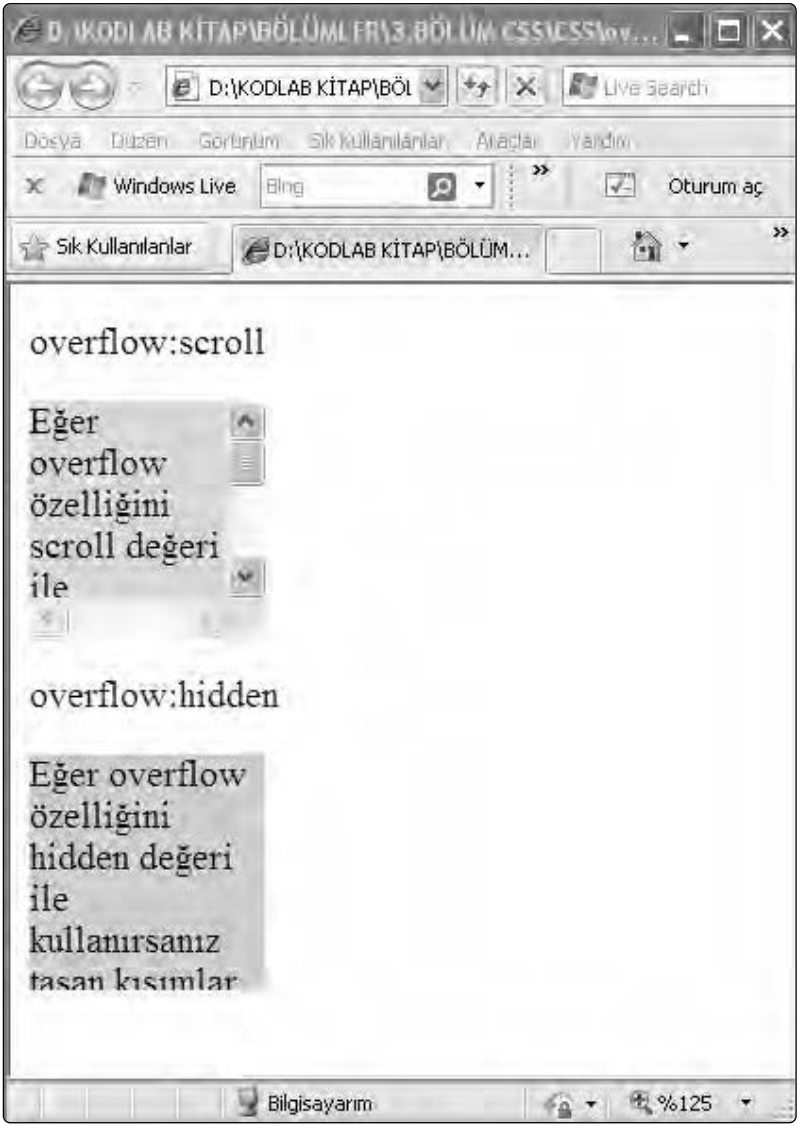
`scroll`: değeri ile içeriğin taşan kısımlarını kaydırma çubuğu ile gösteririz.

```
<head>
<style type="text/css">
div.scroll
{
background-color:#00FFFF;
width:100px;
height:100px;
overflow:scroll;
}

div.hidden
{
background-color:#00FF00;
width:100px;
height:100px;
overflow:hidden;
}
</style>
</head>

<body>
<p>overflow:scroll</p>
<div class="scroll">Eğer overflow özelliğini scroll değeri
ile kullanırsanız taşan kısımlar kaydırma çubuğu yardımı
ile gösterilir.</div>

<p>overflow:hidden</p>
<div class="hidden">Eğer overflow özelliğini hidden değeri ile
kullanırsanız taşan kısımlar gösterilmeyip gizlenecektir.</div>
</body>
```


D:\KODLAB KİTAP\BÖLÜMLER\3.BÖLÜM CSS\CSS\ov...
D:\KODLAB KİTAP\BÖL
Live Search
Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım
Windows Live
Bing
Oturum aç
Sık Kullanılanlar
D:\KODLAB KİTAP\BÖLÜM...
overflow:scroll
Eğer overflow özelliğini scroll değeri ile
overflow:hidden
Eğer overflow özelliğini hidden değeri ile kullanırsanız
Bilgisayarım
%125

Kırpma (clip) Özelliği

Bu özellik, içeriğin sadece bir bölümünün gösterileceği şekilde kırpılması için kullanılır.

Kullanımı

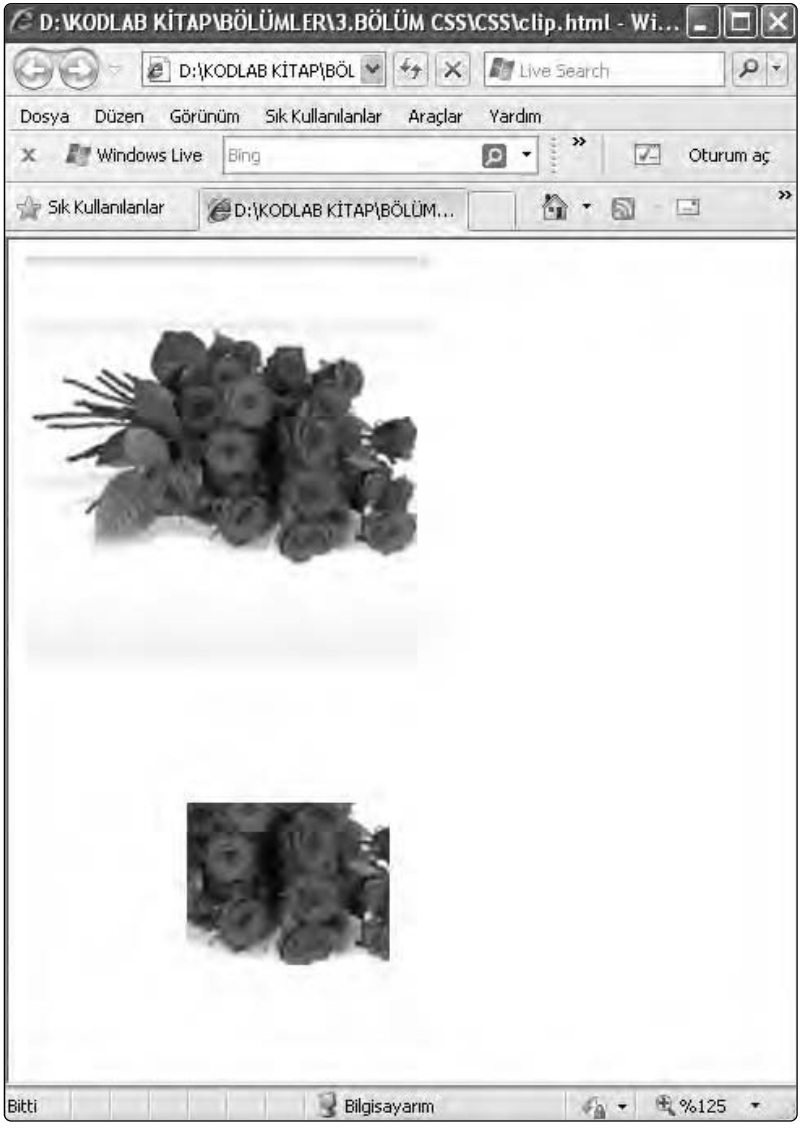
```
img clip: rect(<üst>,<sag>,<alt>,<sol>)
```

şeklindedir.

Kırpma olmayacağı anlamına gelen başlangıç varsayılan değeri ise auto olmaktadır.

```
<head>
<style type="text/css">
.clip
{
position:absolute;
clip:rect(70px,180px,150px,80px);
}
</style>
</head>

<body>
<img src="resim3.jpg" alt="orjinal_resim" />
<br />
<div class="clip">
<img src="resim3.jpg" alt="kırpılan_resim" />
</div>
</body>
```


D:\KODLAB KİTAP\BÖLÜMLER\3.BÖLÜM CSS\CSS\clip.html - Wi...
D:\KODLAB KİTAP\BÖL
Live Search
Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım
Windows Live
Bing
Oturum aç
Sık Kullanılanlar
D:\KODLAB KİTAP\BÖLÜM...
Bitti
Bilgisayarım
%125

Görünülebilme (visibility) Özelliği

Bu özellik öğenin görüntülenip görüntülenemeyeceğini belirlemek için kullanılır.

Kullanımı

```
p { visibility: hidden }
```

şeklindedir.

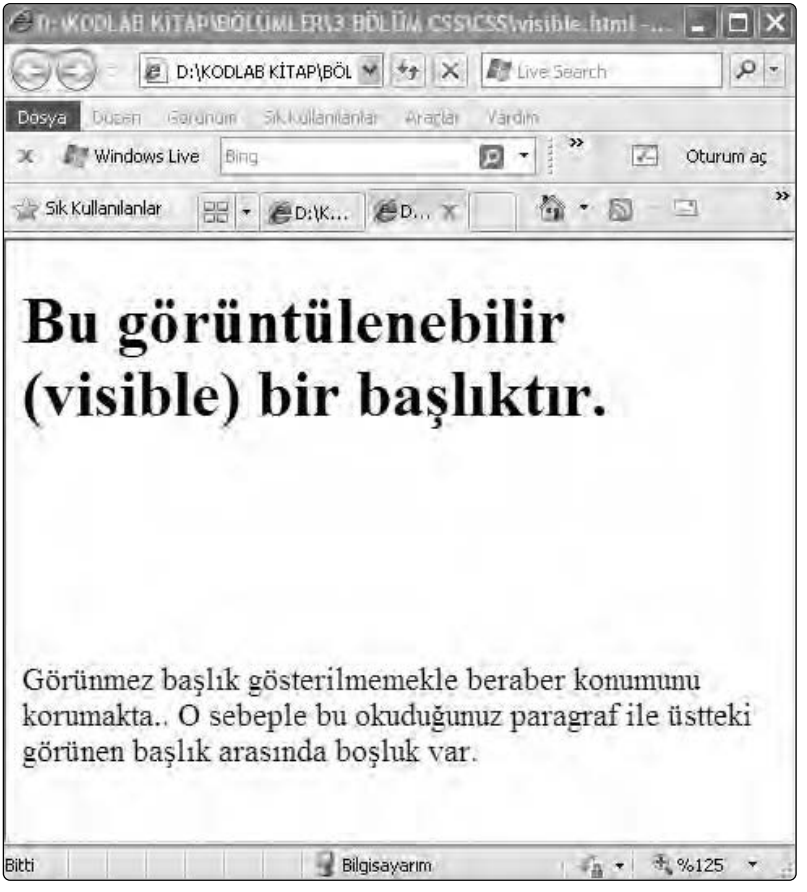
visible: kullanılması demek, öğe görüntülenebilir demektir.

hidden: değeri kullanıldığında öğe görüntülenemez.

collapse: değeri sadece tablo öğeleri için kullanılır. Bir satır ya da sütunu kaldırır, fakat tablonun görünüşüne etki etmez. Yani bu alanlar başka satırlar ya da sütunlar tarafından kullanılır.

```
 <head>
 <style type="text/css">
 h1.visible {visibility:visible}
 h1.hidden {visibility:hidden}
 </style>
 </head>

 <body>
 <h1 class="visible">Bu görüntülenebilir (visible) bir baş-
lıktır.</h1>
 <h1 class="hidden">Bu görüntülenemez(invisible) bir baş-
lıktır.</h1>
 <p>Görünmez başlık gösterilmemekle beraber konumunu korumakta...
O sebeple bu okuduğunuz paragraf ile üstteki görünen başlık
arasında boşluk var. </p>
 </body>
```

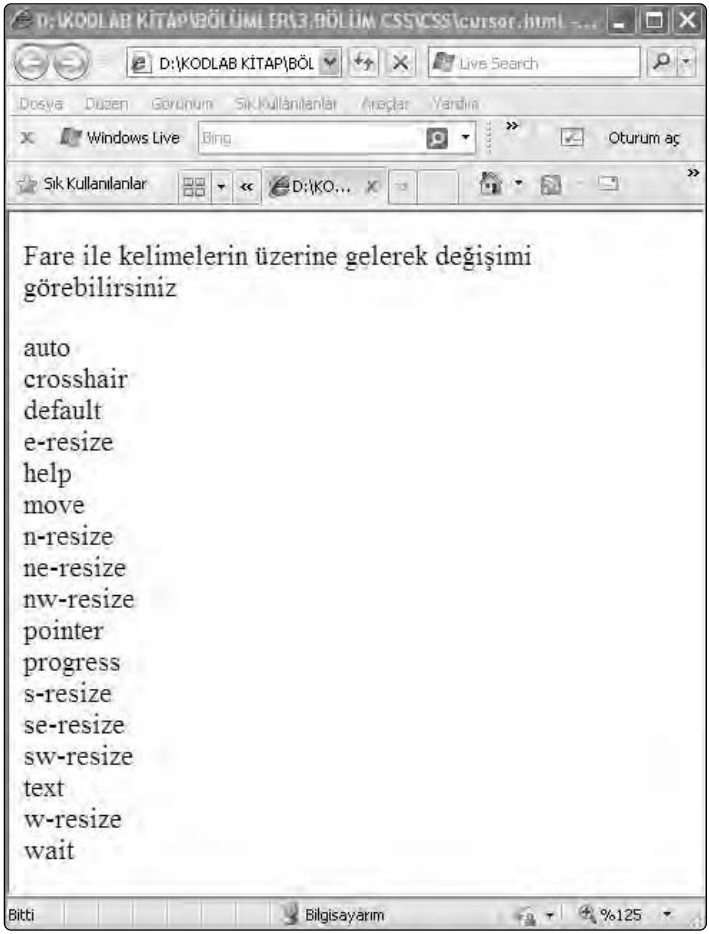


İmleç (cursor) Özelliği

Bu özellik, bir öğenin üzerine fare ile gelindiğinde görüntülenmesini istediğimiz imleci belirlemek için kullanılır.

```
<body>
<p>Fare ile kelimelerin üzerine gelerek değişimi görebilirsiniz.</p>
<span style="cursor:auto">auto</span><br />
<span style="cursor:crosshair">crosshair</span><br />
<span style="cursor:default">default</span><br />
<span style="cursor:e-resize">e-resize</span><br />
<span style="cursor:help">help</span><br />
<span style="cursor:move">move</span><br />
```

```
<span style="cursor:n-resize">n-resize</span><br />
<span style="cursor:ne-resize">ne-resize</span><br />
<span style="cursor:nw-resize">nw-resize</span><br />
<span style="cursor:pointer">pointer</span><br />
<span style="cursor:progress">progress</span><br />
<span style="cursor:s-resize">s-resize</span><br />
<span style="cursor:se-resize">se-resize</span><br />
<span style="cursor:sw-resize">sw-resize</span><br />
<span style="cursor:text">text</span><br />
<span style="cursor:w-resize">w-resize</span><br />
<span style="cursor:wait">wait</span><br />
</body>
```



Sözde Sınıflar (pseudo-classes)

Sözde sınıflar, seçicilere, özel birtakım yetenekler kazandırmak için kullanılır. Bunları ve ne gibi etkileri olduğunu uzun uzun tanımlamak yerine uygulamalar ile göstermek daha anlaşılır olacaktır. Sadece, bu sınıfların kullanımının bize, XHTML öğelerini birtakım özel şartlara göre (örn: bir linkin tıklanmış hali ile tıklanmamış haline farklı stiller uygulamak) farklı farklı şekillendirme imkanı verdiğini belirtmekle yetinelim. Bu durum daha dinamik şekillendirme demektir.

Kullanımı

```
seçici:pseudo-sınıfı {özellik: değer}
```

şeklindedir. Ayrıca sözde sınıfları CSS sınıfları ile de kullanabiliriz. Bu takdirde kullanım;

```
seçici.class:pseudo-sınıfı {özellik:değer}
```

şeklinde olacaktır.

Linkli Sözde-sınıflar (Anchor Pseudo-classes)

Bir linkin fare ile üzerine gelindiğinde, tıklanmadan önce ve tıklandıktan sonraki stil özelliklerini ifade eder.

`a:link`: Tıklanmamış linklere stil atamak için kullanılır.

Kullanımı

```
a:link {color: #FF0000}
```

`a:visited`: Tıklanmış linklere stil atamak için kullanılır.

Kullanımı

```
a:visited {color: #00FF00}
```

`a:hover`: Linkin üzerine fare ile gelindiğinde görüntülenecek stili ifade eder.

Kullanımı

```
a:hover {color: #FF00FF}
```

a:active: Aktif olan linke stil atamak için kullanılır.

Kullanımı

```
a:active {color: #0000FF}
```

```
<html>
<head>
<style type="text/css">
a:link {color: #FF0000}
a:visited {color: #00FF00}
a:hover {color: #FF00FF}
a:active {color: #0000FF}
</style>
</head>

<body>
<p><b><a href="index.html" target="_blank">Lütfen linki
tıklayın.</a></b></p>
<p><b>Not:</b> a:hover, sınıfının etkin olması için a:link
ve a:visited tanımlarından sonra gelmesi gerekir.!!</p>
<p><b>Not:</b> a:active, sınıfının etkin olması için a:hover
tanımlarından sonra gelmesi gerekir.!!</p>
</body>
</html>
```

NOT

a:active, sınıfının etkin olması için **a:hover** tanımlarından sonra gelmesi gerekir.

a:hover, sınıfının etkin olması için **a:link** ve **a:visited** tanımlarından sonra gelmesi gerekir.

İlk-çocuk (first-child) Sözde Sınıfı

`first-child` sözde sınıfı, bir öğenin ilk çocuğu olan (bir alt seviyede bulunan) öğeye stil atamak içindir.

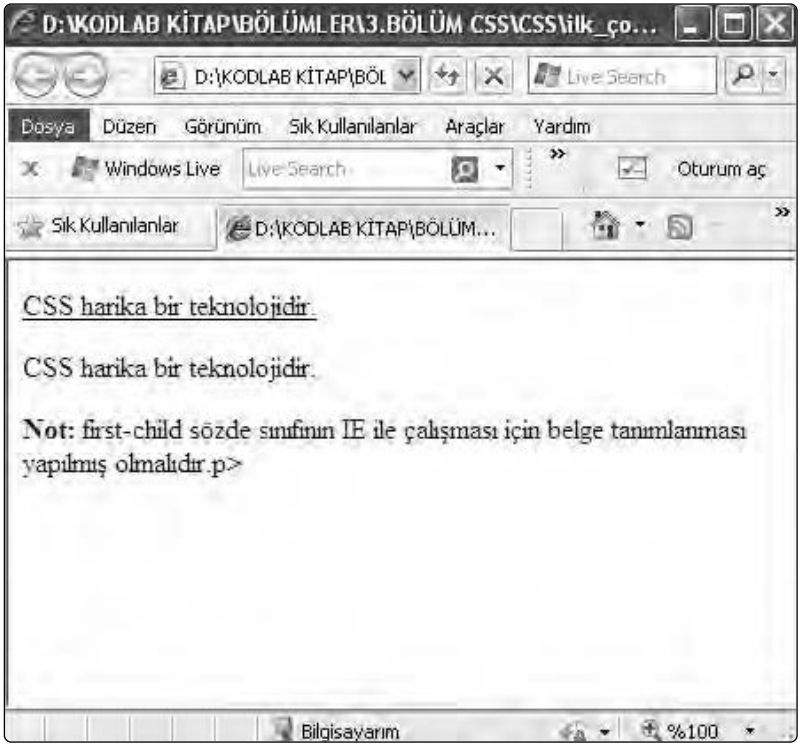
NOT

`first-child` sözde sınıfının IE ile çalışması için `<!DOCTYPE>` belge tanımlanması yapılmış olmalıdır.

Bir sonraki sayfadaki örnekte, herhangi bir öğenin içindeki aranan **ilk p** etiketini bulur ve metni altı çizgili yapar:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<style type="text/css">
p:first-child
{text-decoration:underline ;}
</style>
</head>

<body>
<p>CSS harika bir teknolojidir.</p>
<p>CSS harika bir teknolojidir.</p>
<p><b>Not:</b> first-child sözde sınıfının IE ile çalışması
için <!DOCTYPE> belge tanımlanması yapılmış olmalıdır.p>
</body>
</html>
```

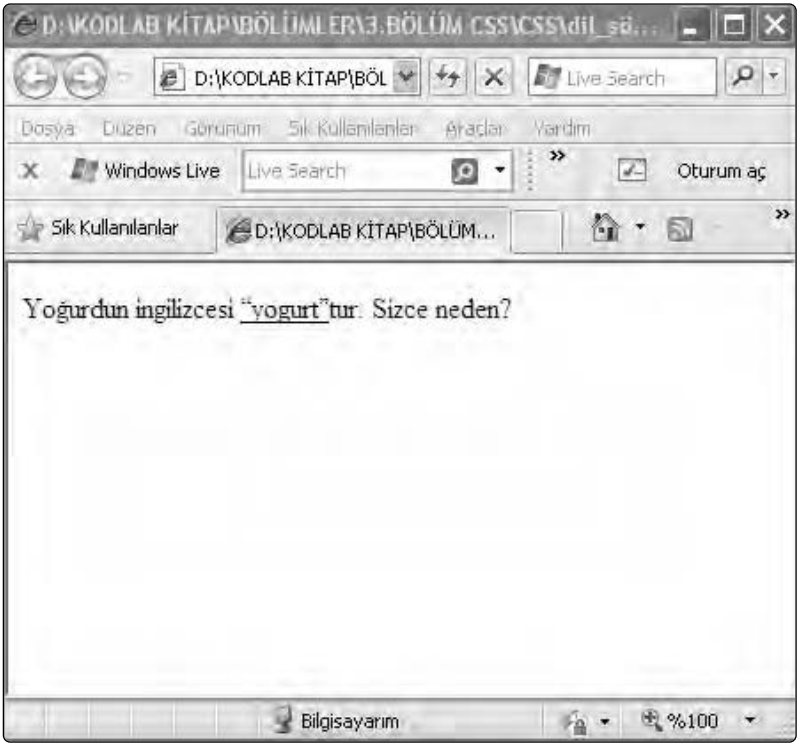


Dil (lang) Sözde Sınıfı

`lang` sözde sınıfı, farklı diller için farklı kurallar tanımlayabilmek içindir. Aşağıdaki örnekte `q` seçicisi için `:lang` sözde sınıfı ve İngilizce için dil kodu (`en`) ile farklı bir kural tanımlanmıştır.

```
<html>
<head>
<style type="text/css">
q:lang(en){text-decoration: underline;}
</style>
</head>

<body>
<p>Yoğurdun ingilizcesi <q lang="en">yogurt</q>tur. Sizce neden? </p>
</body>
</html>
```



Bu kısımda son olarak, **Sözde Sınıflar ile CSS Sınıflar**'ın birleştirilebileceklerini hatırlatalım. Örneğin aşağıdaki uygulamada link kırmızı olarak görüntülenecektir.

```
a.red:visited {color: #FF0000}
<a class="red" href="sınıflar.html">Sözde sınıflar ve CSS
sınıfları</a>
```

Anlattıklarımızı pekiştirmek bakımından bu örneği DVD'den bilgisayarınıza alıp mutlaka deneyin.

```
<html>
<head>
<style type="text/css">
a.one:link {color: #ff0000}
a.one:visited {color: #0000ff}
a.one:hover {color: #ffcc00}

a.two:link {color: #ff0000}
a.two:visited {color: #0000ff}
a.two:hover {font-size: 150%}

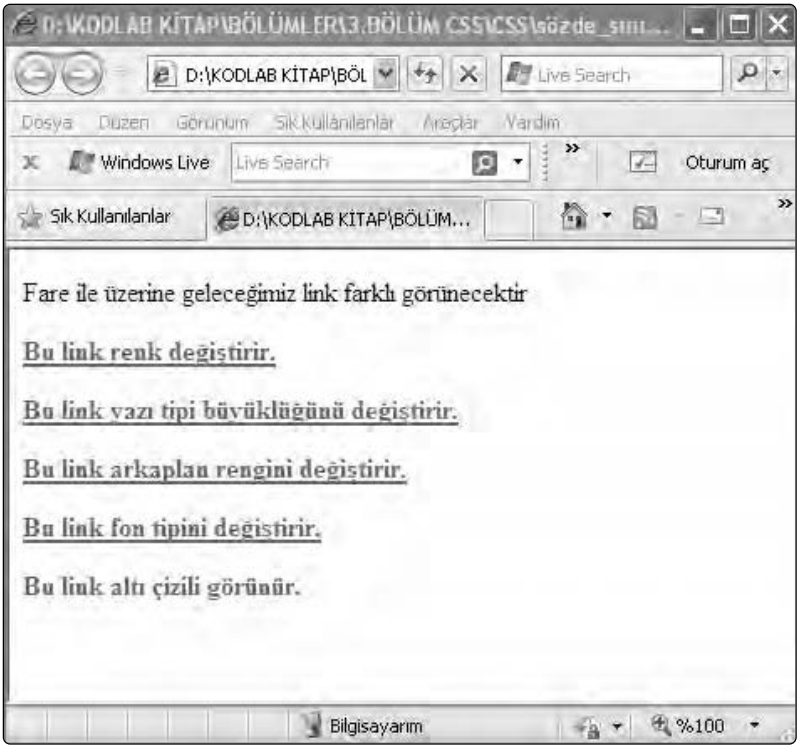
a.three:link {color: #ff0000}
a.three:visited {color: #0000ff}
a.three:hover {background: #66ff66}

a.four:link {color: #ff0000}
a.four:visited {color: #0000ff}
a.four:hover {font-family: monospace}

a.five:link {color: #ff0000; text-decoration: none}
a.five:visited {color: #0000ff; text-decoration: none}
a.five:hover {text-decoration: underline}
</style>
</head>

<body>
<p>Fare ile üzerine geleceğimiz link farklı görünecek-
tir</p>
<p><b><a class="one" href="index.html" target="_blank">Bu
link renk değiştirir.</a></b></p>
<p><b><a class="two" href="index.html" target="_blank">Bu
link yazı tipi büyüklüğünü değiştirir.</a></b></p>
<p><b><a class="three" href="index.html" target="_blank">Bu
```

```
link arkaplan rengini değiştirir.</a></b></p>
<p><b><a class="four" href="index.html" target="_blank">Bu
link fon tipini değiştirir.</a></b></p>
<p><b><a class="five" href="index.html" target="_blank">Bu
link altı çizili görünür.</a></b></p>
</body>
</html>
```



Linklerin üzerine fare ile tek tek gelip değişimi gözlemleyin.

Sözde-Öğeler (pseudo-elements)

Adı üstünde bunlar da sözle öğelerdir. XHTML hiyerarşisi içinde bu öğeler bulunmamaktadır. Belgedeki bir öğenin ilk satırına, hatta ilk harfine yönelik bir tasarımlamayı CSS öğeleri ile yapamayız. Bu durumda sözde öğeler belge dilinden (XHTML) soyutlamalar oluşturmuş olurlar.

Kullanımı

```
seçici:pseudo-öğesi{özellik: değer}
```

şeklindedir. CSS sınıfları, sözde sınıflarla olduğu gibi sözde öğelerle de kullanılabilir;

```
seçici.class:pseudo-öğesi{özellik: değer}
```

İlk Satır (first-line) Sözde Öğesi

`first-line` sözde öğesi, ilk satıra özel stiller atamak için kullanılır.

Yapısı

```
p:first-line {color:#0000ff;font-variant:small-caps}
```

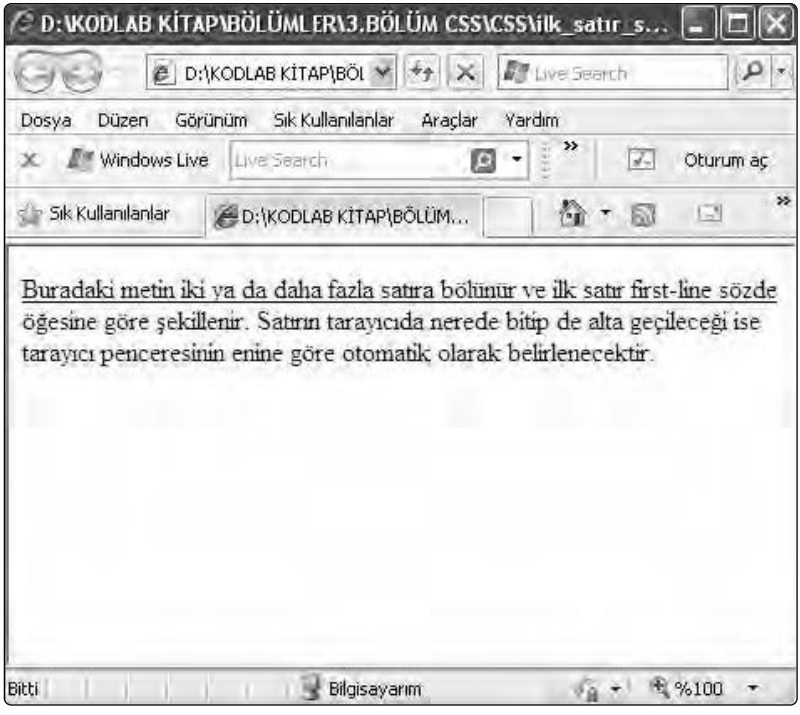
```
<p> Buradaki metin iki ya da daha fazla satıra bölünür ve ilk satır first-line sözde öğesine göre şekillenir. Satırın tarayıcıda nerede bitip de alta geçileceği ise tarayıcı penceresinin enine göre otomatik olarak belirlenecektir.</p>
```

NOT

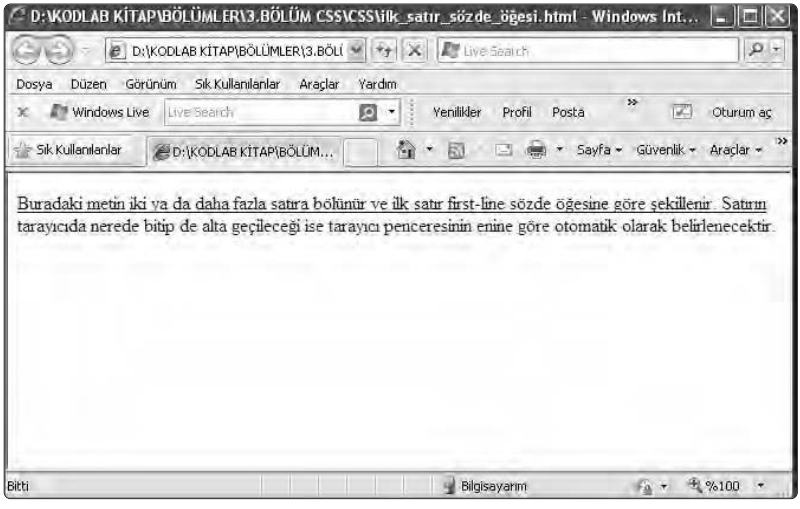
first-line sözde öğesine uygulanabilecek özellikleri; font properties, color properties, background properties, word-spacing, letter-spacing, text-decoration, vertical-align, text-transform, line-height, clear, olarak sayabiliriz.

```
<html>
<head>
<style type="text/css">
p:first-line {text-decoration: underline;}
</style>
</head>

<body>
<p>Buradaki metin iki ya da daha fazla satıra bölünür ve ilk satır first-line sözde öğesine göre şekillenir. Satırın tarayıcıda nerede bitip de alta geçileceği ise tarayıcı penceresinin enine göre otomatik olarak belirlenecektir.</p>
</body>
</html>
```



Tarayıcının eni dar olduğundan metin 3 satır oldu.



Tarayıcının enini büyüttüğümüzde 2 satıra düştü.

İlk Harf (first-letter) Sözde Öğesi

`first-letter` sözde öğesi bir belge öğesinin içindeki yazının ilk harfine özel stil atamak için kullanılır:

```
p:first-letter {color:#ff0000;font-size:xx-large}

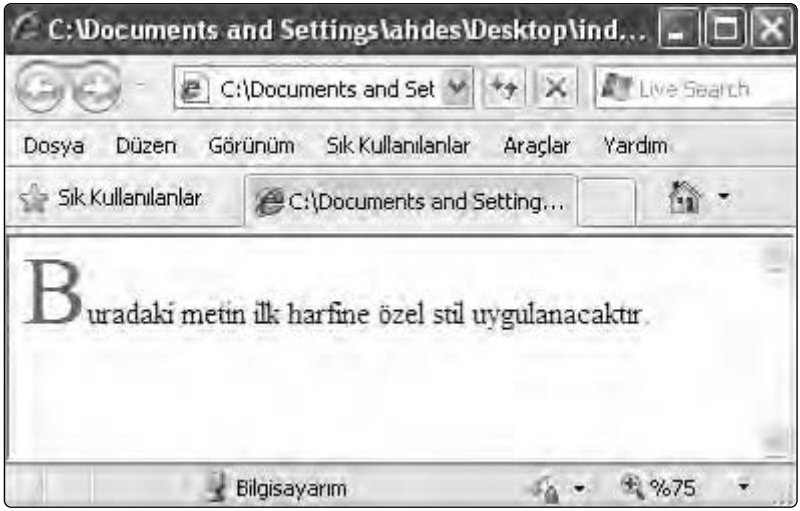
<p>İlk harfine stil uygulanacak metin.</p>
```

NOT

first-letter sözde öğesine uygulanabilecek özellikleri; font properties, color properties, background properties, margin properties, padding properties, border properties, text-decoration, vertical-align, text-transform, line-height, float, clear, olarak sayabiliriz.

```
<html>
<head>
<style type="text/css">
p:first-letter {color: #ff0000; font-size:xx-large}
</style>
</head>

<body>
<p>Buradaki metin ilk harfine özel stil uygulanacaktır.</p>
</body>
</html>
```



BEFORE SÖZDE ÖĞESİ

before sözde öğesi bir XHTML öğesinin önüne metin, ses ya da resim eklemek için kullanılır.

Ses eklemek: `p:before {content: url(ses.wav)}`

Resim eklemek: `p:before{content:url(smiley.gif)}`

Metin eklemek: `p:before {content:"Not:"}`

AFTER SÖZDE ÖĞESİ

after sözde öğesi, bir XHTML öğesinin arkasına metin, ses ya da resim eklemek için kullanılır. Yapısı ve kullanımı aynen `before` sözde öğesi gibidir.

STİLLERDE ÖNCELİK HAKKI

Bazı durumlarda bir XHTML öğesine uygulanmak üzere atanmış birden fazla stil söz konusu olabilir. Bu noktada CSS'in ilk C'si yani **Cascading** kavramı devreye girer. **Basamaklı** anlamına gelen **Cascading**, CSS'in stiller arasındaki katmanlı yapısını temsil eder.

Peki böyle bir durumda bu basamaklı yapıda öncelik hakkı hangi stilde olacaktır? İlk öncelik hakkı içinde **!important** özelliği barındıran stil-

dir. **!important** özelliği barındıran bir stil yoksa en altta yer alan stil üsttekilere göre öncelikli olacaktır.

```
<style type="text/css">
h1 {color: red !important;}
h1{color:blue;}
</style>

<body>
<h1> Stillerde Öncelik Hakkı</h1>
</body>
```

Yukarıdaki örneği denediğinizde metin renginin kırmızı olduğunu göreceksiniz. Eğer ilk stil tanımlaması içinden **!important** özelliğini silerseniz, bu sefer de metin rengi mavi olacaktır.

Nokta (.) Özelliği

Nokta özelliği, kitabın CSS bölümünü bitirmek için kullanılır. Pardon kitabımızın CSS bölümü bitti demek istemiştim. Böyle bir CSS özelliği yok galiba. Sanırım hızımı alamadım!

Neyse siz bana aldırmayın ve bir sonraki bölüme geçmeden önce biraz deneme yapın. Öğrendiklerinizin bilgi, yapacağınız denemelerin ise tecrübe ve ikisinin birden harika siteler demek olduğunu da lütfen unutmayın!

4

XML

Neden XML 149
Tarihçesi 150
HTML 151
XML Belge Yapısı 152
XML Belgesinin Bölümleri 154
İyi Oluşumlu XML Belgeleri (well-formed) 157
Geçerli XML Belgesi (Valid Document) 159
Belge Tipi Tanımlaması (Document Type Definition-DTD) 159
DTD Genel Yapısı 160
Öğe (element) Bildirimi 161
Öznitelikler 165
Öznitelik Türleri 166
Varsayılan Değerler 169
XML Şemaları (XML SCHEMA) 171
XML Belgelerini Biçimlendirme (CSS-XSL) 178
XSL Kullanımı 183

Evet işte şimdi de geldik **XML** bölümüne... Kısa, kolay anlaşılır –tıpkı önceki bölümler gibi!– ama çook önemli bir bölüm başlıyor. Neden çok önemli olduğunu biraz merak edin bakalım. Şaka bir yana, biraz sonra bunu detaylı olarak açıklayacağım. Diğer bölümlerde olduğu gibi bu bölümdeki örnek uygulamaların dosyalarını da DVD'de bulabilirsiniz. İyi okumalar.

Neden XML

Gelişen elektronik, bilgisayar ve yazılım teknolojileri birçok yeniliğin yanında doğal olarak birçok farklılığı da beraberlerinde getirdi. Farklı yapılar arasında veri iletişimi söz konusu olduğunda, doğal olarak ortaya, –bu farklılıklar sebebiyle– birçok sorun çıkıyordu. Bu durum **platform bağımsız veri iletimi** konusunu daha da önemli hale getirdi.

Platform bağımsız veri iletimi, veri alışverişi yapan sistemler arasındaki farklılıklara rağmen sağlıklı veri iletimini gerçekleştirebilmek anlamına gelmektedir.

Mesele sadece internet sitelerini ve web tasarımcıları ilgilendiren bir konu değildir. Masaüstü sistemlerden, yerel ağlara, hatta değişik elektronik cihazlara kadar uzanan çok geniş bir yelpazeye hitap eden bir konudur –bunlara cep telefonları ve son sistem çamaşır makinelerinin programları da dahildir– Fakat biz sadece webcileri (ne demekse!) ilgilendiren kısmından söz edeceğiz.

Evet, internet ortamında sunucularda kullanılan platformlar (*Windows*, *Linux*), birçok sunucu yazılımları, PHP, ASP.NET... vb web programlama dilleri, bir çok popüler web tarayıcıları derken tüm bu devasa yapının içinde bir platform bağımsız işaretleme dili mutlaka gerekmektedir.

Buraya kadar yaptığım açıklamalarla umarım sizi, bir platform bağımsız yapıya ihtiyaç olduğuna ikna edebilmişimdir. O zaman şimdi de, yeryüzünde bu konuya nasıl yaklaşıldığına ve ne gibi çalışmalar yapıldığına bakalım.

TARİHÇESİ

İşaretleme dili konusunu **XHTML** bölümünden hatırlayacaksınız. Tekrar etmek gerekirse; "işaretleme dili, metinlerin nasıl yapılandırılacağını ve biçimlendirileceğini birtakım işaretlerle (etiket) belirleyen dildir" diyebiliriz.

İlk işaretleme dili **GML** *(**G**eneralized **M**arkup **L**anguage)* 1960 yılında IBM firması tarafından geliştirilmiştir. 1978 yılında ise GML bir grup araştırmacı tarafından daha da geliştirilerek **SGML** (**S***tandardized* **G***eneralized* **M***arkup* L*anguage*) işaretleme dili oluşturuldu. SGML 1986 yılında **ISO** (*the* **I***nternational* **O***rganization for* **S***tandardization*) kurumu tarafından uluslararası bir standart olarak kabul edildi. Çok güçlü bir dil olan SGML; otomotiv, basın... vb bir çok endüstri kolunda kullanılmasına karşın, karmaşık bir yapıya sahip olması ve bu dille uygulama geliştirme maliyetinin yüksek olması sebebiyle kullanımı çok da fazla yaygınlaşamamıştır.

Word Wide Consortium (W3C, *www.w3.org*) tarafından 1996 senesinde, SGML altyapısı ile ve veri iletimini kolaylaştıracak basit bir işaretleme dili oluşturmak için yapılan çalışmalar, XML dilinin tasarlanmasıyla sonuçlandı. Yapılan çalışmalar neticesinde 1998 yılında XML dili,

XML 1.0 olarak bir standart haline geldi. Microsoft, Oracle... vb birçok büyük firma kısa sürede bu dili benimsediler ve uygulamalarını bu standarda uygun hale getirdiler.

Kısaca tarihçesinden bahsettiğimiz XML (*Extensible Markup Language*), kullanıcılar tarafından yapılandırılan ve iletilecek bilginin tanımlanmasına yarayan bir işaretleme dilidir. XML, SGML meta dilinin (dil oluşturmayı sağlayan bir üst dil) bir alt kümesidir. XML aynı zamanda SGML gibi bir meta dildir. XML ile de başka alt diller oluşturmak mümkündür. Örneğin cep telefonlarında kullanılan WAP sayfalarının bilgi tanımları, XML meta dilinden türetilmiş olan WML dili ile yazılmaktadır.

HTML

1989 yılında Tim Berners-Lee ve Anders Berlung tarafından geliştirilen **HTML** (**H***yper***t***ext* **M***arkup* **L***anguage*) dili de yukarıda bahsettiğimiz SGML dilinin alt yapısını kullanarak hazırlanmış bir işaretleme dilidir. O halde bu işi HTML ile yapsak da başka bir dille uğraşmasak nasıl olur?

Bu sorunun cevabı hiç kuşkusuz "gayet güzel olur" olacak değil mi? Peki size; "HTML dili, internet ortamında görüntülenecek belgelerdeki metin başlığı, yazı tipi, metne eklenen resim, tablo... vb materyalleri biçimlendirir. Bu amaçla standart olarak belirlenmiş sabit bir (*tag*) etiket kümesi sunar. Verilerin tanımlanması, paylaşımı ve iletimi konularında genişleyebilir bir dil değildir" desem bu size neyi hatırlatır?

Evet doğru cevap. İşte biz bu yüzden (X)HTML görmüştük diyorsunuz değil mi? (X)HTML standartlarının HTML ve XML standartlarını evlendiren (!?) bir sistem olduğunu ve geleceğin bu standartta olduğunu siz zaten biliyorsunuz. İşte bu geleceğin standardının bir kanadını oluşturan XML dilinin temellerini şimdi öğrenmeye başlayacağız.

Önce bir html bir de XML örneği yapalım. HTML örneği ile hafızamızı tazeleyip, xml örneği ile de XML diline giriş yapmış oluruz.

Örnek HTML belgesi

```
<html>
<head>
<title> Örnek html belgesi </title>
```

```
</head>
<body>
   <strong><h1>DENEME YAZISI </h1></strong></body>
</html>
```

Örnek HTML belgesi

```
<kitap>
   <adı>10 Adımda Web Tasarımı</adı>
   <yazarı>Uğur GELİŞKEN</yazarı>
   <sayfa_sayısı>248</sayfa_sayısı>
   <ebatları>13.7x19.5</ebatları>
   <fiyatı>19.50 TL</fiyatı>
</kitap>
```

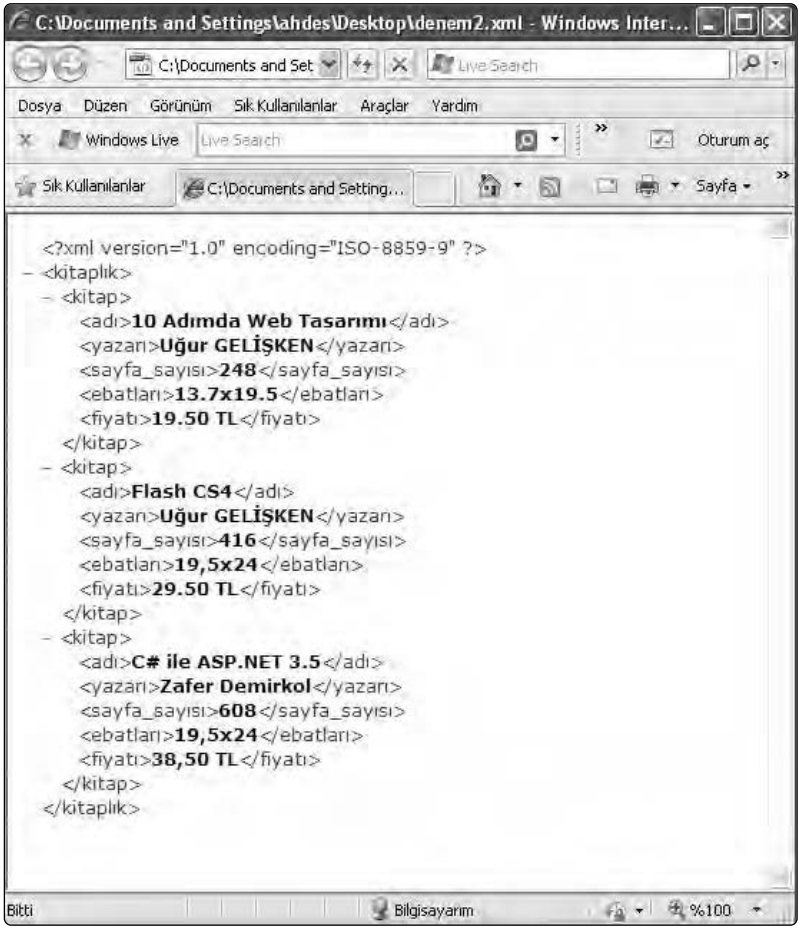
XML Belge Yapisi

XML, belgenin bir ağaç yapısında oluşturulmasına olanak tanır. Öncelikle bu yapının anlaşılması XML belge mimarisinin ve öğeler arasındaki hiyerarşinin kavranması açısından önemlidir. Aşağıdaki ilk XML örneğimizde, bir kitaplıktaki çeşitli bilgisayar kitaplarının XML belge hiyerarşisine göre nasıl yapılandırılması gerektiğini görüyorsunuz.

```
<?xml version="1.0" encoding="utf-8"?>
<!—Bu bir yorum cümlesidir ve açıklama amaçlıdır.—>
<kitaplık>
   <kitap>
          <adı>10 Adımda Web Tasarımı</adı>
          <yazarı>Uğur GELİŞKEN</yazarı>
          <sayfa_sayısı>248</sayfa_sayısı>
          <ebatları>13.7x19.5</ebatları>
          <fiyatı>19.50 TL</fiyatı>
   </kitap>
   <kitap>
          <adı>Flash CS4</adı>
          <yazarı>Uğur GELİŞKEN/yazarı>
          <sayfa_sayısı>416</sayfa_sayısı>
          <ebatları>19,5x24</ebatları>
          <fiyatı>29.50 TL</fiyatı>
   </kitap>
   <kitap>
```

```
            <adı>C# ile ASP.NET 3.5</adı>
            <yazarı>Zafer Demirkol</yazarı>
            <sayfa_sayısı> 608</sayfa_sayısı>
            <ebatları>19,5x24</ebatları>
            <fiyatı>38,50 TL</fiyatı>
   </kitap>
</kitaplık>
```

Oluşturduğumuz XML belgesinin tarayıcıdaki görünümü aşağıdaki gibidir.



XML Belgesinin Bölümleri

XML Bildirimi

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
```

Bu bildirim dosyanın bir XML dosyası olduğunu belirtir. XML versiyonu ve dosyanın karakter kodlamasını da belirler. Belgenin ilk satırında mutlaka bulunmalıdır.

Ayrıca; `standalone` özelliği belgenin dış bildirim durumunu belirler. Dış işaretleme bildirimlerinin XML belgemizin içeriğini etkilemediği durumlarda bu özelliğe `yes` değerinin atanması ile dış dosyaların gerekmediği halde işlenmesi engellenmiş olur. Tersi durumda ya `no` değeri kullanılır ya da `standalone` özelliği hiç kullanılmaz.

İşlem Yönergeleri

Bir işlem yönergesini diğer işaretlemelerin haricinde belgedeki herhangi bir yerine yerleştirebiliriz.

```
<?xml-stylesheet type="text/css" href=xmlbelgemizinadı.css"?>
```

Örneğin yukarıdaki işlem yönergesi ile, XML uygulamamıza dahil edilen verileri sağlayan dosyaları XML bildiriminin altına ekleyebiliriz.

Öğeler (Elements)

```
<kitaplık>
   <kitap>
          <adı>10 Adımda Web Tasarımı</adı>
          <yazarı>Uğur GELİŞKEN</yazarı>
          <sayfa_sayısı>248</sayfa_sayısı>
          <ebatları>13.7x19.5</ebatları>
          <fiyatı>19.50 TL</fiyatı>
   </kitap>
   <kitap>
          <adı>Flash CS4</adı>
          <yazarı>Uğur GELİŞKEN/yazarı>
          <sayfa_sayısı>416</sayfa_sayısı>
          <ebatları>19,5x24</ebatları>
```

```
        <fiyatı>29.50 TL</fiyatı>
    </kitap>
    <kitap>
        <adı>C# ile ASP.NET 3.5</adı>
        <yazarı>Zafer Demirkol</yazarı>
        <sayfa_sayısı> 608</sayfa_sayısı>
        <ebatları>19,5x24</ebatları>
        <fiyatı>38,50 TL</fiyatı>
    </kitap>
</kitaplık>
```

Bu bölüme belgemizin ana gövdesidir diyebiliriz. Bunu html'deki <body> </body> etiketleri arasındaki kısma benzetebiliriz. Yani belgenin asıl içeriğinin yer aldığı bölüm burasıdır. Yukarıdaki örnekte de gördüğünüz gibi bir XML belgesi iç içe birçok öğe grubu barındırabilir. Ancak öğeler **yuvalama** (*nesting*) kurallarına uymalıdırlar.

Örnek

```
<kitap><yazarı> Bu kullanım yanlıştır.</kitap></yazarı>
<kitap><yazarı> Bu kullanım doğrudur.</yazarı></ kitap>
```

Etiketler (*tag*) ise <kitap> </kitap> örneğinde de görüldüğü gibi öğelerin başlangıç ve bitiş tanımlamalarıdır. Etiketlerin arasında ise içeriği oluşturan bilgiler bulunur. Bu bölüm **karakter verisi** (*character data*) olarak adlandırılır.

```
[Öğeler= etiketler + içerikler(karakter verisi)]
```

İçeriklerde, **küçüktür** işareti (<), **ampersand** (&) karakteri ya da]]> dizesi kullanılamaz. Çünkü bunlar öğe tanımlamalarında kullanılır.

XML dilinde etiketlerin içindeki veride bulunan boşluklar (*White Space*) yok sayılmaz. Fakat etiket ya da öznitelik tanımlarındaki boşluklar hesaba katılmaz.

CDATA Bölümleri

CDATA bölümleri belgemizde veri olarak algılanacak olan bölümlerdir. Bir CDATA bölümü <![CDATA[karakter dizesiyle başlayıp]]> karak-

ter dizesi ile biter. Buraya yazacağımız karakterler XML işaretlemeleri olarak değil, bilgi bölümü olarak algılanacaktır.

```
<bilgi>
<![CDATA[ & ve < karakterlerini içeriklerde kullanmayınız.]]>
</bilgi>

<bilgi <![CDATA[ & ve < karakterlerini içeriklerde kullanmayınız.]]> >
</bilgi>
<!— CDATA bölümleri XML işaretlemelerinin içine konulamazlar.—>
```

Bu sebeple (işaretleme olmadığından) CDATA bölümleri XML işaretlemelerinin içine konulamazlar. Öğe içeriklerinin konulduğu yere yazılırlar. Bu itibarla yukarıdaki ikinci kullanım şekli yanlıştır. Ayrıca iki ayrı CDATA bölüm iç içe de yapılandırılamaz.

YORUMLAR

```
<!—Bu bir yorum cümlesidir ve açıklama amaçlıdır.—>
```

XML belgelerinde yorumlar <!— —> dizeleri arasına eklenirler. Sadece XML işaretlemelerinin içine konulamazlar ve iç içe konumlandırılamazlar.

ÖZNİTELİKLER (ATTRIBUTE)

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<kitaplık>
   <kitap>
          <adı>Flash CS4</adı>
          <yazarı>Uğur GELİŞKEN</yazarı>
          <sayfa_sayısı>416</sayfa_sayısı>
          <ebatları>19,5x24</ebatları>
          <fiyatı tür="parakende">29.50 TL</fiyatı>
   </kitap>
</kitaplık>
```

Öznitelikler öğeler ile ilgili birtakım bilgilerin paylaşımını sağlamak için kullanılır. Yukarıdaki örneğimizde `tür` özniteliği `<fiyatı>29.50 TL</fiyatı>` öğesi ile ilgili bir bilgi (kitaba ait fiyatın perakende fiyatı olduğu bilgisi) iletmektedir.

```
<kitap>
        <fiyatı tür="perakende"29.50 TL</fiyatı>
</kitap>

<kitap>
        <fiyatı tür="perakende" tür="toptan" >19.50 TL</fiyatı>
</kitap>
<!— Aynı başlangıç etiketi içinde bir öznitelik iki kez
kullanılamaz.—>
```

Aynı başlangıç etiketi içinde bir öznitelik adını sadece bir kez kullanabiliriz. Dolayısıyla ikinci örnekte kullanım şekli yanlıştır.

İyi Oluşumlu XML Belgeleri (well-formed)

Web tarayıcıları için gerekli olan standartlara sahip bir XML belgesine "iyi oluşumlu (well-formed)" XML belgesi diyoruz.

İyi oluşumlu bir XML belgesinde

- Mutlaka diğer tüm etiketleri (*tag*) kapsayan bir üst etiket bulunur. Örneğimizde `<kitaplık> </kitaplık>` üst etikettir. XML bu şekilde bir ağaç yapısına sahiptir.

```
<kitaplık>
          <kitap>
                  <adı>10 Adımda Web Tasarımı</adı>
                  <yazarı>Uğur GELİŞKEN</yazarı>
                  <sayfa_sayısı>248</sayfa_sayısı>
                  <ebatları>13.7x19.5</ebatları>
                  <fiyatı>19.50 TL</fiyatı>
          </kitap>
</kitaplık>
```

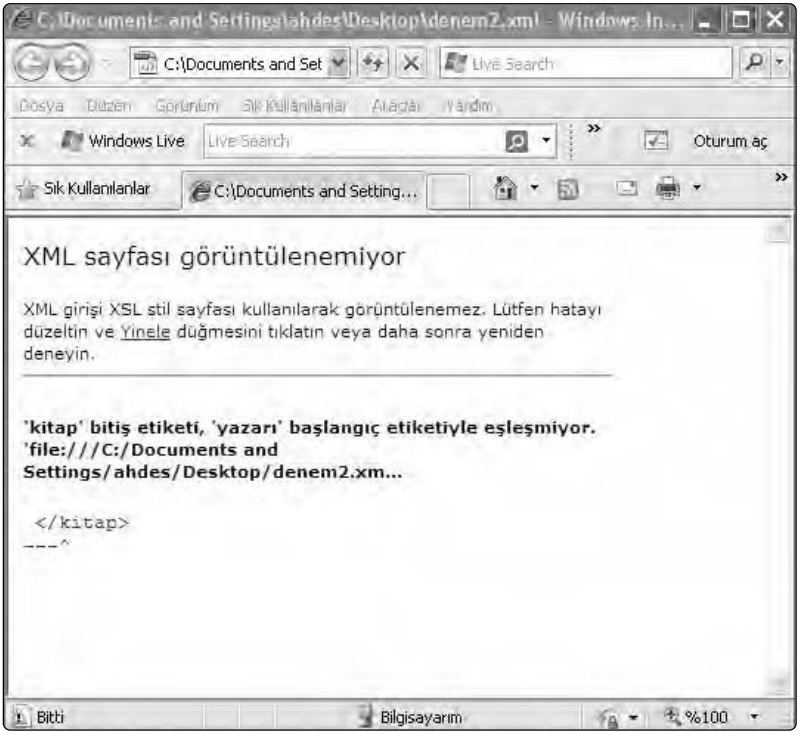
- Her öğenin başlangıç ve bitiş etiketleri içine konumlandığı üst öğenin etiketleri içinde bulunmalıdır. Örneğimizdeki `<kitap></kitap>` etiketleri gibi.

```
<kitaplık>
          <kitap>
                  <adı>10 Adımda Web Tasarımı</adı>
                  <yazarı>Uğur GELİŞKEN</yazarı>
```

```
                    <sayfa_sayısı>248</sayfa_sayısı>
                    <ebatları>13.7x19.5</ebatları>
                    <fiyatı>19.50 TL</fiyatı>
                </kitap>
</kitaplık>
```

- Her öğenin mutlaka başlangıç ve bitiş etiketi olmak zorundadır. Yanı sıra her öğe adı başlangıç ve bitiş etiketlerinde aynı olmalıdır. Öğe adları büyük/küçük harfe duyarlı olduğundan başlangıç ve bitiş etiketlerinde bu hususa da özellikle dikkat edilmelidir. `<kitap> </KİTAP >` ve benzeri kullanımlar hatalıdır.

Bir XML belgesi "iyi oluşumlu (well-formed)" olmadığında ekran çıktısı aşağıdaki resimdeki gibi olacaktır. Belgemiz yerine bir hata mesajıyla karşılaşacağız. Yani belgemiz "iyi oluşumlu (well-formed)" olmadığında, onu tarayıcılara görüntülettiremezsiniz. Onun için belgenizi bu standarta uyduracaksınız başka yolu yok.



Geçerli XML Belgesi (Valid Document)

Geçerli (*Valid*) XML belgesi kavramı, iyi oluşumlu XML belgesi kavramını da içine alan ama ondan daha genel ve artı bazı ek kuralları ifade eden bir kavramdır. Tüm XML belgeleri iyi oluşumlu olmak zorundadır. Bir belgeye geçerli XML belgesi diyebilmemiz için iyi oluşumlu olmasının yanı sıra;

- Belgemizin başlangıcında belgenin yapısını tanımlayan bir (**DTD: D***ocument* **T***ype* **D***efinition*) olmalıdır.
- Belge, başlangıcındaki DTD'de tanımlanan yapıya uygun olmalıdır.

Burada elde edilmek istenen amaç, belgeyi standartlara uymaya zorlamaktır. XML işlemcisi, belgenin geçerliliğini denetlediğinde belgenin bir yerinde DTD'ye uymayan bir kullanım varsa bir hata iletisi görüntülenmesini sağlar. Yani standartlara uymazsanız, burada da yine başınıza biraz evvelkine benzer bir durum gelecek.

Belge Tipi Tanımlaması (Document Type Definition-DTD)

XML dilinde yazacağımız belgelerde, belge standartlarını biz oluştururuz. Burada XML dillinin standartları ile XML dilinde oluşturulmuş bir belgenin standartlarını birbirine karıştırmamalıyız. XML dilinin standartlarını **W3C-Word Wide Consortium** belirliyor. Biz ise kendi oluşturacağımız belgelerin standartlarını belirliyoruz. **Extensible**, yani genişletilebilir olmak işte bu. Yani belgemizi yapma amacımıza göre esnetip şekillendirmek imkânına sahibiz.

HTML dilindeki etiketlere de mahkûm değiliz. Bundan iyisi Şam'da kayısı. İşte bu sebeple belgemizin standartlarını oluştururken, hangi öğelerin ve özniteliklerin kullanılabileceğini, hangilerinin ise mutlaka kullanılması gerektiğini tanımlamamız gerekir. Bu kurallar kümesini de **şema** (*schema*) olarak adlandırıyoruz. Eğer belgemiz, hazırlayacağımız şemanın kurallarına tam olarak uyuyorsa bu belgeye, detaylarına yukarda değindiğimiz **geçerli** (*valid*) belge diyoruz.

Söz konusu bu şemaları oluşturabilmek için iki yolumuz var. DTD ve XML şemaları... Biz bu bölümde DTD 'leri inceleyeceğiz. XML şemaları ise bir sonraki bölümde sizinle olacak.

DTD Genel Yapisi

İlk olarak, bu tanımlama bilgisinin

```
<?xml version="1.0" encoding=" ISO-8859-9" standalone="yes" ?>
```

şeklindeki XML bildiriminden sonra gelmek koşuluyla belgenin her hangi bir bölümünde kullanılabileceğini belirtelim.

Bu bildirim iki şekilde olabilir. İlk yol, belgenin içinde **iç DTD** olarak kullanmaktır. İkinci yol ise belge dışındaki bir başka kaynaktaki bildirimi **dış DTD** bildirimi olarak kullanmaktır.

İç DTD genel biçimi

```
<!DOCTYPE öğe_adı DTD>
```

şeklindedir.

Dış DTD genel biçimi ise

```
<!DOCTYPE öğe_adı SYSTEM " kaynak belgenin yolu">
```

şeklindedir.

DTD "[" ayıracı ile başlar, "]" ayracı ile biter.

DTD Dilbilgisi

DTD'ler üç adet temel bileşen içermektedirler.

Bunlar:

- Öğe (element) bildirimi
- Öznitelikler (attributes)
- Varlık (entity) bildirimleri

Öğe (element) Bildirimi

Geçerli bir belgede, belgede kullanılacak öğelerin bildirimlerinin yer alması gerekir. Bunun sebebi, bu konunun başında da söylediğimiz gibi belgeye uygulanacak kuralları belirleyebilmektir. Bu kuralları belirlememiş olsak, takdir edersiniz ki o zaman bu belge herkes tarafından kullanılabilir platform bağımsız bir sistem olamayacaktır.

Genel biçimi

```
<!ELEMENT Öğe_Adı İçerik>
```

şeklindedir.

Bir öğenin (element) içeriği dört farklı şekilde belirlenebilir.

1. **<!ELEMENT Öğe_Adı EMPTY>** şeklindeki içeriğin boş olduğu durumdur.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!DOCTYPE kitaplık [
   <!ELEMENT kitaplık (kitap)>
   <!ELEMENT kitap (ilk_kitap)>
   <!ELEMENT ilk_kitap EMPTY>
]>

<kitaplık>
   <kitap>
          <ilk_kitap />
   </kitap>
</kitaplık>
```

2. **<!ELEMENT Öğe_Adı ANY>** olduğu durumdur. Serbest içerik tanımlamasıdır. İçerik kısıtlaması yoktur ve birden çok alt öğe içerebilir. Alt öğeler birbirlerinden , (*virgül*) işareti ile ayrılır. Öğe bir değere sahipse parantez içinde #PCDATA (**PCDATA: P***arsed* **C***haracter* **Data**) etiketi de yer almalıdır.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!DOCTYPE kitaplık [
   <!ELEMENT kitaplık (kitap)>
   <!ELEMENT kitap ANY>
```

```
   <!ELEMENT adı (#PCDATA)>
   <!ELEMENT yazarı (#PCDATA)>
   <!ELEMENT fiyatı (#PCDATA)>
]>
<kitaplık>
   <kitap>
          <adı>Flash CS4</adı>
          <yazarı>Uğur GELİŞKEN</yazarı>
          <fiyatı>29.50 TL</fiyatı>
          <fiyatı>28,00 TL</fiyatı>
   </kitap>
</kitaplık>
```

3. **<!ELEMENT Öğe_Adı (Yuvalanmış Öğe) >** olduğu durumdur. Öğenin kendi içeriğinin olmadığı sadece alt öğeleri içerdiği (*Element Only*) anlamına gelir.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!DOCTYPE kitaplık [
   <!ELEMENT kitaplık (kitap)>
   <!ELEMENT kitap (adı, yazarı)>
   <!ELEMENT adı (#PCDATA)>
   <!ELEMENT yazarı (#PCDATA)>
   <!ELEMENT fiyatı (#PCDATA)>
]>

<kitaplık>
   <kitap>
          <adı>Flash CS4</adı>
          <yazarı>Uğur GELİŞKEN</yazarı>
   </kitap>
</kitaplık>
```

XML dilinde çeşitli amaçlarla kullanılan bazı öğe içerik tanımları ve anlamları yan sayfada listelenmiştir:

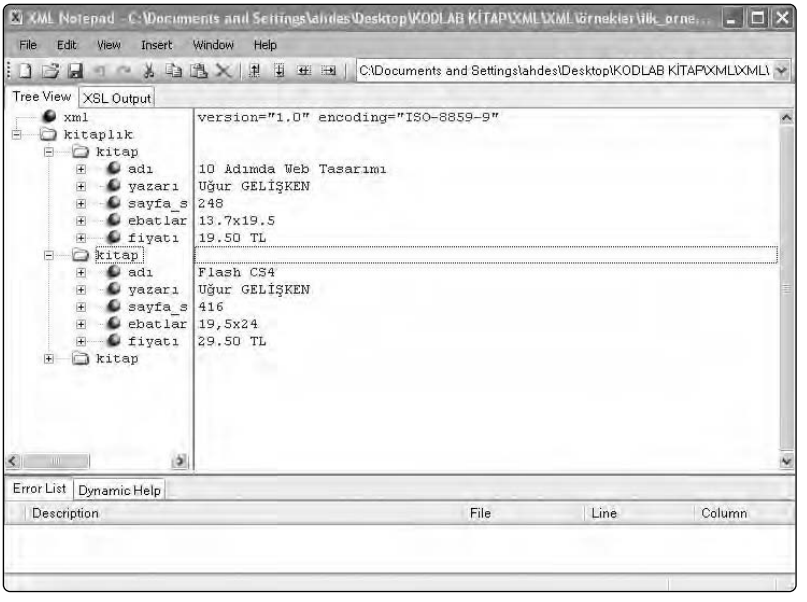
Öğe Tanımı	Anlamı
A?	A alt öğesi bir defa kullanılabilir ya da hiç kullanılmayabilir.
A+	A alt öğesi mutlaka en az bir kez ya da daha fazla kullanılmalıdır.
A, B	B öğesi A öğesinden sonra kullanılmalıdır.
A | B	A ya da B öğesinden sadece her hangi bir tanesi kullanılabilir.
A*	A öğesi istenildiği kadar kullanılabilir.

Bizim örneğimizde `<!ELEMENT kitap (adı, yazarı)>` "adı" öğesi "yazarı" öğesinden önce kullanılacaktır.

4. **`<!ELEMENT Öğe_Adı (#PCDATA, Yuvalanmış Alt Öğe)>`** olduğu durumdur. Yapısından da anlaşılacağı gibi hem öğe verisi hem alt öğe içeren **karışık** (*mixed*) bir yapıdır.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!DOCTYPE kitaplık [
   <!ELEMENT kitaplık (kitap)>
   <!ELEMENT kitap (adı, yazarı, fiyatı)>
   <!ELEMENT kitap (#PCDATA)>
   <!ELEMENT adı (#PCDATA)>
   <!ELEMENT yazarı (#PCDATA)>
   <!ELEMENT fiyatı (#PCDATA)>
]>
<kitaplık>
   <kitap>
          Kitap hakkındaki bilgiler
          <adı>Flash CS4</adı>
          <yazarı>Uğur GELİŞKEN</yazarı>
          <fiyatı>29.50 TL</fiyatı>
   </kitap>
</kitaplık>
```

Bu noktada yeri gelmişken, hem belgelerinizin geçerliliğini kontrol edebilmeniz için hem de xml belgeleriyle daha rahat çalışabilmeniz için **XML Notepad** programını kullanabileceğinizi hatırlatmak isterim. Programı, **www.microsoft.com** sitesinden ücretsiz olarak indirebilirsiniz.



Şimdi isterseniz bir de harici dtd örneği verelim.

```
<?xml version="1.0" encoding="ISO-8859-9"?>

   <!ELEMENT kitaplık (kitap)>
   <!ELEMENT kitap (adı,yazarı,fiyatı)>
   <!ELEMENT kitap (#PCDATA)>
   <!ELEMENT adı (#PCDATA)>
   <!ELEMENT yazarı (#PCDATA)>
   <!ELEMENT fiyatı (#PCDATA)>
```

Yukarıdaki belge **harici_dtd.dtd** adıyla kaydettiğimiz harici bir belge. XML belgemiz ise aşağıdaki gibidir.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!DOCTYPE kitaplık SYSTEM "harici_dtd.dtd">

<kitaplık>
   <kitap>
          Kitap hakkındaki bilgiler
          <adı>Flash CS4</adı>
```

```
        <yazarı>Uğur GELİŞKEN</yazarı>
        <fiyatı>29.50 TL</fiyatı>
   </kitap>
</kitaplık>
```

Görüldüğü gibi DTD'yi XML belgemize dahil etmeyip sadece **harici belgemizi** (*harici_dtd.dtd*) tanımladık.

ÖZNİTELİKLER

Özniteliklerin öğeler ile ilgili birtakım bilgilerin paylaşımını sağlamak için kullanıldığını daha önce söylemiştik. DTD bildirimlerinde öznitelikler için de kurallar tanımlanır.

Genel yapısı

```
<!ATTLIST Öğe_Adı Öznitelik_Adı Öznitelik_Türü Varsayılan_Değer >
```

şeklindedir.

ATTLIST ifadesi XML işlemcisi tarafından bu yapının bir öznitelik tanımlaması olarak algılanıp işlenmesini sağlar.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!DOCTYPE kitaplık [
   <!ELEMENT kitaplık (kitap)>
   <!ELEMENT kitap (adı,yazarı,fiyatı)>
   <!ELEMENT kitap (#PCDATA)>
   <!ELEMENT adı (#PCDATA)>
   <!ELEMENT yazarı (#PCDATA)>
   <!ELEMENT fiyatı (#PCDATA)>
<!ATTLIST fiyatı satış CDATA  #IMPLIED>
]>

<kitaplık>
   <kitap>
          Kitap hakkındaki bilgiler
          <adı>Flash CS4</adı>
          <yazarı>Uğur GELİŞKEN</yazarı>
          <fiyatı satış="parakende">29.50 TL</fiyatı>
   </kitap>
</kitaplık>
```

Yukarıdaki örneğimizde `<!ATTLIST fiyatı satış CDATA #IMPLIED> ;`

```
fiyatı - Öğe_Adı
satış - Öznitelik_Adı
CDATA  - Öznitelik_Türü
#IMPLIED - Varsayılan_Değer;
```

olarak bildirimde yer almaktadır.

Öznitelik Türleri

Yukarıdaki örnekte gördüğünüz CDATA'nın yanı sıra, aşağıdaki öznitelik türleri de belgedeki öznitelik değerlerini kontrol edebilmek amacıyla DTD içerisinde kullanılırlar.

- CDATA
- ENTITY, ENTITIES
- Enumarated
- ID
- IDREF, IDREFS
- NMTOKEN, NMTOKENS
- NOTATION

Şimdi bunları tek tek inceleyeceğiz.

CDATA

En geniş kapsamlı öznitelik türüdür. **Küçüktür** (<), **ampersand** (&) ve **tırnak** işareti (“”) dışında kalan tüm karakterleri ve rakamları içerebilir.

```
<!ATTLIST fiyatı satış CDATA  #IMPLIED>
<fiyatı satış=”parakende”>29.50 TL</fiyatı>
```

ENTITY-ENTITIES

ENTITY özniteliği, dış varlık başvurularının yalnızca DTD'de tanımlanan varlıkların adlarını ihtiva edecek şekilde kullanılmasıdır. **ENTITIES** özniteliği de ENTITY özniteliğine benzer. Ancak birden fazla varlık başvurusu eklenmesine olanak tanır. Bu konuya varlıklar kısmında tekrar döneceğiz.

Enumarated (Sıralı) Öznitelikler

Bu öznitelikler öznitelik değerini belli bir ad kümesinden alırlar. Değerler birbirlerinden (|) işaretleri ile ayrılırlar.

```
<!ATTLIST kitap fiyatı (20,75 | 32,25 | 19,25) >
```

Örneğimizde kitap öğesini birçok fiyatının olabileceği görülmektedir.

ID Öznitelikleri

Öğeler eşsiz bir ID öznitelik değeri almalıdırlar. Diğer bir ifadeyle belgede yalnızca bir kez geçebilirler. ID değeri #REQUIRED ya da #IMPLIED ile tanımlanmak durumundadır. #FIXED ile kullanılmaz.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!DOCTYPE kitaplık [
   <!ELEMENT kitaplık (kitap)>
   <!ELEMENT kitap (adı,yazarı,fiyatı)>
   <!ELEMENT kitap (#PCDATA)>
   <!ELEMENT adı (#PCDATA)>
   <!ELEMENT yazarı (#PCDATA)>
   <!ELEMENT fiyatı (#PCDATA)>
<!ELEMENT kayıt_no (#PCDATA)>
<!ATTLIST kayıt_no isbn ID  ##REQUIRED>

]>


<kitaplık>
   <kitap>
          Kitap hakkındaki bilgiler
          <adı>Flash CS4</adı>
          <yazarı>Uğur GELİŞKEN</yazarı>
          <fiyatı satış="parakende">29.50 TL</fiyatı>
          <kayıt_no isbn="975-297">1</kayıt>
   </kitap>
</kitaplık>
```

IDREF-IDREFS Öznitelikleri

IDREF ve IDREFS öznitelik türleri belgemizde kullanılan (örn: yukarıdaki belge) ID öznitelik türüne başvurabilmek yani o özniteliğe referans

verebilmek içindir. IDREFS öznitelik türünün farkı birden çok ID özniteliğine başvurabilir.

NMTOKEN-NMTOKENS Öznitelikleri

NMTOKEN yada NMTOKENS öznitelikleri kullanılabilecek değerleri geçerli xml adları ile sınırlar. Değer, bir harf ya da alt – (çizgi) ile başlamalıdır. NMTOKEN özniteliğin değerlerini boşlukla ayırmanıza izin vermez.

Eğer bir özniteliğin değerinin yalnız geçerli bir XML adı değil, aynı zamanda da belgede eşsiz olmasını istiyorsanız özniteliğin türünü NMTOKEN değil ID kullanın.

NOTATION Özniteliği

NOTATION öznitelik türü resim, ses... vb dosyaların çalıştırılabilmeleri için gereken uygulamaları tanımlamaya yarar.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!DOCTYPE kitaplık [
   <!ELEMENT kitaplık (kitap)>
   <!ELEMENT kitap (adı,yazarı,fiyatı)>
   <!ELEMENT kitap (#PCDATA)>
   <!ELEMENT adı (#PCDATA)>
   <!ELEMENT yazarı (#PCDATA)>
   <!ELEMENT fiyatı (#PCDATA)>
   <!ELEMENT imaj EMPTY>
   <!ATTLIST imaj kapak ENTITY #REQUIRED>
   <!ENTITY resim SYSTEM "kapak_resmi.jpg" NDATA kapak_resmi>
   <!NOTATION kapak_resmi SYSTEM "jpgviewer.exe">
]>

<kitaplık>
   <kitap>
          Kitap hakkındaki bilgiler
          <adı>Flash CS4</adı>
          <yazarı>Uğur GELİŞKEN</yazarı>
          <fiyatı>29.50 TL</fiyatı>
          <imaj kapak="resim"/>
   </kitap>
</kitaplık>
```

Varsayılan Değerler

#required

Özniteliğin mutlaka bir değer içermesi ve içerdiği değerin de mutlaka kullanılması gerektiği anlamına gelir. Örneğin resim dosyalarının mutlaka belirlenmiş ebatlarının olması ve belge düzenlemelerinde resim ebatları yüzünden birtakım sorunların çıkmaması için bu özelliği kullanmak çok işlevseldir.

#implied

Özniteliğin değerinin kullanılıp kullanılmayacağının tercihe bağlı olması durumunu ifade eder. Öznitelik kullanılmazsa varsayılan bir değer atanmaz.

#fixed

Özniteliğe varsayılan bir değer atanmak istendiğinde kullanılır. Dolayısıyla varsayılan bir öznitelik değeri atanmışsa, farklı bir öznitelik değeri tanımlanırsa tanımlama geçersizdir.

Varlıklar

Varlıklar (*entities*), XML belgesinde kullanılacak verileri isimlendirerek başka bir içerikle ilişkilendirmek için kullanılır. Varlık tanımlamaları bir defaya mahsus yapıldıktan sonra bir çok kereler kullanılabilirler.

- **İç (internal) Varlıklar:** Metin olarak bir kez DTD içinde tanımlandıktan sonra belgenin her yerinde kullanılabilir. **Dahili genel varlıklar** (*internal general entities*) olarak adlandırılır.

Genel yapısı

```
<!ENTITY varlık_adı  "değer">
```

şeklindedir.

Yerleşik iç varlıklar

< (<) sol köşeli parantezi ifade eder.

> (>) sağ köşeli parantezi ifade eder.

& (&) ampersand sembolünü ifade eder.

' (‘) tek tırnak sembolünü ifade eder.

" (“) çift tırnak işaretini ifade eder.,

Varlıklar belgelerde **&** (*ampersand*) ile **;** (*noktalı virgül*) arasına yazılır.

Örnek

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!DOCTYPE kitaplık [
   <!ELEMENT kitaplık (kitap)>
   <!ELEMENT kitap (adı,yazarı,açıklama)>
   <!ELEMENT kitap (#PCDATA)>
   <!ELEMENT adı (#PCDATA)>
   <!ELEMENT yazarı (#PCDATA)>
   <!ELEMENT açıklama (#PCDATA)>
   <!ENTITY detay  "Kodlab Yayınları Türkiye'nin en iyi
Flash kitabını gururla sunar">

]>

<kitaplık>
   <kitap>
          Kitap hakkındaki bilgiler
          <adı>Flash CS4</adı>
          <yazarı>Uğur GELİŞKEN</yazarı>
          <açıklama>&detay;</açıklama>

   </kitap>
</kitaplık>
```

- **Dış (external) Varlıklar: Dış** (*external*) **Varlıklar**, harici bir dosya ile ilişkilendirilen tanımlamalardır. Varlık tanımlamaları *.dtd* uzantılı dosyalarda bulunur. Bu durumda XML belgemizde bu dosyayı `DOCTYPE` ile tanımlarız. Bu şekilde tanımlanmış varlıklara da harici genel varlıklar (external parameter entities) olarak adlandırılır. Bu noktada unutmadan harici dosyanın DTD dosyası olmasının şart olmadığını, örneğin bir .txt dosyası olabileceğini de hatırlatayım.

Genel yapısı

```
<!ENTITY varlık_adı  SYSTEM "   Varlık_Yeri">
```

şeklindedir.

Örnek

```
<?xml version="1.0" encoding="ISO-8859-9" standalone="no"?>
   <!ELEMENT kitaplık (kitap)>
   <!ELEMENT kitap (adı,yazarı,açıklama)>
   <!ELEMENT kitap (#PCDATA)>
   <!ELEMENT adı (#PCDATA)>
   <!ELEMENT yazarı (#PCDATA)>
   <!ELEMENT açıklama (#PCDATA)>
   <!ENTITY detay  "Kodlab Yayınları Türkiye'nin en iyi
Asp.net kitabını gururla sunar">
```

Yukarıdaki parçada **dış_varlık.dtd** dosyasını, alt parçada ise XML belgesini görmektesiniz.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!DOCTYPE kitaplık SYSTEM "dış_varlık.dtd">
<kitaplık>
   <kitap>
          Kitap hakkındaki bilgiler
          <adı>C# ile ASP.NET 3.5</adı>
          <yazarı>Zafer Demirkol</yazarı>
          <açıklama>&detay;</açıklama>
   </kitap>
</kitaplık>
```

XML Şemaları (XML SCHEMA)

XML şema (*schema*) **DTD** gibi işlev görür. Fakat DTD'nin bazı zayıflıkları sebebiyle **W3C** tarafından tanımlanmıştır. XML şemaları öğenin içeriği demek olan veri tipleri üzerinde çok daha etkin bir kontrol imkanı sağlamaktadır.

XML şemaları **basit** ve **karmaşık** olmak üzere iki tipe ayrılırlar. Sadece metin içerenler basit, metin, öznitelik ve başka öğeleri içerenler ise karmaşık tip olarak adlandırılır.

Basit tipteki bir XML şemasında bir öğenin içereceği metnin türünü basit tip tanımlaması ile atayabilirsiniz. Aynen kullanabileceğiniz bazı sabit tiplerin de (`tarih [date]`, `tamsayı [integer]`... vb) olduğunu hatırlatalım.

Basit Tipteki Şemalar

Şema Yapısı

Metin yapısında olan basit şemaların uzantısı *.xsd* olmaktadır. Şimdi ilk olarak şemaların genel yapısına bakalım.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

.........
</xs:schema>
```

Temel yapısı yukarıdaki gibi olan XML şemasında ilk aklıma gelen husus bu şemayı xml belgemizden (şema dosyamızın adının **kitaplık.xsd** olduğunu varsayıyorum) çağırmak oldu.

Bunun için XML belgemizi aşağıdaki gibi düzenliyoruz.

```
<?xml version="1.0"?>
<kitaplık xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="kitaplık.xsd">
Benim Kitaplarım</kitaplık>
```

Yukarıda ilk aklıma gelen deyince aklıma, şemaları daha etkin kullanabilmek için bazı durumlarda (örn: bazı düzenlemeleri unutmamak...vb) yapmamız gereken şey geldi aklıma. Yapmamız gereken; şemamıza bir not eklemek.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:annotation>
<xs:documentation>BU ARAYA NOTUNUZU EKLEYİN.</xs:documentation>
</xs:annotation>
</xs:schema>
```

Şemamızdaki documentation makine tarafından yorumlanacak bir not değil de insanlar tarafından okunacak bir not eklememiz içindir.

Basit Tip Tanımlama

Basit tip öğelerin sadece metin içerebileceğini biliyoruz. Fakat her zaman sadece "Metin" içeriyor olması yeterli değildir. Bize lazım olan metnin bizim istediğimiz belli bir türde olmasıdır. Bu amaçla sabit tanımlı basit bir tip kullanabileceğimiz gibi kendi özel tipimizi de oluşturabiliriz.

NOT

Çok kullanılan bazı tipleri; string, boolean, URL, tarih ve sayı formatları... vb olarak sayabiliriz.

Sabit Tanımlı Tipler

Bu noktada artık örnek yapmaya başlayabiliriz.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="fiyat" type="xs:string"/>
</xs:schema>
```

Yukarıda fiyat adlı öğemize string (string bir dizi harf rakam ve sembollerden oluşan bir yapıyı ifade eder) atadığımız şema dosyasını, aşağıda ise sayfa öğesine sahip XML dosyasını görüyorsunuz.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<fiyat xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ilk_basit_şema.xsd">50
TL</fiyat>
```

Şimdi de tam sayı bildirimine bir örnek yapalım.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="sayfa" type="xs:integer"/>
</xs:schema>
```

Örneğimizde `sayfa` adındaki bir öğeye integer (tam sayı kullanımını ifade eder) değerini atadık. Aşağıda ise **tam_sayı.xsd** adlı şemanın XML dosyasını görmektesiniz.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<sayfa xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="tam_sayı.xsd">900</sayfa>
```

Sayfa öğemizin değerinin `900` olduğu görülüyor. Bu değer bir tam sayı olduğundan belgemiz geçerli bir belgedir.

Şimdiki örneğimiz tarihler konusunda olacak. Bizim burada söz edeceğimiz üç adet tip var. Bunlar `date`, `month` ve `year` tipleridir. Söz konusu tiplerin yapıları; `date` tipi, biçim olarak Yıl-Ay-Gün şeklinde, `month` tipi Yıl-Ay şeklinde, `year` tipi ise Yıl şeklindedir.

Bu konuda da bir örnek yapalım. Aşağıdaki örnek şemada `baskı_tarihi` öğesine `month` tipi atanmış.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="baskı_tarihi" type="xs:month"/>
</xs:schema>
```

Aşağıdaki XML belgesi ise tarih adlı şema dosyasına ait. Tip olarak `month` seçildiğinden sadece yıl ve ay şeklinde bir öğe değeri girilmiş.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<baskı_tarihi xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="tarih.xsd">2009-01</baskı_tarihi>
```

Özel Tip Oluşturma

Şu ana kadar hep hazır tipler kullandık. Şimdi ise kendi tipimizi oluşturmayı öğreneceğiz. Evvela bir **anonim tip** oluşturalım. **Anonim tip**, sadece tanımlandığı öğede kullanılabilir. Aşağıdaki `ilk_özel_tip` adını verdiğim şemadan yola çıkarak anlatmaya çalışayım.

Gördüğünüz gibi `kanaat` adında bir öğe oluşturduk. Sonra kendi tipimizi oluşturacağımızı belirten simpleType etiketini kulandık. Ardından `restriction base` ile tipimizin temel alacağı basit tipi (örneğimizde string) seçtik. Daha sonra da Tip'imizi `Olumlu` olarak belirledik.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
   <xs:element name="kanaat">
          <xs:simpleType>
                 <xs:restriction base="xs:string">
                        <xs:pattern value="Olumlu"/>
                 </xs:restriction>
          </xs:simpleType>
   </xs:element>
</xs:schema>
```

Bu şemanın XML belgesini de hemen inceleyerek sıcağı sıcağına olayı pekiştirelim...

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<kanaat xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ilk_özel_tip.xsd">Olumlu</kanaat>
```

Öğemizi (kanaat) ve **Olumlu** olarak atadığımız özel anonim tipimizi görüyorsunuz. Belgemizi doğruladığımızda **Olumlu** tipi **string** bir tip olduğundan belgemiz doğrulamayı başarı ile geçecektir. Bu tip anonim olduğundan, tipimize bir isim vermedik ve bunu sadece bir defalık, bir başka ifade ile sadece tanımlandığı öğede kullanabiliriz.

Bu sefer ise **isimlendirilmiş tip** oluşturalım ki, şemamızda bir kere tanımladıktan sonra istediğimiz kadar kullanabilelim. Aşağıdaki örneğin yukarıdaki anonim tip örneğinden tek farkı, tipimize `rey` adında bir isim vermemizdir.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="kanaat">
          <xs:simpleType name="rey">
                 <xs:restriction base="xs:string">
                        <xs:pattern value="Olumlu"/>
                 </xs:restriction>
          </xs:simpleType>
   </xs:element>
</xs:schema>
```

Şemamızın bağlı olduğu XML belgesini de ekleyelim.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<kanaat xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="isimlendirilmiş_tip.xsd">Olumlu</kanaat>
```

Bu örnekte sadece bir tane tip oluşturduk. Peki, öğemizin içerebileceği birden çok değer kümesi olsa nasıl olur? İyi olur diyorsanız okumaya devam edin.

Aşağıdaki örnekte farklı olarak `kanaat` öğesi burada tanımladığımız üç değerden (`Olumlu`, `Olumsuz`, `Nötr`) herhangi birini içerebilir.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="kanaat">
          <xs:simpleType name="rey">
                 <xs:restriction base="xs:string">
                        <xs:enumeration value="Olumlu"/>
                        <xs:enumeration value="Olumsuz"/>
                        <xs:enumeration value="Nötr"/>
                 </xs:restriction>
          </xs:simpleType>
   </xs:element>
</xs:schema>
```

Burada dikkat edilmesi gereken husus; değerlerin eşsiz olması gerektiğidir. Artık XML belgemize de bakabiliriz.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<kanaat xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="tip_kümesi.xsd">Nötr</kanaat>
```

Karmaşık Tipteki Şemalar

Karmaşık tipteki şemaları da anonim ya da isimlendirilmiş olarak tanımlayabiliriz. Genel yapı olarak anonim bir karmaşık şema yapısı aşağıdaki gibidir.

Bu yapının basit tipten farklarına bakalım. İlk evvela `simpleType` yerine `complexType` kullanıldığını ve ayrıca basit tipten farklı olarak başka öğelere de yer verildiğini görüyoruz.

```
<?xml version=”1.0” encoding=”ISO-8859-9”?>
<xs:schema xmlns:xs=”http://www.w3.org/2001/XMLSchema”>
<xs:element name=” kitaplık”>
<xs:complexType>
          <xs:sequence>
                         <xs:element name=” tasarım “ .. />
                         <xs:element name=” programlama “
.. />
          </xs:sequence>
   </xs:complexType>
</xs:element>
```

Bu da isimlendirilmiş tipe örnek bir şema...

```
<?xml version=”1.0” encoding=”ISO-8859-9”?>
<xs:schema xmlns:xs=”http://www.w3.org/2001/XMLSchema”>
<xs:element name=”kitaplık”>
<xs:complexType name=”bilgisayar”>
          <xs:sequence>
                         <xs:element name=”tasarım” ↓
type=”xs:string” />
                         <xs:element name=”programlama” ↓
type=”xs:string” />
          </xs:sequence>
   </xs:complexType>
</xs:element>
</xs:schema>
```

İki örnek için de geçerli olan XML belgesi de huzurlarınızda...

```
<?xml version=”1.0” encoding=”ISO-8859-9”?>
<kitaplık xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
xsi:noNamespaceSchemaLocation=”karmaşık_tip.xsd”>
   <tasarım>web tasarım rehberi</tasarım>
     <programlama>php</programlama>

</kitaplık>
```

XML Belgelerini Biçimlendirme (CSS-XSL)

XML belgelerini görüntülemek için kullandığımız, **XML Parser** ve **XML Processor** görevi gören web tarayıcılar belgelerimizi varsayılan temel bir stil kullanarak görüntülerler.



XML belgelerini web tarayıcılarında söz konusu basit stilin dışında daha detaylı olarak biçimlendirebilmenin bir yolu CSS ile oluşturulmuş biçem sayfalarını XML belgemize bağlamak, diğeri ise XSL dosyalarını

kullanmaktır. Biz ilk evvela CSS yöntemini daha sonra da XSL yöntemini inceleyeceğiz.

CSS İLE BİÇİMLENDİRME

Bu yöntemde kullanacağımız iki dosya var. Birincisi biçimlendirme kurallarını belirleyeceğimiz CSS dosyası, diğeri de haliyle XML dosyamız.

Özellikle basit işlemlerde CSS komutlarını XML belgemize;

```
<style type="text/css"
<!-
seçici {bildirim}
->
</style>
```

şeklinde gömmek mümkünse de, biz ayrı bir CSS dosyası oluşturacağız. Bunu tercih etmemizin temel nedeni, XML belgemize esneklik kazandırmak içindir. Bir başka ifade ile, gerektiğinde XML belgemize hiç dokunmadan ve onu yeniden yapılandırmadan mevcut biçimlendirmemizi değiştirebilmek için bu yöntemi tercih ediyoruz. Ayrıca bu biçemi, ayrı bir CSS dosyası şeklinde oluşturduğumuzdan başka bir XML belgesinde de kullanabiliriz.

Haliyle bu noktada ilk iş CSS dosyamızı oluşturmaktır. Fakat kitabımızda CSS ile ilgili müstakil bir bölüm bulunduğundan ve burada yapacağımız iş tamamen aynı olduğundan bu konuya burada değinmeyeceğiz. Eğer CSS bölümünü çalıştıysanız, aşağıdaki örnek CSS dosyamızın içeriğini zaten hemen anlayacaksınız.

CSS dosyasını oluşturmayı bildiğinizi var sayarak hemen yönteme geçelim. Yöntem gayet basit ve etkilidir. Yapacağımız iş XML belgemizde CSS dosyamıza bir bağlantı oluşturmaktan ibarettir.

Kullanacağımız yönergenin genel yapısı:

```
<?xml-styleheet type="text/css" href=CSS_Dosya_Yolu ?>
```

şeklindedir.

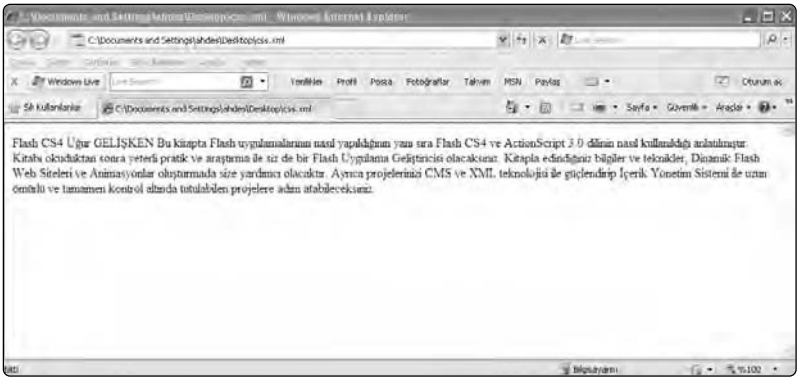
Bu anlattıklarımız ışığında bir örnekle konuyu pekiştirelim.

Örnek XML belgesi

```
<?xml version="1.0" encoding="ISO-8859-9" standalone="yes"?>

<kitap>
   <ismi>Flash CS4</ismi>
   <yazar>Uğur GELİŞKEN</yazar>
   <detay>
   Bu kitapta Flash uygulamalarının nasıl yapıldığının yanı
sıra Flash CS4 ve ActionScript 3.0 dilinin nasıl kullanıldığı
anlatılmıştır. Kitabı okuduktan sonra yeterli pratik ve
araştırma ile siz de bir Flash Uygulama Geliştiricisi olacaksınız.
Kitapla edindiğiniz bilgiler ve teknikler, Dinamik Flash Web
Siteleri ve Animasyonlar oluşturmada size yardımcı olacaktır.
Ayrıca projelerinizi CMS ve XML teknolojisi ile güçlendirip
İçerik Yönetim Sistemi ile uzun ömürlü ve tamamen kontrol altında
tutulabilen projelere adım atabileceksiniz.
   </detay>
</kitap>
```

Örnek belgemizin bu haliyle yani stil uygulamadan tarayıcıda nasıl göründüğüne bir bakalım.



Resimde de gördüğünüz gibi belgemiz yalın halde görüntülendi. Belgemize **xml.css** adındaki biçem dosyasını dahil eden yönergeyi `<?xml-stylesheet type="text/css" href="xml.css"?>` ekleyelim. Yani XML belgemiz aşağıdaki şekilde olacak...

```
<?xml version="1.0" encoding="ISO-8859-9" standalone="yes"?>
<?xml-stylesheet type="text/css" href="xml.css"?>
<kitap>
   <ismi>Flash CS4</ismi>
   <yazar>Uğur GELİŞKEN</yazar>
   <detay>
   Bu kitapta Flash uygulamalarının nasıl yapıldığının yanı
sıra Flash CS4 ve ActionScript 3.0 dilinin nasıl kullanıldığı
anlatılmıştır. Kitabı okuduktan sonra yeterli pratik ve araştırma
ile siz de bir Flash Uygulama Geliştiricisi olacaksınız. Kitapla
edindiğiniz bilgiler ve teknikler, Dinamik Flash Web Siteleri ve
Animasyonlar oluşturmada size yardımcı olacaktır. Ayrıca
projelerinizi CMS ve XML teknolojisi ile güçlendirip İçerik
Yönetim Sistemi ile uzun ömürlü ve tamamen kontrol altında
tutulabilen projelere adım atabileceksiniz.
   </detay>
</kitap>
```

Bu aşamada, kitabımızın CSS bölümünde öğrendiğiniz bilgiler ışığında CSS dosyası oluşturup xml.css adıyla kaydedin. Ben örnek amaçlı olarak, aşağıda gördüğünüz basit biçemi hazırladım.

```
kitap {
width:800px;
padding:1em;
background-color: #ffffcc;
border-width:1px;
border-color:navy;
border-style:solid;
font-family:Verdana;
}

ismi {
  display: block;
  font-size: 16pt;
  font-weight: bold;
  color:red;
  margin-bottom:1em;
}

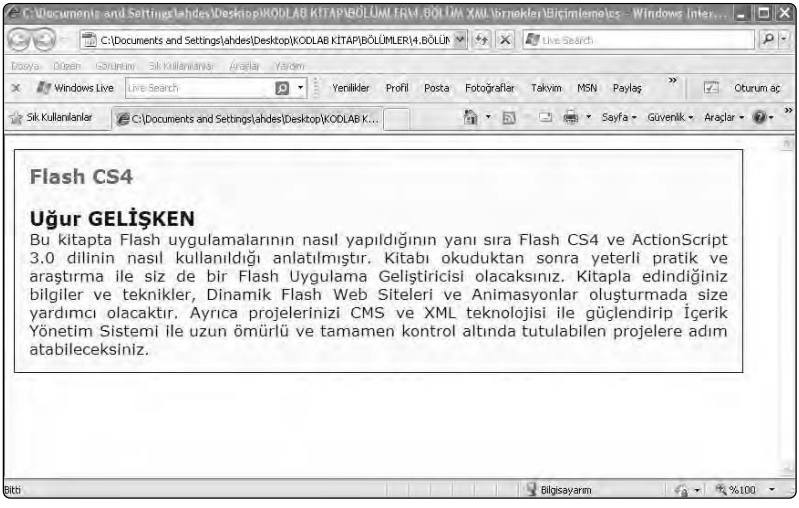
yazar {
```

```
  display: block;
  font-size: 16pt;
  font-weight: bold;
  color:navy;
}

detay {
  display: block;
  font-size: 13pt;
  text-align: justify;
}
```

Burada dikkat edeceğimiz incelik, CSS dosyasındaki stil seçicilerinin adlarını XML belgemizdeki etiketlerle aynı yapmaktır. Örnek vermek gerekirse, belgemizdeki <yazar></yazar> etiketleri arasındaki veriyi biçimlendirmek istiyorsak, CSS dosyasındaki stil seçicisi de yazar { } şeklinde olmalıdır. Böylece uygulayacağınız tasarım özellikleri, seçtiğiniz etiket içindeki veriye uygulanacaktır.

Şimdi de belgemizin tasarım uygulanmış halinin tarayıcıda nasıl göründüğüne bir bakalım.



XSL Kullanımı

XSL ile de yukarıda CSS ile yaptığımız gibi biçimleme yapacağız. Fakat XSL sadece biçimleme yapmakla kalmaz aynı zamanda çevirme işlemi de yapar. Çevirme işlemi, bir XML belgesinin HTML ya da bir başka işaretleme diline dönüştürülmesidir.

XSL stil sayfasının genel şekli aşağıdaki gibidir:

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" ↓
version="1.0">

.............
</xsl:stylesheet>
```

Hazırlayacağımız bu XSL stil sayfasını XML belgemize aşağıdaki bildirim vasıtası ile ekleriz.

```
<?xml-stylesheet type="text/xsl" href="xsl_sayfasının_adı.xsl"?>
```

Şimdi bu konuda da bir örnek yapalım. Aşağıda örnek XML dosyamızı görüyorsunuz:

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<?xml-stylesheet type="text/xsl" href="kitaplar.xsl"?>
<kitaplık>
   <kitap>
          <adı>10 Adımda Web Tasarımı</adı>
          <yazarı>Uğur GELİŞKEN</yazarı>
          <sayfa_sayısı>248</sayfa_sayısı>
          <ebatları>13x19</ebatları>
   </kitap>
   <kitap>
          <adı>Flash CS4</adı>
          <yazarı>Uğur GELİŞKEN</yazarı>
          <sayfa_sayısı>416</sayfa_sayısı>
          <ebatları>19x24</ebatları>
   </kitap>
   <kitap>
          <adı>C# ile ASP.NET 3.5</adı>
```

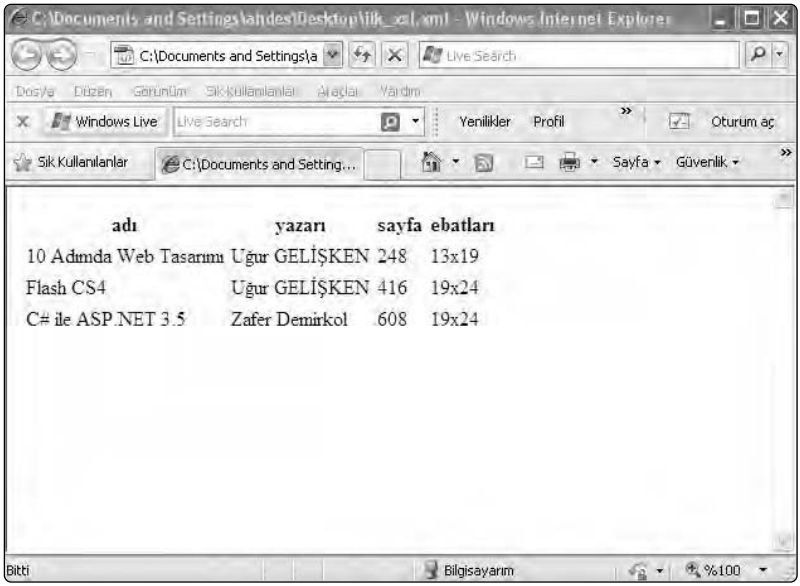
```
            <yazarı>Zafer Demirkol</yazarı>
            <sayfa_sayısı> 608</sayfa_sayısı>
            <ebatları>19x24</ebatları>
   </kitap>
</kitaplık>
```

Gördüğünüz gibi biçimlendirmeyi **kitaplar.xsl** ile yapacağız. Peki o zaman nerde bu? Bakın hemen aşağıda:

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" ↓
version="1.0">
   <xsl:template match="/">
            <table cellspacing="4">
                     <tr>
                              <th>adı</th>
                              <th>yazarı</th>
                              <th>sayfa</th>
                              <th>ebatları</th>
                     </tr>
                     <xsl:for-each select="kitaplık/kitap">
                     <tr>
                              <td><xsl:value-of select="adı"/></td>
                              <td><xsl:value-of select="yazarı"/></td>
                              <td><xsl:value-of ↓
select="sayfa_sayısı"/></td>
                              <td><xsl:value-of ↓
select="ebatları"/></td>
                     </tr>
                     </xsl:for-each>
            </table>
   </xsl:template>
</xsl:stylesheet>
```

XSL dosyamızdaki **for-each** öğesi, oluşturduğumuz şablonu XML belgemizdeki her bir **düğüme** (*node*) uygulayabilmemizi sağlamaktadır. (XML belgemizdeki her bir <kitap></kitap> arası bir düğüm olarak adlandırılır.)

Artık belgemizin tarayıcıda nasıl göründüğüne bakma zamanı geldi.



Evet, XML bölümümüz burada sona erdi. Ama XML sona ermedi. Diğer bölümlerde XML dili ile yeniden karşılaşacağız. Burada aldığınız temel bilgilerle o bölümlerdeki XML uygulamalarını da gayet başarılı bir şekilde yapabileceksiniz. Siz yeter ki okumaya devam edin.

5 DREAMWEAVER

Dreamweaver Çalışma Alanı	188	Formlar	236
Dreamweaver Tercihleri	195	Şablonlar	237
Yeni Bir Site Oluşturmak	207	Spry Kullanımı	241
Yeni Bir Sayfa Oluşturmak	212	Dreamweaver ile Ziyaretçi Defteri	245
Varlıklar	214	Dreamweaver ile Üyelik Sistemi	257
Belge Oluşturmak	215		
Tablolar	231		
Çoklu Ortam Dosyaları	235		

Dreamweaver öğrenilmesi kolay ve kullanması çok keyifli olan bir program. Bu bölümü ilgiyle okuduğunuzda kısa sürede programa adapte olacağınızdan şüpheniz olmasın.

Dreamweaver'ın CS4 sürümünde sahip olduğu arayüzüyle bize sunduğu kullanışlılık gerçekten programın tamamı için de geçerlidir. Üst barda çalıştığınız dosya ile bağlantılı diğer dosyaları görüp hızlı bir şekilde ulaşabilmenizden tutun da araç çubukları ve menülerin hızlı ulaşılabilir ve özelleştirilebilir olmasına, yazdığımız kodların önizlemesinin programdan çıkmadan anında görülebilmesine kadar tamamen kullanıcı dostu bir programdır diyebiliriz.

Madem Dreamweaver bu kadar kullanıcı dostu ve keyifli bir program, o zaman baştan neden Dreamweaver ile başlamadık da **NOTPAD** ile uğraştık, kodlar ile boğuştuk dediğinizi duyar gibiyim.

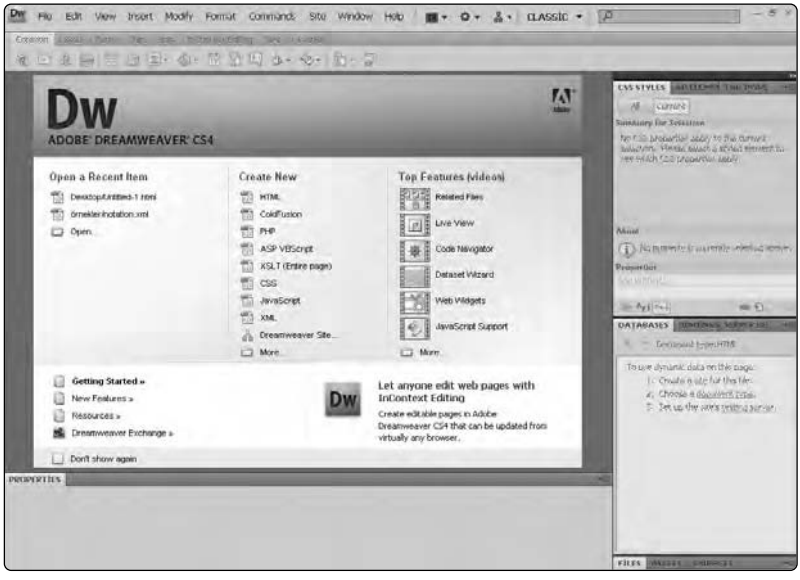
Evet haklısınız, ama öğrendiğimiz konular temel konular olduğundan böyle yaptık. Amaç öğrendiklerimizin pekişmesiydi. Bu aşamadan sonra yapacağımız çalışmalarda Dreamweaver programının tüm nimetlerinden yararlanacağız. Yani kısaca ne yaptımsa sizin için yaptım.

İlk olarak Dreamweaver'ın arayüzünü, menülerini ve genel yapısını inceleyerek başlayacağız. Bu Dreamweaver programını tanıma amaçlı ısınma turunun ardından şu ana kadar öğrendiğimiz bilgilerimizi Dreamweaver programında kullanarak hem bilgilerimizi daha da pekiştireceğiz hem de sıkı bir Dreamweaver kullanıcısı olacağız.

Çok fazla girizgah yapıp sizi sıkmaya niyetim yok merak etmeyin. Bir an evvel başlamak sanırım en iyisi olacak. İşte başlıyoruz.

Dreamweaver Çalışma Alanı

Dreamweaver programını başlattığınızda ilk olarak sizi aşağıdaki ekran karşılar.



Bu sayfa yardımıyla son çalıştığınız belgelere tek bir tıkla ulaşabileceğiniz gibi yeni bir belge oluşturmak için de ortadaki **Create New** bölümünden çalışacağınız dili seçerek başlayabilirsiniz. Eğer Dreamweaver'a bir dahaki girişinizde bu sayfanın açılmamasını isterseniz, sayfanın sol alt kısmında bulunan **Don't show again** kutucuğunu tıklamanız yeterli olacaktır. Artık bir dahaki girişinizde doğrudan Dreamweaver çalışma alanına geçeceksiniz.



Şimdi çalışma alanını tanımaya başlayabiliriz. Yukardan aşağıya doğru sayfamızdaki öğelere bakalım.

En üst kısımda resimde gördüğünüz Dreamweaver **menüleri** ve **çalışma alanı değiştiricisi** bulunmaktadır. **Çalışma alanı değiştiricisi** bir açılır pencere şeklindedir ve **classic**, **coder**, **designer**... vd değişik çalışma alanı seçenekleri sunar. Bunlardan hangisini kullanacağınıza Dreamweaver'ı tanıdıkça ve ihtiyaçlarınıza göre karar verebilirsiniz. Ben şimdilik **classic** görünümü kullanacağım.



Menülerin hemen altında **Insert** paneli yer alır. Bu panel aracılığıyla dosyanıza metin biçimlendirme etiketleri, form, tablo... vb birçok kod parçalarını kolaylıkla ekleyebilirsiniz. **Insert** panelini yan bölüme ya da üst bölüme sürükleyerek taşıyabilirsiniz.

Buradaki sekmelerden **common** (*genel*); tablo, tarih, link, resim... vb yaygın kullanılan özellikleri sayfamıza eklemek için kullanılır. **Layout** (*mizanpaj*) sekmesi `div` ve `iframe` eklemek için kullanılır. Bir sonraki forms sekmesini ise sayfamıza `form` eklemek istediğimizde kullanacağız. **Text** sekmesinden de metin biçimlendirme etiketlerini eklemek istediğimizde yararlanacağız. Son olarak **Favorites** sekmesi ile, istediğimiz tuşları burada gruplayıp sık kullandığımız özellikleri bir araya getirebileceğimizi belirtelim.



Menülerin hemen altında ise **Belge** araç çubuğu yer alır. Belgenizi farklı görünümlerde açarak çalışabilmeni sağlayan araçlar içerir. Mesela sadece kodlar ile çalışmak istediğinizde **Code** tuşuna basarak belgenizi sadece kod görünümünde açabilir ya da **Design** tuşuyla sadece tasarım şeklini görüntüleyerek çalışabilirsiniz. Dilediğinizde her iki görünümü de çalışma penceresinde **Split** tuşu ile görüntüleyebilirsiniz.

Bir diğer tuş olan **Live View** (*Canlı Görünüm*) ile sayfanızın tarayıcı ön izlemesini görebilirsiniz. Bu sayede belgenizin tarayıcılarda nasıl göründüğüne bakmak için Dreamweaver programından ayrılıp tarayıcınızı açmanız gerekmez. Yalnız bu bir önizleme sayfası olduğundan burada herhangi bir düzenleme yapamazsınız. Title kısmından ise belgemizin başlığını gireceğiz. Buraya yazacağımız başlık belgemizdeki `<title> </title>` etiketleri arasında yer alacaktır.



Bunların detaylarını ilerleyen sayfalara bırakarak sağ tarafta gördüğünüz panel gruplarına geçelim. **Paneller**, işlemlerinizi daha etkin bir biçimde izleme ve daha hızlı bir şekilde değiştirme imkanı verirler.

Örneğin **CSS Style** (*CSS stil*) paneli etkin bir CSS yönetimi sağlarken **Files** (*Dosyalar*) paneli sitenizde kullandığınız dosyaların her an ulaşılabilir olmalarına ve daha hızlı bir şekilde düzenlenebilmelerine imkan tanır.

Snippets paneli ise bize bir çok hazır kod parçacığı sunar. Bunları sürükle bırak yaparak kolaylıkla belgenize ekleyebilirsiniz. **Databases** (*Veritabanı*) paneli ile veritabanı işlemlerini yürüteceksiniz.

Sol köşede ise sadece kod görünümünde çalışırken çıkan **kodlama araç çubuğu** bulunmaktadır. Kod yazarken işinizi kolaylaştıracak birçok tuştan oluşmaktadır.



Çalışma alanının hemen altında yer alan **Durum Çubuğu**, üzerinde çalıştığımız belge hakkında bazı yardımcı bilgiler içerir. Sol başta sayfamıza ait seçili etiketleri görüyorsunuz.

Etiketi tıklayarak belgemizdeki ilgili kısmın seçili hale gelmesini sağlayabilirsiniz. Etikete **SAĞ** tıkladığınızda ise eğer varsa etiketle ilgili ID ve class niteliklerini, açılacak menüden ayarlayabilirsiniz.

Sağa doğru yer alan diğer özellikler; belge penceresinde belgemizi sürüklemeye yarayan el seçim aracı, belgeyi yakınlaştırma büyüteç aracı, pencere boyutu ayarlama kutucuğu, belge boyutu ve yükleme süresi bilgisi ile belgede kullanılan karakter seti tipi olarak sıralanmaktadır.



Durum çubuğunun hemen altında ise **Properties** (*Özellikler*) paneli yer alır. Belgenizde yer alan öğelerin en temel özelliklerini burada görebilir ve düzenleyebilirsiniz.



MENÜLER

Dreamweaver'da bir özelliğe, genellikle birçok yöntem ile (örn: menüler, kısayol tuşları, paneller) erişilebilmektedir. Bunlardan standart erişim biçimi olan menüler ve bu menülere bağlı alt menüler ile neler yapabileceğimize bakalım.

FILE/DOSYA

New: Yeni bir belge açmak istediğinizde

Open, Open Recent: Mevcut bir belgeyi açmak istediğinizde

Save: Belgelerinizi kaydetmek için

Save As: Belgelerinizi farklı kaydetmek için

Save to Remote Server: Belgenizi oluşturduğumuz yerel sunucuya (test sunucusu) kaydetmek için.

Import: Dışardan veri ithal etmek için.

Export: Dışarıya veri ihraç etmek için.

Preview in Browser: Belgenizi değişik tarayıcılarda test edebilmek için.

Check Page, Validate: Belgenizi; erişilebilirlik, linklerin durumu, belgenizin tarayıcılardaki durumu, etiketlerin geçerliliği gibi konularda test edebilmek için.

EDIT/DÜZEN

Undo: Yapıp ta vazgeçtiğiniz bir işlemi geri almak için.

Copy: Kopyalama işlemi için.

Paste, Paste Special: Yaptığınız kopyalamayı yapıştırmak için.

Keyboard Shortcuts: Kısayol tuşlarını değiştirmek ya da yeni kısayol tuşu eklemek için.

Preferences: Program ile ilgili birçok ayarın (General [genel ayarlar], Accessibility [erişilebilirlik], CSS Styles [CSS Stilleri], Fonts [Fontlar] New document [Yeni belge Ayarları] Site [Site Ayarları]...vd) yapılması için.

Edit with External Editor: Belgelerinizde kullanacağınız resim, flash... vb nesneleri örneğin Photoshop ve Flash programıyla düzenlemek istediğinizde bu menü yardımı ile istediğiniz programları belirleyebilirsiniz. Düzenleyeceğiniz nesneyi seçip düzenleme işlemini başlattığınızda seçtiğiniz program otomatik olarak açılacaktır.

Tag Libraries: Etiket kütüphanesine erişmek içindir.

VIEW/GÖRÜNÜM

Code: Çalışma alanının sadece kod görünümünde olması için.

Design: Çalışma alanının sadece tasarım görünümünde olması için.

Code and Design: Çalışma alanının hem kod hem de tasarım görünü-

münde sayfada yarı yarıya gösterilmesi için. (bu üç seçeneği çalışma alanı değiştiricisi ile de belirleyebilirsiniz.)

Switch Views: Üstte bahsedilen görünümler arası geçiş için. Bunu fare ile bir tık yaparak da başarabilirsiniz.

Ruler: Çalışma sayfanıza cetvel eklemek için.

Grid: Çalışma sayfanıza ızgara eklemek için.

INSERT/EKLE

Ekle menüsü ile belgenize **tag** (*etiket*), **image** (*resim*), **image objects** (resim nesnesi- örn: *rollover image*), **media** (*swf, flv*), **table** (*tablo*), **Layout** (*mizanpaj-sayfa düzeni*), **form**, **link**, **date** (*tarih*), **named anchor** (*çapa*) **comment** (*yorum*)... ve daha bir sürü şey ekleyebilirsiniz.

Bu işlemlerin çoğunu insert panelinden de yapabileceğinizi tekrar hatırlatmak isterim. Bir de **Customize Favorites** ile belgelerinize sıklıkla eklediğiniz nesneleri bir araya toplayarak kendi sık kullanılanlarınızı oluşturabileceğinizi belirteyim.

MODIFY/DEĞİŞTİR

Bu menü yardımı ile en alttaki **Section Properties** ve **CSS Style** panellerini açıp kapatabilir, **Page Properties** ile **CSS** ve **HTML** görünüm ayarları, başlık ve karakter seti ayarları, resim transparan oranı gibi ayarları yapabilirsiniz. **Make Link** ile sayfanıza köprü atayabilir ve köprünün hedefini belirleyebilirsiniz. Ayrıca etiketler, tablolar, frameler ve resimler ile ilgili bazı düzenlemeleri yapabilirsiniz.

FORMAT/BİÇİMLE

Bu menü yardımı ile paragraf biçimlendirme, metin hizalama, font seçimi, liste oluşturma, renk atama, CSS seçicisi oluşturma gibi biçimlendirmeye yönelik alt menüler yer almaktadır.

COMMAND/KOMUTLAR

Dreamweaver'a yeni yetenekler kazandıran küçük programcıklar olarak tanımlayabileceğimiz **extensions** (*eklentiler*) yönetimini sağlayan Manage **Extension** bu menüde yer alır. Açılan sayfa yardımı ile bilgisaya-

rınızdaki eklentileri programa yükleyebilir ya da adobe firmasının resmi sitesinden eklentilere bakabilirsiniz.

Fireworks programı bilgisayarınızda kurulu ise **Create Web Photo Album** menüsünden albüm oluşturabilirsiniz. XHTML belgelerinizdeki ve word programında oluşturulmuş HTML belgelerinizdeki gereksiz kodları tarayarak temizleyen **Clean Up Word HTML** ve **Clean Up XHTML alt** menüleri de özellikle büyük dosyalarla çalışırken çok işinize yarayacak özelliklerdir.

Site/Site

Dreamweaver'de birkaç sayfa sonra değineceğimiz şekilde bir site oluşturmak ve oluşturduğumuz siteleri yönetmek için kullanacağımız yöntemlerden biri de **Site** menüsü olacaktır.

Window/Pencere

Pencere menüsü Dreamweaver'de işimizi çok kolaylaştıran ve hızlandıran panelleri yöneteceğimiz, açıp kapayacağımız ve aralarında tercih ile geçiş yapabileceğimiz menüdür.

Dreamweaver Çalışma Alanini Özelleştirme

Dreamweaver programının kullanışlılığından söz etmiştik. Bu kullanışlılığı sağlayan özelliklerin başında Dreamweaver'ın bize özelleştirilebilir bir çalışma alanı sunması gelmektedir. Şimdi de biraz kendimize en uygun çalışma ortamını oluşturmak için bu özelleştirme işlemini nasıl yapacağımız üzerinde duralım.

Öncelikle birçok belge aynı anda açıksa, bunların sırasını değiştirmek istediğinizde, belge başlığını fare ile sürükleyip istediğiniz sırada konumlandırabileceğinizi söyleyelim. Aynı şekilde panelleri de fare ile sürükleyerek istediğiniz yere konumlandırabilirsiniz.

Panel grubundan ayırdığınız bir paneli sağ üst köşesindeki çarpı işaretine tıklayarak tamamen kapatabilirsiniz. Daha sonra paneli tekrar açmak istediğinizde yapmanız gereken, **Window** menüsünden istediğiniz paneli tıklamaktan ibaret olacaktır.

Bir panel grubunu simge durumuna küçültmek için panelin adına çift tıklamanız yeterli olacaktır. Panellerin tamamını bir seferde daraltmak içinse panellerin en üstünde yer alan çift ok işaretini tıklayın. Panel grubu içindeki panellerin sırasını da fare ile tutup sürükleyerek değiştirebilirsiniz. Panelleri genişletmek ya da daraltmak için herhangi bir köşesinden tutup sürükleyebilirsiniz.

Peki, bir başkası programı kullandığında ve o kişi de programı kendi isteğine göre özelleştirmek istediğinde bu kadar uğraşıp oluşturduğumuz tüm bu özelleştirmeleri kayıp mı edeceğiz? Tabi ki hayır. Dreamweaver bize çalışma alanımızı kaydetme imkanı da sunuyor. Bunun için yapmamız gereken **Window** menüsünden **Workspace Layout** seçeneğinin altında bulunan **New Workspace** menüsünü seçmek ve bir isim belirlemek olacaktır.

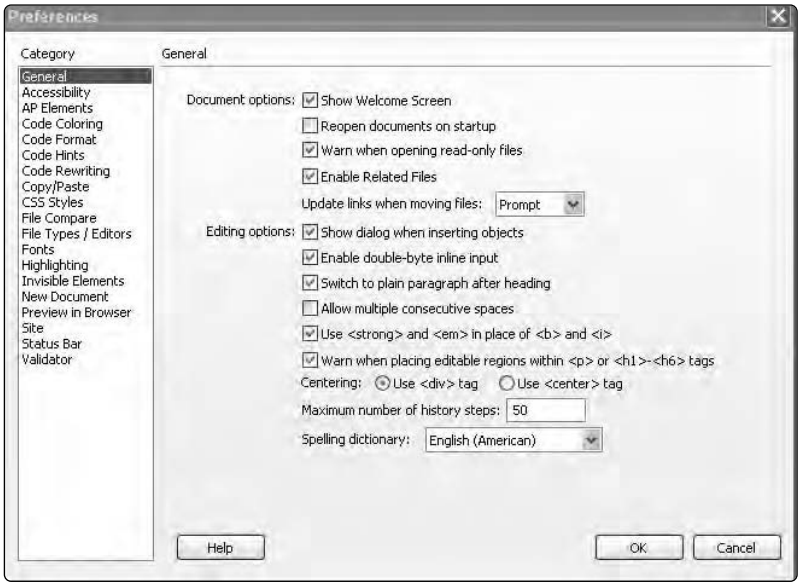
Artık özelleştirdiğimiz çalışma alanımızın adını sayfamızın üst kısmındaki çalışma alanı değiştiricisinde görebileceğiz. Buradan istersek mesela tekrar **Classic** görünümü seçebilir istediğimiz zaman da tekrar özelleştirdiğimiz çalışma alanına geri dönebiliriz. İstersek birden fazla sayıda çalışma alanı da hazırlayabiliriz. Mesela PHP çalışmamız için ayrı XHTML çalışmamız için ayrı çalışma alanları hazırlayabiliriz.

DREAMWEAVER TERCİHLERİ

Edit/Preferences menüsü altında yer alan bu pencereyi kullanarak Dreamweaver ile ilgili tüm temel ayarlarımızı yapacağız. Şimdi penceredeki tüm kategorileri tek tek inceleyelim.

GENERAL (GENEL DREAMWEAVER TERCİHLERİ) KATEGORİSİ

Dreamweaver'da **Preferences** (*Tercihler*) menüsünde yer alan **General** (*Genel Tercihler*) kategorisi, programın genelinde geçerli olmasını istediğimiz özellikleri belirlemek için kullanılır.



Show Welcome Screen: Bu pencerede ilk olarak karşımıza Dreamweaver açılış penceresinin gösterilmesini isteyip istemediğimizi belirlememizi sağlayan seçim kutucuğu çıkıyor. Kutucuğu tıklı duruma getirsek Dreamweaver'ı ilk açtığımızda bizi, solda son çalıştığımız dosyaları, yanında da Dreamweaver'ın desteklediği dilleri gösteren ve yeni belge açmamızı sağlayan **Create new** bölümünün yer aldığı açılış ekranı karşılayacak.

Reopen Documents on Startup: Bu kutucuk ise Dreamweaver'ı ilk başlattığımızda, programı en son kapattığımızda hangi dosyalar açıksa onları tekrar açmak içindir.

Warn When Opening Read-Only Files: Salt okunur dosyaları açarken programın bizi uyarması içindir.

Enable Related Files: Üzerinde çalıştığımız belgeye bağlı olan dosyaları görme imkanı verir. Görüntülenen dosya adını tıkladığımızda ise dosyayı açar. Mesela bir XHTML dosyası ile çalıştığımızı ve bu dosyaya harici bir CSS dosyasının bağlı olduğunu farz edelim. Bu CSS dosyasının adını belgemizin üst kısmında görürüz. Tek tıklama ile dilediğimiz

anda CSS dosyasına, sonra da işimiz bitince tekrar XHTML dosyamıza döneriz.

Update Links When Moving Files: Bu açılır seçim kutucuğu ile de, bir belgeyi taşıdığımızda ya da kaldırdığımızda program, sayfaya bağlı olan linkleri bu kutucuktan yapacağımız tercihe göre ya güncelleyebilir, ya güncellemeyebilir ya da güncellenip güncellenmeyeceğini bize sorabilir.

Show Dialog When Inserting Objects: Sayfanıza bir nesne eklediğinizde eklediğiniz nesne ile ilgili parametreleri girebilmeniz içindir.

Enable Double-Byte Inline Input: "double-byte" olarak tabir edilen (örn: japonca) metinleri belgelerinize ekleyebilmek için bu kutucuk seçili olmalıdır. Aksi takdirde bunları belgelerinizde düzgün bir biçimde görüntüleyemezsiniz.

Switch to Plain Paragraph After Heading: Bu seçenek sayesinde `h1`, `h2`, `h3`... vd başlıklardan sonra **ENTER** tuşu ile hemen yeni bir paragraf oluşturabilirsiniz. Kutucuk seçili olmazsa **ENTER** ile yeni bir paragraf değil yeni bir başlık oluşturmuş olursunuz.

Allow Multiple Consecutive Spaces: HTML içinde peş peşe birden fazla boşluk oluşturabilmek içindir.

Use and in place of <b> and <i>: **Format/Style** menüsünden **bold** ya da *italic* seçeneklerini çalıştığınız metne uyguladığınızda, World Wide Web Konsorsiyumu tarafından önerildiği gibi `<b>` ve `<i>` etiketlerinin yerine program tarafından `<strong>` ve `<em>` etiketlerinin kullanılmasını sağlayan kutucuğun tıklı olması gerektiğini vurgulayalım.

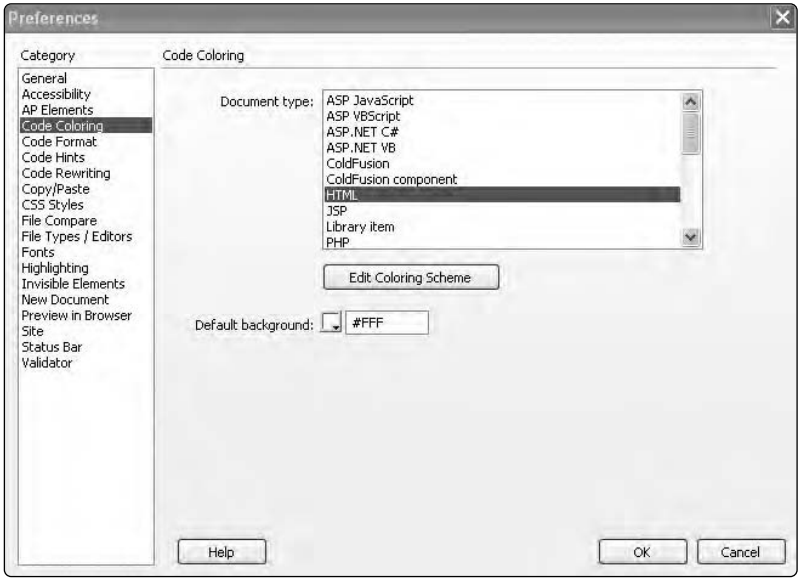
Warn when placing editable regions within <p> or <h1><h6> tags: Dreamweaver şablonu ile çalışırken **Editable Region** denilen şablondaki düzenlenebilir alanların bir paragraf `<p>` ya da bir başlık `<h1>` içinde olması durumunda, bu alanda bir başka paragaf ya da başlık oluşturulamayacağı konusunda bir uyarı mesajı görüntülenmesini isteyip istemediğinizi belirlemek içindir.

Maximum Number of History Steps: Buraya gireceğimiz rakam, Dreamweaver tarafından geriye dönük olarak en fazla kaç işlemin sak-

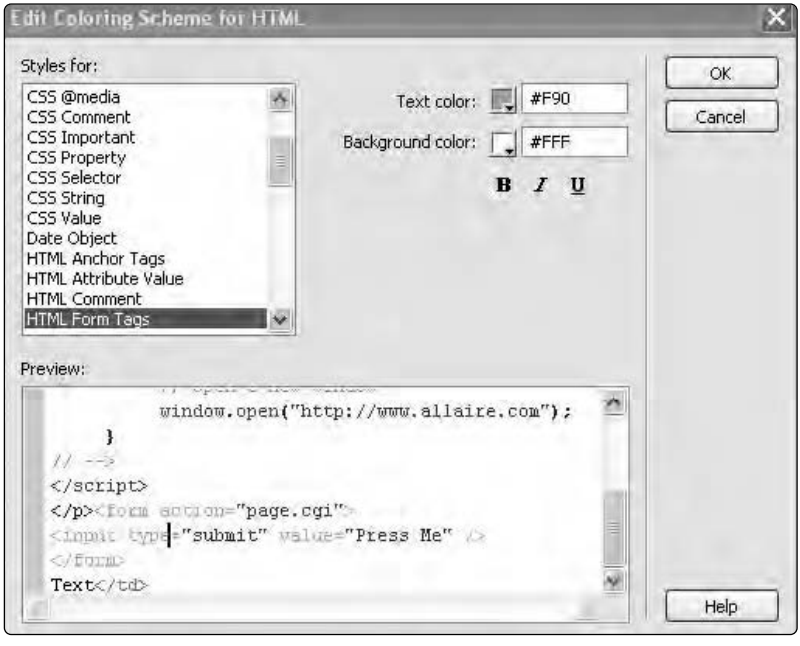
lanacağını belirlemenize yarar. Yapmış olduğunuz eski işlemlerin listesini ise **History** panelinden takip edebilirsiniz.

Spelling Dictionary: Dreamweaver ile yazım kontrolü yapmak içindir. Açılır kutucuktaki diller arasında maalesef Türkçe bulunmuyor.

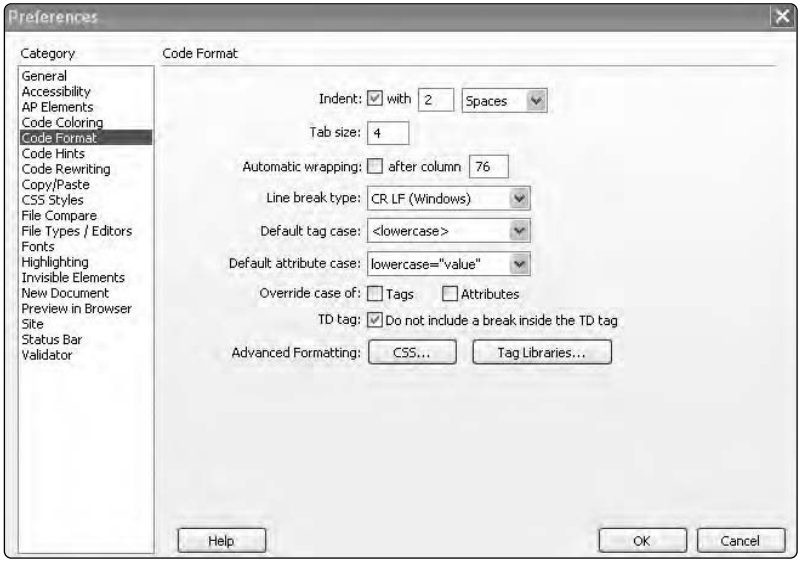
Code Coloring (Kod Renklendirme) Kategorisi



Bu sayfada iken renklerini düzenlemek istediğiniz dili seçip Edit Coloring Scheme tuşuna basarak, aşağıda resmini gördüğünüz sayfada kodları istediğiniz gibi renklendirebilirsiniz.



Code Format (Kod Biçimlendirme) Kategorisi



Indent-With: Kodları girintili bir biçimde oluşturabilmek içindir. Bu yapı çok sayıda kodun olduğu bir sayfanın daha iyi anlaşılmasını sağlayacaktır. **Tab Size** ile girinti için istediğimiz değeri gireceğiz.

Automotaic wrapping-after column: Kod yazarken satırları belli bir karakter sayısı ile sınırlayabilirsiniz. Örneğin bu sayı `76` ise bu karakter sayısına ulaşıldığında kod yazımı bir alt satırdan devam edecektir.

Line Break Type: Alt satıra inme standardıdır. Buraya çalıştığınız sistemi (*CR LF Windows*, *CR Macintosh*, *LF Unix*) girmelisiniz.

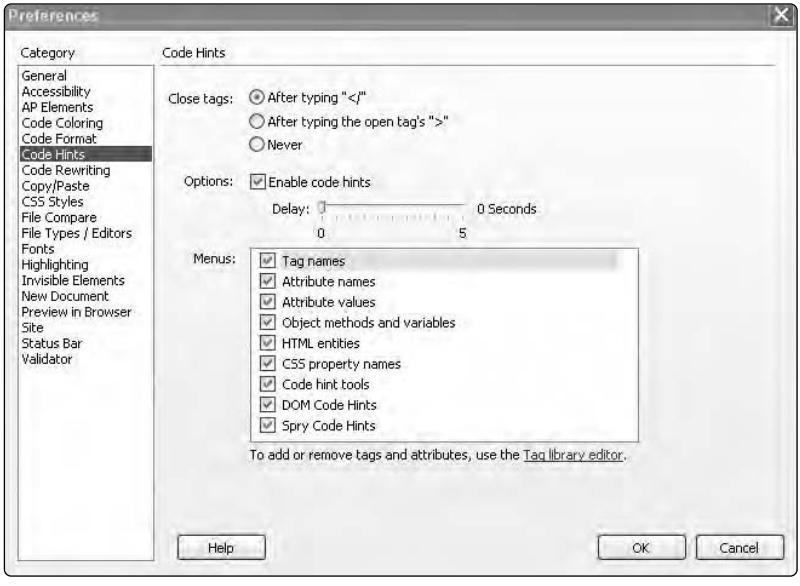
Default Tag Case: Varsayılan etiket biçimidir. `<UPPERCASE>` ile etiketler büyük harfle, `<lovercase>` seçtiğinizde ise küçük harfle yazılacaktır.

Default Attiribute Case: Öznitelik içeren etiketlerin biçimlendirilmesi içindir. `<UPPERCASE="value">` ile büyük harfle, `<lovercase=""value">` seçtiğinizde ise küçük harfle yazılacaktır.

Overwrite Case Of-Tags-Attributes: Yazdığınız kodlardaki etiketlerin ya da özniteliklerin biçimlerinin (küçük-büyük harf) yazım esnasında değiştirilebilmesi içindir.

Do not include a break inside the TD tag: Tablolarda kullandığımız TD etiketlerinden sonra girinti yapılıp yapılmayacağını belirlemek içindir.

Code Hints (Kod Tamamlama) Kategorisi

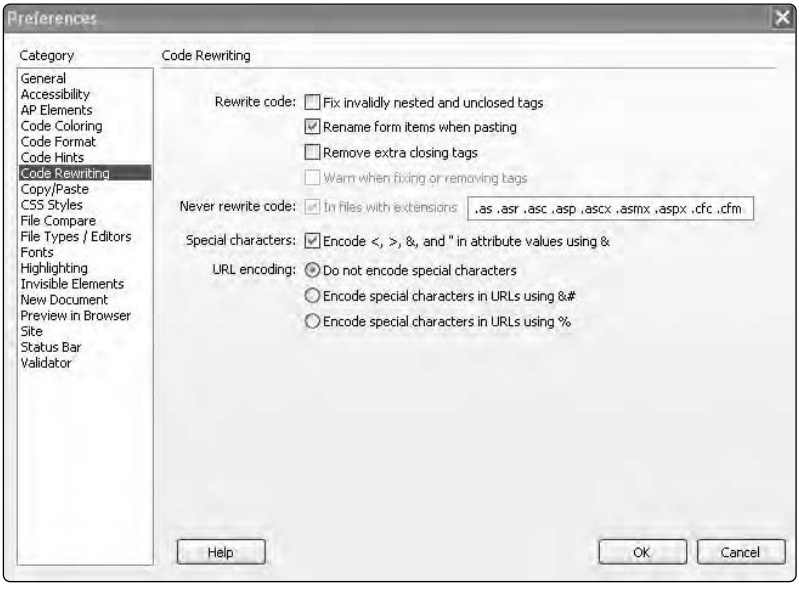


Close Tags: Otomatik etiket kapatma işlevi görür. Bu özelliği kullanarak kapanmamış etiket kalma riskinden kurtulmuş olursunuz. Dilerseniz bu özelliği never radyo düğmesini seçerek kullanmayabilirsiniz.

Options-Enable Code Hints: Otomatik kod tamamlama özelliğini kullanabilmenizi sağlar. Kod yazarken kodun yanında, yazdığınız kodun harfleri ile başlayan kod gruplarını listeleyen bir menü açılır. Buradan kodunuzu seçerek zaman kazanmış olursunuz.

Menus penceresinden ise kod tamamlama özelliğinin geçerli olacağı etiketleri, öznitelikleri, nesneleri seçebilirsiniz.

Code Rewriting (Kod Yeniden Düzenleme) Kategorisi



Rewrite Code: Sizin belirlediğiniz kriterler haricinde oluşturulmuş kod kümelerini, sizin kriterlerinize göre yeniden düzenlemek içindir.

Fix Invalidly Nested and Unclosed Tags: Hatalı yuvalandırılmış ve kapatılmamış etiketleri düzeltmek içindir.

Rename Form Items When Pasting: Kopyaladığınız form öğelerini yapıştırırken yeniden adlandırmak içindir.

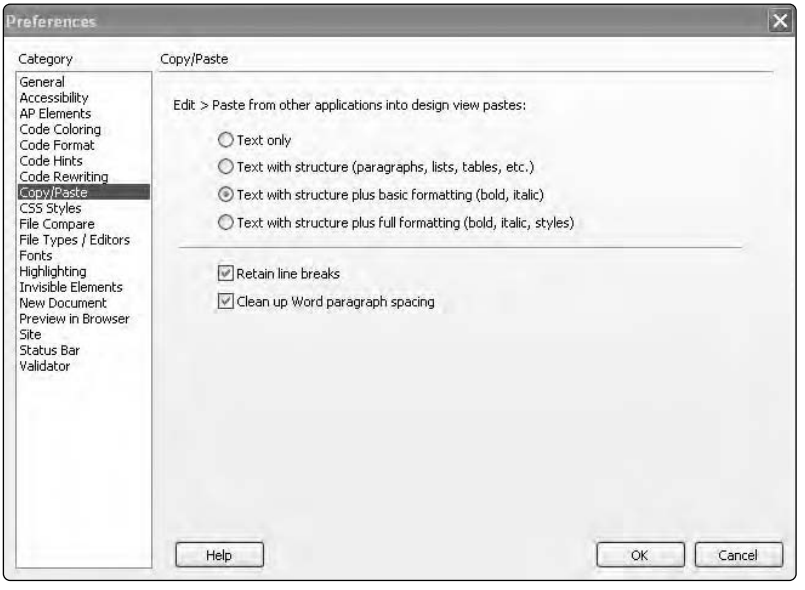
Remove Extra Closing Tags: Fazladan yazılmış kapama etiketlerini programa kaldırtmak içindir.

Warn When Fixing or Removing Tags: Yukarıda bahsettiğim düzeltme işlemlerini yaparken Dreamweaver'ın bizi uyarmasını isteyip istemediğimizi belirlemek içindir.

Never Rewrite Code-In Files with Extensions: Buradaki kutucuğa yazılan uzantılara sahip dosyalara yeniden düzenleme işlemleri uygulanmaz.

Special characters: <, >, & ve " karakterlerini bir öznitelik değeri olarak kullanacağınız zaman, bunları otomatik olarak **encode** eder. Mesela & işaretini `&` şekline çevirir.

Copy/Paste (Kopyala/Yapıştır) Kategorisi



Belgemize **kopyala-yapıştır** yaptığımız içerikler ile ilgili ayarları içerir.

Text only: Kopyalanan metni tüm biçimlendirmelerden arındırır.

Text with structure: Metin ile beraber tablo, paragraf, liste... vb yapısal özellikleri de korur.

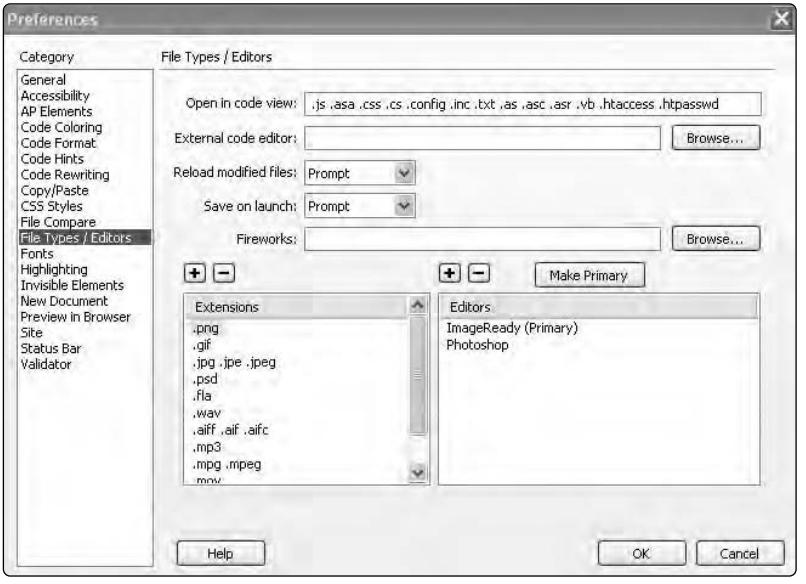
Text with structure plus basic formating: Metin ve yapı ile beraber bold, italik gibi vurgu özelliklerini de alır.

Text with structure plus full formating: Metni tüm biçimlendirme özellikleri ile beraber alır.

Retain line breaks: Yeni satırları korur.

Clean up Word paragraph spacing: Bir Word belgesini kopyalamışsak, belgedeki paragraf boşluk stillerini temizler.

File types/Editors Kategorisi

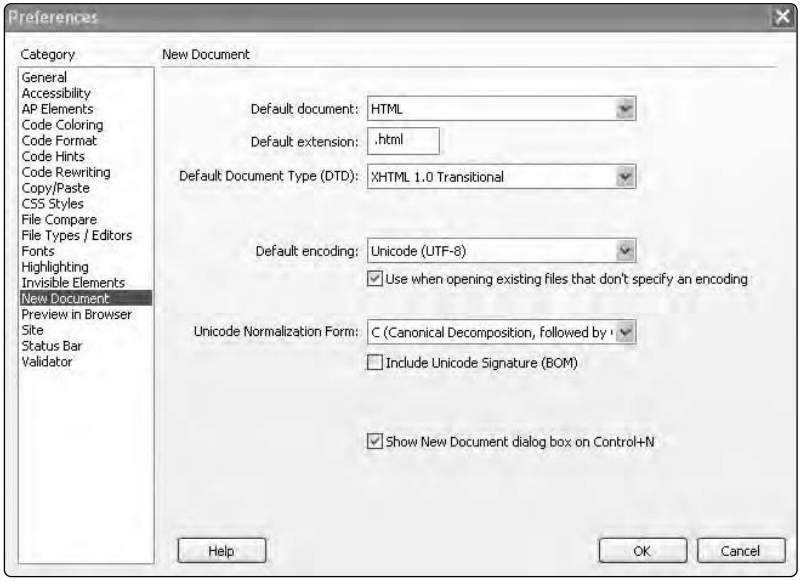


Open in code view: Bu kutucuğa eklenen uzantılara sahip dosyaları program kod görünümünde açacaktır.

External code editor: Kullandığınız bir harici kod editörü varsa buradan programa tanıtabilirsiniz.

Fireworks: Bilgisayarınızda Fireworks yazılımı varsa düzenlemelerinizde kullanmak üzere buradan programa tanıtabilirsiniz. Alttaki kutulardan kullanacağınız dosya uzantılarını ve bunları düzenleyeceğiniz programları belirleyebilirsiniz.

New Document (Yeni Belge) Kategorisi



Yeni oluşturacağımız belgelere ilişkin ayarları içerir.

Default document: Varsayılan belge türünü belirlemek içindir.

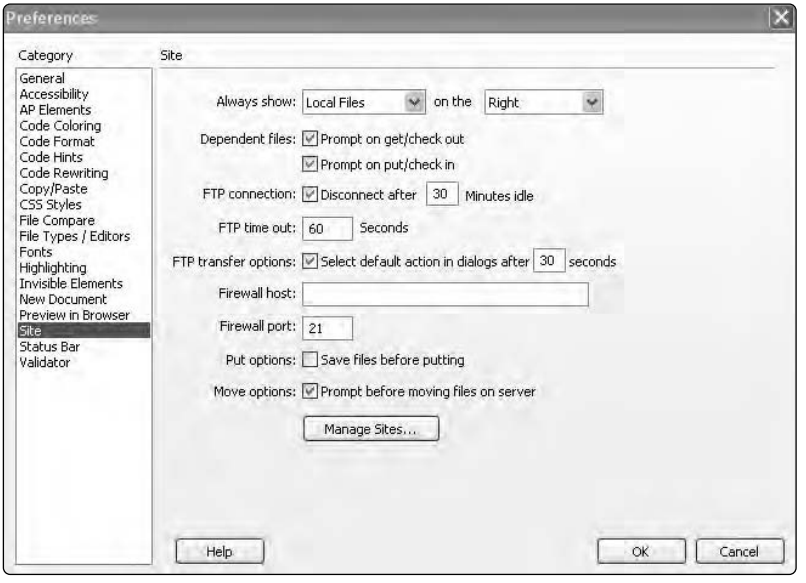
Default extension: Varsayılan dosya uzantısını belirler.

Default document type (DTD): Belge türünü (*Document Type Definitions*) seçmek içindir. Bildiğiniz gibi biz şu ana kadar **XHTML 1.0 Transitional** kullandık.

Default Encoding: Belgede kullanılacak karakter kodlamasını belirler.

Show New Document Dialog on Control+N: `CTRL+N` tuş kombinasyonu ile **New Document** penceresinin açılması içindir.

SİTE KATEGORİSİ



Always Show: Yerel ya da uzak sunucunun ekranın neresinde görüntüleneceğini belirler.

Dependent Files: Belgemize bağlı dosyaların (örn: harici CSS ya da JavaScript dosyası, resim... vb) uzak sunucuya gönderilmesi ya da uzak sunucudan alınması işlemleri olduğunda Dreamweaver'ın bizi uyarması içindir.

FTP Connection-Disconnect after: Bu kutucuğa yazacağınız süre zarfında bir işlem yapılmadığında bağlantıyı keser.

FTP Time Out: FTP bağlantınızın zaman aşım süresini belirlemek içindir.

Put Options-Save files before putting: Dreamweaver'ın dosyalarınızı sunucuya göndermeden önce kaydedip kaydetmeyeceğini belirlemek içindir.

Move options-Prompt before moving files on server: Dosyalarınızı sunucuya göndermeden önce Dreamweaver'ın sizi uyarmasını sağlamak içindir.

Manage Sites: Bu tuşa tıkladığınızda açılan pencereden oluşturduğunuz sitelerinizi yönetebilirsiniz. Bu pencereye **Site/Manage sites** menüleri aracılığıyla da ulaşabilirsiniz.

Yeni Bir Site Oluşturmak

Dreamweaver ile verimli bir çalışma yürütebilmek için ve Dreamweaver'ın nimetlerinden etkili bir biçimde yararlanabilmek için çalışmanıza başlamadan evvel bilgisayarınızda tüm çalışma dosyalarınızı ve sitenizde kullanacağınız resim, video... vb tüm materyalleri içeren bir klasör oluşturmalısınız. Hemen uygulamaya geçelim ve **Site>New Site** menüsünden site oluşturma penceresini açalım.



İlk olarak sitemizin adını belirleyelim. Mesela sitemizin adı `ilk sitem` olsun. Sitemizi kendi bilgisayarımızda hazırlayacağımız için URL adresi kısmını boş bırakıyoruz ve **Next** tuşu ile bir sonraki aşamaya geçiyoruz.



Gelen sayfada bize sunucu taraflı (örn: ilerde öğreneceğimiz PHP) bir teknoloji ile çalışıp çalışmayacağımızı soruyor. Biz şimdilik XHTML ve CSS ile çalışacağımız için bu soruya cevap olarak "No..." radyo düğmesini işaretleyip yine **Next** tuşu ile yolumuza devam ediyoruz.



Bu sayfada ise program bize geliştirme sırasında dosyalarımızı bilgisayarımızda mı tutacağımızı yoksa doğrudan uzak sunucuda mı çalışmak istediğimizi soruyor.

Bu aşama biz kendi bilgisayarımızda çalışacağımız için ilk sıradaki ve aynı zamanda **Recommended** (*önerilen*) radyo düğmesini seçerek dosyalarımızı kendi bilgisayarımızda oluşturmayı seçiyoruz. Ardından alttaki kutucuktan ya da kutucuğun yanındaki klasör işaretine tıklayarak, site klasörümüzün (ilk sitem) bilgisayarımızda nerede bulunacağını seçip ilerliyoruz.



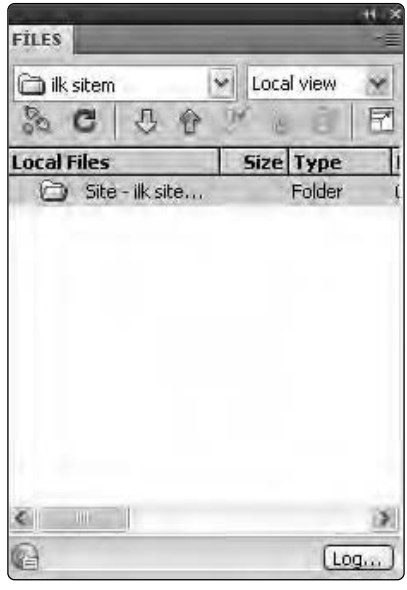
Bir sonraki aşamada uzak sunucuya nasıl bağlanacağımızı soran bir sayfa gelecektir. Dreamweaver ile herhangi bir FTP programına gerek kalmadan tüm sitenizi uzak sunucuya aktarabilir, istediğiniz zaman yeni dosya gönderebilir ya da mevcut dosyaları güncelleyebilirsiniz.

Dosyalar panelindeki **connect** tuşu ile sunucunuza bağlanabilir ya da ilk kez bağlanacaksanız gerekli tanımlamaları yapabilirsiniz. Biz şimdilik kendi bilgisayarımızda çalışacağımız için **none**'ı seçelim. İşte connect tuşuna tıkladığınızda, yine resimde gördüğünüz ekran karşınıza gelecek ve buradaki açılır kutucuktan FTP seçeneğini aktif ederek gerekli FTP bilgilerinizi gireceksiniz. Bu açıklamadan sonra biz **Next** ile bir sonraki aşamaya geçelim.



Bu sayfa şu ana kadar programda yaptığımız seçenekleri gösteren bir sayfadır. Amacı işlemi bitirmeden tercihlerimizi son bir kez kontrol etmemize imkan vermektir.

Bu sayede gerektiğinde önceki aşamalara geri dönerek tercihlerimizi değiştirebiliriz. Biz **Done** tuşuna basarak işlemimizi tamamlayalım.



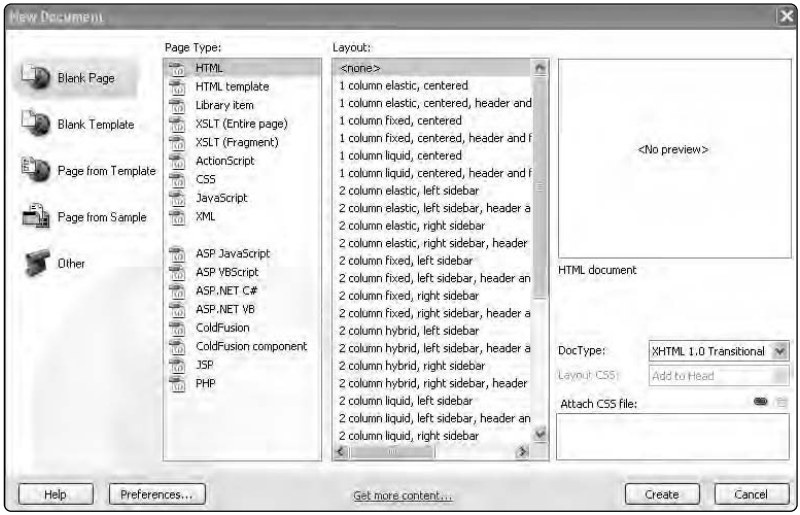
Artık dosya panelinde kendi site klasörümüzü görebiliyoruz. Artık tüm dosyalarımız ve sitemizde kullanacağımız tüm materyaller bu klasör içinde tutulacak. Bu panel ile hızlı bir şekilde bir dosyayı açabilir ya da fare ile **SAĞ** tıklayıp açılan menü yardımıyla **yeni bir dosya** (*New File*) yada **yeni bir klasör** (*New folder*) açabiliriz.

Dilersek mevcut bir dosyayı ya da klasörü yine aynı menü yardımıyla (*Edit/Delete*) silebiliriz. **Edit** menüsü aynı zamanda dosyamızın adını **değiştirmek** (*Rename*), **taşımak** (*Cut)* ya da **kopyalamak** (*Copy*) için de kullanılır.

Yeni Bir Sayfa Oluşturmak

Yerel sitemizi tanımladığımıza göre artık ilk sayfamızı oluşturmaya başlayabiliriz. **File>New** menüsü yardımıyla yeni belgemizi seçeceğimiz **New Document** sayfasınız açıyoruz.

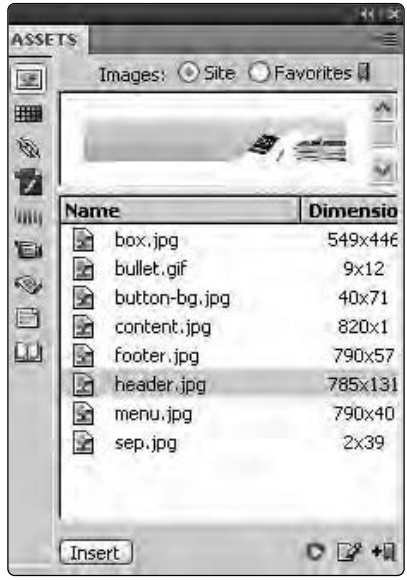
Yan sayfadaki resimde de gördüğünüz gibi Dreamweaver tarafından desteklenen tüm teknolojilerin bir listesi ile karşılaştık. Tahmin edeceğiniz gibi belgemizi hangi teknoloji ile (örn: html, PHP...vd) yazacak isek onu seçeceğiz. Biz şimdi **Page Type** olarak **HTML**, **Layout** olarak da `<none>` seçelim. Sağ alt kısımdan da `DOCTYPE` olarak önceki bölümlerde olduğu gibi yine **XHTML 1.0 Transitional** seçiyoruz. Artık **Create** tuşuna basarak dosyamızı oluşturabiliriz.



Bu aşamada ilk yapmamız gereken iş dosyamızı kaydetmek olmalıdır. Bunu alışkanlık haline getirelim. Hemen **File>Save** menüsü ile kayıt sayfasını açıyoruz.

Dosyamıza bir isim vererek ilk sitem klasörünün içine kaydediyoruz. Biz dosyamıza index adını verdik çünkü bu bizim sitemizin ana dosyası olacak. Dosyanıza isim verirken bu ismin siteyi yayınladığınızda, bu sayfaya ulaşmak için gereken URL'nin bir parçası olacağını unutmayalım. Bu sebeple dosya isminde `?`, `%`, `*`, `>`, `/` gibi özel karakterleri ve boşluk kullanmayın. Dosya adı iki kelimeden oluşuyorsa boşluk yerine _ (alt tire) işaretini tercih edin.

VARLIKLAR



Varlıklar (*assets*) terimi, sitenizdeki belgelerde kullanacağınız resim, video, renk, link... vb tüm materyaller için kullanılır. Dreamweaver **assets** paneli tüm bu materyallerin elimizin altında olmasını ve bunları gerektikçe belgelerimize kolaylıkla eklememizi sağlar.

Bir varlığı, örneğin bir resmi belgemize ekleyebilmek için, resmi fare ile tutup, belgemizde eklemek istediğimiz yere sürüklememiz yeterli olacaktır.

Aynı işlemi, fare imlecini belgede varlığı eklemek istediğiniz yere konumlandırdıktan sonra assets panelinin alt kısmındaki Insert tuşuna basarak da yapabilirsiniz. Renk eklemek istediğimizde ise listeden rengi seçtikten sonra, örneğin bir metnin rengini değiştirmek istediğimizde, renklendireceğimiz metni fare ile seçip **Apply** tuşuna basmamız gerekir. Birden çok varlığı seçmemiz gerektiğinde **SHIFT** tuşuna basmalısınız.

Resimde gördüğünüz gibi her bir varlık türü için ayrı bir tuşumuz mevcut. Ayrıca üst kısımda **Site** ve **Favorites** olmak üzere iki radyo düğmesi görüyorsunuz. Bunlardan **Site** düğmesi, sitenizdeki tüm varlıkları gösterir. Diğer düğmeyi tıkladığınızda ise sık kullandığınız varlıkların listesini göreceksiniz.

Bir varlığı sık kullanılanlarınıza ekleyebilmek için site düğmesi aktif iken listeden bir varlık seçip panelin sağ alt kısmındaki **add to favorites** tuşuna basmanız yeterli olacaktır. Varlıkları sıralamak için listenin baş tarafındaki sütun başlıklarına (örn: name, size, type... vd) tıklayabilirsiniz. Mesela varlıkları ada göre sıralamak için name yazan alanı tıklamalısınız. Varlık listesinde değişiklikler yaptığınızda bunları görmek için panelin alt kısmındaki **refresh** tuşuna basın.

KİTAPLIK

Kitaplık, varlıklardan oluşan Dreamweaver'a has bir dosyadır. Bir kitaplık öğesi oluşturmak için, **Assets** panelinde **Kitaplık** tuşuna basarak aktif ettikten sonra belgenizin kitaplık öğesi olarak kaydetmek istediğiniz bölümünü seçip panelin alt kısmındaki **New Library Item** tuşuna basın. Kitaplık öğesine istediğiniz bir ismi verin. Dreamweaver kitaplık öğelerini (*.lbi* dosya uzantısı ile) bir dosya olarak kaydeder.

Dilerseniz daha sonra düzenlenmek üzere boş bir kitaplık öğesi de oluşturabilirsiniz. Bunun için anlattığım işlemi belgenizde herhangi bir şey seçili değilken yapmanız yeterli olacaktır.

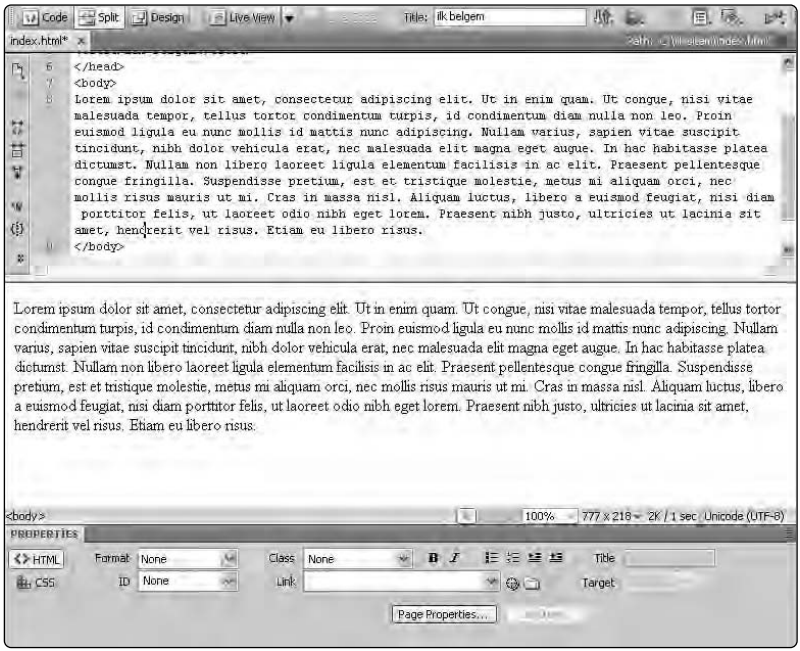
Daha sonra bu kitaplık öğesini düzenlemek istediğinizde öğe seçili iken panelin alt kısmındaki **Edit** tuşuna basın. Dosya yeni bir sayfada açılacaktır.

Bu sayfada gereken düzenlemeleri yapabilirsiniz. Tüm varlıklarda olduğu gibi kitaplık öğelerini de belgeye eklemek istediğinizde, belgenizde kitaplık öğesini ekleyeceğiniz yeri fare ile belirledikten sonra öğeyi sürükleyebilir ya da panelin alt kısmındaki **Insert** tuşunu kullanabilirsiniz.

BELGE OLUŞTURMAK

Artık ilk belgemizi oluşturmaya hazırız. Öncelikle **Title** kısmından belgemizin başlığını (örn: ilk belgem) girelim. Dilediğimiz görünümü seçip çalışmaya başlayabiliriz. Biz **Split** görünümünü seçerek hem tasarım hem de kod görünümü ile çalışarak olup bitenleri daha iyi izleyebilelim.

Dreamweaver ile oluşturacağınız metni sayfanın alt kısmındaki **Properties** (*Özellikler*) penceresi ile rahatça biçimlendirebilirsiniz.



Resimde gördüğünüz gibi ilk belgemize ham bir metin ekledik. Alttaki properties penceresinde de hiçbir biçimlendirme olmadığına dikkat edin. İlk olarak Design görünümünde iken **ENTER** tuşuna basarak ya da split görünümünde iken **Format** kısmından **Paragraph**'ı seçerek paragrafımızı oluşturalım. Ardından başlığımızı seçip **Heading2** (h2) ile başlığımızı biçimlendirelim.

Şimdi de tüm metni seçip iki yana yaslı şekilde hizalama yapalım. Bunun için metni taradıktan sonra gene properties penceresinden CSS tuşuna tıklayalım. Ardından hizalama tuşuna tıklayalım. **New CSS Rule** penceresi açılacaktır. Bu pencere gerçekten harika bir pencere. Çünkü bu pencere sayesinde şıp diye tasarım işini halledeceğiz.

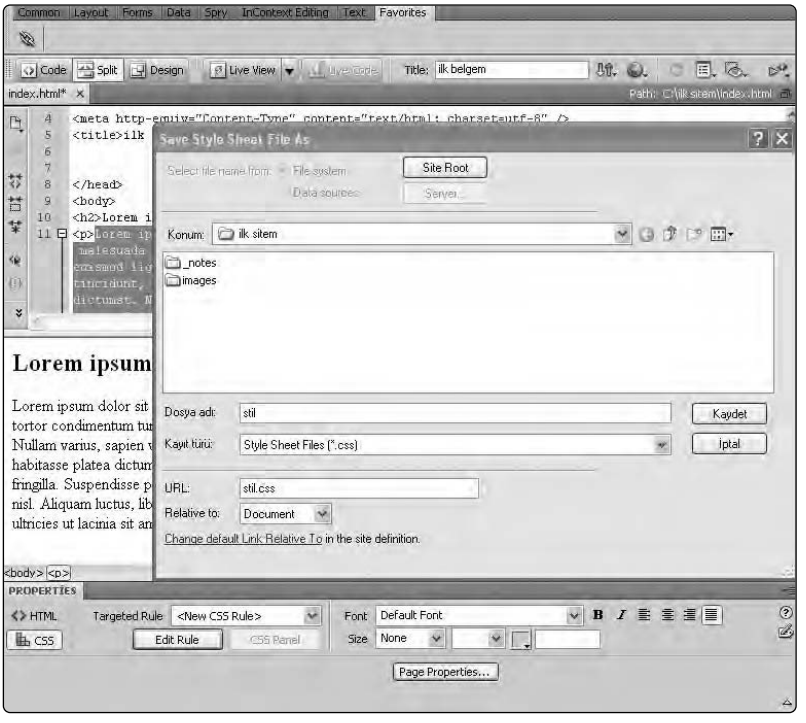
Selector Type yani seçici açılır kutucuğundan seçimimizin türünü belirleyeceğiz. Bir html etiketi mi, id seçicisi mi yoksa bir class mı tanımlayacağımıza karar vermeliyiz. Eğer html etiketi seçersek alttaki **Selector Name** kısmına paragraf etiketi olan p etiketini program kendisi ekleyecektir. **Class** ya da **id** seçicisini tercih edersek bir seçici adı girme-

miz gerekecek. Örnek olarak biz class'ı seçelim ve seçici adı olarak da `paragraf` girelim.

Burada biz seçici adını nokta koymadan giriyoruz çünkü noktayı Dreamweaver halledecek. Son olarak alttaki **Rule Definition** kısmından bu kuralımızın sadece bu belge için mi geçerli olacağını yoksa harici bir CSS dosyası mı oluşturacağımızı soruyor.



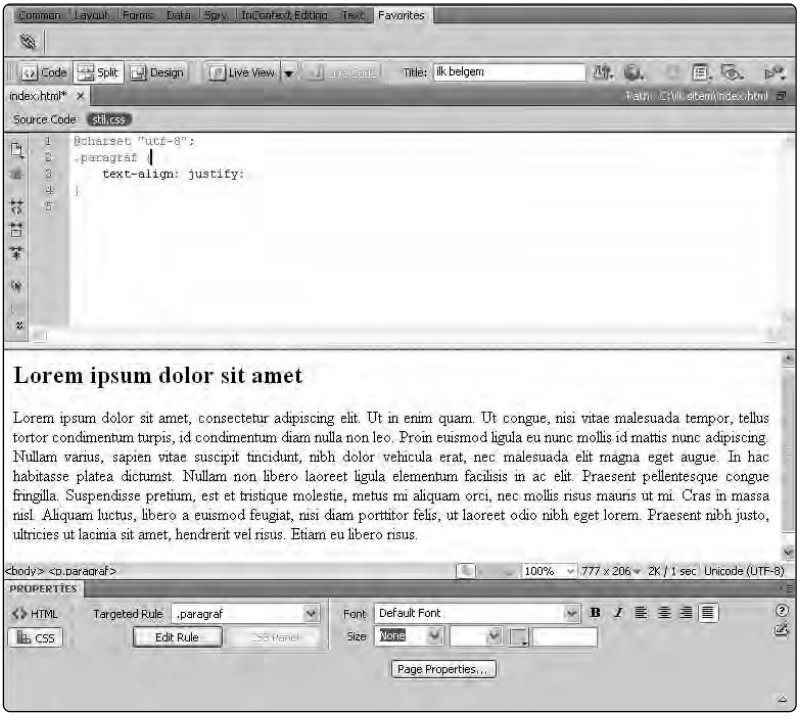
Burada ilk olarak **New style sheet file**'ı seçelim ve **okey** (**OK**) tuşuna basalım. Evet tam işimiz bitti derken bir sayfa daha açıldı. Biz yeni bir CSS sayfası oluşturmak istediğimiz için program şimdi bizden CSS dosyamızı oluşturmamızı istiyor.



Bu sayfada dosyamıza bir isim seçelim (ben stil dedim) ve site klasörümüzün (ilk sitem) altına kaydedelim. Neler oldu gördünüz mü? Evet şimdi isterseniz beraber görelim.

Çalışma sayfamızın üst kısmında source **code** ve **stil.css** ifadesi çıktı. Dreamweaver çalıştığımız belgeye bağlı dosyaları da bize gösteriyor ve işimizi kolaylaştırıyor.

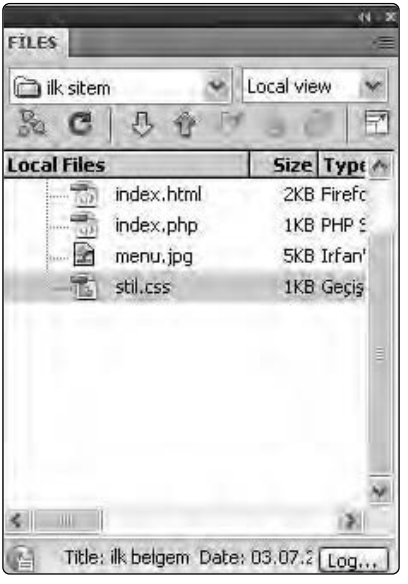
Şimdi **stil.css** dosya adına tıklayın.



İşte bu gerçekten harika. Biz kod falan yazmamıştık ama birisi bizim yerimize yazmış galiba. Ve alt kısımda da gördüğünüz gibi metnimiz iki yana yaslı duruma geldi.

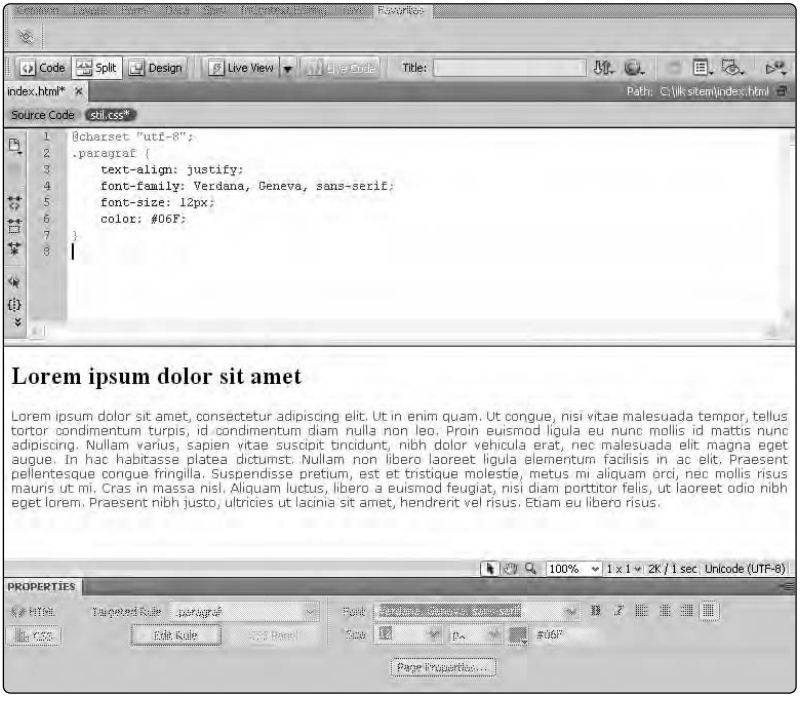
Şimdi **source code**'a tıklayıp belgemize dönebiliriz.

Şimdi de bu resme bakın. Evet gördüğünüz gibi **stil.css dosyamız** files panelinde yerini almış. Buradan da gerektiğinde dosyamızı açma silme taşıma... vb işlemleri yapabileceğiz.



Şimdi **Properties** penceresi yardımıyla birkaç işlem daha yapalım. Mesela font ve font büyüklüğünü seçelim.

Bir de metnimizi mavi renkte yapalım. Tekrar CSS dosyamızı tıklayalım.



O da ne? Birisi gene bir sürü kod yazmış buraya. Siz bu kodların neler olduklarını zaten biliyorsunuz. Biz devam edelim ve ilginç bir şey yapalım. Başlığımızı seçip ona da mesela renk verelim. Bunun neresi ilginç şimdi göreceğiz. Haydaa gene **New CSS Rule** penceresi açıldı. Çünkü başlık `h2` etiketi ile şekillenen ayrı bir etiket de ondan. Gene class seçelim ve baslik adını verelim. **Rule Definition** açılır kutucuğunda bu sefer bizim **style.css** dosyamız da var. Tabiki onu seçiyoruz ve **OK**'liyoruz. CSS dosyamıza `baslik` seçicisi de öylece eklenmiş oldu. Resimde de gördüğünüz gibi properties penceresinde HTML ve CSS tuşlarına basarak yaptığımız işlemleri görebilirsiniz.



Peki biz harici stil sayfası kullanmasa idik ne olurdu. İsterseniz şimdi de onu görelim. Önce **files** panelinden **stil.css** dosyamızı bir güzel silelim. Bunun için dosya seçili durumda iken fare ile **SAĞ** tıklayıp **Edit/Delete** menülerini kullanın. Haliyle dosyamızı silince tüm stillerimiz uçtu ve belgemiz ilk ham haline geri döndü. Olsun biz onu şimdi anında eski haline getiririveririz.

Tekrar metnimizi seçip iki yana yaslama tuşuna basalım. Açılan pencereden **class**'ı seçip seçicimize paragraf adını verelim. En alttan da **This document only** seçeneğini tıklayıp ok diyelim. Bu işlemle CSS kuralımızın sadece bu belge için geçerli olacağını belirtmiş olduk.

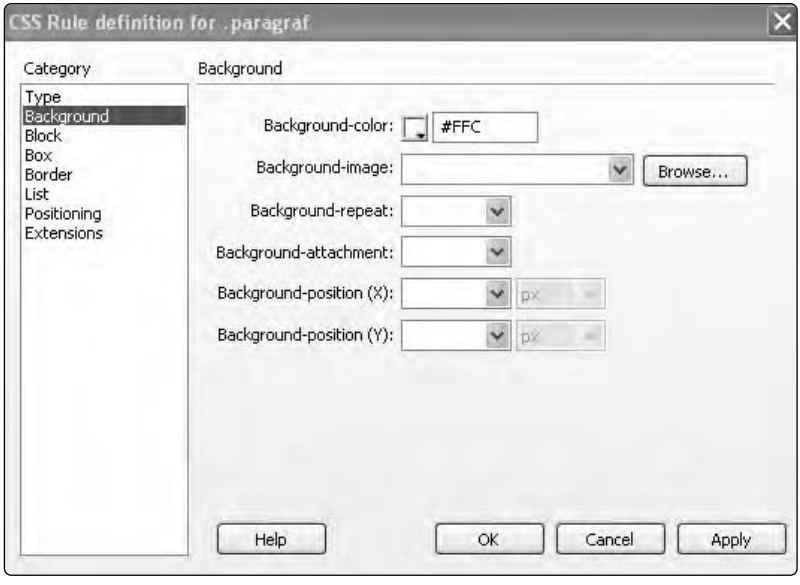


Evet gördüğünüz gibi bu sefer kodlarımız belgemizin <head> </head> etiketleri arasına eklendi. Evet tıpkı bizim CSS bölümünde öğrendiğimiz gibi.



Bir dahili stili harici stil dosyasına dönüştürmek isterseniz stil üzerindeyken **SAĞ** tıklayıp açılan menüden **CSS Styles/Move CSS Rules** alt menüsünü seçin. Bu pencere yardımıyla mevcut bir stil dosyasını seçebilir ya da yeni bir stil dosyası oluşturarak stili oraya kaydedebilirsiniz.

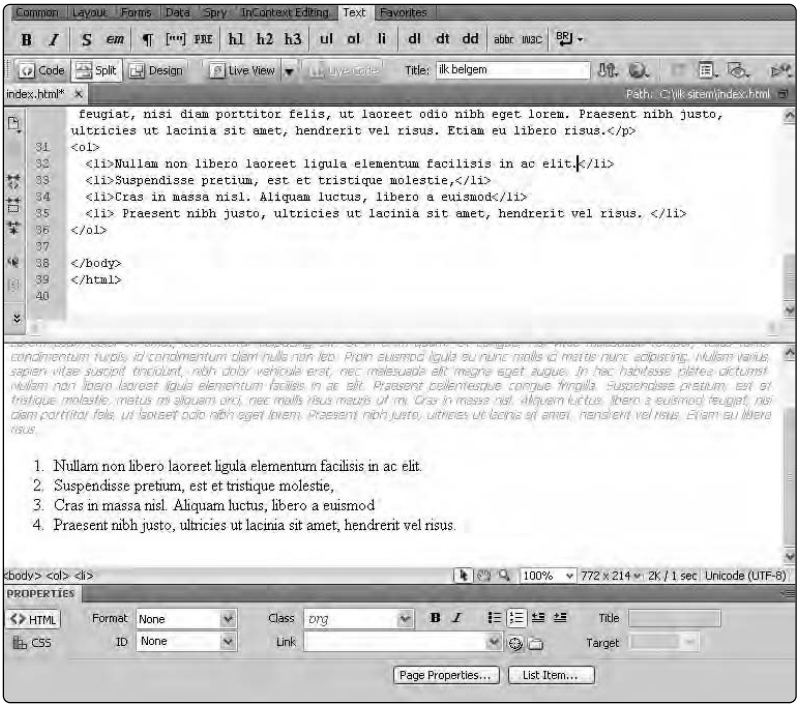
Biz stillerimiz sayfa içine ekleyerek çalışmamıza ve çalışmalarımızda split görünümünü kullanmaya devam edelim. Yeni bir paragraf yapmak istediğimizi var sayalım. Bunun için split ekranının alt kısmında, yani tasarım görünümünde iken bir kez **ENTER** tuşuna basmamız yeterli olacaktır. Program anında paragraf etiketini ve seçicimizi yerine yerleştirecektir. (`<p class="paragraf"> </p>`) Paragrafta gördüğünüz ` ` boşluk ifadesinin yerine biz yeni paragrafımızı ekleyeceğiz.



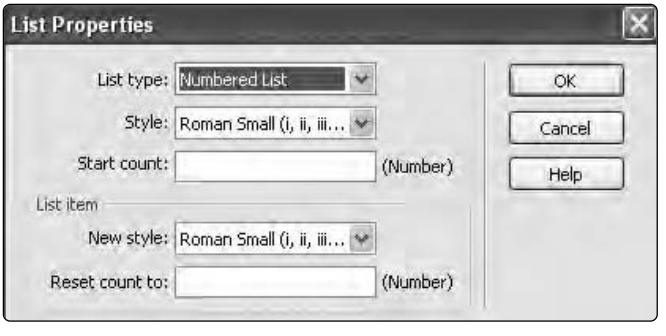
Bir metne arkaplan eklemek istediğimizde, metni seçip **Properties** penceresindeki E**dit Rule** tuşuna tıklayıp açılan pencereden renk seçimi yaparak istediğiniz rengi ekleyebilirsiniz. Bu işlemi **CSS Style** paneli ile de yapabilirsiniz. Yalnız seçeceğiniz renk `.paragraf` adlı seçicinin atandığı paragrafa uygulanacaktır.

Tüm sayfaya arkaplan rengi vermek için **Properties** penceresindeki **Page Properties** tuşuna tıklamalısınız. Açılan sayfada yapacağınız tüm değişiklikler sayfanın tamamında geçerli olacaktır.

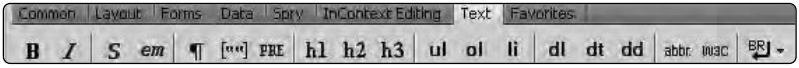
Madde İmleri Eklemek (Listeler)



Belgemize madde imleri eklemek istediğimizde de yine properties penceresini kullanacağız. Penceredeki madde imleri tuşlarıyla bunu rahatlıkla yapabiliriz. Her bir paragraf ayrı bir madde imi olarak algılanacaktır. Madde imlerinden sonra başka bir satıra geçmek için **CTRL+ENTER** tuşlarını kullanın.



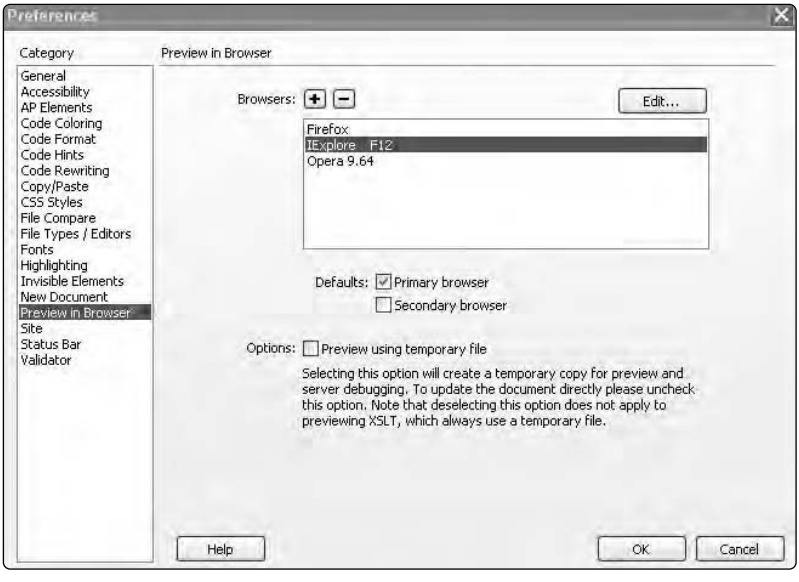
Oluşturduğunuz listede değişiklik yapmak istediğinizde **Properties** penceresindeki **list Item** tuşuna basabilir ya da **Format>List>Properties** menüsünü kullanabilirsiniz. Açılan pencereden istediğiniz değişiklikleri yapabilirsiniz. Dreamweaver'ın mantığını sanırım anlatabildim. Siz tasarım ekranında bir kelime işlem programında çalışır gibi tasarımınızı yaparken Dreamweaver programı gerekli etiketleri ve kodları belgeye ekleyecektir. Tabi ki dilerseniz kod görünümü kısmından kodları kendiniz de yazabilirsiniz.



Metninizi şekillendirmek ya da metninize özel karakterler eklemek için çalışma alanının üst kısmındaki **Ekleme Paneli**'nin **Text** sekmesini de kullanabilirsiniz.

ÖNİZLEME

Yaptığımız çalışmaları her aşamada tarayıcılarda ön izleme yaparak test etmelisiniz. Bu sayede muhtemel tarayıcı farklılıklarını zamanında fark edip düzeltebilirsiniz. Bunun en pratik yolu **F12** tuşuna basarak sayfamızın tarayıcıda açılmasını sağlamaktır.



Varsayılan tarayıcınız haricindeki diğer tarayıcılarda da sayfanızı test etmek için **File/Preview in Browser** menüsünden istediğiniz tarayıcıyı seçerek kontrolünüzü yapabilirsiniz. Listeye yeni bir tarayıcı eklemek için aynı menüdeki **Edit Browser** alt menüsünü seçin. Bu pencerede varsayılan tarayıcınızı değiştirebileceğiniz gibi **+** (artı) işaretine basarak yeni tarayıcı da ekleyebilirsiniz.



File>Preview in Browser menüsünden **Device Central** alt menüsü ile sayfanızın cep telefonlarında nasıl görüneceğini de test edebilirsiniz.

Link (Bağlantı) Eklemek

Link, bildiğiniz gibi, bir sayfa içindeki iki noktayı ya da farklı sayfaları birbirleri ile ilişkilendirmek için kullanılır. Gidilecek nokta ve çıkış noktası olmak üzere her bağlantıda iki unsur söz konusudur. Biz burada, sayfa içi ve sayfa harici link verme işlemlerine değineceğiz.

Dahili Link Eklemek

Uzun bir sayfanın bir bölümünden o bölüm ile ilişkili bir başka bölümüne tek bir tıklama ile gidebilmek için bu iki nokta arasında bir bağlantı oluşturmamız gerekir. Bunun için sayfamızda bir çapa oluşturacağız.

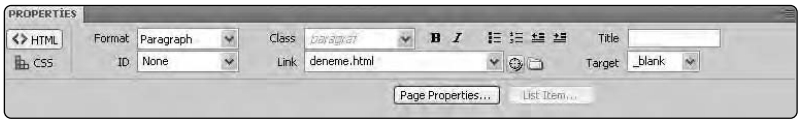


Çapamızı sayfanın gidilmesini istediğimiz yerine atıyoruz. Örneğin sayfanın başından alt kısmına gitmesini istiyorsak, sayfanın alt kısmında istediğimiz noktaya imleci getirdikten sonra **Insert>Named Anchor** menüsü ile ya da insert paneli common sekmesindeki **Named Anchor** tuşu ile çapımızın adını vereceğimiz pencereyi açıyoruz. Çapamıza bir isim verip onaylıyoruz. Şimdi sayfamızın başındaki link vereceğimiz metni seçip properties penceresinde **Link** kutucuğuna çapamıza verdiğimiz adı, başına "#" işareti koyarak (örn: `#capa`) yazıyoruz. Artık linkimiz hazır. Linki tıkladığımızda kendimizi sayfanın en altında bulacağız. Resimlere link vermek için de yine aynı yöntemi kullanabilirsiniz.

E-posta Linki Eklemek

Sayfanızda bir e-posta linki oluşturmak için link vereceğiniz metni seçik Link kutucuğuna e-posta adresini, başına **mailto** ibaresi ile beraber (örn: `mailto:info@siteismi.com`) yazmanız yeterli olacaktır. Bu işlemi de yine insert panelindeki **Email Link** tuşu ile yapabilirsiniz.

Harici Sayfalara Bağlanti Vermek (Hyperlink)



Öncelikle link vereceğimiz metni seçiyoruz. Örneğin sitemizin iletişim sayfasına gitmek için anasayfamızdaki menümüzden iletişim yazısını seçelim ve yine **properties** penceresinden **Link** kutusunun yanındaki klasör işaretine tıklayalım. Açılan pencereden dosyamızı seçelim. Eğer link verdiğimiz sayfanın yeni bir tarayıcı penceresinde açılması istiyorsak **Target** açılır kutucuğundan **_blank** seçeneğini seçin. Bu işlemi

Link kutusunun yanındaki **Point to file** tuşuna basılı tutarak **Files** panelinde gördüğünüz link vereceğiniz dosyanın adının üstüne sürükleyerek de yapabilirsiniz.

Resim Eklemek

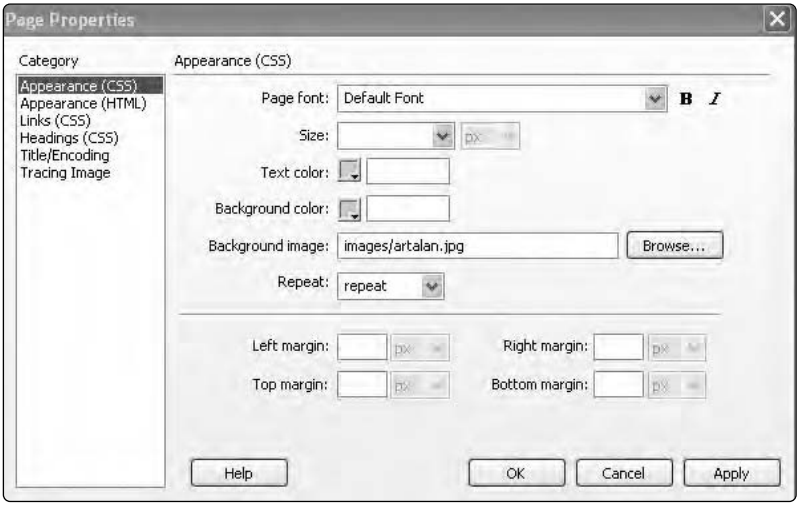


Sayfanıza resim eklemek için imleci resmi ekleyeceğimiz yere konumlandırdıktan sonra **Insert Paneli Common** sekmesinden **Images** tuşuna tıklayalım. Açılan pencereden resmimizi seçip **OK**'leyelim. Resmimiz sayfamıza eklenecektir.

Alttaki **Properties** penceresinden resimle ilgili ayarlamaları kolaylıkla yapabilirsiniz. Resmi hizalayabilir, ebatlarını değiştirebilir, boşluk değerlerini seçebilir, border ekleyebilir, resme link verebilir, resmi kırpabilir, kontrast ve keskinlik değerini ayarlayabilirsiniz.

Sayfaya Arkaplan Eklemek

Sayfaya arkaplan eklemek için **Properties** penceresindeki **Page Properties** tuşuna tıklayın. Açılan pencerede **Background Image** bölümünden **Browse** (*Gözat*) tuşu ile programa arkaplan resminin yerini gösterin.



Ayrıca bu pencereden repeat değerini ve margin değerlerini de girebilirsiniz. Burada bir ince hususu dikkatlerinize sunmak isterim. Eğer seçimi resimde gördüğünüz gibi **Appearance (CSS)** bölümünde yaparsanız, kod `<head></head>` etiketleri arasına ya da varsa harici CSS sayfasına CSS kodu olarak, **Appearance (HTML)** penceresinde eklerseniz `<body background="images/resim.jpg">` şeklinde html etiketine eklenecektir.

Image Map (Resim Haritası) Oluşturmak

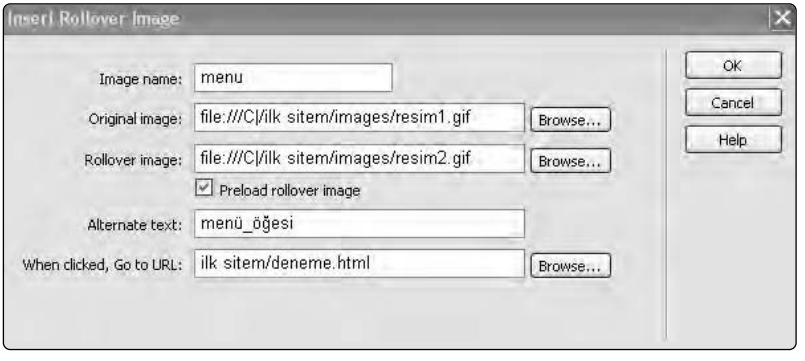
Bazen bir resmin tamamına değil de bir kısmına link vermek isteyebiliriz. Örneğin illeri tanıtan bir site yaptığımızı ve sitemizde her ili tanıtan ayrı bir sayfamız olduğunu varsayalım. Anasayfamızdaki Türkiye haritası resmimizden bir ili tıklayıp o ilin sayfasına gitmek istiyoruz. Bu durumda, resmin sadece sayfasına gitmek istediğimiz ili (örn: Konya) gösteren bölümüne ayrı bir link vermemiz gerekir.



Bu işlem için **Properties** penceresindeki **Hotspot** tuşlarından uygun olanı tıklayıp (bizim işaretleyeceğimiz bölge düzgün bir şekil olmadığından Polygon şeklini seçelim) resmin link vereceğimiz bölgesinin etrafını (örneğimizde Konya il sınırlarını) farenin **+** işareti ile tıklayarak belirliyoruz. Bu bölgede yarı saydam mavi bir alan belirecektir. Resimdeki gibi **Properties** penceresi yardımı ile linklimizi oluşturup işlemimizi tamamlıyoruz.

ROLLOVER IMAGE EKLEMEK

Rollover kavramını kısaca, faremizi bir metnin ya da bir resmin üzerine götürdüğümüzde resimde meydana gelen değişimdir diye özetleyebiliriz.



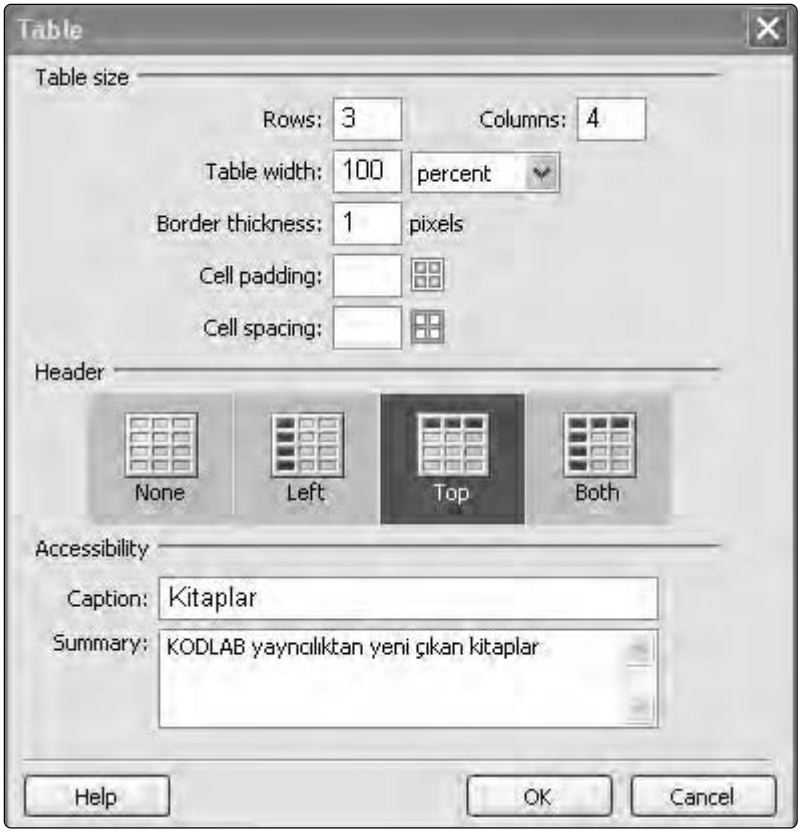
Bir rollover imege oluşturabilmek için ilk olarak **Insert** paneli **Common** sekmesindeki **Rollover** tuşuna tıklayalım. Açılan pencerede resme bir isim verelim. Önce orijinal resmi sonra da resmin üzerine geldiğimizde görüntülenecek resmi **Browse** tuşları ile seçelim. **Preload rollover image** kutucuğunun tıklı olmasına dikkat edin.

Bu özellik, ikinci resmin daha ona ihtiyaç olmadan sayfaya yüklenmesini, dolayısıyla da orijinal resmin üzerine tıklandığında ikinci resmin açılması için beklemememizi sağlar. Son olarak alt metnini ve resme

tıklandığında gidilecek bağlantıyı girip **OK**'lediğinizde Dreamweaver gereken kodları sayfanıza ekleyecektir.

TABLOLAR

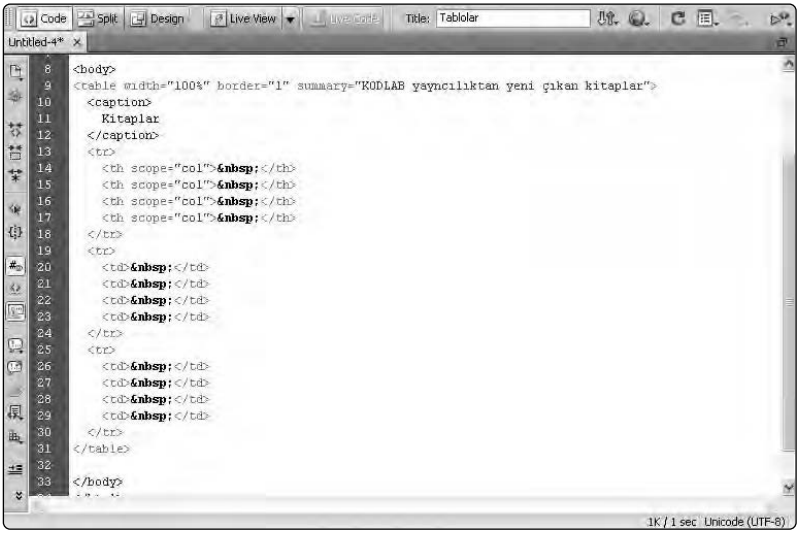
Dreamweaver tablo oluşturma konusunda da bize bir çok kolaylıklar sağlamaktadır. Tablo oluşturmak için **Insert>Table** menüsünü ya da **Insert** panelindeki **Common** sekmesinden **Table** tuşunu kullanabilirsiniz. **Table** tuşuna tıkladığınızda resimdeki sayfa ile karşılaşacağız.



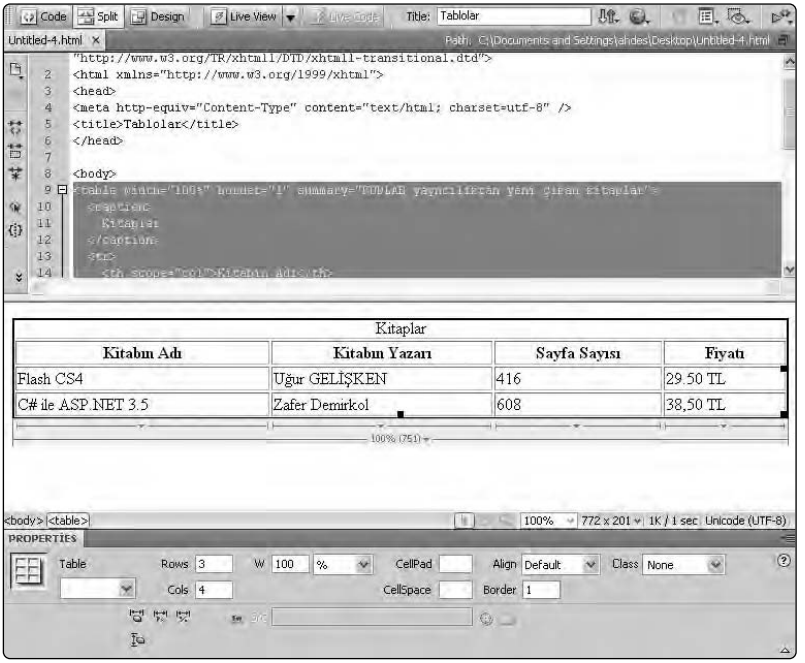
Bu sayfa üç kısımdan oluşmaktadır. **Table Size** bölümünden tablomuzun kaç satır ve sütundan oluşacağını, tablomuzun genişliğini, border değerini, tablomuzun etrafındaki ve tablo hücreleri arasındaki boşlukla-

rı ayarlayacağız. **Header** bölümünden tablomuzda satır ya da sütun başlığı olup olmayacağını, olacaksa konumunu belirleyeceğiz...

Accessbility bölümünden ise erişilebilirlik ayarlarını yapacağız. Bunun için bir tablo açıklaması ve tabloyla ilgili bir özet açıklama metni girebilirsiniz. **Caption** kutucuğuna ekleyeceğiniz başlık, tablonun başında görünecektir. Fakat **Summary** bölümüne ekleyeceğiniz özet açıklama metni sayfada görünmez. Özellikle engelliler için bu açıklama çok yararlı olacaktır. Biz resimdeki gibi gerekli değerlerimizi girelim ve **OK**'leyelim.



Dreamweaver, resimdeki gibi tabloyu bizim için oluşturacaktır. Hücreleri ise **boşluk** (` `) değeri ile dolduracaktır.

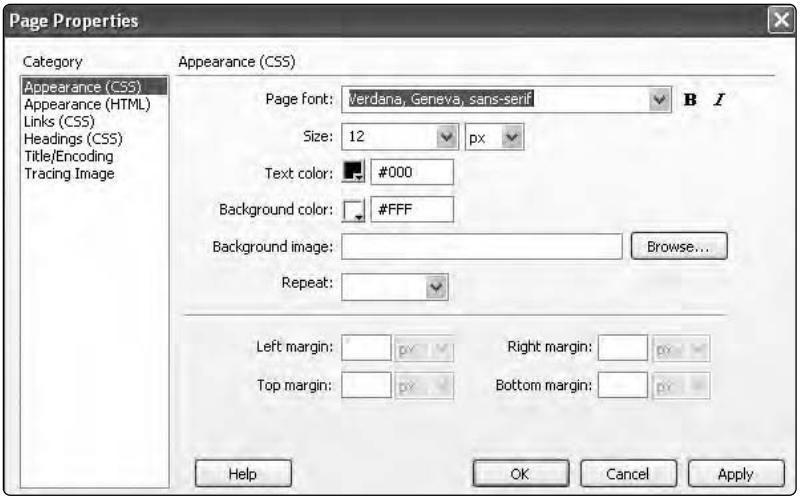


Şimdi de **Split** görünümünde tasarım kısmından içeriklerimizi hücrelere girelim ve resimdeki sonuca ulaşalım. Gördüğünüz gibi tablomuzu oluştururken **Caption** kısmına girdiğimiz açıklama, tablomuzun üst kısmında ve ortalı olarak yer aldı. **Table** penceresinde de **Header** kısmından **Top** seçeneğini aktif ettiğimiz için tablomuzun ilk satırındaki veriler ortalı ve **bold** olarak işlendi. İlk satırdaki veriler sütunların başlıkları olduğundan seçimimizi böyle yaptık.

Bu aşamada dikkatimizi **Properties** penceresine odakladığımızda, tablomuzu oluştururken belirlediğimiz değerleri buradan kolaylıkla değiştirebileceğimizi görüyoruz. Ayrıca, tablonun ebatlarıyla oynamak için, tablonun istediğiniz kenarını fare ile tutup sürükleyerek tablo ölçülerini değiştirebileceğinizi de belirteyim.

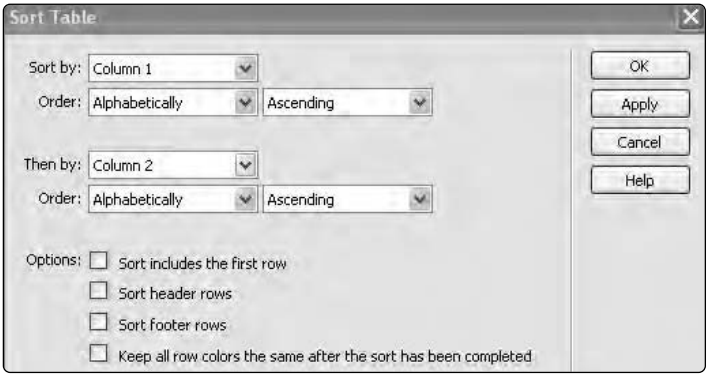


İmlecimizi tablomuzun hücrelerinden herhangi birine konumlandırdığımızda, Properties penceresi resimdeki gibi değişecektir. Pencere, önce tablo özelliklerini gösterirken, hücre özelliklerini gösterecek şekilde değişecektir. Burada dilerseniz tek bir hücrenin özelliklerini değiştirebileceğiniz gibi, dilerseniz de **CTRL** tuşuna basılı tutarak birden çok hücre seçerek düzenleyebilirsiniz.



Tablonun tamamına uygulanacak özellikleri de **Properties** penceresindeki **Page Properties** tuşuna basarak açılan sayfadan ayarlayabilirsiniz. **Apply** ve **OK** dediğinizde değişiklikler **CSS** kodlarına eklenecektir.

TABLO İÇERİKLERİNİ SIRALAMAK



Tablo içeriklerini sıralamak için **Commands>Sort Table** menüsünü seçin. Açılan pencereden sıralamanın hangi sütun esas alınarak yapılacağını (*Sort By*), sıralama ölçütünü (alfabetik ya da sayısal), sıralamanın artan sırada mı azalan sırada mı olacağını, iki hücrenin benzeşmesi halinde sıralamaya esas olacak ikincil sütunun hangisi olacağını (*Then By*), ilk satırın sıralamaya dahil edilip edilmeyeceğini (*Sort includes first row*), başlık satırlarının sıralamaya dahil edilip edilmeyeceğini (*Sort header rows*) belirleyip **Apply** ve **OK** diyelim. Bu uygulamadan sonra tablomuz, kitap adlarına göre alfabetik sırada olacaktır.

Dilerseniz tablomuzun her hangi bir hücresine tıklayarak bu hücre içerisinde bir başka tablo daha oluşturabilirsiniz. Yapmanız gereken ilk tablomuzu oluştururken yaptığımız işlemleri tekrar etmekten ibarettir. Ayrıca, bir Excel tablosunu kopyalayıp Dreamweaver'da sayfanıza yapıştırıp kullanabileceğinizi belirteyim.

Çoklu Ortam Dosyaları

Çoklu ortam dosyalarımızı (örn: video, ses, flash animasyon...vd) belgelerimize ekleyerek, site içeriğimizi daha da zenginleştirebiliriz. Şimdi bu işlemi Dreamweaver ile nasıl yapabileceğimizi göreceğiz.

Bu işlemi **Insert/Media** menüsünden yapabileceğimiz gibi **Insert** panelindeki Media tuşunun yanındaki ok işaretini tıklayarak açılan pencereden, belgemize ekleyeceğimiz çoklu ortam dosya türünü seçerek de yapabiliriz. Bu dosya, sitenizin klasöründe yer alıyorsa bunu varlıklar paneli'ni kullanarak da belgenize ekleyebilirsiniz.



Biz bu yöntemle belgemize bir swf dosyası ekledik. Dosyayı eklediğinizde **Properties** penceresi resimdeki gibi olacaktır. Pencere yardımıyla dosyanızın boyutunu, çevresindeki boşluk değerini, dosyanın kalitesini, arkaplan rengini ayarlayabilir, animasyonu oynatabilir ya da düzenleyebilirsiniz. Diğer dosya türlerini sayfamıza eklemek için kullanacağımız mantık da aynıdır.

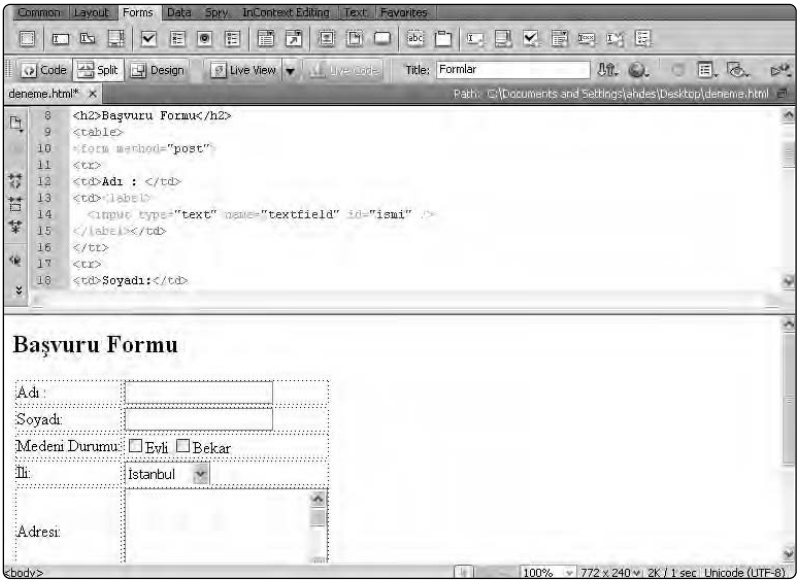
FORMLAR



Dreamweaver ile sayfanıza ekleyeceğiniz formları kolaylıkla oluşturabilirsiniz. Bunun için **Insert/Form** menüsünü ya da **Insert** panelindeki Form sekmesini kullanabilirsiniz. Bu sekme yardımıyla form öğelerini tuşlar yardımıyla formunuza hızlı bir şekilde ekleyebilirsiniz.

İlk olarak en baştaki **Form** tuşuna tıklayın. Bu şekilde formun ana gövdesini sayfanıza eklemiş olacaksınız. Her zaman olduğu gibi yine sihirli Properties penceresine müracaat edeceğiz. Hemen buradan yapabileceğimiz düzenlemelere bakalım.

`Form ID`, formumuzun eşsiz adıdır. `Action`, form girdilerinin **Gönder** tuşuna basıldığında nereye gönderileceğini belirler. (örn: *form.php*). `Method`, ile form girdilerinin nasıl gönderileceğini seçiyoruz. Target, form sayfasının açılma şeklini seçmek içindir. (`_blank` ile yeni pencere; `_self` ile aynı pencerede).



Artık XHTML bölümünde öğrendiğiniz gibi fakat Insert panelindeki tuşlar yardımıyla hiç elle kod yazmadan formunuza istediğiniz alanları ekleyebilirsiniz. Dikkat edeceğiniz husus, tüm form elemanlarını, tasarım görünümünde gördüğünüz kesik kırmızı çizgili alanın içine eklemektir.

Bu çizgiler formun sınır çizgileri olduğundan bu alanın dışına ekleyeceğiniz form elemanları formumuzun bir parçası olmayacaktır. Bu söylediklerimizin kod taraflı ifadesi;

```
<form action="" method="post">FORM ELEMANLARI BURAYA</form>
```

şeklindedir.

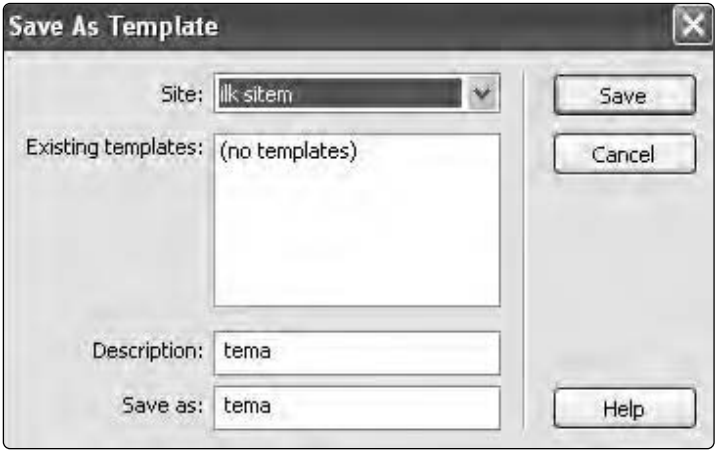
ŞABLONLAR

Çok sayıda sayfadan oluşan bir site tasarlıyorsanız, tasarımınızı her sayfa için ayrı ayrı yapmak yerine, bir şablon oluşturarak bu şablonu sitenizin tüm sayfaları için kullanabilirsiniz. Bu size zaman kazandıracak pratik bir yöntemdir.

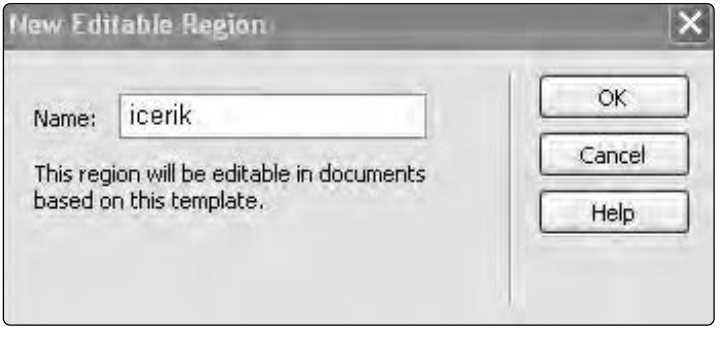
Şablon kullanabilmeniz için öncelikle Dreamweaver'da bir site tanımlamış olmalısınız. Bunun nasıl yapılacağını önceki sayfalarda anlatmıştım.



Öncelikle sitenizin tüm sayfalarında kullanacağınız bir sayfa tasarımı yapmalısınız. Şu ana kadar öğrendiğiniz bilgilerle bunu rahatlıkla yapabilirsiniz. **DVD** içinde **Uygulamalar>Dreamweaver** klasöründe örnek bir şablon bulacaksınız. Biz bu bölümde bu şablonu baz alacağız.



Hazırladığımız sayfayı **File>Save as Template** menüsü yardımıyla şablon olarak kaydedelim. Menüye girdiğimizde açılan pencereden Site kutucuğuna şablonumuzu kaydedeceğimiz siteyi seçelim. **Save** tuşu ile kaydettiğimizde linklerin güncellenmesinin gerektiğini gösteren bir onay kutusu açılacak. **OK** dediğimizde şablonumuz site klasörümüzün içinde, (Dreamweaver tarafından otomatik olarak oluşturulan **Templates** klasörüne) açılır pencerede girdiğimiz isimle ve *.dwt* uzantısıyla kaydedilecektir.

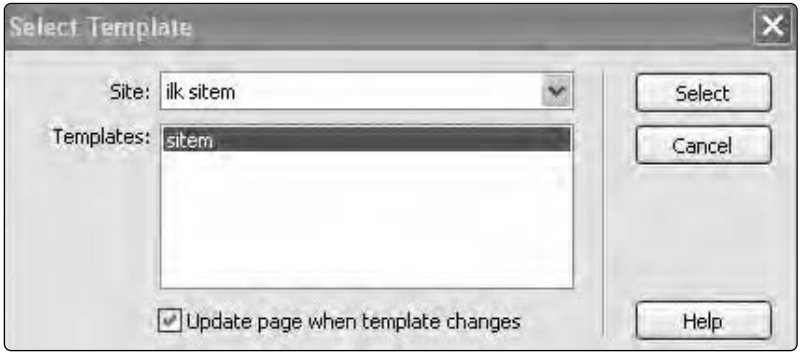


Şablonumuzda içeriğin bulunduğu kısmı seçili duruma getiriyoruz. **Insert>Template Objects>Editable Region** (**CTRL+ALT+V**) menüsü ile seçili alanımızı düzenlenebilir alan haline getiriyoruz. Düzenlenebilir alanımız için bir isim belirliyoruz. Düzenlenebilir alanımızın etrafında, üst kısmında alana verdiğimiz ismin yer aldığı yeşil bir çerçeve oluşacaktır.

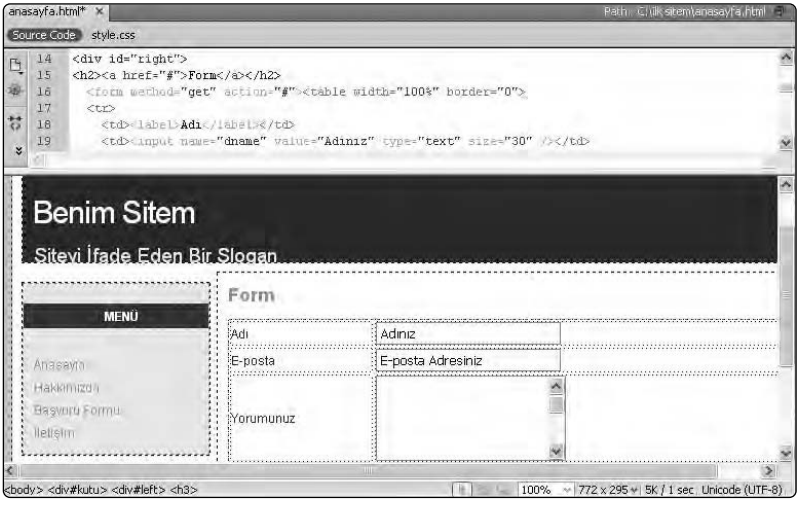
Peki biz bu işlemle ne yapmış olduk? Şablonumuzu sitemizde kullanacağımız dosyalara uygularken, şablonumuzun her sayfada aynı kalacak olan bölümlerini korumuş olduk. Artık yeni sayfalar oluşturulurken sadece düzenlenebilir (*Editable*) alanlara müdahale edilebilecek, sayfanın diğer kısımları ise değiştirilemeyecektir.

Biz örnek olarak sadece şablonumuzun içerik kısmını düzenlenebilir alan olarak belirledik. Dilerseniz şablonda bir çok düzenlenebilir alan belirleyebilirsiniz.

Artık bir şablonumuz olduğuna göre, bu şablonu baz alan yeni bir sayfa oluşturabiliriz. Hemen **File>New** menüsünü kullanarak yeni bir sayfa açalım. Adını örneğin anasayfa koyup, title kısmına da bir isim girerek kaydedelim.

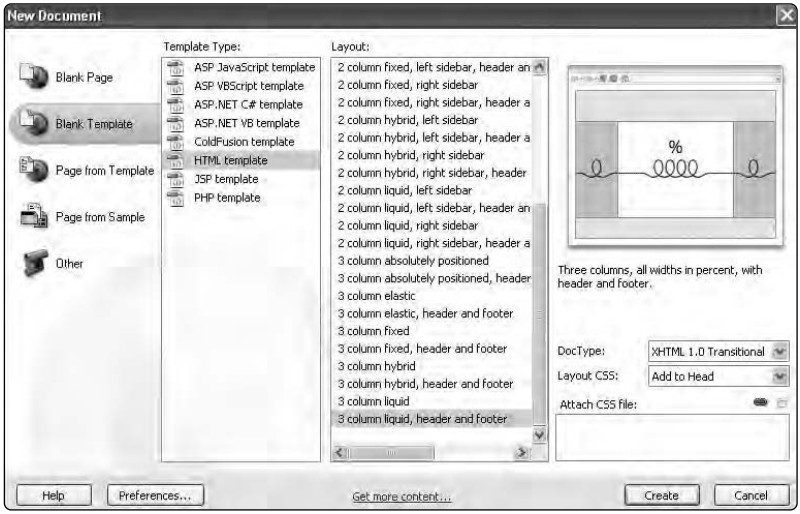


Ardından **Modify>Template>Apply** template to page menüsü yardımıyla, yeni açtığımız sayfaya, şablonumuzu uygulama işlemine başlıyoruz. Açılan pencereden, belgemize uygulanacak şablonu seçiyoruz. **Update page when templat**e onay kutucuğunun tıklı olmasına dikkat ediyoruz çünkü bu kutucuk tıklı olmazsa, ilerde şablonumuzda bir değişiklik yapmamız gerektiğinde, yaptığımız değişiklik bu sayfaya uygulanmayacaktır.



Evet sayfamız aynı şablonumuz gibi oldu. Farklı olarak, sayfanın sadece içerik bölümüne müdahale edebildiğini, diğer bölümlerde hiçbir değişiklik yapılamadığını göreceksiniz. Kod kısmında da, düzenlenebilir alanların haricindeki alanlara ait kodlar soluklaşacaktır. Şimdi sayfanın düzenlenebilir alanını istediğiniz gibi değiştirip sayfayı kaydedebilirsiniz.

Bu şablonla bir işiniz kalmadığında ya da artık onu kullanmak istemediğinizde ve sitenizin sayfalarının düzenlenebilir alanlar dışında kalan kısımlarını da istediğiniz gibi değiştirmek istediğinizde ne yaparsınız? Böylesi bir durumda **Modify>Templates>Detach from template** menüsü yardımıyla hemen sayfanızı normal bir sayfa haline çevirebilirsiniz.



Şimdi **File>New** menüsünden **New Document** penceresini açalım. Burada yandaki **Page from Template** sekmesinden artık şablonumuzu görebiliriz ve gerektikçe kullanabiliriz. Bu pencereye gelmişken, **Page from Sample** sekmesinden hazır bazı CSS stillerini seçip kullanabileceğinizi de belirteyim.

Yine bu pencerede **Blank Template** sekmesinden Dreamweaver'ın hazır şablonlarına ulaşıp kullanabilirsiniz. Özellikle öğrenme evresinde bu hazır şablonların çok faydalı olacağını düşünüorum. Şablonlardan istediklerinizi seçin ve üzerinde denemeler yapın. Gerçekten keyif alacağınızı umuyorum.

SPRY KULLANIMI

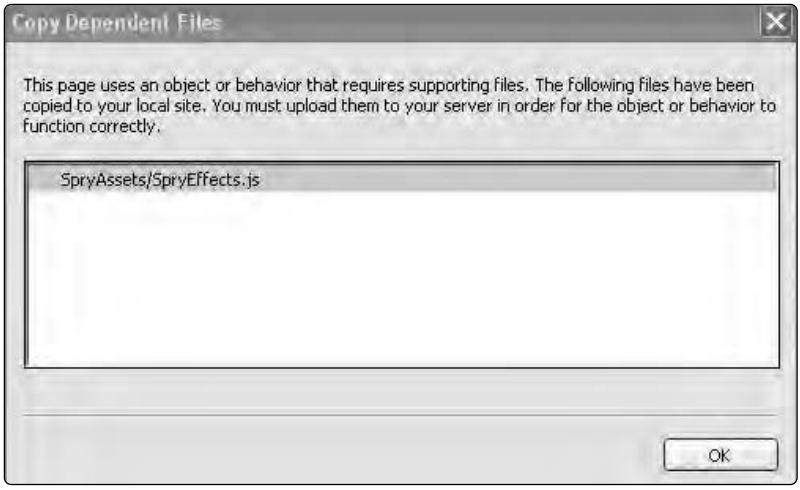
Spry, Dreamweaver için geliştirilen bir JavaScript kütüphanesidir (*framework*). Ajax teknolojisini Dreamweaver içinde kullanmamızı sağlar. Bu sayede sayfalarımıza birçok işlevsel özelliği birkaç fare tıklaması ile eklemiş oluruz. Javascript ve Ajax teknolojilerine ayrı bir bölüm olarak ilerleyen sayfalarda değineceğiz. Biz şimdilik işin kod kısmından çok işlevlerine odaklanalım.

EFEKTLER

TAG Inspector panelindeki Behaviors sekmesini tıklayalım. Sekmedeki **+** (artı) tuşuna bastığımızda açılacak olan menüden **Effects**'i seçelim. Burada ne var öncelikle onu söyleyeyim. Buradan, yukarda bahsettiğim Spry Framework sayesinde uygulamalarımıza bazı güzel özellikler kazandırabiliriz. Biz, bu hazır kod parçalarını Dreamweaver yardımıyla sayfamıza eklemeyi göreceğiz.

Javascript bilmeden de Dreamweaver yardımıyla uygulamalarımızda bazı zenginleştirmeler yapabiliriz. Ama tabi gelişmiş uygulamalar oluşturmak ya da bunları istediğimiz gibi düzenleyebilmek için Javascript öğrenmemiz gerekecek.

Hemen başlayalım ve Dreamweaver ile yeni bir html sayfa açalım. İlk yapmamız gereken bir **div** oluşturmak. Çünkü efekti bu div etiketine uygulayacağız. Etiketimizin içinde içeriğimizi oluşturalım. Mesela "Shake efekti için tıklayın" şeklinde bir metin oluşturup biraz şekillendirelim. Şimdi yukarıda anlattığım şekilde menüyü açıp **Shake**'i seçin ve belgeyi kaydedin.



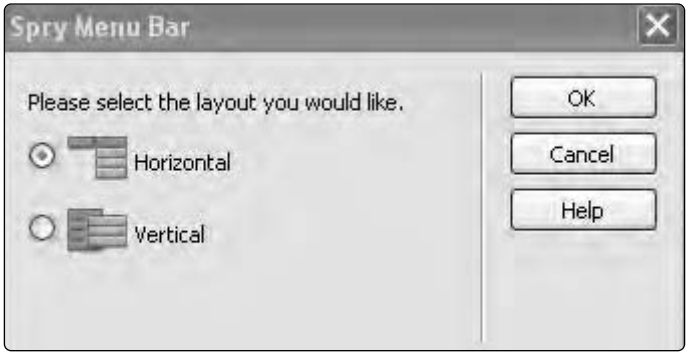
Fakat o da ne. Bir uyarı penceresi ile karşılaştık. Evet bu pencerede nesnemizin çalışabilmesi için pencerede belirtilen dosyaya ihtiyacımız ol-

duğu belirtiyor. Yani biz işlemi gerçekleştirince yani shake efektini uygulayınca Dreamweaver gereken kodları oluşturup bir dosyaya yazar ve head etiketleri arasında da script etiketlerini açıp dosyayı belgemize eklemiş olur.



Şimdi tarayıcıdan dosyayı çağırıp metne tıkladığınızda, metnin kıpraştığını göreceksiniz. **Effects** menüsündeki diğer efektleri de aynı yöntemle kullanabilirsiniz.

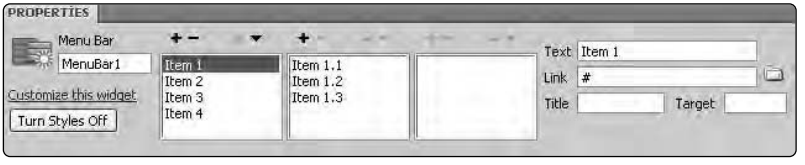
MENÜLER



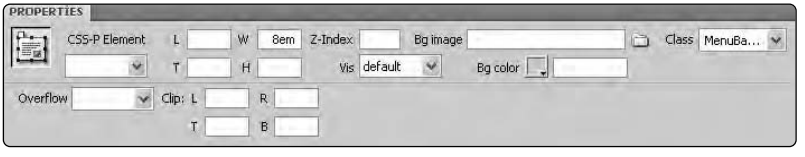
Şimdi de spry kullanarak menü yapmasını görelim. Hemen **Insert** panelindeki **Spry** sekmesini, ardında da **Spry Menu Bar** tuşunu tıklayalım. Resimdeki, menünün yatay mı, dikey mi olacağını seçmemizi isteyen pencere ile karşılaşacağız. Buradan istediğiniz birini seçip devam edin.



Pencerede **OK** tuşunu tıkladığınızda menünün ham hali karşınıza gelecek. Burada da yine dosyayı kaydederken gerekli JavaScript dosyalarını gösteren bilgi penceresi ile karşılaşacağız. Ve aynı şekilde Dreamweaver, gereken dosyaları oluşturup kaydedecektir.



Şimdi yapmanız gereken **Properties** penceresinden menünüzü istediğiniz gibi düzenlemek. Menünün üzerine gelip, sol üstte bulunan ve menünün adını gösteren mavi kısma tıkladığınızda, **Properties** penceresinde menünün genelini ilgilendiren ayarları göreceksiniz.



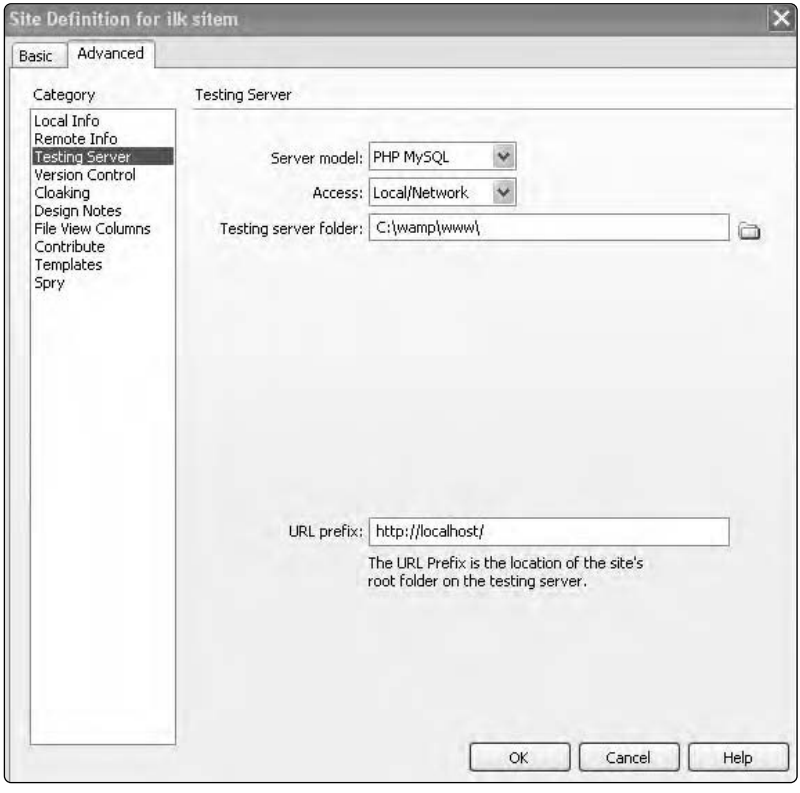
Menüdeki her hangi bir öğenin (örn: `Item 1`) çerçevesini tıkladığınızda ise **Properties** penceresi değişecek ve öğeyi ilgilendiren ayarları buradan yapabileceksiniz. Artık menünüzün öğelerinin isimlerini girmeye başlayabilir, menünün rengini, konumunu, alt menüsü olup olmayacağını ve diğer ayarlamaları yapabilirsiniz. Size kolay gelsin.

Aynı yöntemleri kullanarak ve yine **Insert** panelindeki **Spry** sekmesinin yardımıyla tab menü ve akordion menü de yapabilirsiniz. Tab menü için **Spry Tabbed Panels** tuşunu, akordion menü içinse **Spry Accordion** tuşunu kullanın.

DREAMWEAVER İLE ZİYARETÇİ DEFTERİ

Şimdi de Dreamweaver'da biraz PHP çalışalım. İsterseniz bu konuyu, PHP bölümünü çalıştıktan sonra dönmek üzere atlayabilirsiniz.

TEST SUNUCUSU OLUŞTURMA



Dreamweaver ile PHP çalışabilmek için öncelikle Dreamweaver'a yerel sunucumuzu tanıtmak durumundayız. Çünkü Dreamweaver ile bir PHP sayfa oluşturup tarayıcıda görmek için her zaman yaptığımız gibi **F12** tuşuna bastığımızda, bir test sunucusuna ihtiyacımız olduğu ve oluşturmak isteyip istemediğimizi soran bir uyarı penceresi ile karşılaşırız. **OK** ile devam ettiğimizde resimde gördüğünüz pencere açılacaktır.

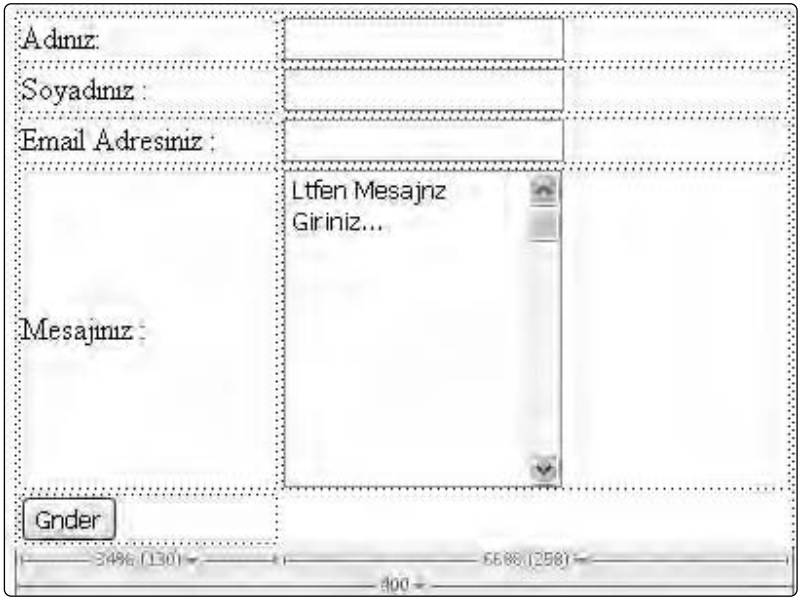
Buradan **Server Model** olarak **PHP**, **MySQL**, **Access** olarak da **Local>Network'**ü seçelim. Böylece program olarak PHP MySQL ile çalışacağımızı ve localde yani kendi bilgisayarımızda çalışacağımızı belirtmiş olduk.

Ardından test sunucumuzun konumunu (*C:\wamp\www*) **Testing Server Folder** kısmından gösteriyoruz. Site URL'si olarak da *http://localhost/* girip **OK**'lediğimizde test sunucumuz hazır olmuş olacak.



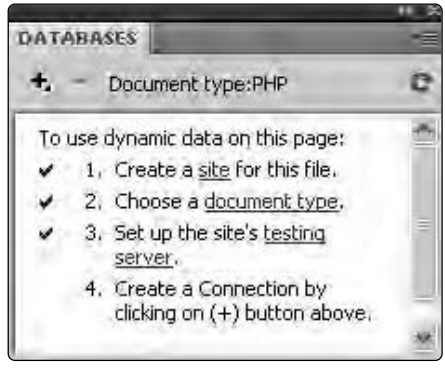
Artık **Files** panelindeki kutucuktan sunucunuzu seçip içindeki dosyalarla ya da yeni bir dosya oluşturarak çalışmalarınıza başlayabilirsiniz. Tabi MySQL bağlantısına ihtiyacınız olmadığı sürece. Bir ziyaretçi defteri yapıyorsunuz ve haliyle hazırladığınız forma site ziyaretçileri görüşlerini girip gönderdiklerinde, girilen verilerin veritabanında depolanması gerekir. Bunun için Dreamweaver da MySQL bağlantısı yapmamız gerekmekte.

MySQL Bağlantısı

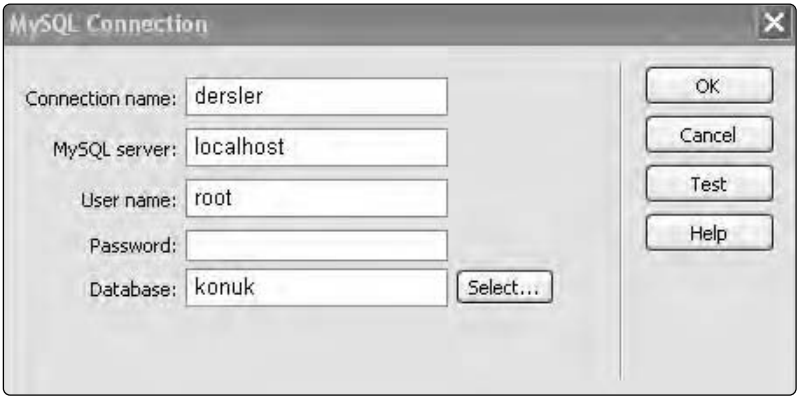


Bu işlem için önce ziyaretçi defterimizi oluşturmak üzere yeni bir PHP dosyası açalım. Ve "Adınız", "Soyadınız", "E-mail Adresiniz" ve "Mesajınız" alanlarından oluşan bir form hazırlayalım. Formu sunucumuzun kök dizininde açtığımız bir klasöre koyacağız.

Bundan sonrası için bir veritabanı oluşturmamız ve veritabanımızda gerekli tabloları hazırlamamız gerekiyor. Bu konuda PHP bölümünde yeterli bilgi mevcuttur. Biz bu aşamada bir veritabanı ve veritabanında gerekli olan tabloyu oluşturduğunuzu varsayalım. (PHP bölümünde hazırladığımız ziyaretçi defteri örneğini DVD'de PHP uygulamalar klasöründe bulabilirsiniz.)



Formumuzu hazırladık. Ardından veritabanımızı ve veritabanımızda tablomuzu oluşturduk. Şimdi **Databases** panelindeki **+** (artı) tuşuna basıp **MySQL Connection** öğesini tıklayalım. Karşımıza bağlantı penceresi gelecektir.



Şimdi gerekli bilgileri doldurmaya başlayalım.

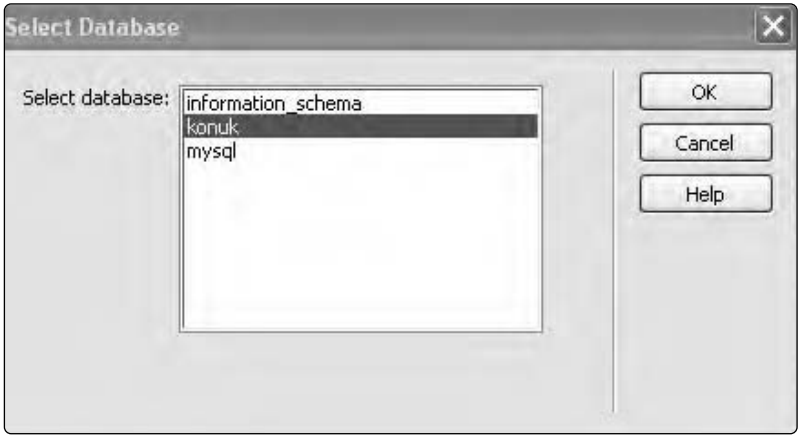
Connection Name: Bu kutucuğa bir bağlantı adı giriyoruz. Buraya herhangi bir isim girebilirsiniz.

MySQL Server: Biz kendi bilgisayarımızda çalıştığımız için buraya `localhost` ibaresini giriyoruz.

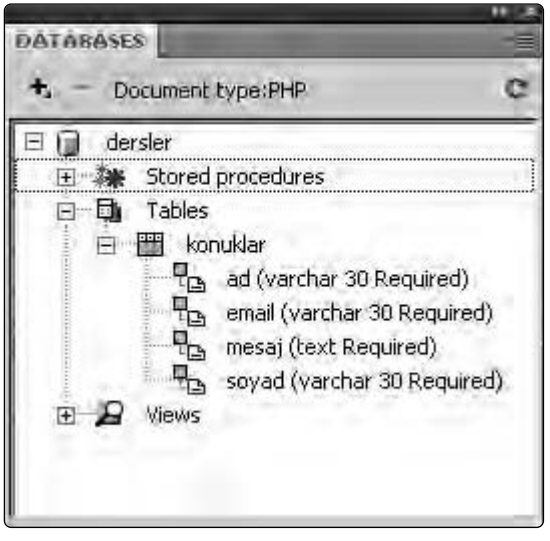
UserName: Bu bölüme de bildiğiniz gibi `root` giriyoruz...

Password: Şifre bölümünü boş geçebiliriz.

Database: Burada **Select** tuşunu tıklıyoruz. Sunucumuzda bulunan veritabanlarını listeleyen bir pencere açılacaktır.



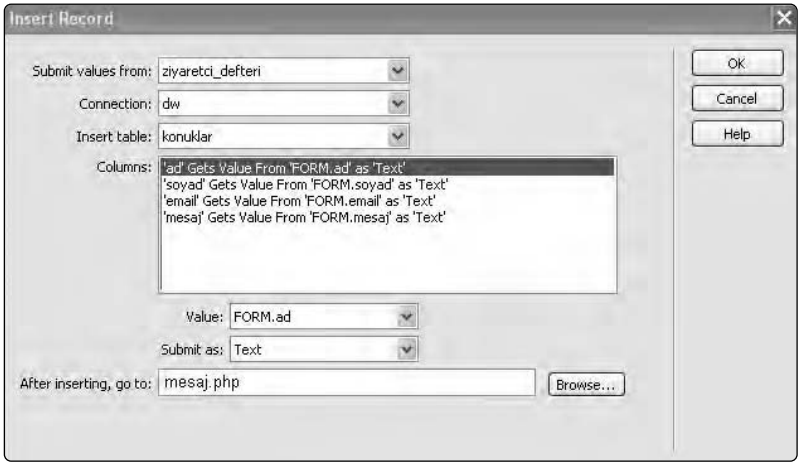
Pencereden veritabanımızı seçip **OK** diyoruz. Artık işlemi sonlandırabiliriz. Ama önce **Test** tuşu ile bağlantımızı test edelim. Şifreyi boş geçtiğimiz için bir uyarı penceresi açılacaktır. Bunu da **OK** ile geçelim ve tekrar **Test** tuşuna basalım. Bu sefer bağlantının başarılı olduğunu bildiren bir bilgilendirme penceresi gelecek. Bunu da **OK** diyelim. Ardından **connection** penceresi için de son bir kez **OK** tuşuna basıp işlemi tamamlayalım.



Şimdi tekrar **Databases** paneline bakarsanız veritabanı tablomuzu ve içindeki alanları görebileceksiniz. Evet veritabanımızı Dreamweaver'ın içinden görmek ve kurcalamak harika bir duygu.

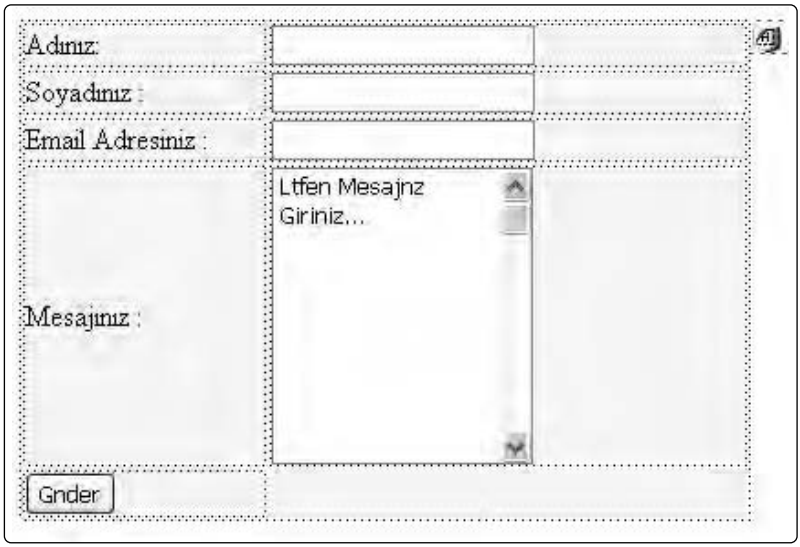
FORMLAR İLE VERİTABANINA VERİ GİRİŞİ

Şimdi **Behaviors** panelini açalım ve **+** (artı) uşuna basıp açılan menüden **Insert Record**'ı seçelim. Açılan pencereden gerekli ayarlamaları birlikte yapalım. Tabi bu işlemi yapabilmek için yerel sunucumuzun başlatılmış olması gerekir. Biz **PHP** bölümünde wampserver kullandık.

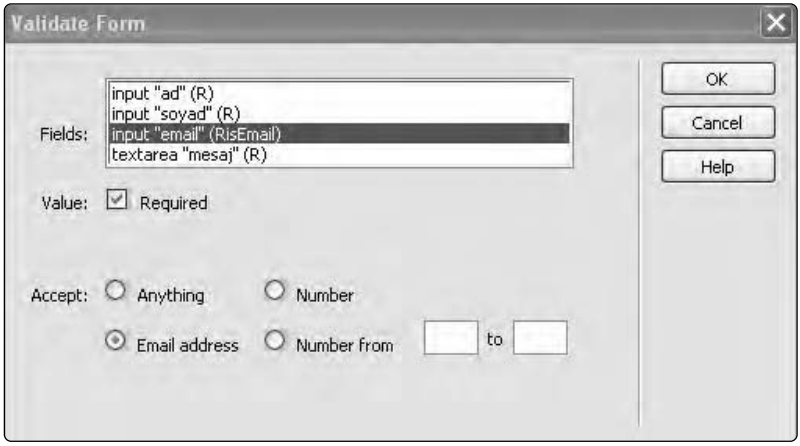


- **Submit values from:** Buradan veri girişi yapılacak formu seçiyoruz. Bizim bir tek formumuz olduğundan Dreamweaver açık ve seçili olan formumuzun name değerini burada gösterecektir. Birden çok form olsaydı bu açılır kutucuktan seçebilecektik.
- **Connection:** Burada mysql bağlantısı başlığında anlattığım şekilde oluşturduğunuz bağlantıyı seçiyoruz.
- **Inser Table:** Bu açılır kutucuktan veritabanındaki tablomuzu seçiyoruz.
- **Columns:** Hazırladığımız veritabanı tablosunun alanları burada listeleniyor.
- **Value:** Seçtiğimiz alanı temsil eden değeri seçiyoruz.

- **Submit as:** Veri türünü (sayı (integer), metin (text) ,tarih (date)...vd) seçiyoruz.
- **After inserting go to:** Ziyaretçi defteri doldurulup gönderildikten sonra kişinin yönlendirileceği sayfayı seçiyoruz. Mesala **mesaj.php** diye bir sayfa hazırlayıp kişiyi bu sayfaya yönlendirebiliriz.



Resimde gördüğünüz gibi **Insert Record** penceresindeki işlemleri tamamlayıp **OK** ile devam ettiğimizde hazırladığımız formun mavi renk alacaktır. Bunun anlamı, artık formumuzun dinamik veri girişi yapılacak bir form olduğudur. Sayfamızın kod kısmına bakarsanız Dreamweaver'ın gerekli PHP kodlarını sayfaya eklediğini göreceksiniz.



Ziyaretçi defteri formumuz hazır ancak biz formumuzun boş olarak gönderilmesini engellemek istiyoruz. Peki bunu nasıl yapacağız. Bu aşamada imdadımıza **Tag Inspector** panelindeki **Behaviors** sekmesi yetişiyor. Buradaki **+** (artı) tuşuna bastığınızda açılacak olan menüden **Validate Form** menü öğesini seçin.

Açılan pencerede **Fields** alanından doldurulması zorunlu alanları belirliyoruz. Bunun için istediğimiz bir alanı seçip **Value** değerini *Required* yapmak için kutucuğu tıklıyoruz. Seçtiğimiz alanın yanında bir **R** harfi belirecektir. Bu işlemi tek tek doldurulması zorunlu olacak tüm alanlar için yapıyoruz.

Ayrıca e-mail alanı seçili iken alttaki **Accept** kısmından **Email Address** radyo düğmesini tıklıyoruz. Bunun yapmaktan maksadımız, email alanına girilecek verinin gerçekten bir mail adresi olup olmadığının kontrolünü gerçekleştirebilmek. **OK** tuşu ile işlemi tamamladığınızda Dreamweaver sayfaya gereken JavaScript kodlarını ekleyecektir. Bu işlem için dilerseniz **Inset** panelindeki **Forms sekmesinden** de yararlanabilirisiniz.

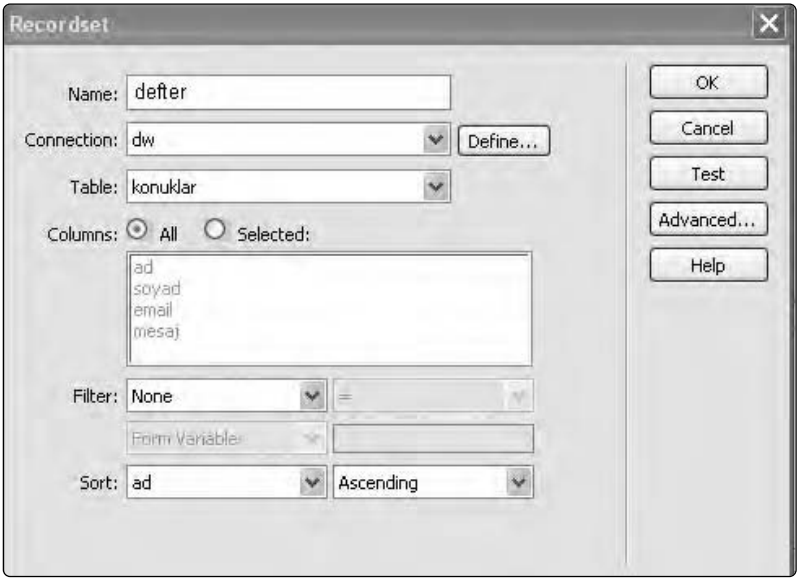
Ziyaretçi formunu artık deneyebilirsiniz. Bir deneme gönderimi yapıp ardından veri tabanınızı kontrol ederseniz girdiğiniz verilerin tabloya kaydedildiğini göreceksiniz. Bir kez de alanlardan birini boş bırakarak deneyin. Bu sefer bu alan boş olamaz uyarısı alacaksınız. Yani **validate form** uygulamamız tas tamam gerçekleşmiş, bunu kontrol etmiş olduk.

Veritabanındaki Bilgileri Yazdırma

Şimdi de formumuz aracılığıyla girilen verileri veritabanımızdan çekip yazdırmayı öğreneceğiz. İlk iş olarak yeni bir PHP belgesi açalım. Ardından bir isim belirleyip yerel sunucumuza (www klasörü içine) kaydedelim.

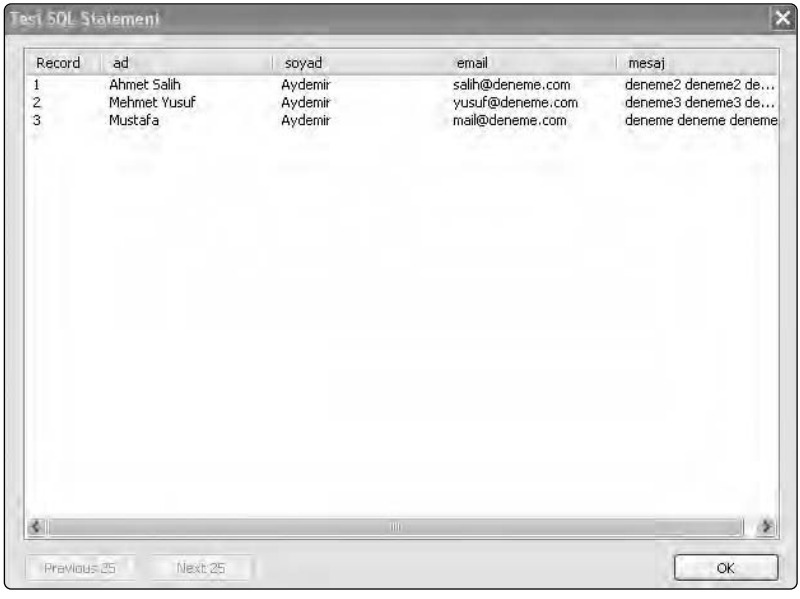
Şimdi belgemizde alt alta iki tablo oluşturalım. İlk tablomuz iki sütun ve dört satırdan, ikinci tablomuz ise tek sütun ve tek satırdan oluşacak. İlk tablomuzun ilk sütununa alt alta Adı, Soyadı, E-mail ve Mesaj ibarelerini girelim. İkinci sütuna ise veritabanından gelen veriler yazdırılacak. Şimdi artık başlamaya hazırız.

Veritabanından gelen verileri almak için bir **kayıt dizisi** (*Recordset*) oluşturmamız gerekiyor. Bunun için **Bindings** panelinden **+** tuşuna basıldığında açılan menüden **Recordset** (*Query*) seçeneğine tıklayarak **Recordset** penceresini açıyoruz.



Pencerede yapacağımız ayarlamalar ile kayıt dizimizi oluşturmuş olacağız. Hemen penceredeki detayları incelemeye başlayalım.

- **Name:** Kayıt dizimiz için bir isim belirliyoruz. Bu alanı şimdilik aynen bırakabilirsiniz. Fakat bir çok kayıt dizisi oluşturmaya başladığınızdan bunları birbirinden ayırt edebilmek için her birine açıklayıcı bir isim vermeniz en mantıklısı olacaktır.
- **Connection:** Daha önce oluşturduğumuz veritabanı bağlantısını bu alandan seçiyoruz. Dilersek yeni bir bağlantı oluşturmak için **Define**... tuşunu kullanarak, açılan pencereden yeni bir bağlantı tanımlayabiliriz.
- **Table:** Sorgulayacağımız ve içindeki verileri alacağımız veritabanı tablosunu da bu alandan seçiyoruz.
- **Columns:** Seçtiğimiz tablo içinde yer alan veri alanlarını da burada liste halinde görebilirsiniz. Eğer **All** radyo düğmesini tercih ederseniz tüm alanlar seçili hale gelecektir. Fakat tüm veri alanlarında değil de seçeceğiniz alanlarda sorgu gerçekleştirmek için **Selected** radyo düğmesini tercih etmelisiniz.
- **Fitler:** Sorgulayacağınız ve biraz evvel **Table** alanından seçmiş olduğumuz tabloyu, sorgu sonunda olduğu gibi değil de belli kriterlere göre süzerek sıralamak isterseniz bu alanı kullanmalısınız. Mesela sadece ismi **M** ile başlayan kişilere ait verileri bu alanı kullanarak rahatlıkla süzebilir ve sıralayabilirsiniz.
- **Sort:** Sıralayacağımız kayıtları nasıl listeleyeceğimizi buradan seçiyoruz. Mesela sıralamayı ad'a göre yapalım. Hemen yandaki açılır kutucuktan da **Ascending** seçeneğini tercih ettiğimizde sıralama A'dan Z'ye doğru olacaktır.



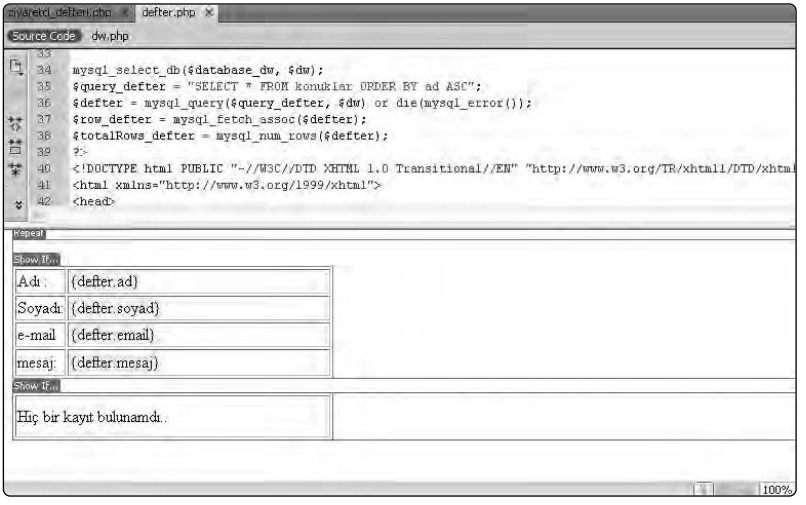
Pencerede yapmamız gereken işlemler bu kadar ancak pencereyi **OK** ile kapatmadan önce isterseniz sonucu bir test edelim. **Test** tuşuna tıkladığımızda açılan pencerede veritabanı tablomuzdaki kayıtları göreceksiniz. Biz süzme yapmadığımız için tablodaki tüm kayıtların listesini aldık. Süzme yapmadık ama sıralama yaptık.

Verilerin ad'a göre alfabetik sırada olduğu dikkatinizi çekmiştir sanırım. Her şey yolunda olduğuna göre **OK** ile pencereyi kapatabiliriz. **OK** tuşuna bir kez de **Recordset** penceresi için tıklayıp işlemi başarıyla sonlandırıyoruz.



Bindings paneline şimdi tekrar bir göz atalım. Panelde hazırladığımız kayıt dizisini ve içindeki alanları artık görebileceğiz. Bundan sonrası için işimiz epey kolaylaşmış oldu.

Artık yapmamız gereken, **Bindings** panelindeki kayıt alanlarını hazırladığımız birinci tablodaki yelerine fare ile sürükleyip bırakmak.



Mesela panelden ad alanını seçip tablomuzdaki "Adı:" satırının yanındaki hücreye sürükleyip bırakalım. Diğer alanlar için de aynı işlemi tekrar edeceğiz. Resimde gördüğünüz gibi bu işlemi yaptığımızda alanlarımız tabloda `{defter.ad}, {defter.soyad}` ...vd şeklinde görünecektir. Burada defter ibaresi kayıt dizimize verdiğimiz isimdir. Diğer kısım da tahmin edeceğiniz gibi alanımızın adıdır.

Peki biz belgemizde tek satır ve tek sütundan oluşan ikinci tabloyu neden yaptık? Bu tabloyu veritabanında herhangi bir kayıt olmadığı durumlarda bir hata mesajıyla karşılaşmamak için kullanacağız. Şimdi yapacağımız işlem, eğer veritabanı tablomuzda herhangi bir kayıt yoksa çıktı olarak ekrana "Hiç bir kayıt bulunamadı" şeklinde bir bilgilendirme mesajı yazdırmak içindir.

İkinci tablomuzu seçtikten sonra, **Server Behaviors** panelinden **+** tuşunu tıkladığımızda açılan menüden **Show Region/Show If Recordset Is Empty** menü öğesini tıklıyoruz. Böylece metnimizin sadece kayıt dizimiz boş iken görüntülenmesini sağlamış olduk.

Şimdi de birinci tablomuzu seçip, yine **Server Behaviors** panelinden bu sefer **Show Region/Show If Recordset Is Not Empty** menü öğesini seçelim. Bu işlemle de tablomuzu sadece kayıt dizimiz boş değilken göstermiş olduk. Kayıt dizisi boşken bu ilk tablomuz ekranda görüntülenmeyecektir.



Evet resimdeki gibi tablolar içinde verilerimiz ekrana geldi. Ve tüm bu işlemleri tek satır PHP kodu yazmadan yaptık. Tabi bu işlem için mutlaka tablo kullanmanızın gerekmediğini mesela bir div kullanarak da yapabileceğinizi belirteyim. Ben konunun anlaşılmasını kolaylaştırmak adına tablolardan yararlandım.

Dreamweaver ile Üyelik Sistemi

Dreamweaver ile bir üyelik sisteminin nasıl yapılacağını anlatmaya başlamadan önce bize gerekli olan veri tabanı tablomuzu oluşturmamız gerekmektedir. Bunun için yine phpMyAdmin programından yaralanacağız. Bu program ile tablo oluşturma işlemini nasıl yapacağımıza ilişkin

bilgi için php bölümü okumanız gerekiyor. Bu konuda gereken bilgiyi edindiğinizi varsayıyorum. Aşağıdaki sql, bu örnek için kullanacağımızı tabloyu (uyeler) ve alanlarını göstermektedir.

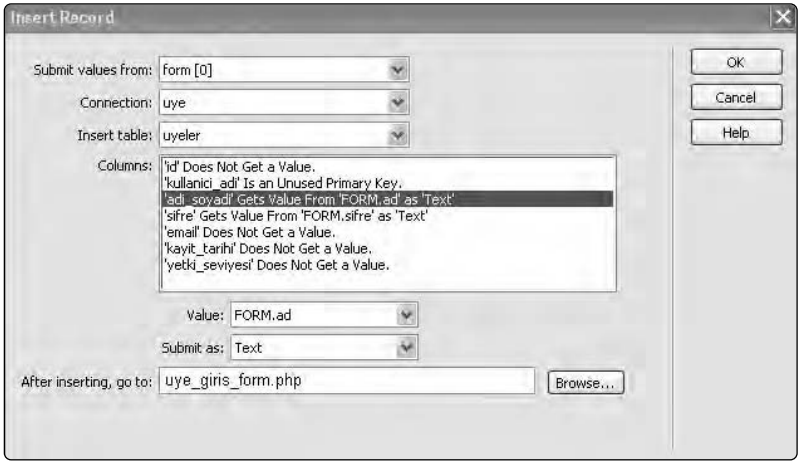
```
CREATE TABLE IF NOT EXISTS `uyeler` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `kullanici_adi` varchar(25) COLLATE utf8_turkish_ci NOT NULL,
  `adi_soyadi` varchar(60) COLLATE utf8_turkish_ci NOT NULL,
  `sifre` varchar(30) COLLATE utf8_turkish_ci NOT NULL,
  `email` varchar(30) COLLATE utf8_turkish_ci NOT NULL,
  `kayit_tarihi` datetime NOT NULL,
  `yetki_seviyesi` varchar(10) COLLATE utf8_turkish_ci NOT
NULL DEFAULT 'user',
  UNIQUE KEY `kullanici_adi` (`kullanici_adi`),
  UNIQUE KEY `email` (`email`),
  KEY `id` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_turkish_ci
AUTO_INCREMENT=1 ;
```

Üyelik Formu

Veritabanımızda gerekli tabloyu ve alanları oluşturduğumuza göre sitemize üye olmak isteyen ziyaretçilerin üye olmak için dolduracakları üyelik formunu hazırlamaya geçebiliriz. Formumuzun adı mesela **uye_form.php** olsun. Dreamweaver ile daha önce öğrendiğimiz gibi bir form hazırlayalım.

Formun metodunu post olarak seçelim. Farklı olarak Insert panelinden forma üye olacak kişinin üyelik tarihini gösteren bir gizli alan oluşturalım. Gizili alanımızın Value değerine `<?php echo date("Y-m-j H:i:s"); ?>` yazalım. Bu kod, üyenin kayıt tarihini almak için gerekli.

Ziyaretçi defteri için yaptığımız gibi **Databases** panelindeki artı + (artı) tuşuna basıp **MySQL Connection** öğesini tıklayalım. Açılan pencereden veritabanı bilgilerimizi girip yeni bir bağlantı oluşturalım.

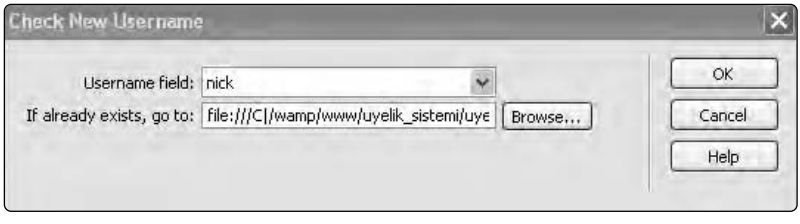


Ve yine ziyaretçi defteri için yaptığımız gibi **Behaviors** panelini açalım ve **+** tuşuna basıp açılan menüden **Insert Record**'ı seçelim. Bu pencere de sizin bildiğiniz bir pencere. Burada tek söyleyeceğim, **After inserting go to** kutucuğuna üyelik işlemi başarıyla tamamlandığında gidilecek olan sayfayı göstermeniz gerektiğidir.

Üyelik işlemi başarıyla tamamlandıysa kişinin siteye giriş yapacağı forma yönlenmesini sağlamak için bir dosya oluşturalım ve adına da mesela **uye_giris_form.php** diyelim. Bu dosyayı daha sonra düzenleyeceğiz.

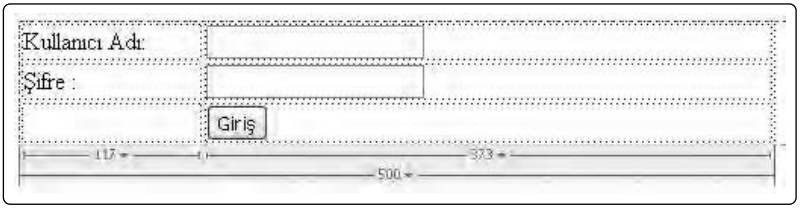
Üyelik Kontrolü

Sitemize aynı üye adı ile üye olunmamasını sağlamak zorundayız. Bu açıdan yapacağımız bir işlem var. **Server Behaviors** panelinden **User Authentication>Check New Username** seçeneği tam da bu iş için.

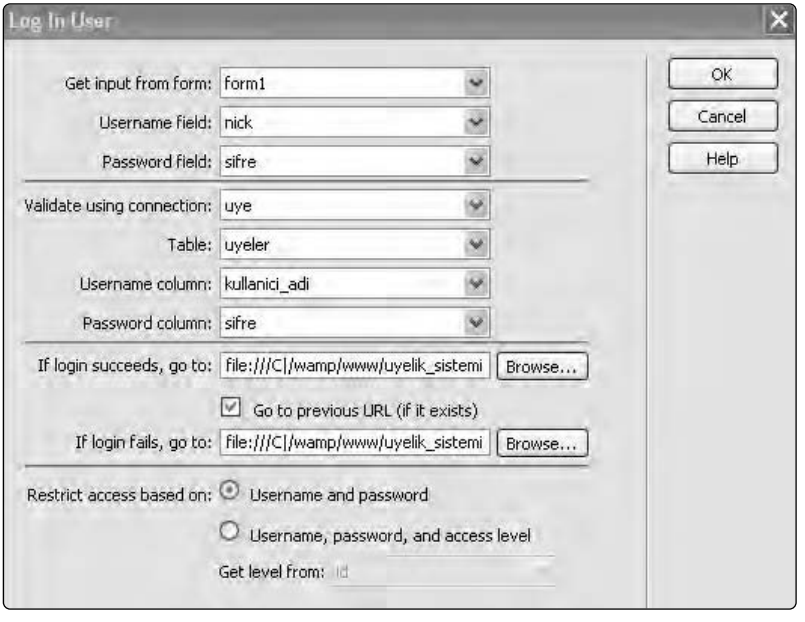


Açılan pencerede **Username field** alanından kullanıcı adı için girdiğimiz name değerini seçelim. **If already exist**, **go to** alanından ise mevcut bir kullanıcı adı ile kaydolmaya çalışan bir üye olduğunda söz konusu üyenin yönlendirileceği sayfayı belirliyoruz.

Üye Giriş (Login)



Şimdi sıra boş olarak oluşturduğumuz **uye_giris_form.php** dosyasına bir giriş formu hazırlamaya geldi. Dosyamızı açıp hemen bir tablo hazırlayalım. Tablomuzda `Kullanıcı adı` ve `Şifre` satırları olacak. Resimdeki gibi bir form konuyu anlamamız için yeterli olacaktır. Burada yine, şifre alanının `type` değerini `password` olarak ayarlamayı unutmuyoruz. Formun `action` değerini `uye_giris_form.php` yapıyoruz.



Formumuzu bu şekilde oluşturduktan sonra **Server Behaviors** panelinden **User Authentication>Log In User** menüsüne tıklıyoruz. Şimdi açılan pencereden yapmamız gereken ayarları inceleyelim.

- **Get input from form:** Oluşturduğumuz üye giriş formunun name değeri.
- **Username field:** Formdaki kullanıcı adı alanının name değeri.
- **Password field:** Formdaki şifre alanının name değeri.
- **Validate using connection:** En başta oluşturduğumuz veritabanı bağlantısı.
- **Table:** Veritabanındaki üye kayıtlarının tutulacağı tablo.
- **Username column:** Veritabanı tablosundaki kullanıcı adı alanı.
- **Password coulmn:** Veritabanı tablosundaki şifre alanı.
- **If login succeeds, go to:** Üye başarıyla giriş yaptığında hangi sayfaya yönlendirilecekse bu alana o sayfanın adını gireceğiz. Eğer kullanıcıyı giriş yaptığı sayfaya döndürmek istiyorsak, **Go to previous URL** (*if it exists)* kutucuğunun tıklı olması gerekir.
- **If login fails, go to:** Üye giriş işlemi başarısızlıkla sonuçlandığında üyenin gitmesini istediğimiz sayfanın adını buradan belirliyoruz.
- **Restrict access based on:** Üyelerinize farklı yetki seviyeleri vermek istiyorsanız (örn: üye, mod, yönetici... vb) **Username**, **password and Access level radyo** düğmesini seçmelisiniz. Böyle bir ayrım gerekmiyorsa Username and password düğmesini seçin.

Artık tek yapmamız gereken **OK** tuşunu tıklamak. Böylece giriş sayfası olayını halletmiş olduk. Sıra geldi çıkış işlemine.

Çıkış (Logout)

Bu işlem için yeni bir sayfa oluşturalım ve adına da mesela **cikis.php** diyelim. Yine **Server Behaviors** panelinden bu sefer **User Authentication>Log Out User** alt menüsünü tıklıyoruz.



Açılan pencerede **Log out when** kısmından **Link Clicked** radyo düğmesine tıklarsak, çıkış işlemi sayfadaki bir linke tıklanması ile olacaktır. Eğer **Page loads** radyo düğmesine tıklarsak, bu sefer de çıkış işlemi bulunulan sayfadaki çıkış linki ile yapılabilecektir. **When done**, **go to** kısmından ise çıkış yapan kullanıcının yönlendirileceği sayfayı belirleyeceğiz.

Evet artık tastamam bir üyelik sistemimiz var. Kısaca özetlersek; önce ziyaretçilerin üye olabilmeleri için gereken bir üye kayıt formu yaptık. Ardından üyelik işlemi başarılı olduğunda, üyenin siteye giriş işlemini yapacağı giriş formunu hazırladık. Ve sonra nasıl çıkış yapılacağı meselesine değindik. Üyelik sisteminizi güle güle kullanın.

6

PHP

PHP'nin Yapısı	268	Diziler	295
Temel Sözdizimi	269	Formlar	302
Değişkenler	270	Dosya ve Dizin İşlemleri	313
PHP'de Program Denetimi	279	Veritabanı İşlemleri	324
Fonksiyonlar	285	PHP ve XML	342

Bildiğiniz gibi günümüz web sitelerinde; ziyaretçi defteri, abonelik formları, üyelik sistemleri, arama özellikleri, yorum özelliği gibi pek çok özellikler bulunmaktadır. Bu özellikler, site ziyaretçilerinin sadece pasif bir biçimde kendilerine sunulan içeriği okumakla yetinmeyip, çeşitli şekillerde site içeriğine katkıda bulunmalarını ve birbirleriyle de iletişim halinde olmalarını sağlamaktadır. Tüm bunları bir HTML site ile yapamayız.

Bahsettiğimiz işlemleri gerçekleştirmek için bir web programlama diline ihtiyaç duyarız. Çünkü bahsettiğimiz tüm bu işlemler için ziyaretçilerin girdikleri verilerin bir yerde depolanması, işlenmesi ve istendiğinde tekrar kullanılabilmesi gerekecektir. HTML dilinin ise verileri yorumlamak için değil, verileri göstermek için kullanıldığını kitabımızın ilgili bölümünde anlamıştık.



Sanırım bir web programlama dili kullanmamız gerektiği konusunda anlaştık. Bu iş için kullanılabilecek bir çok web programlama dili (örn: PHP, ASP.NET, PERL, RUBY... vb) mevcuttur. Biz kitabımızda PHP web programlama dilini anlatacağız.

PHP, yukarıda bahsettiğimiz tüm işlemleri ve çok daha fazlasını yapabilen ve dünyada en yaygın olarak kullanılan web programlama dillerinden birisidir. Üstelik gerek PHP dili, gerekse PHP dili ile kullanılan bir çok üst düzey veritabanı sistemi tamamen ücretsizdir. Evet, dünya çapında bir dili herhangi bir lisans ücreti ödemeden kullanmanın keyfini birazdan birlikte yaşayacağız.

PHP ücretsiz olmasına karşın, çok hızlıdır ve üstün bir performansa sahiptir. Özellikle MySQL veritabanı sistemi ve Apache web sunucu yazılımıyla birlikte her türlü üst düzey projenin altından rahatlıkla kalkabilmektedir.

Ayrıca PHP'nin kolay öğrenilen bir dil olması da gerçekten büyük bir avantaj. Lisans ücreti olmayan, üstün bir performansa sahip ve kolay öğrenilebilir bir dil... İşte PHP...-

Bu aşamada PHP'nin geçmişi hakkında biraz bilgi verelim. Amacı internet üzerinde kişisel bir site yapmak olan Rasmus Lerdorf tarafından geliştirilmeye başlandı. Rasmus Lerdorf, PERL dilinden uyarladığı bu yazılıma **Personal Home Page** (*Kişisel Ana Sayfa*) adını verdi. Bir süre sonra bir çok geliştiricini katkılarıyla PERL dilinden farklı kendine has yapısı olan bir dil haline geldi. Günümüzde PHP 5. sürümüne ulaşmıştır ve Unix, Linux, Windows tabanlı tüm sistemlerde çalışabilmektedir.

PHP dilini öğrenmeye başlamadan önce yapmamız gereken bazı işlemler var. Öncelikle bilgisayarımıza bir **web sunucu** (*Apache*), bir **veritabanı sitemi** (MySQL) ve bir de haliyle **PHP** kurmamız gerekir. Yapacağımız çalışmalar için kendi bilgisayarımızda bir sunucu ortamı gereklidir. Aksi takdirde PHP kodlarımızı çalıştıramayız.

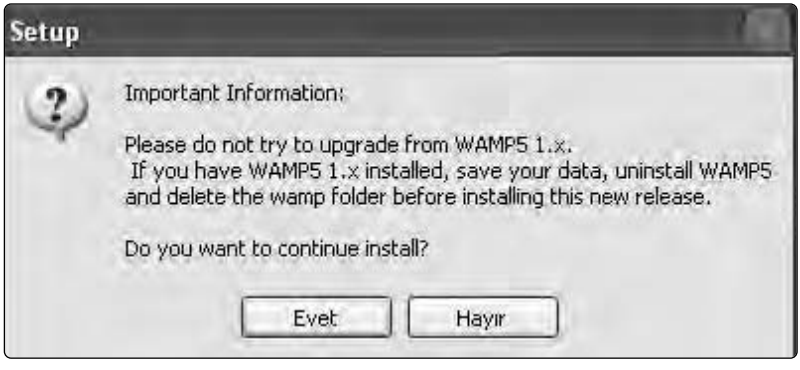
Sitemizi bitirip bir sunucu hizmeti alacağımız zaman, işin bu tarafı bizi ilgilendirmeyecektir. Çünkü alacağımız sunucuda tüm bu yazılımlar kurulu olacağından, biz sitemizin dosyalarını sunucuya yükleyip kullana-

bileceğiz. Sunucu yapılandırma, sunucu yazılımları kitabımızın içeriğine dahil olmadığından bu konulara değinmeyeceğiz. Bize lazım olan PHP kodlarımızı çalıştırabilecek bir ortamdır.

Yeri gelmişken bu yazılımları kendi sitelerinden tek tek indirip sırasıyla kurabileceğinizi belirteyim. Bunlardan **apache** web sunucusunu **www.apache.org** sitesinden, **PHP**'yi **www.php.net** sitesinden, **MySQL** veritabanı sistemini ise **http://dev.mysql.com/** adresinden indirebilirsiniz. Ayrıca veritabanı işlemlerinizi yaparken, sizi tek tek veritabanı komutları yazmaktan kurtaracak olan **phpMyAdmin** programını da **www.phpmyadmin.net** sitesinden indirebilirsiniz.

Fakat merak etmeyin, daha tek bir satır kod yazmadan sizi bu kadar programı kurmakla uğraştıracak değilim. Peki, o zaman biz çalışmalarımızı nasıl yapacağız. Bu sorunun cevabı çoklu paket kurulumudur. Yani tüm bu programların tek bir pakette toplamış halidir diyebilirim. Biz tek bir paket yükleyeceğiz ve tüm gereken programlar o pakette olacak. Bunun için de bir çok alternatif mevcut.

Bunlardan **Wamp Server www.wampserver.com/en/** adresinden, **easyphp**, **www.easyphp.org** adresinden, **xampp server** ise **www.apachefriends.org/en/xampp.html** adresinden temin edilebilir. Biz çalışmalarımızda **wamp server paketini** kullanacağız.

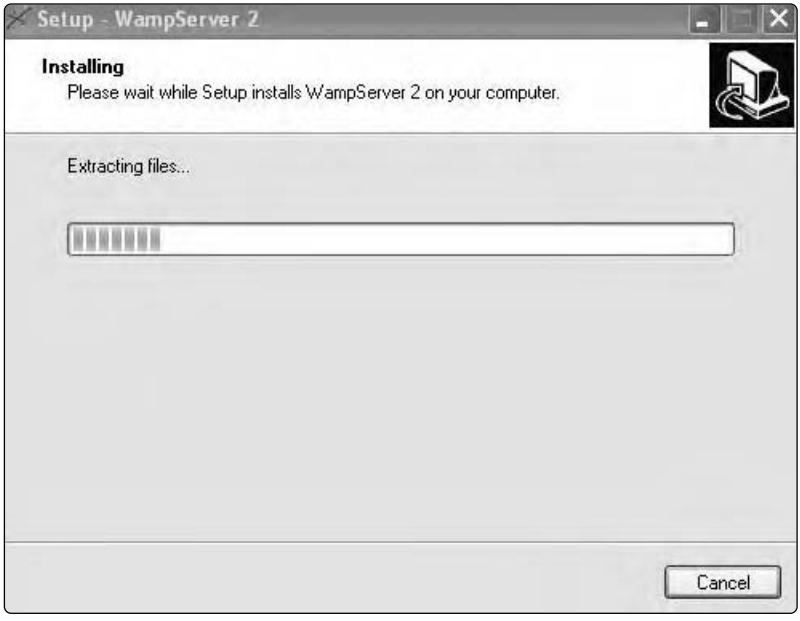


Şimdi birlikte paketimizi kuralım. İndirdiğimiz dosyaya tıkladığımızda ilk olarak karşımıza resimde gördüğünüz pencere çıkacak. Bu pencerede, **Wamp**'ın eski sürümlerini kullanıyorsak öncelikle bu eski versiyo-

nu kaldırıp, dosyasını da sildikten sonra yeni sürümün kurulumuna başlamamız gerektiği söylenmekte... Biz ilk kez kurduğumuz için, "Evet" deyip devam ediyoruz.

Daha sonrasında setup penceresi gelecek; bu aşamayı **Next** ile geçiyoruz. Bir sonraki pencereyi de lisans anlaşmasını kabul ettiğimizi gösteren radyo düğmesini işaretledikten sonra yine **Next** ile geçiyoruz. Sonraki aşamada wamp'ın kurulacağı dizini seçeceğimiz ekran gelecek. Dizini belirleyip (örn: *c:\wamp*) **Next** ile devam ediyoruz.

Gelen pencereden masaüstüne kısayol oluşturmamızı sağlayan kutucuğu tıklayalım ve yolumuza devam edelim ve bir hamle daha yapıp **Install** diyerek kurulumu başlatalım.



Kurulum tamamlandığında gelen pencerede **SMTP** kutucuğunda `localhost` ibaresini, Email kutucuğuna da bir örnek e-posta adresi göreceksiniz. Bunları aynen bırakıp devam ettiğimizde, kurulumun bittiğini gösteren pencere ile karşılaşacağız. Bunu da tıklayınca, sunucumuz artık çalışmaya başlayacaktır. Sunucunun çalıştığını, bilgisayarımızın

başlat çubuğundaki simgeden de görebiliriz. Hemen başlat çubuğundaki sunucu simgesini **SAĞ** tıklayalım ve dil olarak Türkçe'yi seçelim.



Artık sunucumuz emrimize amade. Sunucumuzun yönetim penceresini açmak için **Başlat** çubuğundaki simgeye tıkladığımızda pencere açılacaktır. Pencereden **Localhost** menüsünü tıkladığımızda tarayıcı penceresinde sunucumuzun ana sayfası açılacaktır.



Açılan sayfada sunucumuzun özellikleri ile apache, PHP, MySQL yazılımlarının sürümlerini görmektesiniz. Bu sayfaya, doğrudan tarayıcınızın adres çubuğuna **http://localhost/** yazarak da ulaşabilirsiniz. İşte burası bizim sitemizin ana sayfası. Hazırlayacağımız tüm PHP dosyalarımızı bu adresle çağıracağız. PHP dosyalarımızı da *c:\wamp\www* kla-

sörünü içine atacağız. Tabi önce oluşturmamız lazım değil mi? Şimdi PHP ile ilgli genel bilgiler vererek PHP'ye başlamanın tam sırası...

PHP'NİN YAPISI

PHP NASIL ÇALIŞIR?

Bildiğiniz gibi web sayfaları HTML dili ile tarayıcılarda görüntülenir. Bir web sitesini tıkladığınızda tıkladığınız sayfa sizin bilgisayarınıza indirilir ve tarayıcınız tarafından yorumlanarak ekranda gösterilir. Bu sebeple HTML için, kullanıcı taraflı dil denilir.

PHP ise sunucu taraflı bir dildir. Yani sayfanızda yer alan PHP kodları sunucudaki PHP yorumlayıcısı tarafından işlenir. Olduğu gibi istemciye gönderilmez. Sonuçlar istemciye HTML olarak gönderilir.

Şimdi ilk belgemizi yazalım. **ilk_belge.php** adında bir dosya oluşturun ve sunucunuzun kök dizinine (*www*) atın. Ardından belgeyi açın ve aşağıdaki kodları ekleyin.

```
<html>
 <head>
  <title>İlk Deneme</title>
 </head>
 <body>
 <?php echo '<p>Merhaba Dunya</p>'; ?>
 </body>
</html>
```

Burada ilk dikkat çeken şey, PHP kodlarını `html` etiketleri arasına gömmüş olmamızdır. Bunu yapmasanız da bu PHP kodları çalışacaktır. Çünkü PHP, `açılış <?php` ve `?>` kapanış etiketleri dışındaki şeyleri göz ardı edecektir. Fakat yapacağımız birçok detaylı çalışmada html etiketlerini de kullanacağımız için PHP kodlarını html etiketlerinin arasına yazmayı alışkanlık haline getirin derim.

Bir sonraki aşama olarak tarayıcınızın adres satırına **http://localhost/ilk_belge.php**, yazıp sayfanızı çalıştırın. Ekrana, `Merhaba Dünya` yazısı gelecektir. Tebrikler, bu arada ilk PHP sayfanızı da yapmış oldunuz!

Sunucu, dosyanın uzantısının *.php* olmasından bunun bir PHP dosyası olduğunu anlar ve dosyayı PHP'ye havale eder. Böylece dosyamız işlenir ve tarayıcınıza aşağıdaki html çıktısı üretilerek gönderilir.

```
<html>
 <head>
 <title> İlk Deneme </title>
 </head>
 <body>
 <p>Merhaba Dunya</p>
 </body>
</html>
```

Evet yanlış okumadınız. Veriler tarayıcıya HTML çıktısı olarak gönderilir. Konunun başında bahsettiğimiz sunucu taraflı olma durumu işte budur. Yani tarayıcılarından sitenizi çağıranlar, sizin yazdığınız PHP kodlarını göremeyecekler, sadece HTML olarak görecekler. Tabi sayfada istemci taraflı bir başka dil daha kullanılmışsa (örn: *JavaScript*) onu da görebilirler, ama sunucu taraflı dilleri göremezler.

Temel Sözdizimi

PHP kodları **<?php ?>** etiketleri arasına yazılır. Bunlara açılış ve kapanış etiketleri diyoruz. Bu etiketleri `<script language="php"> </script>` şeklinde de yazabiliriz. Fakat bu, bazı düzenleyicilerde sorun çıkarabilmektedir. İki yazım şekli daha söz konusudur. Bunlar; `<? ?>` ve `<% %>` şeklindedir. Son iki kullanım ancak **php.ini** yapılandırma dosyasından açılarak kullanılabilirler. Fakat bir çok sunucuda bu kullanımlar açık değildir ve bu da size sorun çıkartacaktır. Dolayısıyla biz, teamüllere uygun olarak ilk kullanım şekli ile (`<?php ?>`) çalışacağız.

PHP'de her bir müstakil komut kümesini deyim olarak adlandırıyoruz. Deyimler birbirlerinden ; (noktalı virgülle) ayrılırlar. (örn: `echo 'Merhaba Dünya';`)

Kodladığınız bir belgede, özellikle de belge uzunsa, belgenin daha sonra yapacağınız incelemelerde daha açıklayıcı olabilmesi için, açıklama satırları ekleyebilirsiniz. Bu açıklamalar sadece size değil belge üzerinde çalışacak başka programcılara da kolaylık sağlayacaktır. Tek satırlık

bir açıklama için cümlenin başına **//** ya da **#** işareti konur. Çok satırlı bir açıklama yazacaksak açıklamanın başına **/*** işaretini, sonuna da ***/** işaretini koyarız. Bu işaretler arasındaki içerik PHP yorumlayıcı tarafından dikkate alınmayacaktır.

```
<?php
    echo 'Merhaba Dünya'; // Bu tek satırlık açıklama örneğidir.
    /* Bu, ise çok satırlı
       bir açıklama örneğidir. */
    echo 'Merhaba Evren'; # Bu da bir başka tek satırlık
açıklama
?>
```

DEĞİŞKENLER

Değişkenleri bir konteynır yani bir kutu olarak düşünebiliriz. Evet içi boş bir kutu. İçine bir sürü değişik şey koyabileceğimiz bir kutu. İşte PHP'de buna değişken, içine koyduğumuz şeylere de değer diyoruz. Yani PHP'de bir sürü kutumuz olabilir ve kutulara da bir sürü değişik şey koyabiliriz. Gerçekten esnek bir dille karşı karşıyayız.

Değişkenler verileri tutarlar ve bellekte bir yer kaplarlar. Bir değişkenin içine belgenizin farklı yerlerinde değişik değerler atayabilirsiniz. Bu durumda yeni eklediğiniz değer geçerli olacaktır. İşte bu sebeple bu yapıya değişken adı verilmektedir.

Değişkenler başlarına **$** (dolar işareti) konularak tanımlanırlar. Türkçe karakterler hariç harflerden, rakamlardan ya da **_** işaretinden oluşturulabilirler. Rakamlar değişkenlerin içinde yer alabilirler fakat en başta bulunamazlar. Değişkenlerin içinde **_** işaretinden başka hiçbir işaret kullanılamaz. Değişkenlere bir değer atanırken değişkenden sonra **=** (eşittir) işareti kullanılır.

Değişkenlerin DOĞRU Kullanımı	Değişkenlerin HATALI Kullanımı
`$iki_1 = 12;`	`$1iki = 12;`
`$isim = "Mustafa";` `$eni = 2.24;`	`$işi = "Muhasebe";` `$veri%=1000;`

Veri Tipleri

PHP değişkenlere değer olarak girdiğiniz veriyi işleyebilmek için, verinin türünü bilmek zorundadır. Örneğin girdiğiniz veri bir metin mi yoksa mesela alfanümerik bir veri mi, bunun belirlenmesi gerekir. Haliyle bunu programa biz söyleyeceğiz. Böylece program, söz konusu veri ile yapmak istediğimiz işlemleri (örn: aritmetiksel işlemler) yapabilecektir. Şimdi PHP'de kullanılan veri türlerini inceleyelim.

Tam Sayı (İnteger) Veri Tipi: Pozitif ya da negatif tam sayılardır.

Örnek

```
<?php $bir=8; echo ($bir); ?>

<?php $iki=-51; echo ($iki); ?>

<?php $uc=547; echo ($uc); ?>
```

Bu işlemlerin sonucu sırasıyla `8`, `-51` ve `547` olarak tarayıcıda görüntülenecektir.

Noktalı (Double) Veri Tipi: Ondalık sayılar içindir ve sayının ondalık kısmı için . (nokta işareti) kullanılır.

Örnek

```
<?php $bir=8.4; echo ($bir); ?>

<?php $iki=-51.22; echo ($iki); ?>
```

Bu işlemlerin sonucu sırasıyla, `8.4` ve `-51.22` olacaktır.

Karakter Dizileri (String) Veri Tipi: Harfler, rakamlar ve işaretlerden oluşur. (örn: `ms251`, `01/01/2009` ...vb) çift tırnak (") ya da tek tırnak (') içinde kullanılabilirler. Fakat tırnak içinde başka bir değişken olacaksa mutlaka çift tırnak (") kullanılmalıdır.

```
<?
$isim="Mustafa";
$tam_isim="$isim Aydemir";
echo($tam_isim);
?>
```

Değişken Tipleri

Nesne (Objects) Değişkenler: Bir sınıfa (*class*) ait değişkenlerdir. Önce bir class sonra da bir nesne (*object*) değişkeni tanımlanır ve class'ın sahip olduğu fonksiyonları değişkene atanır.

```
<?php
class kitap {
function oku ( ) {
echo "Web Tasarım Rehberi";}
}
$bilgisayar = new kitap;
$bilgisayar->oku();
?>
```

Bu örnekle kitap sınıfının sahip olduğu fonksiyonu bilgisayar değişkenine aktardık.

Dizi Değişkenler: Bir çok değişkenin art arda kullanılmasıdır.

Kullanımı

```
$dizi[indis]=değer;
```

şeklindedir.

Örnek

```
<?php
$ay[0]= "Ocak";
$ay[1]= "Şubat";
$ay[2]= "Mart";
$ay[3]= "Nisan";
?>
```

Parantez içinde gördüğünüz **indis** değeri `1`'den değil `0`'dan başlamaktadır. Uygulamalarınızda buna dikkat etmelisiniz. Değişkenlerin fonksiyonlarla kullanımını sonraki sayfalarda göreceğiz.

Mantıksal (Boolean) Değişkenler: Öteki PHP değişkenlerini mantıksal değişkene çevirir. Ve sonuç olarak geriye, içi boş string değeri için ve `0` rakamı için, `0` yani `FALSE` (*yanlış*) ya da `null` (*boş*), diğer değerler için `1` yani `TRUE` (*doğru*), değerlerini döndürür.

Tip Değiştirmek

Değişkenlerin türünü değiştirmek istediğimizde ise `settype( )` fonksiyonunu kullanacağız.

```
<?php
$rakam=333;
settype($rakam,'boolean');
echo($rakam);
?>
```

Örneğimizi denediğinizde ekrana `333` değerinin değil `1` değerinin basıldığını göreceksiniz.

```
<?php
$rakam=3.33;
settype($rakam,'integer');
echo($rakam);
?>
```

Bu örneği denediğinizde ise ekrana, settype ile tamsayı tipi olan integer'i kullandığımız için `3.33` değerinin değil `3` değerini basacaktır.

Bazı durumlarda bir değişkenin değerini tanımsız yapmak gerekebilir. Bu tip durumlarda `Unset` fonksiyonu kullanılır.

Örnek

```
$a=1
$b=2
$c=3
unset ($a,$b,$c);
```

Bu örnekteki `$a`, `$b` ve `$c` değişkenlerine ait değerler yok olmuş oldu.

Değişken Değerlerinden Boşluk Kaldırma

Değişkenlerin değerleri boşlukla başlayıp, boşlukla bitebilir. Fakat bazen bu istenmeyen bir durumdur. Örneğin formlarda gereksiz boş gönderileri engellemek gerekecektir. Bu işlem için `trim()` fonksiyonunu kullanacağız.

Fonksiyon	İşlevi
`trim()`	Değişken değerlerindeki hem sağdaki hem de soldaki boşlukları siler.
`ltrim()`	Değişken değerlerindeki soldaki boşlukları siler.
`rtrim()`	Değişken değerlerindeki sağdaki boşlukları siler.

```
<?php
$degisken=" Deneme Metni ";
$degisken1 = trim($degisken);
$degisken2 = ltrim($degisken);
$degisken3 = rtrim($degisken);

echo $degisken1;
echo $degisken2;
echo $degisken3;
?>
```

Bu fonksiyonun kullanımı, boşlukları kaldırmanın ötesinde değişken değerlerinde bir farklılığa yol açmaz.

Süper Global Değişkenler

Süper Global	İşlevi
`$_POST`	`Post` metodu ile gönderilen form gönderilerinde gelen değişkenin değerini almak için kullanılır.
`$_GET`	`Get` metodu ile gönderilen form gönderilerinde gelen değişkenin değerini almak için kullanılır.
`$_COOKIE`	Çerez kullanarak oturum açma işlemi için kullanılır.
`$_SESSION`	Oturum açmak için kullanılır.
`$_REQUEST`	`$_POST`, `$_GET` ve `$_COOKIE` süper globallerinin değerini almak için kullanılır.
`$_FILES`	Dosya yüklemek için kullanılır.
`$_SERVER`	Ip adresi, server ve tarayıcı türü gibi çevre değişkenlerini tutar.
`$_ENV`	Temp dizini, dosya yolu gibi sistem değişkenlerini tutar.
`$_GLOBALS`	Değişkenin adı ile tanımlanır. Örnek: `$deger=1 $_GLOBALS["veri"]`.

SABİTLER

Değişkenlerle uğraşırken bir de başımıza sabitler çıktı diyebilirsiniz. Haklısınız. Ama bazı durumlarda bir değişkenin değişmemesini isteyebiliriz. Böylesi durumlarda *define ( )* fonksiyonunu kullanacağız. Mesela en basitinden bir matematiksel işlemde Pi sayısının `3.14` olarak sabit kalması gerekmez mi?

Diğer değişkenlerden farklı olarak sabitlere yeni bir değer atamaya kalktığımızda hata mesajıyla karşılaşırız.

Kullanımı

```
define ("sabitinismi",değeri);
```

şeklindedir. Aşağıdaki örneğin çalıştırdığınızda sonuç `300` olacaktır. Sabitlerin önlerinde değişkenlerde olduğu gibi **$** işareti bulunmaz.

```
<?php
define ("ebat",300);
echo(ebat);
?>
```

Şimdi de PHP'de kullanılan bazı sabitlerden bahsedelim.

__FILE__: Çalışan PHP dosyasının dosya yolu ile birlikte ismi

__LINE__: Bu sabitin bulunduğu satırın numarası

__DIR__: Dosyanın dahil olduğu dizin

__FUNCTION__: Fonksiyon ismi

__CLASS__: Sınıf ismi

__METHOD__: Metod ismi

__NAMESPACE__: Geçerli isim alanının ismi

Bir örnek yapacak olursak;

```
<?php
echo __FILE__ ;
?>
```

Örneği denerseniz; "C:\wamp\www\deneme.php" gibi dosya adını ve dosya tam yolunu gösteren bir sonuç alırsınız.

OPERATÖRLER

Operatörler için, değerleri nasıl kullanacağımızı ve ne gibi işlemler yapacağımızı belirlemeye yarayan birtakım işaretlerdir diyebiliriz. Şimdi operatörleri inceleyelim.

STRING OPERATÖR

Stringleri birlikte kullanmamızı sağlayan **(.)** nokta operatörüdür.

```
<?php
$ilk="Çanak"; //Çanak değeri verdiğimiz bir değişken oluşturduk.
$son="kale"; //kale değeri verdiğimiz bir değişken oluşturduk.
$il=$ilk.$son; /* il adında yeni bir değişken oluşturup
 (.) nokta operatörü ile birleştirdik*/
echo $il; // il değişkenini ekrana yazdırdık.
?>
```

Yaptığımız programı çalıştırınca ekrana **Çanakkale** değerini basacak.

ATAMA OPERATÖRLERİ

İlk olarak "=" (eşittir) işaretini görelim. En temel operatörümüz budur ve bir değişkene bir değer atamak için kullanılır.

```
$isim = "Mustafa"; //  $isim değişkenine Mustafa değerini atadık.
$eni = 2.24; // $eni değişkenine 2.24 değerini atadık.
```

Diğer atama operatörleri için de aşağıdaki tabloyu inceleyelim.

Operatör	Kullanım	Anlamı
+=	$a +=$b	$a=$a + $b
-=	$a -=$b	$a=$a - $b
-=	$a +=$b	$a=$a * $b
-=	$a /=$b	$a=$a / $b
-=	$ilk .=$son	$ilk=$ilk.$son

Karşılaştırma Operatörleri

Operatör	Kullanım	Anlamı
= =	eşittir	$x = = $y
= = =	denktir	$x = = = $y
! =	eşit değil	$x ! = $y
< >	eşit değil	$x < > $y
<	küçüktür	$x < $y
>	büyüktür	$x > $y
<=	küçük ya da eşit	$x <= $y
>=	büçük ya da eşit	$x >= $y

Mantıksal Operatörler

Operatör	Anlamı	Kullanım	Açıklama
!	DEĞİL	!$x	$x değişkeninin değerinin tersine işlem yapar.
&&	VE	$x && $y	Değişkenlerin değerlerinin hepsi doğru olmalıdır.
||	VEYA	$x || $y	Değişkenlerin değerlerinden sadece birinin doğru olması yeterlidir.

Aritmetik Operatörler

Operatör	Anlamı
+	Toplama
–	Çıkarma
*	Çarpma
/	Bölme
%	Mod (Kalan)

Şimdi aritmetik operatörler ile ilgili bir örnek yapalım.

```
<?php
$sayi1=20;
$sayi2=2;

$topla = $sayi1+$sayi2;
$cikar = $sayi1-$sayi2;
$carp = $sayi1*$sayi2;
$bol = $sayi1/$sayi2;
$kalan = $sayi1%$sayi2;

echo $topla. "<br />";
echo $cikar. "<br />";
echo $carp. "<br />";
echo $bol. "<br />";
echo $kalan;
?>
```

Örneğimizde, ilk önce `$sayi1` adında bir değişken oluşturup `20` değerini atadık. Bir de `$sayi2` değişkeni oluşturup ona da `2` değerini atadık. Ardından `$topla` adlı bir değişken daha oluşturup `$sayi1` ve `$sayi2` değişkenlerini toplattık. Diğer işlemler için de aynı metodu izledik. Ardından da hepsini ekrana yazdırdık.

Ayrıca `.` (nokta) operatörü yardımıyla XHTML'den bildiğiniz `<br />` etiketini ekleyerek sonuçların alt alta görüntülenmesini sağlamış olduk. Programı localhostta denerseniz sırasıyla; `22`, `18`, `40`, `10` ve `0` sonuçlarını alt alta yazdırdığını göreceksiniz.

Yapacağını aritmetik işlemlerde değerleri sadece `1` arttırmak için `$a++` `1` azaltmak için de `$a--` yazmanız yeterli olacaktır. Bu kullanımda PHP değişkeni işleme sokup ardından da değeri bir artırır ya da bir azaltır. Eğer işaretleri değişkenden önce kullanırsak, yani `++$a` ve `--$a` şeklinde kullanırsak, önce değişkenin değeri bir arttırılır ya da azaltılır ve ardından değişken işleme konulur.

Bir de işlem önceliği konusuna değinelim. Eğer ikiden fazla sayıyla işlem yapıyorsak, (örn: `2+2*3`) sonuç ne olacaktır? Burada yapmamız gereken parantez kullanmaktır. Yani işlemi `(2+2)*3` şeklinde yazarsak

sonuç 12 olur. İşlemi 2+(2*3) şeklinde yazdığımızda ise sonuç 8 olacaktır. Bir başka ifade ile önce parantez içindeki işlemler yapılır.

Son olarak bazı aritmetiksel fonksiyonları görelim. En sık kullanılanları aşağıdadır:

Pow () Üstel fonksiyonlar
Max () Maksimum (en yüksek) değer
Min () Minimum (en küçük) değer
Sqrt () Karekök
Sin () Sinüs
Cos () Kosinüs
Pi () Pi sayısı

İsteseniz bir örnekle kullanımlarını görelim.

```
<?php
print "Pi Sayisi :".pi()."<br />";
print "Sinüs99:".sin(99)."<br />";
print "Kosinüs 99:".cos(99)."<br />";
print "2 üssü 4 :".pow(2,4)."<br />";
print "Karekök 99 :".sqrt(99)."<br />";
?>
```

Örneğin ekran çıktısı da aşağıdaki gibidir.

```
Pi Sayisi :3.1415926535898
Sinüs99:-0.99920683418635
Kosinüs 99:0.039820880393139
2 üssü 4 :16
```

PHP'de Program Denetimi

Programlarımızı yazarken, programın yapmasını istediğimiz şeyleri hangi sırayla ve hangi şartlarla yapacağını ona söylememiz gerekir. Bu şekilde bir nevi bir akış planı hazırlamış oluruz ve böylece programı kontrol altına almış oluyoruz. PHP'de bu işlemler için kontrol yapılarını ve döngüleri kullanıyoruz.

KONTROL YAPILARI

IF DEYİMİ

Türkçeye eğer şeklinde çevirebileceğimiz `if` deyimi, programda istediğimiz koşulları oluşturmamızı sağlar.

Kullanımı

`If (şart) {komut;}` şeklindedir.

İstediğimiz şart sağlandığı takdirde komut kısmındaki işlem gerçekleştirilir. Koşulun sağlanmadığı durumlarda program bu satırı atlayıp yoluna devam edecektir.

Örnek

```
<?php
$deger=1000;
if ($deger == 1000)
echo "Sonuç Doğrudur.";
?>
```

Örneğimizde programdan, eğer girdiğimiz değer `1000`'e eşitse sayfaya "Sonuç Doğrudur" ibaresini yazdırmasını istedik. Gerçekten de `$deger` değişkenine atadığımız değer `1000` olduğundan koşulumuz yerine geldi ve program ekrana "Sonuç Doğrudur" ibaresini yazdırdı.

Örneğimizdeki `$deger` değişkenine atadığımız değeri `1001` olarak değiştirirseniz, koşul sağlanamayacak ve program bu komut satırını atlayacağından ekrana hiçbir şey yazdırmayacaktır. Tabi programımızda başka komutlar da olmuş olsaydı bu satırı atlayıp diğer komutlarla devam edecekti. Eğer programdan istediğimiz koşul sağlanmadığından satırın atlanmasını değil de vereceğimiz bir başka komutu yerine getirmesini istersek ne yapmamız gerekir? Tam da bu noktada devreye `If ...else` kalıbı girmektedir.

Kullanımı

```
If (şart) {komut;
}else{
ikinci komut;}
```

şeklindedir.

```
<?php
$sonuc=180;
if ($sonuc>=190){
echo “Sınavı Kazandınız.”;
}else{
echo “Sınavı Kazanamadınız.”;
}
?>
```

Örneğimizi denerseniz ekrana `Sınavı Kazanamadınız.` ibaresinin yazdırıldığını göreceksiniz. Çünkü koşulumuz `$sonuc` değişkenine atadığımız değerin en az `190` ya da daha büyük bir değer olmasıyla gerçekleşecektir. Dolayısıyla bu örnekte koşul sağlanamadığından `else` ile verilen komut geçerli olmuş oldu.

Tabi yukarıdaki örnekte olduğu gibi her zaman uygulamamız gereken tek bir koşul olmayabilir. Bunun için de `elseif` kalıbını kullanacağız. Aşağıdaki örneği inceleyin.

```
<?php
$rakam=1;
if($rakam==”1”){
echo “Doğru”;

} elseif($rakam==”0”){
echo “Yanlış”;
}else{
echo “Anlamsız”;
}
?>
```

Gördüğünüz gibi bu örnekte `elseif` ile bir ikinci koşul daha tanımladık. Koşullardan hangisi sağlanıyorsa işleme o sokulacaktır. İki koşul da sağlanmıyorsa bu sefer `else` ile belirtilen komut işleme alınacaktır.

SWITCH DEYİMİ

Bir değişken değerinin mevcut şartlardan hangisine uyduğunu denetlemek için kullanılır. Uygun koşul hangisi ise onunla ilgili komutlar devreye sokulacaktır. Uygun koşul gerçekleştirildiğinde `break` kalıbı ile `switch` yapısından çıkılır. Eğer hiçbir koşul sağlanamıyorsa o zaman `default` kalıbı ile kullanılan komutlar gerçekleştirilecektir.

```
<?php
$mevsim=”YAZ”;
switch ($mevsim) {
case “İLKBAHAR”;
echo “Mevsimlerden İlkbahar”;
break;
case “SONBAHAR”;
echo “Mevsimlerden Sonbahar”;
break;
case “KIŞ”;
echo “Mevsimlerden Kış”;
break;
default:
echo “Hava Çok Sıcak”; }
?>
```

Örneğimizde koşulların hiçbiri sağlanamadığından ekrana `Hava Çok Sıcak` ibaresi yazdırılacaktır. Eğer değişkenimizin değeri `İLKBAHAR` olsaydı PHP ilk komut bloğunda aradığını bulmuş olacaktı ve `break` ile işlemi sonlandıracaktı. Diğer koşulları atlayıp, yoluna (eğer belgede başka komutlar da varsa) `switch`'in kapama süslü parantezinden sonrasından devam edecekti.

DÖNGÜLER

İstenilen belli bir koşul sağlanana kadar işlemin tekrar etmesini sağlamak için kullanılır.

WHILE DÖNGÜSÜ

Döngü içerisinde tanımlanan koşul sağlandığı müddetçe devam eder. Koşulun sağlanmadığı durumda döngü başlamayacaktır.

Kullanımı

```
while(koşul)
{
komut
}
```

şeklindedir.

```
<?php
$hesap=5;
while ($hesap<=10)
{
echo $hesap."<br />";
$hesap++;
}
?>
```

Değeri 5 olan hesap değişkeni döngünü başlangıç noktasıdır. Döngü çalıştırılır ve `$hesap++;` ile bir arttırdığımız için `$hesap<=10` koşulu sağlanana kadar her seferinde 1 artırılarak devam eder. Son seferde 1 arttırıldığında değer 11 olacaktır ve `11<=10` bizim koşulumuzu sağlamadığından döngü sona erecektir.

DO WHİLE DÖNGÜSÜ

`while` döngüsünden farklı olarak kontrol işlemi döngünün sonunda gerçekleştirilmektedir. Yani tanımladığımız koşul ne olursa olsun döngü en az bir kez çalıştırılacak demektir.

```
do {
komut
}
while (koşul);
```

şeklindedir.

```
<?php
$sayac = 4;
do {
echo $sayac ++."<br />";
}
while ( $sayac <=8 ) ;
?>
```

Bu örnekte de koşulumuz gereği döngü 4 ile başlayıp 8'e kadar (8 dahil) devam edecektir.

FOR DÖNGÜSÜ

Komutlarım istenilen sayıda tekrarlanmasını sağlar. `While` döngüsüne benzer. Bunda değişken ve alacağı değer for döngüsünün başında belirlenir.

Kullanımı

```
for (değişken; koşul; değişim değeri)
{
komutlar
}
```

şeklindedir.

Örnek

```
<?php
for ($hesap=5;$hesap<=10;$hesap++)
{
echo $hesap ."<br />";
}
?>
```

Örneğimizde öncelikle bir değişken oluşturduk ve değişkenin değerini 5 olarak belirledik. Koşul olarak değişken değerimizi 10'dan küçük ya da eşit olacak şekilde ayarladık ve değişim değeri olarak da birer birer artırmak istediğimizi belirttik. Örneği denediğinizde programın ekrana sırasıyla 5, 6, 7, 8, 9 ve 10 değerlerini yazdırdığını göreceksiniz. İlk değerimiz 5 olduğundan haliyle ilk olarak bu değer yazdırıldı. Ardından döngü, $hesap<=10 koşulu sağlanana kadar devam etmiş oldu.

CONTINUE

Continue deyimi döngü yapılarının içinde kullanılır. Continue kullanıldığında, döngünün istenilen kısmı atlanacak ve döngü devam edecektir.

Örnek

```
<?php
for ($sayi = 1; $sayi <= 8; ++$sayi) {
    if ($sayi == 4)
        continue;
    print "$sayi\n";
}
?>
```

Örneğin çıktısı: 1 2 3 5 6 7 8 şeklindedir. Gördüğünüz döngüde 4 rakamı atlanmış.

Fonksiyonlar

Fonksiyonları, girilen değerleri alan, bunları işleyen ve gerekli sonuçları üreten kod blokları olarak tanımlayabiliriz. Bir kere tanımlandıktan sonra her gerektiğinde kullanılabilirler. PHP'de hazır birçok fonksiyon bulunduğu gibi, biz de programımız için gereken fonksiyonları tanımlayabiliriz. Değişkenler için geçerli olan isimlendirme kuralları fonksiyonlar için de aynen geçerlidir. Yani fonksiyon isimleri bir harf ile veya (_) alt çizgi ile başlamalıdır. Fakat büyük/küçük harfe duyarlı değildirler. Yani programın bir yerinde büyük harfle bir yerinde küçük harfle de yazılmış olsalar herhangi bir hata oluşmaz.

Fonksiyon Tanımlama

Fonksiyonları, `function` komutu ile tanımlıyoruz.

Kullanımı

```
function fonksiyonun_adı (argüman1, argüman2)
{
komutlar
}
```

Gördüğünüz gibi değişkenler parantez içinde ve aralarına virgül konularak kullanılır. İstenildiği kadar eklenebilir. Fonksiyonla iletilecek değerler PHP'de **argüman** olarak adlandırılır. Bir argüman olmasa bile yani içi boş olsa bile parantezleri yine de kullanmamız gerekir. Tıpkı aşağıdaki örnekte olduğu gibi.

```
<?php
function slogan()
{
print "Ben PHP Dilini Çok Sevdim";
}
slogan( );
?>
```

Ayrıca, fonksiyon içinde bir başka fonksiyon çağırabileceğiniz gibi, hazır PHP fonksiyonlarını da kullanabilirsiniz. Bu genel tanımlamalardan sonra artık örneklerle konuyu pekiştirebiliriz.

```
<?php
function slogan($metin)
{
print($metin);
}
slogan("Ben PHP Dilini Çok Sevdim");
?>
```

Bu basit örnekte slogan adında bir fonksiyon tanımladık. Ardından, `Ben PHP Dilini Çok Sevdim` değerini fonksiyona vererek yazdırdık. Tahmin edeceğiniz gibi değer `$metin` değişkeninde tutulacaktır. Örnekteki gibi `echo` yerine `print` komutu da kullanılabilir.

```
<?php
function kod($sayi1,$sayi2)
{
if ($sayi1==$sayi2)
print("Doğru Kodlar");
else
print("Hatalı Kodlar");
}
kod(123,124);
?>
```

Bu örneğimizde gördüğünüz gibi if else kontrol yapısını da kullanarak karşılaştırma yaptırdık ve sonucu ekrana yazdırttık. İki değeri, yani iki değişkenimize iletilecek değerleri (`123`, `124`) farklı girdiğimizi için koşulumuz sağlanamayacak ve ekrana `Hatalı Kodlar` ibaresi yazdırılacaktır.

Geriye Değer Döndüren Fonksiyonlar

Bu tür fonksiyonlar `return` ifadesi ile kullanılırlar. Amaç, `return` ile fonksiyonun çalışmasını durdurarak programda fonksiyonu çağıran koda bir başka fonksiyonda kullanılmak üzere değer döndürmektir. Bir aritmetiksel örmek yaparak konuyu açıklığa kavuşturalım:

```
<?php
function topla ($sayi1, $sayi2)
{
$sonuc = $sayi1 + $sayi2;
return $sonuc;
}
```

```
function cikart ($sayi1, $sayi2)
{
$sonuc = $sayi1 - $sayi2;
return $sonuc;
}
$sayi1 = 50;
$sayi2 = 20;
echo topla($sayi1, $sayi2).”<br />”;
echo cikart($sayi1, $sayi2);
?>
```

Örneğimizde gördüğünüz gibi argümanlarımızı her iki fonksiyonda da kullandık. Şimdi de aşağıdaki örneği inceleyin. Burada size bir şey göstereceğim:

```
<?php
function topla ($sayi1 = 3, $sayi2 = 2)
{
$sonuc = $sayi1 + $sayi2;
return $sonuc;
}
$bir = 30;
$iki = 6;
print topla($bir, $iki).”<br>”;
print topla($bir).”<br>”;
print topla();
?>
```

Örneği incelediyseniz bazı şeyler dikkatinizi çekmiştir. Bu sefer argümanlarımıza değerler verdik. `Topla` fonksiyonunu ekrana yazdırırken üç yazdırma komutu kullandık. İlkinde her şey tamam. İkincisin bir, üçüncüsünde ise hem bir hem de iki argümanı eksik. O zaman kodu çalıştırınca hata verir diye düşünebilirsiniz. Hayır hiçbir hata oluşmayacak çünkü `return` ifadesini kullandığımız için değer döndürülecek ve fonksiyonumuzu tanımlarken argümanlarımıza girdiğimiz değerler varsayılan değerler olarak işleme konulacaktır.

Gerçekten de programı çalıştırdığımızda ekranda sırasıyla `36`, `32` ve `5` değerlerini göreceğiz. Çünkü ilk işlemde `30` ve `6`, ikincisinde `30` ile `2`, üçüncüsünde ise `3` ile `2` toplandı ve ekrana basıldı.

Global Değişkenli Fonksiyonlar

Normal şartlar altında fonksiyonlar kendilerine ait değişkenleri kullanabilirler. Bir başka fonksiyon içinde tanımlanmış bir değişkene erişemezler. Bu işlemi yapmak istediğimizde `global` ifadesini kullanmamız gerekir.

```
<?php
$deger = 10;
function goster ()
{
print ($deger);
}
goster();
?>
```

Örnekteki gibi fonksiyon dışındaki bir değişkeni fonksiyon içinden çağırırsanız hata alırsınız. Yukarda anlattığım gibi bunun için `global` kullanacağız. Şimdi de aşağıdaki örneği inceleyin:

```
<?php
$deger = 10;
function goster ()
{global $deger;
print ($deger);
}
goster();
?>
```

Bu örneği denediğinizde ise ekrana `10` değerini yazdıracaktır. Çünkü gördüğünüz gibi değişken fonksiyon dışında olsa da `global` ile çağırdık ve programımız çalıştı.



Bir fonksiyonun programın tamamında değil de, sadece bir tek fonksiyonda kullanılması gerektiği durumlarda değişkeni fonksiyonun içinde tanımlamak daha mantıklı olacaktır.

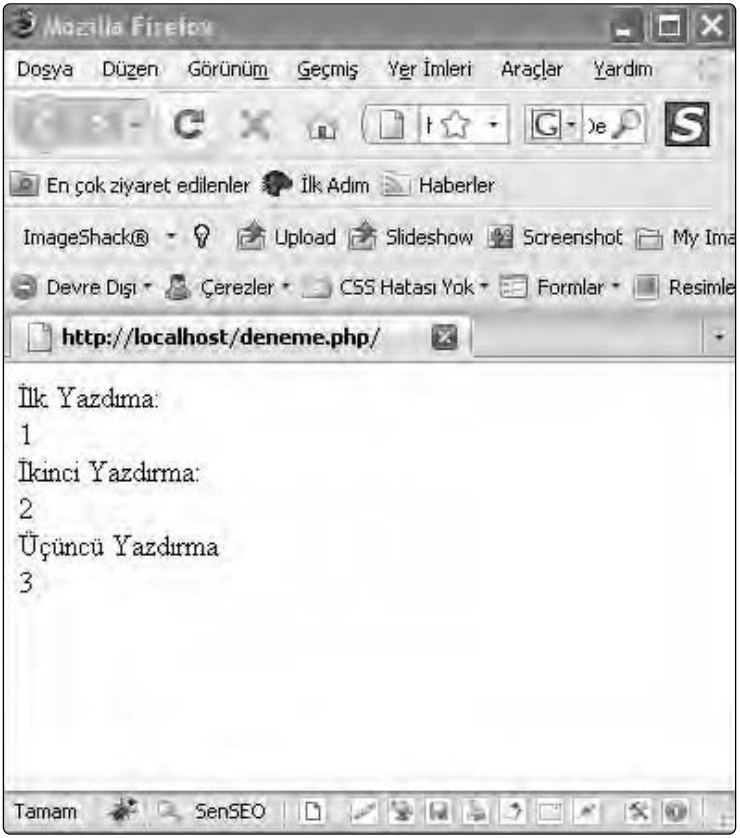
Statik Değişkenli Fonksiyonlar

Bir fonksiyonda kullanılan bir değişkenin değeri işlem bittikten sonra ilk değerine döner. Fakat bazı durumlarda fonksiyonumuzu birden faz-

la çağırmamız gerekir. Eğer bildiğimiz yöntemlerle bunu yaparsak hep aynı değerle yani değişkenin kendi değeriyle karşılaşırız.

Mesela fonksiyonu üçüncü kez çağırmamız gerektiğinde, ikinci çağırışımızdaki değer program tarafından tutulmamışsa yaptığımız işlem anlamsız olacaktır. Bunun için, yani her seferinde değişkenin değerinin başa dönmemesi için yapmamız gereken değişkenin başına `static` ifadesi eklemektir.

```
<?php
function tut() {
static $deger = 0;
$deger++;
echo $deger."<br />";
}
print ("İlk Yazdıma:"."<br />");
tut();
print ("İkinci Yazdırma:"."<br />");
tut();
print ("Üçüncü Yazdırma"."<br />");
tut();
?>
```



Örneği tarayıcıdan çağırdığınızda resimdeki sonucu göreceksiniz. Tam istediğimiz gibi değerler birer birer artmaktadır. Eğer bu örneği `static` ifadesi olmadan denerseniz yukarıda anlattığım sebepten dolayı her seferinde 1 sonucunu alırsınız.

MAIL FONKSİYONU

Web sitenizden zaman zaman üyelerinize mail atmanız, sitedeki yeniliklerden onları haberdar etmeniz gerekecektir. Ya da mesela üyelik işlemleri sırasında üyenin mail adresine aktivasyon maili göndermeniz gerekebilir. Hangi gerekçe ile olursa olsun PHP ile mail göndermek tek bir fonksiyon ile yani `mail` fonksiyonu ile gerçekleştirilmektedir.

Kullanımı

```
mail ( "mailin_gönderileceği_adres" , "mailin_konusu" ,
"mailin_içeriği" ,
        "başlık bilgileri");
```

şeklindedir.

```
<?php
$adres = "filankisi@adres.com";
$konu  = "Mailin konusu";
$mesaj = "Bu bir deneme mesajıdır.";
$header = "From: admin@gondereninsitesi.com
           Reply-To: info@gondereninsitesi.com
           Content-type: text/html; charset=iso-8859-9" . "\r\n";
mail("$adres","$konu","$mesaj","$header");
?>
```

Örneğimizde yer alan değişkenlerin görevlerini kısaca özetlersek:

$adres mailin gönderileceği kişinin mail adresi.

$konu gönderilecek mailin konusu

$mesaj gönderilecek mailin içeriği

$header gönderilecek mailin başlık bilgileri

Burada ek açıklamaya muhtaç bir tek `$header` değişkeni var. Öncelikle bir başlık değişkeni kullanmanın zorunlu olmadığını, opsiyonel olduğunu belirteyim. Bu değişkenle biz, `mail` ile ilgili bazı tanımlamalar yapmış oluyoruz. Bunları da inceleyelim;

From: Bunu belirlemezsek mail sunucudaki varsayılan bir mail adresinden gönderilecektir. Fakat bu tanımlamayı yaparsak mail bizim belirlediğimiz e-mail'den gittiği bilgisi eklenecektir.

Reply-To: Maili gönderdiğimiz kişi kendine gelen e-maili yanıtlamak isterse burada tanımladığınız e-mail adresine mesajını gönderebilecektir.

Content-type: İçerik türünün ve karakter setini tanımlanması içindir. HTML tipinde mail göndereceğimizde bunu eklememiz gerekmektedir.

Ve nihayetinde `mail` komutu ile mailimizi gönderiyoruz. Tabi fonksiyonun içinde yer alacak değişkenleri de aralarına virgül koyarak ekledikten sonra. Bu örneği denediğinizde bir hata mesajı alacaksınız. Bunun sebebi bilgisayarımızda bir mail sunucusunun kurulu olmamasıdır. Kodları bir sunucuda denediğinizde sorunsuz çalışacaktır.

Bu örnekte kullandığımız `\r` ve `\n` karakterleri de neyin nesi diye düşünmüş olabilirsiniz. Bu karakterlerin (mesela bir `($)` işaretinin PHP'de bazı özel işlevleri bulunmaktadır. Örneğin `$` işaretini değişken oluşturmakta kullanıyoruz. PHP bir değişkenin başındaki `$` işaretine bakarak, bunun bir değişken olduğunu anlar. Biz `$` işaretini değişken oluşturmak için değil de sadece dolar işareti olarak kullanmak istersek başına bir `\` koymamız gerekir. Şimdi benzer kullanımı olan işaretlere bir bakalım.

`\'` Tek tırnak
`\"` Çift tırnak
`\\` Ters-bölü
`\$` Dolar işareti
`\n` Yeni Satır (New Line)
`\r` Satır Başı (Return)
`\t` Sekme (Tab) karakteri
`\D` Ondalık işareti
`\s` Boşluk (space)

TARİH-ZAMAN FONKSİYONLARI

Tarih ve zaman fonksiyonları, yapacağınız web sitelerinde en çok ihtiyacınız olacak şeylerden birisidir. PHP ile kullanılabilen çok sayıda zaman fonksiyonu mevcuttur. Dolayısıyla bu konuda PHP bize çok geniş bir hareket alanı sunmaktadır. İşin detaylarını örneklerle ve tablolarla vermeye çalıştım.

GETDATE() FONKSİYONU

Bu fonksiyonu programımıza tarih ve zaman bilgisi eklemek için kullanacağız. İşte bir örnek;

```
<?php
```

```
$gunun_tarihi=getdate();
print $gunun_tarihi['mday']."-".$gunun_tarihi['mon']."-
".$gunun_tarihi['year'];
?>
```

Bu fonksiyon bize, gün, ay ve yıl şeklinde günün tarihini gösterecek. Örneğimizde kullanılan anahtarları (örn: mday) ve diğerlerini işlevleri ile beraber aşağıdaki tablodan inceleyebilirsiniz.

Anahtarlar	Açıklama
Seconds	Saniyeleri sayısal olarak gösterir.
Minutes	Dakikaları sayısal olarak gösterir.
Hours	Saatleri sayısal olarak gösterir.
Mday	Ayın günlerini sayısal olarak gösterir.
Wday	Haftanın günlerini sayısal olarak gösterir.
Mon	Ayların sayısal olarak gösterir.
Year	Yılların sayısal olarak gösterir.
Yday	Yılın günlerini sayısal olarak gösterir.
Weekday	Haftanın günlerini metin olarak gösterir.
Month	Ayları metin olarak gösterir.

DATE() FONKSİYONU

Tarih/saati biçimlendirmek için kullanılır.

Örnek

```
<?php
print date("d-m-Y");
?>
```

Bu örnekte gün, ay ve yıl formatında tarihi görüntülemiş olduk. Tabi biz kendi bilgisayarımızda çalıştığımız için görüntülenecek olan bilgiler bilgisayarımıza ait tarih ve zaman bilgileri olacaktır. Bir sunucuda çalışacağınız zaman ise görüntülenecek değerler sunucunun zaman değerleri olacaktır. Şimdi de tabloyu inceleyelim.

Parametre	Açıklama
d	Ayın günlerini önlerine sıfır koyarak gösterir."01-31"
D	Günlerini 3 harften oluşan metin olarak gösterir.
j	Ayın günlerini önlerine sıfır koymadan gösterir."1-31"
l (küçük 'L' harfi)	Haftanın günlerini metin olarak gösterir.
N	Haftanın günlerinin ISO-8601 standardında sayısal olarak gösterir (PHP 5.1.0 sürümünü ile gelen bir özelliktir.)
S	Ay günlerine 2 karakterli İngilizce sıralama ekler. (st, nd, rd-birinci, ikinci... vd anlamında)
w	Hafta günlerini sayısal olarak gösterir
z	Yılın günlerini rakamsal olarak gösteiri. (0'dan başlar)
W	ISO-8601 standardına göre yılın hafta numarası, (PHP 4.1.0 sürümüyle eklenmiştir.) (Örn: 32.hafta)
F	Ayları tam metin olarak gösterir. (Örn: Ocak, Şubat)
m	Ayları önlerine sıfır koyarak gösterir. "01-12"
M	Ayları 3 harften oluşan metin olarak gösterir
n	Ayları önlerine sıfır koymadan gösterir."1-12"
t	Ayların toplam gün sayısını gösterir. "28-31"
L	Artık yıl olduğunda Boolean (doğru/yanlış) ile, artık yıl ise 1 değilse 0 gösterir.
Y	Yılları sayısal olarak (4 haneli) gösterir.
y	Yılları sayısal olarak (2 haneli) gösterir.
e	Zaman dilimi belirtecidir. (PHP 5.1.0 sürümünü ile gelen bir özelliktir.)
I (büyük ı)	Yaz saati uygulaması olup olmadığını gösterir. Varsa 1, yoksa 0 gösterir.
c	ISO 8601 standardına göre tarih ve saat gösterir. (PHP 5 sürümü ile gelen bir özelliktir.)
P	Saat ve dakika olarak Greenwich'e göre zaman (GMT) farkını gösterir. (PHP 5.1.3 sürümüyle gelen bir özelliktir.)

TIME() FONKSİYONU

Zamanı saniye cinsinden gösterir.

Örnek

```
<?php
$zaman = time() + (1* 24 * 60 * 60);
echo 'Bugün:'. date('d-m-Y') ."<br />"."\n";
echo 'Yarın:'. date('d-m-Y',$zaman)."\n";
?>
```

Örneğimizde `time` fonksiyonunu `date` fonksiyonu ile birlikte kullanarak bugünün ve bir sonraki günün tarihini yazdırdık. Fonksiyondaki **+** işareti sonraki **–** işareti ise önceki anlamındadır. Parantez içindeki `1 * 24 * 60 * 60` ifadesi günü, saati, dakikayı ve saniyeyi belirtir. Mesela burada `1` yerine `7` yazarak bir hafta sonrasını görüntüleyebiliriz. Yine fonksiyondaki **+** yerine **–** yazarsak bir önceki günün tarihini bulmuş oluruz.

DİZİLER

Diziler, çok sayıda değer içeren değişkenlerdir. Dizilerimizi `array()` fonksiyonu ile oluşturacağız.

```
<?php
$isimler[] = "Mehmet";
$isimler[] = "Yusuf";
$isimler[] = "Ahmet";
$isimler[] = "Salih";
echo $isimler[2];
?>
```

Bu ilk örneği çalıştırırsanız, `Ahmet` sonucunu üretecektir. Çünkü değişkenlerdeki indisler boş olduğundan `0`'dan başlayarak PHP bunları kendisi sıraladı.

```
<?php
$isimler[2] = "Mehmet";
$isimler[3] = "Yusuf";
$isimler[4] = "Ahmet";
$isimler[5] = "Salih";
echo $isimler[2];
?>
```

Şimdi de bunu deneyin. Bu sefer işi PHP'ye bırakmadık ve değerleri biz girdik. Dolayısıyla bu sefer `Mehmet` çıktısını aldık.

```
<?php
$isimler=array("Mehmet","Yusuf","Ahmet","Salih");
echo $isimler[3];
?>
```

Bu örneğimizde array fonksiyonu ile değer tanımladık ve `$isimler` değişkeninin 4. değeri olan `Salih` değerini yazdırdık. Değişkenin indisi `3` olduğundan, değişkenin 4. değerine (`Mehmet=0`, `Yusuf=1`, `Ahmet=2`, `Salih=3`) karşılık gelmiş oldu.

```
<?php
$isimler=array("Mehmet","Yusuf","Ahmet","Salih");
for($ad=0;$ad<=3;$ad++)
echo $isimler[$ad]."<br>";
?>
```

Bu örneğimizde ise `for` döngüsü ve birde farklı olarak indis değeri için bir başka değişken kullandık. Burada bir döngü olduğundan ve `$ad` değişkenin değeri `0` olduğundan `echo` komutu dört değeri de (`Mehmet`, `Yusuf`, `Ahmet`, `Salih`) ekrana gösterecektir. Döngüdeki `$ad<=3` koşulunu `$ad<=2` olarak değiştirirsek sadece ilk üç değeri (`Mehmet`, `Yusuf`, `Ahmet`) gösterecektir.

```
<?php
$meyveler = array(
   "yaz" => array(
          1 => "Kavun",
          2 => "Karpuz",
          3 => "Kiraz"
          ),
   "kış" => array(
          1 => "Elma",
          2 => "Armut",
          3 => "Portakal"
          )
);
echo $meyveler["yaz"][2];
?>
```

Bu sefer içi içe array oluşturmayı görmüş olduk. Bu örneği çalıştırdığınızda ekrana `Karpuz` sonucu basılacaktır. Burada parantezlerin ve virgüllerin nasıl kullanıldığına lütfen dikkat edin. İçteki array yapılarını birbirlerinden `,` (virgül) ile ayırıyoruz. Fakat ikinci dizimizden sonra bir başka dizi gelmediğinden virgül kullanmadık.

```
<?php
$gunler = array(“Pazar”, “Pazartesi”, “Salı”, “Çarşamba”,
“Perşembe”, “Cuma”, “Cumartesi”);
echo $gunler[date(“w”)];
?>
```

Son olarak `date` fonksiyonu ile array fonksiyonunun kullanımını görmüş olalım. Bu program ile ekrana bilgisayarınızdaki tarih saat ayarına göre çıktı basılacaktır.

FOREACH() FONKSİYONU

Bu fonksiyon işimizi epey kolaylaştıracak olan bir fonksiyondur. Koşulsuz olarak döndürme işlemi yapar ve for döngüsünü kullanmamıza gerek kalmamış olur.

Kullanımı

`foreach(döndürülecek değişken as döngü değişkeni){ komut }`

şeklindedir.

```
<?php
$isimler = array(“Mehmet”,”Yusuf”,”Ahmet”,”Salih”);
foreach($isimler as $diz){
echo $diz.”<br />”;
}
?>
```

Gördüğünüz gibi `as` ifadesi ile `$isimler` değişkenini `$diz` değişkenine aktararak ekrana bastırdık.

```
<?php
$dizi=array(2,4,6,8,10,12);
foreach($dizi as $say=>$dizi) {
echo $say . “: “ . $dizi . “</br>”;
```

```
}
?>
```

Bu sefer indis değerini de işin içine soktuk ve birlikte yazdırdık. Örneğimizin ekran çıktısı aşağıdaki gibi olacaktır.

0: 2
1: 4
2: 6
3: 8
4: 10
5: 12

Örneğimizden sonra yolumuza devam edelim. Evet, dizileri programınızda kullanırken bazı düzenlemeler yapmanız gerekecek. Ben şimdiden söylemiş olayım. Bunları da görmemiz gerekiyor ve ilkiyle başlıyoruz.

SORT()-RSORT() DİZİLERİ SIRALAMA

Dizi değerlerini alfabetik olarak ya da küçükten büyüğe doğru sıralamak için kullanılır. İşte örnek:

```
<?php
$dizi=array("Aaaa","Deddee","Hahaa","Kekee","Bebee",
"Sesee","Lelee","Tetee");
sort($dizi);
foreach($dizi as $say) {
echo $say. "</br>";
}
?>
```

Örneğimizin ekran çıktısı:

Aaaa
Bebee
Deddee
Hahaa
Kekee
Lelee
Sesee
Tetee

Gördüğünüz gibi hepsini alfabetik olarak sıraladı. Eğer tam tersini yapmak isterseniz `sort()` yerine `rsort()` kullanmanız yeterlidir. İsterseniz deneyin! Evet deneyin demişken, yaptığımız örnekleri bilgisayarınızda mutlaka deneyin ve kendinizde benzer örnekler yapın. Konuların pekişmesi için bu gerekli.

ASORT()-KSORT() İLİŞKİLİ DİZİLERİ SIRALAMA

`asort()` sıralamayı dizin değerine göre, `ksort()` ise indise göre yapar. Örnek aşağıda:

```
<?php
$dizi = array(2 => “Aaaa”, 1 => “Kekee”, 6 => “Sesee”, 5 =>
“Bebee”,   4 => “Tetee”, 3 => “Hahaa”);
asort($dizi);

foreach($dizi as $diz => $dizi) {
   echo($diz.”:”.$dizi.”<br />”);
}
?>
```

Örneğimizin ekran çıktısına bakınca olay hemen anlaşılacak.

```
2:Aaaa
5:Bebee
3:Hahaa
1:Kekee
6:Sesee
4:Tetee
```

Evet, sıralama dizi değerine göre yapıldı. Sıralamayı ters, yani büyükten küçüğe yapmak isterseniz `asort()` yerine `arsort()` kullanın.

Şimdi bir de sıralamayı indise göre yapmayı görelim. Tek fark bu sefer `ksort()` kullanacağız.

```
<?php
$dizi = array(2 => “Aaaa”, 1 => “Kekee”, 6 => “Sesee”, 5 =>
“Bebee”,   4 => “Tetee”, 3 => “Hahaa”);
ksort($dizi);
```

```
foreach($dizi as $diz => $dizi) {
   echo($diz.":".$dizi."<br />");
}
?>
```

Ve ekran çıktımızda sıralama indise göre olacak.

```
1:Kekee
2:Aaaa
3:Hahaa
4:Tetee
5:Bebee
6:Sese
```

Bunda da ters sıralama yapmak için yapmanız gereken tek şey `ksort()` yerine `krsort()` kullanmak.

ARRAY_PUSH() Dizilere Değişken Ekleme

Bu fonksiyonu bir diziye yeni değişken ya da değişkenler eklemek istediğimizde kullanacağız. Nasıl mı? İşte örnek;

```
<?php
$ilk_dizi = array("Aaaa","Kekee","Sesee","Bebee","Tetee","Hahaa");
$yeni_dizi = array_push ($ilk_dizi,"Pepee", "Yayaa");
rsort ($ilk_dizi);
foreach($ilk_dizi as $diz) {
echo($diz."<br />");
}
?>
```

İlk dizimizden (`$ilk_dizi`) sonra yeni bir dizi (`$yeni_dizi`) oluşturduk ve rsort ile ters sıraladık. Burada yaptığımız şey, `$yeni_dizi` değişkenine `$ilk_dizi` değişkenini değerlerini aktarmak ve ardından yeni değerleri ilave ettik. Gerisi önceki uygulamalarla aynı.

ARRAY_MERGE() Dizileri Birleştirme

İki dizinin elemanlarını birleştirmek için kullanılır. Tabi istersek ikiden fazla diziyi de birleştirebiliriz.

```
<?php
$ilk_dizi = array("Aaaa","Kekee","Sesee","Bebee","Tetee","Hahaa");
$ikinci_dizi = array ("Pepee", "Yayaa");
$yeni_dizi = array_merge ( $ilk_dizi, $ikinci_dizi );
rsort ($yeni_dizi);
foreach($yeni_dizi as $diz) {
echo($diz."<br />");
}
?>
```

Burada yaptığımız şey, `array_merge` fonksiyonu ile iki dizinin elemanlarını birleştirmekten ibaret. Bunu üçüncü bir dizi değişkeni (`$yeni_dizi`) tanımlayarak yapıyoruz.

ARRAY_SLICE DİZİNİN BİR KISMINI ALMA

Eğer bir dizi değişkeninin tüm değerlerini değil de bir kısmını kullanmamız gerekirse, bu işlem için `array_slice()` fonksiyonunu kullanacağız.

Örnek

```
<?php
$ilk_dizi = array("Aaaa","Kekee","Sesee","Bebee","Tetee","Hahaa");
$yeni_dizi = array_slice($ilk_dizi,2,3);
foreach($yeni_dizi as $diz) {
   echo($diz.",");
}
?>
```

Burada `$yeni_dizi` değişkeni ile, bir kısmı alınacak diziyi (`$ilk_dizi`), hangi değerden itibaren alacağımızı (`2`) ve kaç değer alacağımızı (`3`) belirttik. Dolayısıyla programımızın ekran çıktısı, `Sesee`, `Bebee`, `Tetee` şeklinde olacaktır.

ARRAY_SUM() TOPLAMA

Dizideki değerleri toplamak içindir.

Örnek

```
<?php
$ilk_dizi = array(3,5,7,9);
```

```
echo array_sum($ilk_dizi);
?>
```

FORMLAR

Formlar konusu sizin gayet iyi bildiğiniz bir konu. O zaman bu konuyu geçelim. Tabi iş böyle değil. Biz formları görsel anlamda hazırlamayı öğrendik. Fakat bir formu hazırlayıp **Gönder** tuşuna basıldıktan sonra neler olup bitiyor ve bundan sonraki kısmı nasıl hallediliyor bunlara değinmedik. Değinmedik çünkü bunu yapabilmek için PHP bilmemiz gerekiyordu. Ama artık zamanı geldi ve başlayalım.

Önce bir iletişim formu yaparak işe başlayalım. İlk işimiz tabiki bir form hazırlamak. Siz bunu Dreamweaver ile iki dakikada halledersiniz! Ben uygulamalarımızda kullanmak üzere bir form hazırladım. Formumuz aşağıdaki gibi;

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>İletişim Formu</title>
</head>

<body>
<form id="form" name="form" method="post" action="iletisim.php" >
  <table width="400" border="0" align="left">
    <tr>
      <td width="34%">Adınız Soyadınız:</td>
      <td width="66%"><input name="isim" type="text" /></td>
    </tr>
    <tr>
      <td>Mail Adresiniz :</td>
      <td><input name="mail" type="text" /></td>
    </tr>
    <tr>
      <td>Mesajın Konusu :</td>
      <td><input name="konu" type="text" /></td>
    </tr>
```

```
    <tr>
      <td>Mesajınız :</td>
      <td><textarea name="konu" cols="21" rows="10">Lütfen
Mesajı Giriniz...</textarea></td>
    </tr>
    <tr>
      <td><input name="" type="submit" value="Gönder" /></td>

    </tr>
  </table>
</form>
</body>
</html>
```

Forma girilecek verileri işleyecek olan dosyamız **iletisim.php** dosyasıdır.

Bunu,

```
<form id="form" name="form" method="post" action="iletisim.php" >
```

satırında da görüyorsunuz. Dolayısıyla şimdiki işimiz bu dosyayı hazırlamak. Aşağıda örnek bir *iletisim.php* dosyası görüyorsunuz.

```
<?php
$isim=$_POST['isim'];
$mail=$_POST['mail'];
$konu=$_POST['konu'];
$icerik=$_POST['icerik'];
$header="From: $mail\r\n".
          "Reply-To: info@gondereninsitesi.com\r\n";
          "Content-Type: text/html; Charset=iso-8859-9\r\n";

if ($isim=="" || $mail=="" || $konu=="" || $icerik=="") {
echo "Lütfen Boş Yerleri Doldurunuz.";
}else{
mail("filanca@mail.com",$konu, $icerik, $header);
echo "Mesajınız İletilmiştir Teşekkür Ederiz.";
}
?>
```

Örneği incelerseniz konuyu zaten bildiğinizi göreceksiniz. Çünkü mail fonksiyonunu biz daha önce görmüştük. Yalnız burada iki konuya de-

ğineceğim. Birincisi `$_POST` olayı. Formumuzu gönderirken kullanabileceğimiz iki yöntem var, `post` ve `get` yöntemleri. Bildiğiniz gibi bunu formumuzun method bölümünden seçiyoruz.

Bunlardan **`get`** metodu, verileri gönderirken site URL'sinin arkasına ekler. Dolayısıyla bilgiler tarayıcının adres çubuğunda görüntülenir. Bu sebeple pek güvenli bir yöntem değildir. Özellikle şifre ve benzeri özel bilgileri gönderirken kullanılmamalıdır. Adres satırına girilecek karakter kısıtı nedeniyle zaten istediğiniz uzunlukta bir gönderim de yapamazsınız.

`Post` metodu ise, gönderilen bilgiler açık olarak gösterilmediği için daha güvenlidir. Bu metod ile alabileceğiniz veri miktarı sunucunuzda size ayrılan yer ile sınırlıdır. Yani uzun gönderilerinizi bu yöntemle yapmak durumundasınız.

Eğer formda metod olarak **`get`** seçtiyseniz, `$_GET` değişkeni, formu işleyecek olan PHP dosyasında (bizim örneğimizde iletisim.php) get metodu ile gelen istekleri yakalamaya yarar. Eğer **`post`** metodunu seçtiyseniz de `$_POST` değişkenini kullanmalısınız. Bir de dilerseniz kullanabileceğiniz üçüncü bir değişken daha var, `$_REQUEST` değişkeni. Bu değişkeni her iki metotla da kullanabilirsiniz. Varsayılan olarak `TRUE` değeri `$_GET`'dir. Yani önce get ile gelen bir istek olup olmadığına bakar.

Örneğimizde değineceğim ikinci husus `if` satırı. Burada fark ettiğiniz gibi amacımız formda herhangi bir alanın boş bırakılması halinde kişiyi uyarmak. Bunun için de veya (`||`) operatörünü kullandık.

İçerik Kontrolü

Hazırladığınız formlara sitenizin ziyaretçileri tarafından girilecek bilgilerin kontrolünün gerektiği bir çok durumlar olacaktır. Mesela e-mail adresinin mutlaka girilmesini, bu alanın boş olmamasını isteyebilirsiniz. Ya da formdaki diğer alanların da doldurulmasını isteyebilirsiniz. Formdaki doldurulmasını istediğiniz alanlar boş bırakılarak form gönderilmek istendiğinde, bir uyarı mesajı ile ziyaretçiyi uyarabilirsiniz.

Böylece ardarda binlerce kez boş forumlar gönderilmesini de engelleyeceksiniz. Bu durum ayrıca güvenlik açısından da gereklidir. Şimdi bu işlemi yapabileceğimiz yöntemleri görelim.

If ( ) Komutu

Bu yöntemi son örneğimizde kullanmıştık. Bu sayede formun boş olarak gönderilmesini engellemiştik. Yani belirlediğimiz alanlardan herhangi biri boşsa kişiyi uyar boş değilse mesajı ilet komutu vermiş oluyoruz. İşte örneğimizin ilgili bölümü;

```
if ($isim=="" || $mail=="" || $konu=="" || $icerik=="") {
echo "Lüttfen Boş Yerleri Doldurunuz.";
}else{
mail("filanca@mail.com",$konu, $icerik, $header);
echo "Mesajınız İletilmiştir Teşekkür Ederiz.";
```

Isset ( ) Komutu

Bu fonksiyonun amacı, bir değişkenin içinde değer olup olmadığını kontrol etmektir. Eğer değer varsa isset `()` fonksiyonu `TRUE`, değer yoksa `FALSE` sonucunu verecektir. Şimdi yukarıdaki örneğimizi, bu sefer isset `()` fonksiyonu ile yazalım.

```
<?php
if(!isset($isim)) {
echo "Lütfen İsminizi Giriniz.";
}elseif (!isset($mail)){
echo "Lütfen Mail Adresinizi Giriniz.";
}else{
echo "Mesajınız İletilmiştir Teşekkür Ederiz.";
}
?>
```

Bu şekilde isim ve e-mail alanlarının boş olmasını engellemiş olduk. Dilersek tüm alanlar için de aynı şeyi yapabiliriz. Örneğimizde fonksiyona `!` operatörünü ekledik. Böylece eğer isim yok ise demiş olduk.

Empty ( ) Komutu

`Empty ( )` fonksiyonu da `isset()` fonksiyonu ile aynı işlevi görecektir. Farkı sorgulama mantığındadır. Bu fonksiyon `isset()` fonksiyonunun aksine, değişkene değer atanmışsa `FALSE`, boş ise `TRUE` sonucunu verecektir. Şimdi örneğimize bakalım.

```
<?php
if(empty($isim)) {
echo "Lütfen İsminizi Giriniz.";
}elseif (empty($mail)){
echo "Lütfen Mail Adresinizi Giriniz.";
}else{
echo "Mesajınız İletilmiştir Teşekkür Ederiz.";
}
?>
```

Gördüğünüz gibi burada da yaptığımız iş aynı. Tabi `empty ()` fonksiyonunun önüne `!` operatörünü eklemedik. Çünkü fonksiyonun mantığı gereği, değişken boş iken `TRUE` değerini döndürür.

STRLEN( ) KOMUTU

Hazırlayacağınız formda yapabileceğiniz tek kısıtlama alanların boş olmasını engellemek değildir. Mesela güvenlik açısından form alanlarına girilecek veri miktarını da kısıtlamanız gerekebilecektir. Bunu forma en fazla kaç karakter girilebileceğini belirleyerek yapabilirsiniz.

Bu işlem için `strlen ()` fonksiyonunu kullanacağız. Bu fonksiyon verinin karakter sayısını kontrol eder. Aşağıdaki örneği inceleyerek meseleyi açıklayalım.

```
<?php
if (strlen($sifre)<6){
echo "Şifreniz en az 6 karakter olmalıdır.";
}else{
//Gereken diğer kodlar
echo "Kaydınız Tamamlanmıştır. Teşekkür Ederiz.";
}
?>
```

Diyelim ki, ziyaretçi üyelik formuna üye kaydı işlemi yapıyor. Ve siz, üyenin belirleyeceği şifrenin en az `6` karakterden oluşmasını istiyorsunuz. Örneğimizdeki gibi bir kod ile kişinin karakter sayısı `6`'dan az olan bir şifre belirlemesini engellemiş oluruz. Kişi buna teşebbüs ettiğinde bir uyarı mesajı ile karşılaşacaktır.

Tabi burada biz sadece bu fonksiyonu anlatmaya çalıştık. Şifre 6 karakter ya da daha fazla olduğunda, yani kişi sizin istediğiniz karakter kısıtlamasına girmeyen uygun bir şifre belirlediğinde yapılacak işlemi gösteren bir kod yazmadık.

Ereg( )-eregi( ) Komutu

Şimdi `ereg( )-eregi( )` fonksiyonlarını göreceğiz. Bu fonksiyonları, değer içinden istenilen karakter eşleşmesini yapmak için kullanırız. Eşleşme olduğunda `TRUE`, olmadığında ise `FALSE` değeri verir. Aralarındaki fark, `eregi( )` fonksiyonunun eşleştirmede büyük-küçük harf ayrımı yapmamasıdır. Şimdi forma girilecek mail adresinin gerçek bir mail adresi olup olmadığını doğrulama işlemine bir örnek yapalım.

```
<?php
if (eregi("^.+@.+\\..+$", $mail,$tut)) {
echo "E-Posta Adresiniz:".$tut [0];
}else {
echo "E-posta adresiniz hatalı.!";
}
?>
```

Fonksiyonun içeriği, `eregi("^.+@.+\\..+$", $mail)` şeklindedir. Burada önce eşleştirilecek karakterleri girdik. Ardından da eşleştirilecek karakterleri tutan değişkenin adını yazdık. Bir mail doğrulaması yaptığımızı varsayarak konuşursak, buradaki $mail değişkeninde tutulacak değer, kişinin gireceği mail adresidir. Yani kişinin gireceği mail adresiyle bir karakter eşleştirmesi yapmaya çalışacağız.

Bu örneği yukarıdaki form örneği ile birlikte inceleyebilirsiniz. Buraya istersek üçüncü olarak bir değişken ekleyerek sonucu bu değişkene aktarabiliriz. O zaman fonksiyonumuz `eregi("^.+@.+\\..+$", $mail,$tut)` şeklinde olur. Mesela eşleştirme sonucu dizi değişkenimizin (`$tut`) tuttuğu değeri de ekrana yazdıralım. Fonksiyondaki (`^.+@.+\\..+$`) karakterleri hakkında detaylı bilgi için okumaya devam edin!

ereg_replace()-eregi_replace() Komutu

Aradığımız bazı değerleri bir başka değer ile değiştirmek için kullanılır. Program aranılan değeri bulursa yerine bizim belirlediğimiz yeni değe-

ri koyacaktır. Bu fonksiyon özellikle güvenlik açısından bize çok faydalı olacak.

Mesela bazı site ziyaretçileri hazırladığımız formlarla zararlı kodlar içeren veriler yollayabilirler. Burada programlarda sık kullanılan bazı karakterleri bulup yerine mesela boşluk karakteri yazdırırsak yani karakteri silersek eklenen zararlı kod işe yaramaz hale gelecektir.

Kullanımı

```
ereg_replace ("eşleştirilecek_karakterler", yeni_karakter, $deger)
```

```
<?php
$degistir = '1';
$bul = "Rakamla: bir";
$bul = ereg_replace('bir', $degistir, $bul);
echo $bul;
?>
```

Örnekte metin içindeki "bir" ifadesini "1" rakamıyla değiştirdik. Tabi burada değişkene biz değer atadık. Normalde bu değer kişinin forma girdiği veri olacak. Bir de; `ereg( )-regi( )` fonksiyonlarında olduğu gibi bu iki fonksiyon arasındaki fark `eregi_replace( )` fonksiyonunun eşleştirmede büyük-küçük harf ayrımı yapmamasıdır.

SPLIT() KOMUTU

Bir değeri, değerin içerisinde aradığımız karakterleri baz alarak parçalara ayırır ve parçaları bir başka değişkene aktarır.

```
<?php
$deger = "Bir İki Üç";
$bul = " ";
$diz = split($bul, $deger);
foreach ($diz as $bas) {
print "$bas <br>";
}
?>
```

Örneğimizde `$deger` değişkeninin içinde `$bul` değişkeninin içeriği, yani boşluk karakteri arandı ve bu boşluk değeri baz alınarak `$deger` değişkeninin içeriği parçalara ayrıldı. Ve bu parçalar `$diz` değişkeninin

değeri olarak atandı. Sonunda da `foreach` fonksiyonu kullanılarak değer bastırıldı.

Örneğin ekran çıktısı tam istediğimiz gibi.

```
Bir
İki
Üç
```

SUBSTR() KOMUTU

Bir alfanümerik değişkene ait değerin tanımladığımız kısmını almak için kullanılır.

Kullanımı

```
substr ($degisken, seçim_sıra_değeri, karakter_sayısı);
```

şeklindedir.

Burada `$degisken` ile seçim yapacağımız değeri tutan değişkeni kasdediyoruz. `seçim_sıra_değeri` ile seçimin baştan itibaren kaçıncı karakterden başlayacağını belirliyoruz. `karakter_sayısı` ile de seçeceğimiz karakter sayısını belirliyoruz. Bu son parametre opsiyonel olup, istersek kullanmayabiliriz. Bu parametreyi kullanmadığımızda PHP değişkenin sonuna kadar alır.

```
<?php
$degisken = “123456789”;
$secilen = substr ($degisken, 3, 5);
echo $secilen;
?>
```

Bu örneğin çıktısı; `45678` şeklinde olacaktır. Çünkü programa 3. karakterden sonra başlamasını ve `5` adet karakter almasını söyledik. İstersek `seçim_sıra_değeri` olarak (`-`) negatif bir değer verebiliriz. Bu durumda seçim işlemi değişken değerinin sonundan başlayarak yapılır.

CHR() KOMUTU

ASCII değerini verdiğimiz karakteri gösterir.

Örnek

```
<?php
```

```
echo (chr(64));
?>
```

Bu kod ekranda, @ karakterini görüntüleyecektir.

ORD() KOMUTU

İstediğimiz karakterin ASCII değerini gösterir.

Örnek

```
<?php
echo (ord("X"));
?>
```

Bu kod ekranda, ASCII 88 değerini görüntüleyecektir.

PRINTF()-SPRIINTF() KOMUTU

Bu fonksiyonları, değişkenleri biçimlendirmek istediğimizde kullanacağız. **printf()** ile sonuçlar ekrana basılır. **sprintf()** ile sonuçlar değer olarak döndürülür. Dönen değeri tutan değişkeni de isterseniz ekrana bastırabilirsiniz. Bu fonksiyonlarla aşağıdaki parametreleri kullanabilirsiniz.

Parametre	Açıklama
b	Değişkenin tamsayı olarak işlem görmesi ve ikili sayı olarak dönmesi için kullanılır.
c	Değişkenin tamsayı olarak işlem görmesi ve sayının değerine karşılık gelen ASCII karakterinin dönmesi için kullanılır.
d	Değişkenin tamsayı olarak işlem görmesi ve ondalık sayı olarak dönmesi için kullanılır.
f	Değişkenin kesirli sayı olarak işlem görmesi ve kesirli sayı olarak dönmesi için kullanılır.
o	Değişkenin tamsayı sayı olarak işlem görmesi ve sekiz-tabanlı (octal) sayı olarak dönmesi için kullanılır.
s	Değişkenin alfanümerik olarak işlem görmesi ve alfanümerik olarak dönmesi için kullanılır.
x	Değişkenin tamsayı sayı olarak işlem görmesi ve 16 tabanlı (hexadecimal) sayı olarak dönmesi için kullanılır. (Küçük harf olarak)
X	Değişkenin tamsayı sayı olarak işlem görmesi ve 16 tabanlı (hexadecimal) sayı olarak dönmesi için kullanılır. (Büyük harf olarak)

Şimdi gelelim fonksiyonun kullanılış biçimine. Her iki fonksiyon da aynı biçimde kullanılır.

```
printf (“parametre”, $degisken1, $degisken2,....);
```

şeklindedir.

Yukarıda sıraladığımız parametreleri kullanırken parametrelerin önüne `%` (yüzde) işareti koymamız gerekmektedir. Ayrıca tabloda gösterdiğimiz parametrelere ek olarak kullanılabilecek birkaç özellik daha söz konusudur. Bunlardan (`‘x`) tek tırnak ve bir işaret şeklindeki kullanım, metni belirlediğimiz işaret (ben örnek olarak `x` yazdım, burada istediğiniz karakteri kullanabilirsiniz.) ile tamamlamaya yarar.

Biçimlendirmede `-` (eksi) işareti kullanmak metnin sola hizalanacağı anlamına gelir. Eğer `-` (eksi) işaretini kullanmazsak metin sağa hizalanır. Bir de `sayı nokta sayı` (örn: `20.20`) şeklinde bir biçimlendirme söz konusu. İlk sayı metnin azami uzunluğunu, ikinci sayı ise asgari uzunluğunu gösterir.

```
<?php
$metin = “ Metin “ ;
printf( “%’x-30.30s” , $metin);
?>
```

Örnekte $metin değişkeninin tuttuğu değer (Metin) 30 karakterden az olduğundan program x işareti ile 30 karaktere tamamlayacaktır ve ekran çıktısı: Metin xxxxxxxxxxxxxxxxxxxxxxx şeklinde olacaktır.

Bu konuya burada nokta koyarken bir noktaya parmak basalım. Burada anlattığımız fonksiyonları sadece formlarda değil içerik kontrolü ve biçimlendirmesi yapmamız gereken her yerde kullanacağız.

Düzenli İfadeler (Regular Expressions)

Bu işaretler ile bir verinin karakterlerinin dizilişine ve içerdiği karakterlere bakarak aradığımız metni bulmaya ya da mesela yukarıdaki form örneğimizde olduğu gibi verinin bizim istediğimiz veri olup olmadığını tespit etmemize yarar. Gerekli eşleştirmeyi ise fonksiyonlarla (örn:

ereg – eregi) yaparız. Bu girişten sonra, ifadeleri incelemeye başlayabiliriz.

^veri: Bu ifade, baş tarafı aranan alfanümerik veri ile başlayan değerleri bulur. Örneğin ^abc ifadesi iabc ile başlayan verileri bulur. Fakat içinde abc geçse bile başı abc ile başlamayan verileri bulamayız.

veri$: Bu ifade, sonu aranan veri ile biten değerleri bulur. Örneğin vyz$ ifadesi vyz ile biten verileri bulur. Ama vyz ifadesini sonunda başka bir karakter varsa mesala (.) işareti ya da bir başka harf ya da rakam varsa veriyi bulmaz.

^veri$: Bu ifade, başta ve sonda sıralanışı aynen arar.

Bunların yanı sıra bir de PHP ile karakter grupları oluşturarak arama işlemini daha da etkin bir hale getirebiliriz:

[a-z]	Bir küçük harfi arama
[A-Z]	Bir büyük harfi arama
[a-zA-Z]	Bir küçük ya da büyük harfi arama
[0-9]	Bir rakamı arama
[0-9\\-]	Bir rakamı, noktayı ya da kısa çizgi işaretini arama

Şimdi birlikte kullanım örnekleri yapalım. Mesela; ^[a–z] [0–9] $ ifadesi, başında a harfinden z harfine kadar herhangi bir **küçük** harf olan, sonunda ise 0 ila 9 arası bir rakamla biten verileri bulur.

Bir de ^ işaretinin köşeli parantez içinde kullanımına bakalım. Burada işaret ters anlam kazanır. Yani başında belirtilen karakterin olmadığı verileri bulur. Bir örnek vermek istersek ^[^0–9] [A–Z]$ ifadesi başında rakam **olmayan**, sonunda ise a harfinden z harfine kadar herhangi bir **büyük** harf olan ve iki karakterden oluşan verileri bulacaktır. Buradaki ilk ^ işareti ile başında, köşeli parantez içindeki [^] işareti ile de olmayan demiş oluyoruz.

Son olarak formumuzda da kullandığımız ^.+@.+\\..+$ ifadesine bakalım. Anlam olarak incelersek; .+ işareti verinin başına başında her hangi bir işaret olabilir demektir. Bu kullanım verinin boşluk ile başla-

masını engeller. Ardından gelen @ işareti, verinin bir @ işareti içermesini sağlıyor. Devamında yine herhangi bir veri olmasına izin veriyor ve ardından ters bölü ve nokta ile \. verinin içinde bir nokta işaret olması sağlanıyor. Verinin sonunda da yine .+ işareti ile verinin sonunda her hangi bir işaret olabilir denilmiş oluyor.

Bu ifade ile verinin gerçekten bir mail adresi formatında olup olmadığını kontrol etmiş oluyoruz. Bu ifade bize, girilen metnin bir e-mail adresi olduğunu kesin olarak garanti etmeye yetmez. Ancak birçok mail adresi olmayan ifadeyi bununla belirlemek mümkün olmaktadır.

PREG_MATCH()

Bu fonksiyon, bir string içinde aranılan metni bulmak için kullanılır.

Örnek

```
<?php
$metin= "bir berber bir berbere gel beraber bir berber dükkanı
açalım demiş.";
if (preg_match( "/b.*r/", $metin, $dizi))
print $dizi[0]
?>
```

Örneğimizde /b.*r/ düzenli ifadesi ile b harfiyle başlayan ve sonra herhangi bir harfle devam eden ve r harfiyle biten ifadeleri arıyoruz. Ve sonucu bir dizi değişkenine aktarıp yazdırıyoruz. Örneğimizin ekran çıktısı: "bir berber bir berbere gel beraber bir berber" şeklinde olacaktır.

Eğer düzenli ifade /b.*?r/ şeklinde olursa, yani ? (soru işareti) eklersek bu örneğin çıktısı sadece bir olur. Çünkü soru işareti aradığımız kriterlere uyan sadece ilk ifadeyi bulur.

Dosya ve Dizin İşlemleri

Bu kısımda, programımıza bir başka dosyayı dahil etmeyi, dosyalar hakkındaki çeşitli bilgileri sağlamayı, dosya oluşturma, dosya silme, dosya açma ve dosyaya ek yapma işlemlerini göreceğiz. Ayrıca dizin oluşturma, silme ve dizin içeriğini listeleme işlemlerine değineceğiz. Hadi bakalım başlayalım.

INCLUDE-REQUIRE KOMUTU

PHP programımıza bir başka dosyayı dahil etmek için `include` komutunu kullanacağız. Bu sayede programınıza mesela sık kullandığınız bir metni her belge yazışınızda eklemek yerine bu komutla çağırabilirsiniz. Tabi önce bu metni bir dosya olarak kaydetmiş olmalısınız. Aynı şekilde bir PHP dosyasındaki komutları da programınıza bu şekilde dahil edebilirsiniz.

Önce bir metin dosyası oluşturup içine herhangi bir metin girin ve mesela **metin.txt** adıyla kaydedin. Şimdi bu dosyayı PHP programımıza dahil edecek kodu yazalım.

```
<?php
include (“metin.txt”);
?>
```

Bu kodu kaydedip çalıştırırsanız ekrana **metin.txt** dosyasının içeriğini yazdıracaktır. Bu komutla isterseniz birden çok dosyayı da programınıza dahil edebilirsiniz.

Kullanım olarak `require` komutu da aynı şekilde kullanılır. Farkı, include komutunun sadece bir hata mesajı verip çalışmaya devam ettiği durumlarda `require` komutu **fatal error** hatası verecek ve programın çalışması durdurulacaktır. Programa dahil edilecek olan dosya bulunamadığında programın durdurulmasını istediğiniz hallerde `require` komutunu tercih etmelisiniz.

Şimdi de mesela **deneme.php** adında bir dosya oluşturalım. Dosyanın içeriği şöyle olsun:

```
<?php
$ismi = ‘Mustafa’;
$soy_ismi = ‘Aydemir’;
?>
```

Şimdi bir **index.php** dosyası oluşturup deneme.php dosyasını dahil edelim. Bunun için gereken program:

```
 <?php
echo “Adı: $ismi $soy_ismi”.”<br />”   ;
include ‘deneme.php’;
```

```
echo "Adı: $ismi $soy_ismi";
?>
```

Programın ekran çıktısı:

Adı:
Adı: Mustafa Aydemir

şeklindedir.

Programımıza deneme.php dosyasını dahil edince ikinci `echo` komutu **deneme.php** dosyasındaki değişkenlerin değerlerini de yazdırdı.

FILE_EXISTS

Bu fonksiyon bir dosyanın var olup olmadığını denetler. Eğer dosya varsa `TRUE` yoksa `FALSE` sonucunu verir.

Örnek

```
<?php
$dosya = 'deneme.php';

if (file_exists($dosya)) {
    echo "$dosya adında bir dosya var";
} else {
    echo "$dosya adında bir dosya yok";
}
?>
```

Örneğimiz ekrana, eğer böyle bir dosya varsa "deneme.php adında bir dosya var", böyle bir dosya yoksa "deneme.php adında bir dosya yok" sonucunu yazdırır.

IS_READABLE

Bir dosyanın okunabilir olup olmadığını denetler. Belirtilen dosya okunabilirse `TRUE` aksi takdirde `FALSE` değerini döndürür.

Örnek

```
<?php
$dosya = 'deneme.php';
if (is_readable($dosya)) {
```

```
    echo 'Dosya okunabilir';
} else {
    echo 'Dosya okunamaz';
}
?>
```

IS_WRITABLE

Bir dosyanın yazılabilir olup olmadığını denetler. Belirtilen dosya yazılabilirse `TRUE` aksi takdirde `FALSE` değerini döndürür.

Örnek

```
<?php
$dosya = 'deneme.php';
if (is_writable($dosya)) {
    echo 'Dosyaya yazılabilir';
} else {
    echo 'Dosyaya yazılamaz';
}
?>
```

FILESIZE

Bir dosyanın boyut bilgisini verir.

Örnek

```
<?php
$dosya = 'deneme.php';
echo $dosya . ': ' . filesize($dosya) . ' bytes';
?>
```

TOUCH

Bir dosyanın erişim ve değişiklik zamanını ayarlamak için kullanılır.

Örnek

```
<?php
$saat = time() - 3600;
if(touch('deneme', $saat)){
echo 'Dosyaya erişim zamanı başarıyla değiştirildi.';
}
?>
```

Bu örnekte `touch` fonksiyonunu `time` fonksiyonu ile beraber kullandık. time fonksiyonu ile dosyanın erişim zamanını `1 saat` önceye (`-3600`) ayarladık. Bu arada eğer belirttiğiniz dosya yoksa, program önce belirttiğiniz adda içi boş bir dosya oluşturacaktır. Dolayısı ile bu fonksiyonu **dosya oluşturmak** için de kullanabiliriz. Zaten fonksiyonu bu bölümde ele almamızın sebebi de bu.

UNLINK

Bu fonksiyonu bir dosyayı silmek istediğimizde kullanacağız.

Örnek

```
<?php
unlink (“deneme.php”);
print (“deneme dosyası başarıyla silindi!”);
?>
```

FOPEN()

Bu fonksiyonu istediğimiz bir dosyayı programımız için açmakta kullanacağız. Dosya açıldığında programımızın ne yapacağını da fonksiyonla beraber belirtmemiz gerekir. Bunu aşağıdaki tabloda gördüğünüz kiplerle yapacağız.

Kip	Açıklama
r	Dosyayı okumak için açar.
r+	Dosyayı okumak ve yazmak için açar.
w	Dosyayı başından başlayarak yazmak için açar. Dosya mevcutsa içeriğini siler. Dosya yoksa oluşturur.
w+	Dosyayı başından başlayarak okumak ve yazmak için açar. Dosya mevcutsa içeriğini siler. Dosya yoksa oluşturur.
x	Dosyayı oluşturur. Dosyayı başından başlayarak yazmak için açar. Dosya mevcutsa açılmaz ve fopen fonksiyonu FALSE sonucunu döndürür.
x+	Dosyayı oluşturur. Dosyayı başından başlayarak okumak ve yazmak için açar. Dosya mevcutsa açılmaz ve fopen fonksiyonu FALSE sonucunu döndürür.
a	Dosyayı sonundan başlayarak yazmak için açar. Dosya yoksa oluşturur.
a+	Dosyayı sonundan başlayarak okumak ve yazmak için açar. Dosya yoksa oluşturur.

```
<?php
$dosya = fopen("deneme.php", "a");
?>
```

Örnek kullanımda fonksiyonumuz dosyayı sonundan başlayarak yazmak için açacaktır. Eğer böyle bir dosya yoksa oluşturacaktır.

FCLOSE()

Bu fonksiyonu dosya kapatma işlemi için kullanacağız. Bu fonksiyona parametre olarak kapatacağımız dosyanın adını değil dosyanın tanıtıcısı olan değişkenin adını yazacağız.

Örnek

```
<?php
 $dosya= fopen('deneme.php', 'w+');
 fclose($dosya);
 ?>
```

FWRITE()

Bir dosyaya metin yazdırmak için bu fonksiyonu kullanacağız.

Örnek

```
<?php
$dosya= fopen('deneme.php', 'w');
fwrite($dosya, "Bu bir yazı");
fclose($dosya);
?>
```

Bu örnekte öğrendiğimiz son üç fonksiyon da mevcut. Önce fopen fonksiyonu ile dosyayı yazdırılmak üzere açtık. Ardından `fwrite` fonksiyonu ile istediğimiz veriyi yazdırdık. Son olarak da fclose fonksiyonu ile dosyayı kapadık. İlk aşamada eğer belirttiğimiz isimde bir dosya yoksa oluşturulacaktır.

FGETS()-FEOF()

Açılan bir dosyaya satır satır erişmek için kullanılır. Programımız satırları okurken dosyanın sonuna gelindiğinin belirtilmesi gerekir. Bunun için dosya sonunu saptayacak olan `feof()` fonksiyonunu kullanacağız.

feof fonksiyonu dosya sonuna ulaşıldığında TRUE değeri, aksi takdirde FALSE değeri döndürür.

Örnek

```
<?php
$dosya= fopen(“deneme.php”, “r”) or die (“Dosya açılamadı.”);
while (!feof($dosya)) {
$oku=fgets($dosya,1024);
echo “$oku”;
}
fclose ($dosya);
?>
```

Bu örnekteki while döngüsü, dosya sonuna erişilinceye kadar devam edecektir. Örneğimizde gördüğünüz or die() yapısı, dosyanın var olmaması, dosya izinlerinin okuma iznini içermemesi... vb bir sebeple fonksiyon görevini yapamadığında, fonksiyonu sonlandırmak için kullanılır.

FREAD()

Metni satır satır okutmak değil de, parça parça okutmak istediğimizde bu fonksiyonu kullanacağız.

Örnek

```
<?php
$dosya= fopen(“deneme.php”, “r”) or die (“Dosya açılamadı.”);
while (!feof($dosya)) {
$oku=fread($dosya,6);
echo “$oku<br />”;
}
fclose ($dosya);
?>
```

Bu örnekte veri, 6’şar baytlık parçalara ayrılarak okutulmuş oldu.

FGETC()

Bu fonksiyonun farkı verideki karakterleri tek tek okutur.

Örnek

```
<?php
$dosya= fopen(“deneme.php”, “r”) or die (“Dosya açılamadı.”);
```

```
while (!feof($dosya)) {
$oku=fgetc($dosya);
echo "$oku<br />";
}
fclose ($dosya);
?>
```

FLOCK()

Bu fonksiyon, bir işlem sürecinde dosyayı kilitleyerek, dosyaya bir başkası tarafından erişilip sürecin bir başka işlem süreciyle bozulmamasını sağlamış olur. Bu işlem için fonksiyonla beraber kullanmak istediğimiz kilit türünü de seçeceğiz. Bunlar 3 tanedir. `LOCK_SH`, diğer süreçlerin dosyayı okumasına müsaade eder fakat yazmalarını engeller. Dosya okurken kullanılır. `LOCK_EX`, diğer süreçlerin dosyayı hem okumasını hem de yazmasını engeller. Dosyaya veri yazarken kullanılır. `LOCK_UN` ise kilidi açar.

```
<?php
$dosya= fopen("deneme.php", "a") or die ("Dosya açılamadı.");
flock($dosya,LOCK_EX);
fwrite($dosya, "Bu bir yazı");
flock($dosya,LOCK_UN);
fclose ($dosya);
?>
```

Görüldüğü gibi fopen fonksiyonundan hemen sonra dosyayı kilitliyoruz. Yazma süreci bittikten sonra da kilidi açıyoruz.

OPENDIR()-READDIR()

Bir dizini listeleyebilmek için önce `opendir` fonksiyonu ile dizini açacağız ve `readdir` fonksiyonu ile de dizinin içindekileri okuyabileceğiz.

```
<?php
$dizin="dizin";
$ac= opendir($dizin) or die ("Dosya açılamadı.");
while ($diz=readdir($ac)){
echo $diz."<br />";
}
?>
```

Bu örnekle dizin adlı klasörümüzün içindekileri listelemiş olduk. Önce dizini `opendir` fonksiyonu ile açtık ve ardından `readdir` fonksiyonu ile listeledik.

MKDIR()-RMDIR()

Yeni bir dizin oluşturma işlemini php'de mkdir fonksiyonu ile yaparız.

Kullanımı

```
mkdir (“dizin_ismi”,0755)
```

şeklindedir.

Dizin ismiyle beraber dosya izinlerini de belirtiyoruz. `0755` ile okunabilir ama yazılamaz. Yazılabilir olması için `0777` değerini girmelisiniz.

Bir dizini silmek içinse `rmdir` fonksiyonunu kullanırız.

Kullanımı

```
rmdir (“dizin_ismi “);
```

şeklindedir.

Yönlendirme (Location ve Refresh)

PHP ile sayfa yönlendirme nasıl yapılır, bu başlık altında bu konuyu inceleyeceğiz. Öncelikle Dreamweaver yardımıyla aşağıdaki gibi basit bir form oluşturalım ve **index.php** adıyla kaydedelim...

```
 <!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0 Transitional//EN”
“http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd”>
 <html xmlns=”http://www.w3.org/1999/xhtml”>
 <head>
 <meta http-equiv=”Content-Type” content=”text/html; charset=utf-8” />
 <title>refresh komutu</title>
 </head>
 <body>
 <form action=”index2.php” method=”post” name=”sayi”>
   <label>
     <input name=”textfield” type=”text”  value=”Lütfen bir
sayı giriniz.” />
```

```
    </label>
    <label>
      <input type="submit" name="button"  value="Gönder" />
    </label>
  </form>
  </body>
  </html>
```

Ardından yeni bir belge açıp içine rastgele bir şeyler yazalım. Mesela; "Bu bir deneme sayfasıdır.", yazabiliriz. Ve dosyaya herhangi bir isim verip (örn. **deneme.php**) kaydedelim.

Şimdi son bir dosya daha oluşturacağız. Oluşturacağımız dosya ile yönlendirme işlemini gerçekleştireceğiz. Aşağıdaki kodları **index2.php** adıyla kaydedin. Çünkü **index.php** içinde action olarak bu dosya adını kullandık.

```
<?php
$sayi = $_POST["sayi"];
if ($sayi==1){
header("Location:deneme.php");
}else{
header("Refresh: 5; url=deneme.php");
}
?>
```

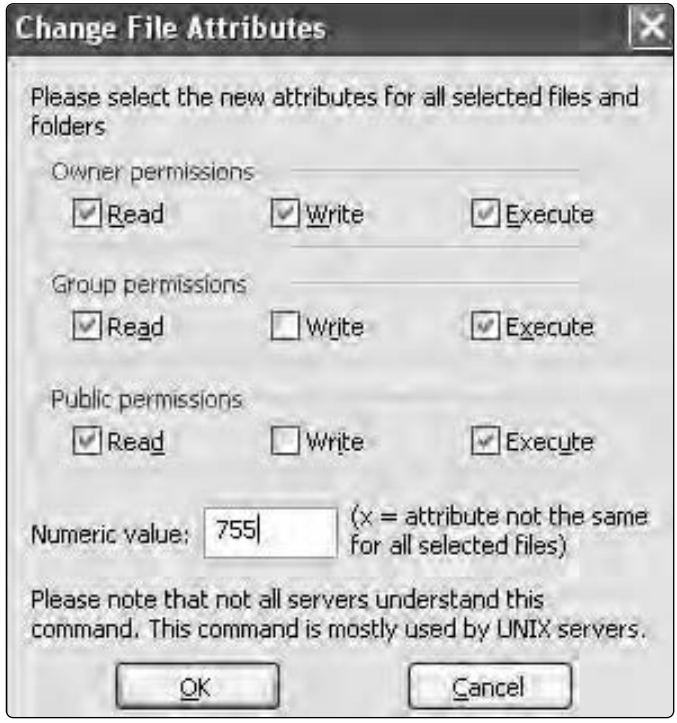
Programımızda gördüğünüz location ve refresh komutlarının ikisi de yönlendirme yaparlar. Farkı, location hemen yönlendirir, refresh ise belli bir süre beklettikten sonra yönlendirme yapar. Bu süreyi biz örneğimizde `5 saniye` olarak belirledik.

Şimdi bu üç dosyayı localhost'unuzda bir klasör oluşturup dosyanın içine kopyalayın. index.php dosyasını çağırıp forma 1 rakamını girerseniz deneme.php sayfasına hemen yönlendirilirsiniz. Başka bir sayı yazarsanız 5 saniye bekledikten sonra yönlendirilirsiniz.

DOSYA VE DİZİN İZİNLERİ (CHMOD)

Yukarıda dosya ve dizin oluşturma işlemi sırasında karşılaştığımız dosya ve dizin izinleri konusuna da biraz değinelim. **CHMOD** İngilizce

"Change Mode" kelimesinin kısaltmasıdır. Teknik bir terim olarak ise; sunucudaki dosyaların yazma, okuma ve çalıştırma izinlerini ifade eder. Chmod değerleri 3 haneli rakamlardan oluşur. İlk hane **Yönetici**, ikincisi **Grup**, üçüncüsü ise **Kullanıcı erişim** izinleridir.



Her bir hanede 1 ile 7 arası rakamlar kullanılır. Bu rakamların anlamı:

1 – Çalıştırma

2 – Yazma

3 – Hem yazma hem çalıştırma

4 – Okuma,

5 – Hem okuma hem çalıştırma

6 – Hem okuma hem yazma

7 – Hem okuma hem yazma hem çalıştırma.

Açıklamalarımızı örneklerle pekiştirecek olursak;

Örneğin `644` değerini ele alırsak; Yöneticiye 6, Gruplara 4 ve Kullanıcılara 4 değeri atanmış.

Yukarıdaki listemizden 6 ve 4 değerlerinin anlamlarına bakarsak, yöneticiye yazma, gruplara ve kullanıcılara sadece okuma yetkisi verdiğini görürüz. Tabi bu konunun **Unix/Linux tabanlı** sunucular için geçerli olduğunu da belirteyim.

VERİTABANI İŞLEMLERİ

Veritabanı programları, verileri depolayıp işleyen ve istenilen çıktıları üreten programlama dilleridir. Günümüz siteleri artık bir veritabanı etkileşimi ile çalışmak zorundadırlar. Aksi takdirde site ziyaretçilerinizin yaptıkları veri girişlerini tutmak, işlemek ve istenilen sonuçları üretmek mümkün olmayacaktır.

Mesela bir forma ya da ziyaretçi defterine girilen verileri almak ve sitede istenilen bir biçimde göstermek, bir alışveriş sitesinde alışveriş işlemi yaptırmak veya bir anketi üyelerinize doldurtup sonuçlarını yayınlamak ve daha bunlar gibi yüzlerce işlem yapılamayacaktır.

PHP, bir çok veritabanı sistemine uyumludur. Fakat genellikle PHP gibi açık kaynak bir yazılım olan MySQL ile kullanılır. Bu sistemle tam uyum içinde ve çok yüksek performansla çalışır.

Veritabanları verileri, içinde değişik veri türlerinde veri alanları barındırabilen tablolarda tutarlar. Tablolara eklenecek veriler farklı veri türlerinde olabilirler.

Şimdi de MySQL'de kullanılan bazı veri türlerine değinelim. Bunlar bizim için gerekli, çünkü hangi verinin hangi veri türünde saklanacağını bilmek demek MySQL'e veri ekleyebilmek demektir ki zaten amacımız da bu.

Veri Türü	Açıklama
INT	–2147483648 ile 2147483648 arası tamsayı.
TINYINT	–128 ile 127 arası tamsayı
VARCHAR(M)	1 ile 255 arası karakter.
CHAR(M)	1 ile 255 arası karakter.
TEXT	En fazla 65535 karakter metin
MEDIUMTEXT	En fazla 16777215 karakter metin
TIMESTAMP	YYYY–MM–DD–HH–MM–SS (Yıl+Ay+Gün+Saat+Dakika+Saniye) biçiminde zaman. 1 Ocak 1970 ile 2037 arası.
TIME	HH–MM–SS biçiminde zaman
DATE	1000–01–01 ile 9999–12–31 arası YYYY–MM–DD biçiminde tarih

MYSQL_CONNECT()

Veritabanına herhangi bir veri girebilmek için öncelikler veritabanı sistemi ile bir bağlantı gerçekleştirmemiz gerekir. İşte `mysql_connect()` fonksiyonu, PHP-MySQL arasındaki bağlantıyı sağlar.

Örnek

```
$baglan=mysql_connect (“localhost” , “root” , “1234”);
```

Fonksiyonumuzla **sunucu adı**, **veritabanı kullanıcı adı** ve v**eritabanı şifresini** belirtiyoruz. Veritabanı ile bağlantı sağlandığında burada oluşturduğumuz değişken (`$baglan`), veritabanından gelen ve veritabanına giden bilgilere aracılık edecektir.

MYSQL_SELECT_DB()

Veritabanı sistemi ile bağlantı sağlandığında, bu fonksiyonla hangi veritabanından veri çekeceğimizi belirleyeceğiz.

Kullanımı

```
mysql_select_db(veri tabanı adı, bağlantı adı);
```

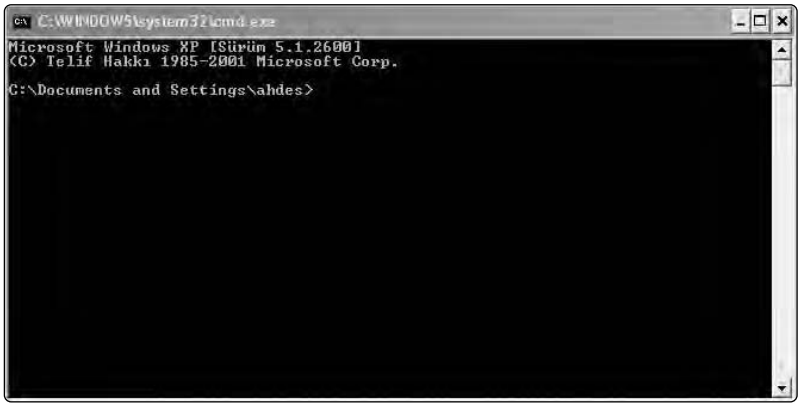
şeklindedir.

Buradaki bağlantı adı `mysql_connect()` fonksiyonundaki verileri tutan değişkendir. Şimdi bu iki fonksiyonu birlikte örnekle görelim.

```
<?php
$baglan=mysql_connect("localhost","root","");
mysql_select_db("deneme",$baglan);
?>
```

Eğer bilgisayarınızda oluşturacağınız veritabanına bir şifre vermedi iseniz, fonksiyonda bu alanı boş bırakmalısınız. İlk satır ile veritabanı sistemine bağlandık ve ikinci satırla da sistemdeki deneme isimli veritabanından veri alışverişi yapacağımızı belirtmiş olduk.

Tabi bu aşamada bize bir veritabanı lazım değil mi. O zaman hemen oluşturalım. Bu işlem için iki seçeneğiniz var. Birincisi bilgisayarınızda **Başlat/Çalıştır** penceresinden **cmd** komutunu çalıştırmak ve aşağıda gördüğünüz DOS ekranında veritabanı oluşturmanızı sağlayacak olan **MySQL komutlarını** yazmaktır.



Bu pencerede mysql dizini de (`mysql>`) iken, `create database <veri tabanı adı>;` şeklinde bir veritabanı oluşturabilirsiniz. Daha sonra use `<database adı>;` kalıbı ile veritabanını seçebilir ve create table `<tablo adı>;` komutu ile ve ilgili parametreleri kullanarak ilk tablonuzu oluşturabilirsiniz. Mesela isim, soy isim ve yaş bilgilerini içeren **personel** adında bir tablo oluşturalım.

```
create table personel (ismi VARCHAR(30), soyismi VARCHAR(30),
yasi INT ) ;
```

Tablo oluştururken veri alanlarına depolanacak bilginin türünü de (örn: VARCHAR, INT... vb) seçiyoruz. Artık veritabanımızda bir tablo var fakat içinde herhangi bir bilgi depolanmıyor. Şimdi de tablomuza veri girişi yapalım. Bu işlemi `insert into` komutuyla yapacağız.

```
insert into personel (ismi, soyismi,yasi) values ('Mustafa',
'Aydemir', '38');
```

Veritabanı oluşturmak ve gereli işlemleri yapmak için biz bu yöntemi kullanmayacağız. İkinci yöntemi kullanacağız. Yani işlemleri **phpMyAdmin** programı ile daha kolay ve hızlı bir biçimde yapacağız.

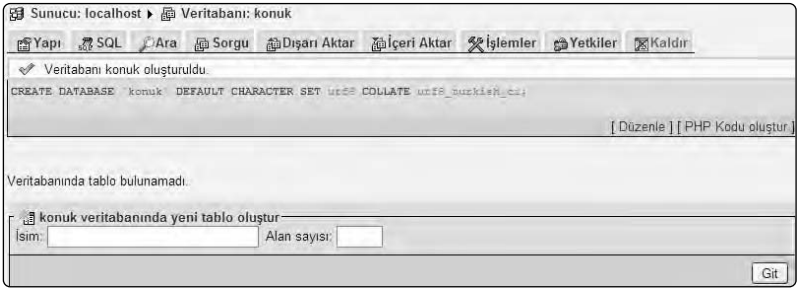
Önce **yerel sunucumuzu** (*WampServer*) çalıştıralım. Ardından hemen tarayıcımızı açıp adres çubuğuna **http://localhost/phpmyadmin/** yazarak **phpMyAdmin**'i başlatalım. Karşınıza aşağıdaki resimde gördüğünüz sayfa gelecektir.



Artık **phpMyAdmin** arayüzü karşımızda ve biz tüm işlemlerimizi bu arayüz sayesinde rahatlıkla yapabileceğiz. Arayüzün sol tarafında sunucumuzdaki mevcut veritabanlarının listesini görüyorsunuz.

Biz kendi veritabanımızı oluşturunca buradaki listede görebileceğiz. Sağ kısımda ise sunucu ve MySQL hakkındaki sürüm bilgileri yer alır.

Evet hiç vakit kaybetmeden ilk veritabanımızı oluşturalım. Bir ziyaretçi defteri için veritabanı hazırladığımızı varsayalım. Sayfanın orta kısmında bulunan **MySQL localhost** bölümündeki "Yeni veritabanı oluştur" kutucuğuna veritabanımızın adını girip (örn: `konuk`) **Oluştur** tuşuna tıklıyoruz. Evet işte bu kadar!



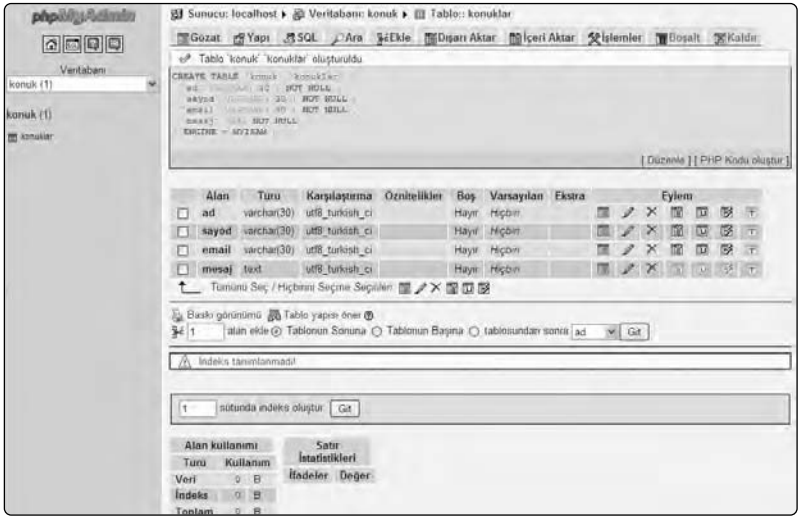
Artık bir veritabanımız var ve şu an resimdeki ikinci sayfadayız. Gördüğünüz gibi sayfanın başında veritabanımızın oluşturulduğu bilgisi yer alıyor ve sol tarafta artık veritabanımızın adını –`konuk(0)`– görebiliyoruz. Veritabanı adının yanındaki `0` veritabanımızda henüz hiçbir tablo olmadığı anlamına geliyor. O zaman ne duruyoruz, hemen oluşturalım!

"konuk veritabanında yeni tablo oluştur" kutucuğuna tablomuzun adını girelim. Mesela bu tablomuzda sitemiz ziyaretçilerine ait bilgilerin yer alacağını varsayarak tablomuzun adını `konuklar` olarak koyalım. Alan sayısı kutucuğuna ise tablomuzda yer alacak alanların sayısını gireceğiz. Mesela tablomuzda `ad`, `soyad`, `e-mail`, `mesaj` olmak üzere toplam 4 alan olsun. Dolayısıyla kutucuğa 4 rakamını girip sağ alttaki **Git** tuşu ile yolumuza devam edelim.



Bu sefer karşımıza tablomuzun içeriği geldi. Tabi boş olarak. Biz şu ana kadar bir veritabanı oluşturduk ve ardından da veritabanımızda, içinde 4 alan bulunan bir tablo oluşturduk. Şimdi sıra geldi tablomuza ait alanları oluşturmaya.

Alanlarımıza sırayla `ad`, `soyad`, `e-mail`, `mesaj` diyelim ve `ad`, `soyad`, `e-mail` için `VARCHAR` veri türünü, mesaj için de `TEXT` veri türünü seçelim. Türlerle ilgili tablodan bunların işlevlerine tekrar bakabilirsiniz. Ardından uzunluk değerlerini girerek üyelerin üyelik işlemleri sırasında bu alanlara girebilecekleri veri miktarını kısıtlayalım. Eğer tablomuza yeni bir alan eklememiz gerekirse sayfanın sağ alt kısmındaki kutucuğa ekleyeceğimiz alan sayısını girip **Git** tuşu ile bu işlemi gerçekleştirebilirsiniz. Biz yeni bir alan eklemeyeceğimiz için **Kaydet** tuşu ile devam edelim.



İşte yeni bir sayfadayız. Gördüğünüz gibi alanlarımıza veriler girilmiş durumda. Eğer bu işlemleri ilk yöntemle manuel olarak yani bizzat MySQL komutlarını girerek yapmak isteseydik şu an sayfanın üst kısmında gördüğünüz aşağıya alıntıladığım kodları yazmamız gerekirdi.

```
CREATE TABLE `konuk`.`konuklar` (
`ad` VARCHAR( 30 ) NOT NULL ,
`soyad` VARCHAR( 30 ) NOT NULL ,
`email` VARCHAR( 30 ) NOT NULL ,
`mesaj` TEXT NOT NULL
) ENGINE = MYISAM ;
```

Evet phpMyAdmin bu işi bizim için halletti. Yani biz verileri ekledik, phpMyAdmin ise bunları düzenleyerek gereken kodları ekleyip mysql programına takdim etti.

Gelelim sayfanın alt kısmına. Burada da tablomuza ait alanları görüyoruz. Alanların başlarındaki kutucukları tıklayıp **Eylemler** sütunundan kalem şeklindeki **Değiştir** tuşuyla alanlara ait düzenlemeler sayfasına geçerek alanlardaki verileri istediğiniz gibi değiştirebilirsiniz.

Evet geriye yapmamız gereken XHTML ile bir ziyaretçi defteri oluşturmak sonra da ziyaretçi defterindeki bilgileri veritabanına yazdıracak PHP programını yazmak. Gördüğünüz gibi geriye pek bir şey kalmamış durumda. Bu işlemleri birlikte yapacağız merak etmeyin. Ama öncesinde öğrenmemiz gereken birkaç şey daha var.

Önce ilginç bir şey yapalım ve süreci tersine çevirelim. Oluşturduğumuz veritabanının içini phpMyAdmin'den boşaltın. Yani konuklar tablosunu silin. Merak etmeyin birazdan geri gelecek! Şimdi aşağıdaki **baglan.php** adını verdiğimiz dosyayı oluşturun. (Tabi siz istediğiniz adı verebilirsiniz.)

```
<?php
$baglan=mysql_connect("localhost","root","");
mysql_select_db("konuk",$baglan);
?>
```

Bu dosya ile bağlantımızı yapacağız. Ardından mesela **veri.php** adında bir dosya daha oluşturun. Ve iki dosyayı da **www** klasörüne kopyalayın.

İçeriği şöyle olmalı:

```
<?php
include “baglan.php”;
mysql_query(“CREATE TABLE `konuk`.`konuklar` (
`ad` VARCHAR( 30 ) NOT NULL ,
`soyad` VARCHAR( 30 ) NOT NULL ,
`email` VARCHAR( 30 ) NOT NULL ,
`mesaj` TEXT NOT NULL
) ENGINE = MYISAM ;
“);
?>
```

Evet bu dosyada, daha önce öğrendiğimiz include komutu ile **baglan.php** dosyasını programımıza dahi ettik. Ardından `mysql_query` fonksiyonu ile daha önce phpMyAdmin den aldığımız MySQL komutlarını sorgulatarak veritabanımıza gönderdik. Demek ki neymiş? PHP komutlarıyla da veritabanında tablo oluşturabiliyormuşuz.

Evet demek ki phpMyAdmin’de tablomuzu oluşturduğumuzda sql komutlarını boşuna almamışız. Böylece sizi bunları tek tek yazma zahmetinden kurtarmış oldum. Bu arada çaktırmadan `mysql_query` fonksiyonunu da öğrenmiş olduk!

Burada aklınıza, bu işlemi iki ayrı dosyada yapmak zorunda mıyız? Şeklinde bir soru gelebilir. Tabi ki hayır. İşte aşağıda tek bir dosyayla işi bitireceğiz.

```
<?php
$baglan=mysql_connect(“localhost”,”root”,””);
mysql_select_db(“konuk”,$baglan);
$sql= “CREATE TABLE `konuk`.`konuklar` (
`ad` VARCHAR( 30 ) NOT NULL ,
`soyad` VARCHAR( 30 ) NOT NULL ,
`email` VARCHAR( 30 ) NOT NULL ,
`mesaj` TEXT NOT NULL
) ENGINE = MYISAM ;”;
$sonuc=mysql_query($sql,$baglan);
echo $sonuc;
?>
```

Yaptığımız şey aslında aynı. Farklı olarak `mysql_query` fonksiyonunu bir değişkene atayarak ekrana bastırdık. Bu dosyayı çalıştırdığınızda eğer işlem gerçekleşirse `TRUE` sonucunu döndüreceğinden ekrana `1` rakamını basacaktır. Eğer bir aksilik olurda işlem gerçekleşmezse ekranda bir tane `0` rakamı göreceksiniz! Evet şimdi de aşağıdaki kodu inceleyin.

```
<?php
$baglan=mysql_connect("localhost","root","");
mysql_select_db("konuk",$baglan);
$sql= "CREATE TABLE `konuk`.`konuklar` (
`ad` VARCHAR( 30 ) NOT NULL ,
`soyad` VARCHAR( 30 ) NOT NULL ,
`email` VARCHAR( 30 ) NOT NULL ,
`mesaj` TEXT NOT NULL
) ENGINE = MYISAM ;";
$sonuc=mysql_query($sql,$baglan) or die ("Veri tabanina
baglanilamadi! Hata: ". mysql_error());
echo $sonuc;
?>
```

Bizim kullandığımızla aynı sadece daha önce öğrendiğimiz `or die` yapısı ve `mysql_error()` fonksiyonu eklendi. Bu fonksiyon eğer bir hata oluşursa bunu rapor eder. Şimdi bu programı istediğiniz bir isimle kaydedip çalıştırın.

Ekrana:

```
Sorgu başarısız. Hata: Table 'konuklar' already exists
```

şeklinde bir hata mesajı gelecektir. Zaten bu tablo veritabanımızda mevcut olduğundan tekrar aynı isimde tablo oluşturmaya kalkınca bu hatayı aldık. Eğer bu fonksiyonu kullanmasak bir an ne olup bittiğini anlayamayabilirdik değil mi? Evet ikinci kez satır arasında bir fonksiyon daha öğrenmiş olduk.

Burada bir soru soralım. Merak etmeyin kolay bir soru. Bu oluşturduğumuz bomboş veritabanına veriler nereden gelecek? Evet bildiniz, ziyaretçi defterinden. Yani bir formdan. Ziyaretçiler formu doldurup gönder dediklerinde oluşturduğumuz tablonun veri alanlarına kaydedilecek. Yani biz konuyu öğrenmek için böyle yaptık. Ama siz, sitenize pek ge-

len giden olmazsa, bu metotla kendiniz birkaç görüş ekleyerek PHP bilgilerinizi pratik etmiş olursunuz.

Şaka bir yana aynı hızla konumuza devam edelim ve bir ziyaretçi formu oluşturup veritabanına bilgileri oradan göndermeye (yani normalde olacağı şekliyle) çalışalım.

Burada uzun uzun form hazırlamaya uğraşmadan aşağıdaki formu kullanalım. Malum konumuz XHTML değil PHP. Aslında burada ad ve soyadı tek bir alana da tanımlayabilirdik. Biz ayrı yaptığımızdan ziyaretçi defterinde de ayrı alanlar oluşturacağız.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>İletişim Formu</title>
</head>

<body>
<form id="form" name="form" method="post" action="konuk.php" >
  <table width="400" border="0" align="left">
    <tr>
      <td width="34%">Adınız:</td>
      <td width="66%"><input name="ad" type="text" /></td>
    </tr>
    <tr>
      <td>Soyadınız :</td>
      <td><input name="soyad" type="text" /></td>
    </tr>
    <tr>
      <td>Email Adresiniz :</td>
      <td><input name="email" type="text" /></td>
    </tr>
    <tr>
      <td>Mesajınız :</td>
      <td><textarea name="mesaj" cols="21" rows="10">Lütfen
Mesajınızı Giriniz...</textarea></td>
    </tr>
```

```
    <tr>
      <td><input name="" type="submit" value="Gönder" /></td>

    </tr>
  </table>
</form>
</body>
</html>
```

Bu dosyayı **ziyaretçi_defteri.html** olarak kaydedin. PHP dosyamızın adı da **konuk.php** olacak. (Çünkü; `action="konuk.php"` yaptık)

```
<?php
$baglan=mysql_connect("localhost","root","") or die ("Veritabanına
     bağlanılamadı");
mysql_select_db("konuk",$baglan);

$ad = $_POST['ad'];
$soyad = $_POST['soyad'];
$email = $_POST['email'];
$mesaj= $_POST['mesaj'];

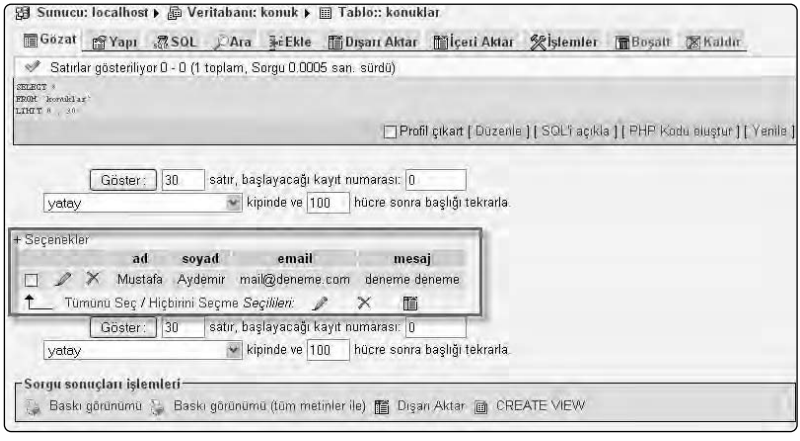
$sorgu=mysql_query("insert into konuklar (ad, soyad, email,
mesaj) values ('$ad','$soyad','$email','$mesaj')")or die
("Veri Yazdırılamadı Hata: ". mysql_error());
if ($sorgu){
echo "Kayıt Başarıyla Eklendi.";
}else{
echo "Kayıt Eklenemedi.";
}
?>
```

Evet **konuk.php** dosyasında önce veritabanı bağlantılarımızı yaptık Ardından aşağıdaki kısımla form alanındaki (`ziyaretçi_defteri.html`) verileri aldık.

```
$ad = $_POST['ad'];
$soyad = $_POST['soyad'];
$email = $_POST['email'];
$mesaj= $_POST['mesaj'];
```

Burada `$_POST` metoduyla aldığımız değerler, formdaki `name="ad"` ve diğer değerleridir. Peşi sıra alttaki kısımla veritabanı sorgumuzu çalıştırarak formdan gelen verileri veritabanına gönderdik.

```
$sorgu=mysql_query("insert into konuklar (ad, soyad, email,
mesaj) values
  ('$ad','$soyad','$email','$mesaj')")or die ("Veri Yazdırılamadı
Hata: ". mysql_error());
```



Ve resimde örneğini gördüğünüz gibi ziyaretçi defterini doldurup Gönder dediğinizde veritabanındaki konuklar tablosuna verilerin yazıldığını göreceksiniz.

Son olarak ziyaretçi defterine yazılan mesajları görüntülemek için ne yapacağız, ona bakalım.

```
$sorgu=mysql_query ("SELECT * FROM konuklar");

while ($veriler=mysql_fetch_array($sorgu)) {
echo "Adı: " . $veriler['ad'] . "<br />Soyadı: " . $veriler['soyad']
. "<br />Mail: ". $veriler['mail'] . "<br />Mesaj: ".$veriler['mesaj']
. "<br />";
}
```

Bunun için yukarıdaki örnekte olduğu gibi bir sorgu işlemi gerçekleştiriyoruz. Sorgudaki (`"SELECT * FROM konuklar"`) kısmı konuklar tablomuzdaki tüm verileri (`*`) sorgulamamızı sağlıyor.

Ardından bir `while` döngüsü ile sonuçları ekrana yazdırıyoruz.

Bunun için `mysql_fetch_array` fonksiyonunu kullanacağız. Bu fonksiyonla sorgu değişkenindeki verileri, `veriler` adlı bir değişkene aktarıyoruz. Peki neden aktarıyoruz. Tabiki ekrana yazdırmak için.

Burada isterseniz

```
(“SELECT * FROM konuklar”);
```

yerine

```
(“SELECT * FROM konuklar where ad = ‘Mustafa’ “);
```

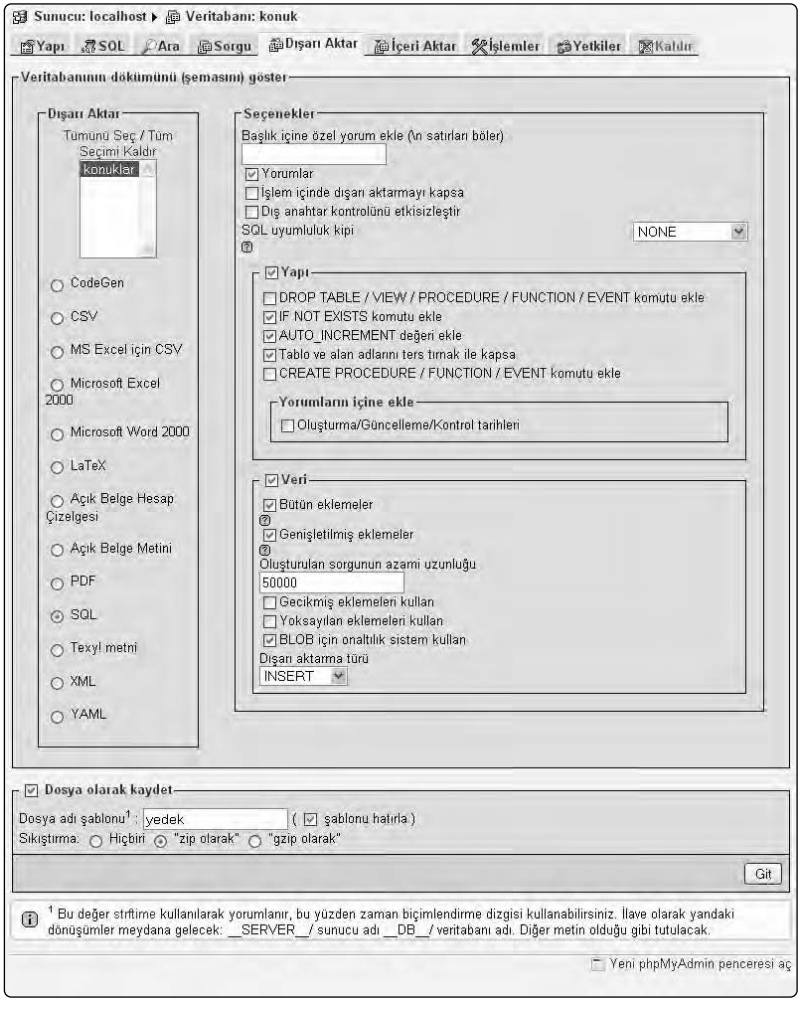
şeklinde şartlı sorgulama yaparak sadece mesela adı `Mustafa` olanların listesini alabilirsiniz.

Şimdi kodların başına veritabanı bağlantı fonksiyonlarını da ekleyip istediğiniz bir adla kaydedip deneyin. Ya da bağlantı fonksiyonlarını ayrı bir dosya şeklinde hazırlayıp `include` fonksiyonu ile dosyanıza dahil edebilirsiniz. Dosyayı çalıştırdığınızda veritabanınızdaki kayıtları göreceksiniz.

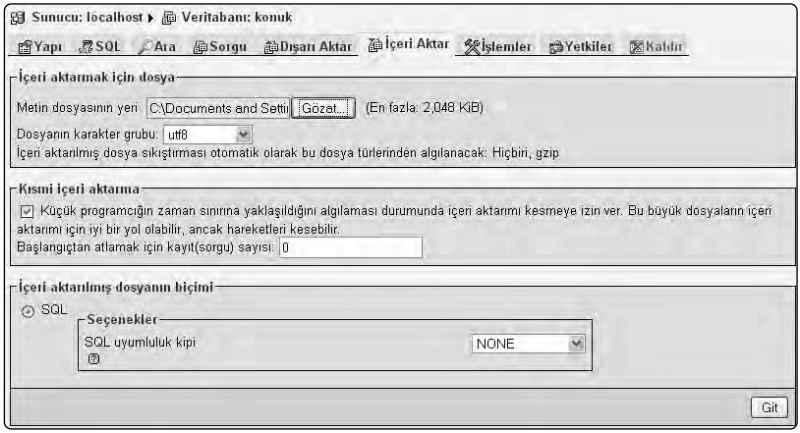
phpMyAdmin İle Yedek Alma ve Yedeği Geri Yükleme

Sitenizi kurdunuz ve hedef kitlenizle buluştunuz. Fakat bu yapmanız gereken işlerin bittiği anlamına gelmiyor. Rutin olarak yapmanız gereken işlemlerden birisi de belli aralıklarla sitenizin yedeğini almak olacaktır. Böylelikle doğabilecek her türlü probleme karşı site içeriğinizi sağlama almış olacaksınız.

Şimdi yedek alma yöntemlerinden biri olan **phpMyAdmin ile Yedek Alma** yöntemini göreceğiz. Ardından da aldığımız yedeği gerektiğinde nasıl geri yükleyebileceğimize bakacağız. Bu işlemler phpMyAdmin ile gerçekten çok kolaydır. O zaman hemen başlayalım ve örnek olarak localhost’umuzda oluşturduğumuz ve anlatımlarımızda kullandığımız konuk adlı veritabanımızın yedeğini bir dosya olarak alalım.



phpMyAdmin'i açın ve üstteki menülerden **Dışarı Aktar** menüsüne tıklayın. Karşınıza resimdeki sayfa gelecektir. Sayfanın en alt kısmında dosya olarak kaydet bölümünde yedeğinize birr dosya ismi belirleyin ve sıkıştırma seçeneklerinden "zip olarak" yazan radyo düğmesine tıklayın. Sayfanın sağ alt kısmında bulunan **Git** tuşuna bastığınızda yedeğini sıkıştırılmış bir SQL dosyası olarak indirilecektir.



Şimdi SQL yedek dosyanızı zip'ten çıkarıp inceleyebilirsiniz. Yedeğinizi geri yüklemeniz gerektiğinde yine phpMyAdmin'den bu sefer **İçeri Aktar** menüsüne tıklayın.

Karşınıza resimdeki sayfa gelecektir. Buradaki **Gözat** tuşu ile programa bilgisayarınızdaki sql yedek dosyasının yerini gösterin. Ve yine sayfanın sağ alt kısmında bulunan **Git** tuşuna tıkladığınızda yedeğiniz geri yüklenecektir.

Peki localhostta değil de gerçek bir hostta yani bir uzak sunucuda yedek alırken phpMyAdmin'e nasıl ulaşacağız. Günümüzde sunucu yönetimleri kişilerin host hesaplarını etkin bir şekilde kontrol edebilmeleri ve yönetebilmeleri için sunucu yönetim yazılımları kullanmaktadırlar. Bunlardan en çok kullanılanları **cPanel** ve **Plesk** yazılımlarıdır.

İşte önce hesabımızla bu sunucu kontrol panellerine erişip oradan da phpMyAdmin'e erişeceğiz. Kitabımız **CMS Sistemleri** bölümünde **cPanel**'de veritabanı oluşturma konusu anlatılmıştır. O bölümü okuduğunuzda bu konuda daha detaylı bilgi sahibi olmuş olacaksınız.

Oturum Yönetimi (Session)

Sitenizin bir üyesi, kullanıcı adı ve şifresini yazarak sitenize giriş yaptığında, sadece üyelerin görebileceği her sayfaya ulaşırken her seferinde tekrar üye girişi yapmaz. Siteye geldiğinde oturum açar, giderken ka-

patır ve gider. Bu iki eylem arasındaki tüm işlemleri üye sıfatıyla yapar. İşte, bir girişle üyelere mahsus tüm sayfaları görebilmemizi sağlayan session fonksiyonudur.

Oturumu Açmak

Oturumu açmak için `session_start()` fonksiyonunu kullanırız. Bu fonksiyonu kullanırken dikkat edeceğimiz en önemli husus, fonksiyonu sayfanın en başında kullanmaktır. Aksi takdirde oturum açma işlemi gerçekleşmeyecek ve siz bir hata mesajı ile karşılaşacaksınız.

Eğer belgenizin yapısı gereği fonksiyonu en başta kullanmak istemiyorsanız belgenin en başında `ob_start()` fonksiyonunu kullanabilirsiniz. Bu fonksiyon sayesinde `session_start()` fonksiyonunu belgenizin herhangi bir yerinde kullanabilirsiniz. Oturumu kapatmak içinse `session_destroy()` fonksiyonu kullanılır.

Şimdi konuyu pekiştirme adına session kullanarak bir üyelik sistemi yapalım. Öncelikle veritabanı bağlantısı için gereken kodları her seferinde tekrar yazmak yerine **ayar.php** adında bir dosya oluşturalım ve diğer dosyalarımızda bu dosyayı çağıralım.

ayar.php

```
<?php
$baglan=mysql_connect("localhost","root","") or die  ↓
("Veritabanına  Bağlanılamadı");
mysql_select_db("uyeler",$baglan) or die ("Veritabanı Seçilemedi");
?>
```

Veritabanı bağlantı dosyamız hazır olduğuna göre bir sonraki aşamaya geçebiliriz. Şimdi sitemizi ziyaret edenlerin sitemize üye olabilmeleri için dolduracakları formu hazırlayalım. Formumuzun adına da mesela **uyelik_formu.html** diyelim. İşte formumuzun kodları.

uyelik_formu.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Üyelik Formu</title>
</head>
<body>
<h2>Üyelik Formu</h2>
<form method="post" action="uyelik_islemleri.php">
Kullanıcı Adınız: <input type="text" name="kullanici"><br />
Şifreniz: <input type="password" name="sifre"><br />
<input type="submit" value="Üye Ol">
</form>
</body>
</html>
```

Şimdi formda gördüğünüz **uyelik_islemleri.php** dosyasını oluşturarak formdan gelen verileri veritabanına yazdıralım.

uyelik_islemleri.php

```
<?php
include("ayar.php");
$kullanici=$_POST['kullanici'];
$sifre=$_POST['sifre'];

if ($kullanici=="" || $sifre=="") {
echo "Lütfen formu tam olarak doldurunuz.";
}else{
$sorgula=mysql_query("insert into uyeler (kullanici, sifre)
values ('$kullanici', '$sifre')");
if ($sorgula) {
header("location: uye_giris_formu.html");
}else{
echo "Üyelik işlemleriniz başarıyla tamamlanmıştır. Teşekkür ederiz.";
}
}
?>
```

Yaptığımız `if` sorgusu ile üyelik formundaki alanların tam olarak doldurulmasını sağlayıp, verileri `uyeler` adlı tablomuza yazdırıyoruz. Sonrasında ise yeni üyemizi üye girişi yapacağı **uye_giris_formu.html** sayfasına yönlendiriyoruz. Şimdi sıra üye giriş formunu hazırlamaya geldi.

uye_giris_formu.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Üyelik Formu</title>
</head>
<body>
<h2>Giriş Formu</h2>
<form method="post" action="uye_giris_islemleri.php">
Kullanıcı Adı: <input type="text" name="kullanici"><br />
Şifre: <input type="password" name="sifre"><br />
<input type="submit" value="Giriş">
</form>
</body>
</html>
```

Formumuzu yukarıdaki gibi oluşturduk. Bir sonraki aşama olarak forma girilen üyelik bilgilerini, üyenin kayıt olurken belirlediği bilgilerle karşılaştırmak ve ona göre üyelik oturumu açmak için **uye_giris_islemleri.php** adlı dosyamızı oluşturacağız.

uye_giris_islemleri.php

```
<?php
session_start();
include("ayar.php");
$kullanici=$_POST['kullanici'];
$sifre=$_POST['sifre'];

if($sifre == "" || $kullanici == "") {
echo "Lütfen bilgilerinizi tam olarak giriniz.";
}else{
$sorgula=mysql_fetch_array(mysql_query("select * from uyeler ↓
where kullanici='$kullanici' and sifre='$sifre'"));
if (($kullanici==$sorgula['kullanici'] && ↓
$sifre==$sorgula['sifre'])) {
$_SESSION['kullanici']=$_POST['kullanici'];
$_SESSION['sifre']=$_POST['sifre'];
```

```
header("location: anasayfa.php");
}else{
echo "Lütfen üyelik bilgileriniz kontrol ederek tekrar giriniz.";
}
}
?>
```

Evet, oturumu açtık ve üyeyi sitemizin ana sayfasına yönlendirdik.

anasayfa.php

```
<?php
session_start();
if(!isset($_SESSION['kullanici'])) {
echo "Lütfen Giriş Yapınız.";
}else{
echo "Hoşgeldiniz Sayın: " . $_SESSION['kullanici']. "<br />";
echo "Çıkış yapmak için lütfen <a href='cikis.php'> tıklayın.</a>";
}
?>
```

Bu kodlarla oturumu anasayfaya taşımış olduk. En nihayetinde üye siteden çıkmak istediğinde linki tıklayarak **cikis.php** sayfasına gidecek ve oturum sonlandırılacaktır.

cikis.php

```
?php
session_start();
if(($_SESSION["kullanici"]) and ($_SESSION["sifre"])) {
session_destroy();
}
echo "Yine bekleriz";
?>
```

PHP VE XML

PHP 5 sürümü ile birlikte gelen **SimpleXML** eklentisi sayesinde PHP ile XML dosyalarını okutabilir ve dosyalara veri yazdırabilirsiniz. Anlatımda aşağıdaki **ornek.xml** adını verdiğim XML dosyasını kullanacağız.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<kitaplar>
```

```
   <kitap>
          <isim>Flash CS4</isim>
          <yazar>Uğur GELİŞKEN</yazar>
          <sayfa>416</sayfa>
          <ebat>19,5x24</ebat>
   </kitap>
</kitaplar>
```

Şimdi örneğimizi inceleyelim İşte örnek:

```
<?php
if (file_exists('ornek.xml')) {
   $xml=simplexml_load_file('ornek.xml');
}else{
echo "Dosya Bulunamadı";
exit ();
}
echo "XML Dosyası Bulundu.";
?>
```

Örnekte gördüğünüz `file_exists` fonksiyonu, **örnek.xml** dosyasının var olup olmadığını kontrol eder. `simplexml_load_file` fonksiyonu ise **örnek.xml** içindeki verileri oluşturduğumuz `$xml` değişkenine aktarır. Artık veriler PHP'de kullanılabilir hale gelmiş demektir. Öz itibari ile XML verilerini çekmenin mantığı budur.

Tabi bizim esas derdimiz dosyayı bulmak değil dosyanın içindeki verileri almak. Onun için bir sonraki aşama ile devam edelim.

```
<?php
if (file_exists('ornek.xml')) {
   $xml=simplexml_load_file('ornek.xml');
}else{
echo "Dosya Bulunamadı";
exit ();
}
$x = simplexml_load_file("ornek.xml");
$a = $x->kitap[0]->isim ."<br />";
$b = $x->kitap[0]->yazar."<br />" ;
$c = $x->kitap[0]->sayfa."<br />" ;
$d = $x->kitap[0]->ebat;
```

```
echo $a . $b.$c . $d ;
?>
```

Örneğimizin çıktısı :

Flash CS4
Uğur GELİŞKEN
416
19,5x24

Şimdi de aşağıdaki örneği inceleyin. Bu örnekte `addChild()` ile XML dosyamıza yeni öğeler ekleyeceğiz.

```
<?php
$xml=simplexml_load_file('ornek.xml');
$kitap=$xml->addChild('kitap');
$kitap->addChild('isim','C# ile ASP.NET 3.5');
$kitap->addChild('yazar','Zafer Demirkol');
$kitap->addChild('sayfa','608');
$kitap->addChild('ebat','19x24');
$xml->asXML('ornek.xml');
?>
```

Dosyayı çalıştırdıktan sonra **ornek.xml** dosyasına bakarsanız yeni verilerin eklendiğiniz göreceksiniz. Verileri dosyaya yazdırmak yerine dilerseniz programın son satırını;

```
echo $xml->asXML();
```

şeklinde değiştirerek ekrana da yazdırabilirsiniz.

7

JAVASCRIPT

JavaScript'in Yapısı 346
Belirleyiciler (Identifier) 348
Anahtar Kelimeler (Keyword) 349
Değerler (Literals) 349
Değişkenler (Variable) 350
Dizi-Değişkenler (Array) 351
Uygulama: Bilgi Giriş Penceresi 353
Kontrol Yapıları 354
Operatörler 354
if (eğer) Deyimi 357
if... Else (eğer... diğer) Deyimi 357
if else if 358
Döngüler 360
While Deyimi 360
Yuvalanmış Döngü 362
Continue 363
Fonksiyonlar 364
Nesneler 365
JavaScript Hazır Nesneleri 367
Nesnelere Ait Özellikler 368
Nesnelere Ait Metotlar 370
Olay Yöneticileri (Event Handlers) 372
onLoad-onUnLoad 372
onSubmit Olayı 373
onMouseOver Olayı 374
onClick Olayı 374
onFocus-onBlur 376
onChange Olayı 376

JavaScript, **Netscape** firması tarafından üretilmiştir. Web sitenizin görünüm olarak zenginleştirilmesi, dinamik yapısının güçlendirilmesi ve işlevselliğinin artırılması için kullanılan **istemci taraflı** (*client-side script*) bir dildir. JavaScript az sonra öğrenmeye başlayacağımız dinamik işlevleri site ziyaretçilerinizin bilgisayarlarında çalıştırır. Zaten istemci taraflı dillerin mantığı da budur.

JavaScript ile yapacağınız işleri, sunucu taraflı bir dille mesela PHP ile yapmak istediğinizde, sunucu taraflı dillerde programlar Web Server'da çalıştığı için, bu durum sunucunuza ek bir yük bindirecektir. Sunucu taraflı dillerin tüm işlevlerini JavaScript ile yapabileceğinizi kastetmiyorum. Burada anlatmaya çalıştığım husus, PHP ile yapabileceğiniz işleri JavaScript ile yapmanın sunucuya daha az yük bindireceğinden ve sayfalarınızın daha hızlı açılmasını sağlayacağıdır. JavaScript kodları HTML etiketleri içine gömülerek yazılır ve tarayıcılar tarafından yorumlanır. Dolayısıyla HTML'de olduğu gibi JavaScript ile de farklı tarayıcılarda farklı sonuçlar alma ihtimali söz konusudur. Tabi JavaScript kodlarını ayrı bir dosya olarak da tutabilirsiniz. Bu takdirde dosya uzantısı **.js** olmalıdır.

JavaScript'in Yapisi

JavaScript kodları HTML sayfalarına gömülür ve sayfa içinde kullanmak için `script` etiketinden yararlanılır.

Kullanımı

```
<script type="text/javascript">
//JavaScript kodları buraya gelecek.
</script>
```

Eğer harici bir sayfa kullanacaksak, söz konusu harici JavaScript dosyasını belgemize eklemek için aşağıdaki kalıbı kullanırız.

```
<script src="dosyaismi.js" type="text/javascript"></script>
```

Bu kalıp head etiketleri arasına yazılır. Şimdi isterseniz ilk JavaScript çalışmamızı yapalım.

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xlmns="http://www.w3org/1999/xhtml">
<head>
<script type="text/javascript">
document.writeln ("MERHABA JAVASCRİPT BEN GELDİM");
</script>
</head>
<body>
</body>
</html>
```

Bu kod ile ekrana;

```
MERHABA JAVASCRİPT BEN GELDİM
```

yazdırmış olduk. Kodda geçen `document`, üzerine bilgi yazılacak olan nesneyi ifade eder. `Writeln` metodu ise, nesneye string (karakter dizisi) bir veri yazdırmak için kullanılır.

Burada `Writeln` yerine `Write` kullanabilirsiniz. Farkı, `Writeln` metodu, yazma işlemini yeni bir satırbaşından itibaren başlatır. `Write` ise bir önceki adımda hangi satırda kalınmışsa o satırdan devam eder.

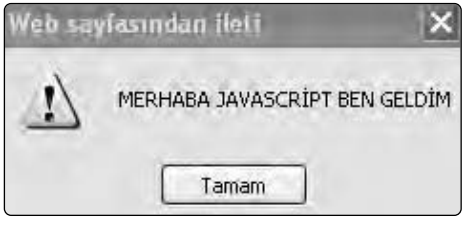
Her bir JavaScript satırı **;** (noktalı virgülle) bitirilir. Ve stringler " " (tırnak) içine alınarak yazılır. Burada istersek HTML etiketlerini de kullanabiliriz.

Şöyle ki:

```
<script type="text/javascript">
document.writeln (<h1>MERHABA JAVASCRİPT BEN GELDİM</h1>");
</script>
```

Şimdi gelin bir uygulama daha yapalım. Bu sefer `window` nesnesini ve `alert` metodunu kullanacağız. Bu uygulamada amacımız ekranda bir bilgi penceresi açtırmak.

```
 <!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
 <html xlmns="http://www.w3org/1999/xhtml">
 <head>
 <script type="text/javascript">
 window.alert ("MERHABA JAVASCRİPT BEN GELDİM");
 </script>
 </head>
 <body>
 </body>
 </html>
```



Uygulamayı çalıştırdığınızda resimdeki gibi bir bilgi penceresi açılacaktır.

Dikkat ettiyseniz, yazdığımız programlarda hep nesnelerden ve metotlardan bahsediyoruz. Çünkü JavaScript, **Object-oriented** (*Nesne-Yönelimli*) bir dildir ve **DOM** (**D***ocument* **O***bject* **M***odel*) denilen kurallar içinde hareket etmek zorundadır. Şimdi bunları açıklayalım.

Malumunuz olduğu üzere, güncel yaşamımızda da bir çok işimizi nesneler aracılığıyla hallederiz. Mesela yemek yemek için kaşık nesnesini kullanırız. Ekmekleri dilimlemek için bıçak nesnesini kullanırız. Her nesnenin kendine has özellikleri vardır. Mesela bıçak kesebilme özelliğine sahiptir ve biz bıçağı bu özelliğe sahip olduğu için kullanırız. Bizim yaptığımız ekmeği dilimle işi, olayı (*event*); bıçağın kesme eylemi, metodu; bıçağın kesebilme özelliğine sahip olması, özelliği; bıçağın kendisi ise nesneyi temsil eder.

Bu örneği JavaScript diline uyarlamak için örneğimizde kullandığımız kod satırından yararlanalım.

```
document.writeln (“MERHABA JAVASCRİPT BEN GELDİM”);
```

Burada kullandığımız writeln, `document` nesnesine ait bir metoddur. Biz bu metodu kullanarak bir iş (yazdırma eylemi) yaptırıyoruz. Bu şekilde, nesneleri ve nesnelere ait özellikleri hedef alan yapıdaki programlara **Object Oriented Program** (**OOP***: Nesne Yönelimli Program*) adı verilmektedir.

Document Object Model (**DOM:** *Belge Nesnesi Modeli*) ise olaylara (event) (örn: ziyaretçinin sayfadaki bir tuşa tıklaması) ve metodlara (method) standart getiren bir kurallar bütünüdür. Tarayıcılar, her bir internet sayfasını bir belge, belgede yer alan öğeleri de (örn: form, resim, metin...vd) nesne olarak kabul ederler. İşte biz JavaScript dilini kullanarak bu nesnelerin bazı özelliklerini düzenleyeceğiz.

Şimdi kaldığımız yerden JavaScript dilinin yapısını öğrenmeye devan edelim. Nesneler konusunun detaylarına ilerleyen sayfalarda değineceğiz.

BELİRLEYİCİLER (IDENTIFIER)

JavaScript dilinde kullanılan nesneleri, değişkenleri ve metodları oluşturmak için kullanılan kelimelere **belirleyici** (*identifier*) denilir. Belirleyiciler oluşturulurken bir harfle ya da _ (alt çizgi) ile başlamak zorundadırlar. Bir başka işaretle ya da rakamla başlayamazlar. Diğer işaretler ve rakamlar ilk sırada yer almamak şartıyla belirleyicilerle kullanılabilirler. Ayrıca belirleyicileri oluşturan kelimeler içinde boşluk bulunamaz. Bir de muhtemel sorunları baştan önlemek adına belirleyicilerde

Türkçe karakter kullanmamanızı öneririm.

JavaScript büyük-küçük harf ayrımına duyarlıdır. Yani "KÜÇÜK" kelimesi ile "küçük" kelimesi JavaScript için aynı şey değildir.

Anahtar Kelimeler (Keyword)

Aşağıdaki tabloda gördüğünüz ve JavaScript için özel anlamı olan kelimeleri belirleyici olarak kullanamazsınız. Bu anahtar kelimelerin işlevlerini yeri geldikçe göreceğiz.

break (kes)	function (işlev)	null (boş değer)	while (iken)
else (aksi takdirde)	if (eğer)	return (dön)	with (ile)
continue (devam)	in (içinde)	this (bu)	typeof (türü)
false (yanlış)	int (integer, tam sayı)	true (doğru)	void (geçersiz)
for (için)	new (yeni)	var (variable, değişken)	switch (değiştir)

Değerler (Literals)

JavaScript ile yazılmış program çalıştırıldığında değişmeyen durağan olarak kalan değerlere **durağan değerler** denilir. **Değerler** (*iterals*) bu özellikleriyle değişkenlerden ayrılırlar.

Tamsayı Değerler (Integer Literals)

Tamsayılar, decimal (ondalık) ve hexadecimal (onaltılık) olarak kullanılabilirler. Onaltılık bir tamsayı kullanılırken önüne `0x` getirilir. Onaltılık tamsayılar `0-9` arası sayılar ve `a-f` aralığındaki harflerden oluşurlar.

Kesirli Değerler (Floating-Point Literals)

Üs kısmı tamsayının arkasına e harfi eklenerek gösterilir. Tamsayının ardından `.` nokta ve kesirli kısım gelir. (örn: `4.12` , `2e4`)

Mantıksal Literaller (Boolean Literals)

Boolean literaller `TRUE` ya da `FALSE` değerlerinden birini alırlar. `TRUE` değeri `1`, `FALSE` değeri `0` rakamıyla tutulur.

Karakter Dizisi (String Literals)

" " (çift tırnaklar) ya da ' (tek tırnaklar) arasına yazılırlar. Burada dikkat edilecek husus stringin başında tek tırnak kullandıysanız sonunda da tek tırnak kullanmalısınız. Başta çift tırnak kullandıysanız, sonda da çift tırnak kullanmalısınız.

Karakter dizileri içinde özel karakterler de kullanabilirsiniz. Şimdi bunlara ve anlamlarına bakalım.

\b Geri (backspace) işlemi
\f Sayfa çıktısı (form feed).
\n Yeni satır (new line)
\r Enter-Return işlemi
\t Sekme (tab)
**** Ters-bölü işareti koyma
\' Tek tırnak işareti koyma
\" Çift tırnak işareti koyma.

Yukarıdaki özel karakterleri JavaScript ile yazdıracağınız bir karakter dizesi içinde kullanacağınız zaman metni <pre></pre> etiketleri arasına yazmalısınız. Aksi halde HTML içinde JavaScript ile yaptığınız bu düzenleme korunmaz.

Değişkenler (Variable)

İçerdiği değer programın yorumlanması esnasında değişebilen yapıdadır. Değişkenler bir değişken adı ve değişkenin aldığı değerden oluşur. Burada değişen değişkenin adı değil değişkenin değeridir.

Bir **belirleyiciyi** (*identifier*) değişken olarak kullanmak için önüne var **anahtar kelimesi** (*keyword*) eklenir.

Örnek

```
var deger1, deger2,
var isim,soyisim
```

Değişkeni belirlerken istersek değişkene değer de atayabilirisiniz. Bir örnek ve bir uygulama:

```
var isim="Mustafa", soyisim="Aydemir"
var x=5,y=4
```

Uygulama ise şöyledir:

```
var en = 8, boy = 12;
var alan = en * boy;
document.writeln(alan);
```

Bu programı çalıştırırsanız tarayıcıdan 96 sonucunu alırsınız.

Dizi-Değişkenler (Array)

Aynı değişken ile çok sayıda veriyi bir arada sıralanmasına ve saklanmasın olanak tanır. Bu da bizi tek tek tanımlama yapmaktan kurtarır. Dizinin adına, dizinin elemanlarını sıralayan bir **indis** eklenir (örn: `x[2]`). Bir dizinin ilk elemanının indisi daima `0`'dır.

Diziler `array` tipinde birer nesnedirler. Diziler `new` operatörü ile tanıtılır. Bu operatör ile bellekte diziye, dizi indisi kadar yer ayrılır. Ayrılır çünkü bu veriler birlikte kullanılacaktır. Dizileri kullanmamızın amacı da zaten budur.

Kullanımı

`var dizininismi = new Array (deger1, deger2, deger3...)`

şeklindedir.>

`var dizi=new Array(4);` şeklindeki bir dizi için `4` indis söz konusudur ve bellekte 4 yer ayrılır. İndisler `0`'dan başladığı için bu dizinin elemanları; `dizi[0]`, `dizi[1]`, `dizi[2]`, `dizi[3]` şeklindedir.

String türünde veri içeren dizi elemanlarını alfabetik olarak sıraya sokmak için `sort` metodu kullanılır.

```
<script type="text/javascript">
var diz = new Array (4);
diz [0]="Bbbbbb";
diz [1]="Dddddd";
diz [2]="Aaaaaa";
diz [3]="Cccccc";
```

```
document.write(diz+"<br />");
document.write(diz.sort()+"<br />");
</script>
```

Örneğimizde dizilerimizi tanımladıktan sonra değişkenimizi yazdırdık. Bir de `sort` metodu kullanarak yazdırdık. Aradaki aşağıdaki ekran çıktısında görebilirsiniz.

```
Bbbbbb,Dddddd,Aaaaaa,Cccccc
Aaaaaa,Bbbbbb,Cccccc,Dddddd
```

Sıralamayı ters, yani büyükten küçüğe yaptırmak istersek bu sefer `reverse` metodunu kullanmamız gerekir.

```
<script type="text/javascript">
var diz = new Array (4);
diz [0]="Bbbbbb";
diz [1]="Dddddd";
diz [2]="Aaaaaa";
diz [3]="Cccccc";
document.write(diz+"<br />");
document.write(diz.sort()+"<br />");
document.write(diz.reverse()+"<br />");
</script>
```

Bu sefer örneğimize `reverse` metodunu da ekledik. Bu sefer programın çıktısı şöyle olacaktır:

```
Bbbbbb,Dddddd,Aaaaaa,Cccccc
Aaaaaa,Bbbbbb,Cccccc,Dddddd
Dddddd,Cccccc,Bbbbbb,Aaaaaa
```

Son olarak `join` metodunu da görelim. Bu metod, dizi elemanlarını bir `string` haline getirir. Elemanlar arasına otomatik olarak `;` (virgul) koyar. İstersek elemanların aralarına konacak işareti biz de belirleyebiliriz.

Örnek

```
<script type="text/javascript">
var diz = new Array (4);
diz [0]="Bbbbbb";
diz [1]="Dddddd";
```

```
diz [2]="Aaaaaa";
diz [3]="Cccccc";
document.write(diz.join()+"<br />");
document.write(diz.join("+")+"<br />");
</script>
```

Örneğimizin çıktısı ise şöyle:

```
Bbbbbb,Dddddd,Aaaaaa,Cccccc
Bbbbbb+Dddddd+Aaaaaa+Cccccc
```

UYGULAMA: BİLGİ GİRİŞ PENCERESİ

Şimdi JavaScript'e biraz daha ısınalım ve hoş bir özelliği bu arada öğrenmiş olalım. Site ziyaretçilerinize istediğiniz türde bilgi girişi yaptırmanızı sağlayacak bir pencere yapacağız şimdi. Bunun için `window` nesnesini ve `prompt` metodunu kullanacağız. Kullanımı yapı itibarıyla şöyle;

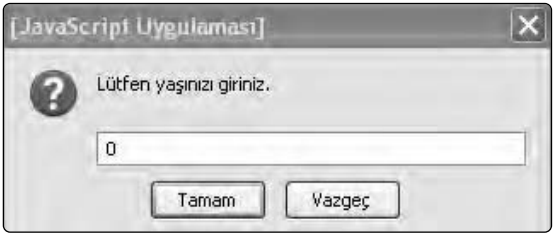
```
Değişken=window.prompt ("açıklama metni", "varsayılan değer");
```

Burada açıklama metninden kastım, ziyaretçiye girmesini istediğimiz bilgi konusunda açıklamada bulunmaktır. Varsayılan değerden kastım ise söz konusu bilgi girilmediği takdirde değişkene atanacak değerdir.

Yalnız burada bilmemiz gereken husus, `prompt` metodunu kullandığımızda pencereden girilecek bilgi **karekter dizisi** (*string*) türünde bir veri olarak algılanacağıdır. Eğer bilgi sayısal veri ise `parseInt` fonksiyonunu kullanarak **tamsayı** (*integer*) bir veri haline dönüştürmemiz gerekir. Şimdi bir örnek yapalım.

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xlmns="http://www.w3org/1999/xhtml">
<head>
<script type="text/javascript">
var yasgir,yas;
yasgir=window.prompt("Lütfen yaşınızı giriniz.","0");
yas=parseInt(yasgir);
document.writeln("YAŞINIZ:"+yasgir);
</script>
</head>
```

```
<body>
</body>
</html>
```



Bu örnekte, ekranda size yaşınızı girmenizi isteyen bir pencere açılıyor ve siz buraya yaşınızı girip tamam tuşunu tıkladığınızda

```
YAŞINIZ:38
```

şeklinde yaşınızı gösteriyor. Nasıl yaptığımızı da sanırım anladınız. Önce `yasgir` ve `yas` isimli iki değişken tanımladık ve `window.prompt` ile pencereyi oluşturup yaşınızı girmenizi rica eden bir mesaj bastırdık. Gireceğiniz değer bir integer değer olduğu içinde `parseInt` ile `yasgir` değişkenin değerini `yas` değişkenine aktardık. Son olarak da `document.writeln` ile ekrana yazdırdık.

KONTROL YAPILARI

Ortada bir program varsa, içinde bir karar süreci de vardır. Yani şu olursa şöyle yap, olmazsa böyle yap şeklinde bir karşılaştırma ve karar süreci yaşanır. İfadenin doğruluğuna ya da yanlışlığına göre belli işlemlerin yapılıp yapılmayacağı belirlenir. JavaScript ile bu tarz işlemleri yapmamızı sağlayan yapılara geçmeden önce, bu yapıları kullanmamızı sağlayan operatörlerin bazılarını sıralayalım.

OPERATÖRLER

ATAMA (ASSIGNMENT) OPERATÖRÜ

Bir değişkene değer atamak için = (eşittir) operatörünü kullanırız.

Örnek

```
var isim="Mustafa", soyisim="Aydemir";
var x=5,y=4;
```

Siz yukarıdaki gibi atama yapmak için = (eşittir) kullandığınızda JavaScript bunu kontrol edecektir. Eğer değişken daha önce tanımlanmış bir değişkense, JavaScript değişkenin yeni değerini geçerli sayar. Eğer değişken adı **belirleyici** (*identifier*) kurallarına uymuyorsa, mesela bir rakamla başlıyorsa JavaScript hata verir. Yukarıda örneğini yaptığımız şekilde, oluşturduğunuz değişkene değer olarak bir başka değişkeni, dolayısıyla o değişkenin değerini de atayabilirsiniz. Örneği tekrar görelim;

```
var en = 8, boy = 12;
var alan = en * boy;
```

Burada alan değişkenine en ve boy değişkenlerinin değerleri atanmış oldu.

ARTTIRMA VE AZALTMA OPERATÖRLERİ

JavaScript ile aritmetiksel işlemlerde toplama için `+`, çıkarma için `-`, bölme için `/` ve çarpma için `*` operatörleri kullanılır.

Örneğin; `x=2+4` işleminde `x` değişkenine `6` değeri aktarılacaktır. Bir değişkenin değerine, `x=x+4` şeklinde artı değer ekleyebilirsiniz. Bir işlemde bir aritmetiksel değeri birer birer arttırmak gerektiğinde `++x`, birer birer azaltmak gerektiğinde de bu işlemi `--x` şeklinde yapabilirsiniz. Bu şekilde bir yazım önce `x`'in değerini `1` arttır ya da azalt sonra kullan anlamına gelir.

Eğer işlemleri `x++` ve `x--` şeklinde yazarsak, önce `a`'nın değerin `al` ve sonra da `1` arttır anlamına gelir. Ayrıca `x=x+y` işlemini kısaca `x += y` şeklinde, eğer çıkarma işlemi ise `x = x - y` işlemini `x -= y` şeklinde yazabilirsiniz. Aynı şekilde `x=x/y` işlemini `x/=y` olarak ve `x=x%y` işlemini de `x%=y` olarak yazabilirsiniz.

Aritmetiksel işlemlerde işlem önceliği `()` (parantez) içindeki işlemde olur. Mesela `x= (2+4)*2` işleminde önce toplama işlemi, ardından çarpma işlemi yapılır. Birden fazla parantez söz konusu ise önce en içteki parantez işlem önceliğine sahip olur. Örneğin;

```
var alan = (2+3)*3;
document.writeln(alan);
```

işleminin sonucu `15` olacaktır.

Ayrıca, çarpma, bölme ve mod aynı işlem önceliğine sahiptirler ve toplama, çıkarma işlemlerinden önceliklidirler. Aynı işlem önceliğine sahip operatörler peşpeşe kullanıldıklarında işlemler soldan sağa doğru sıra ile yapılır.

Mantıksal Operatörler

&& (Ve): İki koşul da sağlanmalı.

|| (Veya): İki koşuldan biri sağlanmalı.

! (**Değil):** Tek koşul için uygulanır. Koşul yanlış olduğunda sonuç `TRUE` (*doğru*), doğru olduğunda ise sonuç `FALSE` (*yanlış*) olacaktır.

String Operatörleri

Toplama operatörü (+) karakter dizilerini (*string*) birleştirmek için de kullanılır.

```
<script type="text/javascript">
var metin1="Mustafa",metin2="Aydemir";
document.writeln(metin1+metin2);
</script>
```

Bu örnekte iki değişkenin değerlerini birleştirip yazdırdık. Ekran çıktısı:

```
MustafaAydemir
```

olacaktır.

Karşılaştırma Operatörleri

= = Eşittir .

!= Eşit değildir.

> Büyüktür.

< Küçüktür.

>= Büyük ya da eşit.

<= Küçük ya da eşit.

IF (EĞER) DEYİMİ

if deyimi ile koşullar arasında seçim yapacağız. Daha doğrusu programa yaptıracağız. Seçimin sonucunu ifadenin doğruluğu yada yanlışlığı belirleyecek.

```
If (koşul)
   //koşul doğruysa ilk satırı işleme koy
  // koşul doğru değilse ikinici satırdaki kodu işleme koy
```

Koşul doğru olduğunda birden çok işlem yaptırmak istersek if deyimini aşağıdaki gibi kullanacağız. Yani uygulanacak işlem gurubunu süslü parantez içinde yazacağız.

```
if (koşul)
{
//koşul doğruysa ilk satırı işleme koy
//koşul doğruysa ikinci satırı da işleme koy
...
...
}
// koşul doğru değilse bu satırı işleme koy.
```

Örnek

```
<script type="text/javascript">
var rakam;
rakam=window.prompt("Lütfen bağış miktarını giriniz.","0");
girilen=parseInt(rakam);
if (girilen>=50)document.writeln("Teşekkür Ederiz.");
</script>
```

Bu örnekte verdiğimiz koşulun gerçekleşmesi halinde yapılacak işlemi belirleyip ekrana yazdırıyoruz.

IF ... ELSE (EĞER ... DİĞER) DEYİMİ

Yukarıdaki yapıyı if...else kalıbına uyarlayalım. Bu takdirde kullanım aşağıdaki gibi olmalıdır.

```
If (koşul)
{
```

```
//koşul doğruysa ilk satırı işleme koy
//koşul doğruysa ikinci satırı da işleme koy
{
else
{

// koşul doğru değilse bu satırı işleme koy.
{
```

`if` ile ilgili örneğimizde koşulun gerçekleşmesi halinde yapılacak işlemi belirlemiştik. Şimdi yapacağımız eğlenceli örnekte `if...else` kalıbıyla koşul gerçekleşmezse yapılacak işlemi de belirleyebileceğiz.

```
<script type="text/javascript">
var rakam;
rakam=window.prompt("Lütfen bağış miktarını giriniz.","0");
girilen=parseInt(rakam);
if (girilen>=50)document.writeln("Teşekkür Ederiz.");
else document.writeln("Daha Cömert Olabilirdiniz.");
</script>
```

IF ELSE IF

Bu yapı diğerlerinden farklı olarak bize birden fazla koşul belirleme imkanı verir. İşte eğlenceli bir örnek daha;

```
<script type="text/javascript">
var gelir,miktar;
gelir=window.prompt("Lütfen Aylık Gelirinizi Giriniz.","0");
miktar=parseInt(gelir);
if (miktar>=2000)document.writeln("Kazancınız Yerinde Güle
Güle Harcayın.");
else if (miktar>=1000)document.writeln("Kazancınız İdare
Eder Halinize Şükredin.");
else document.writeln("Durumunuz Zor Görünüyor. Daha İyi
Bir İş Bulmalısınız.");
</script>
```

Koşullu Operatör

Bu operatörü burada zikretmemin sebebi, `if...else` yerine kullanılabilecek bir alternatif olmasındandır.

Kullanılışı

```
(koşul)?değer1:değer2;
```

şeklindedir.

Eğer koşul **doğru** (TRUE) ise değer1, koşul **yanlış** (FALSE) ise değer2 işleme konulur. Şimdi bu yapıyı, if...else için verdiğimiz örneğe uyarlayalım.

```
<script type="text/javascript">
var rakam;
rakam=window.prompt("Lütfen bağış miktarını giriniz.","0");
rakam=(rakam>=50)? document.writeln("Teşekkür Ederiz.") :
                   document.writeln("Daha Cömert Olabilirdiniz.");
</script>
```

Bence daha kullanışlı bir yapı. Denemeye değer ne dersiniz?

SWITCH DEYİMİ

Switch deyimi ile birçok koşuldan her biri için yapılacak işlemleri belirleyebiliriz. Bunu case ile yaparız. Koşul TRUE değerini sağladığında break ile switch yapısından çıkılır. Böylece diğer case'ler hiç değerlendirmeye alınmamış olur. Hiçbir koşul doğru değilse default değer işlenir.

```
<script type="text/javascript">
var soru
soru = prompt("Hangi Takımın Taraftarısınız: Fenerbahçe Galatasaray
             Beşiktaş", "");
switch (soru)
   {
   case "Fenerbahçe" :
         alert ("Siz Bir Kanaryasınız");
         break
   case "Galatasaray" :
         alert ("Siz Bir Aslansınız");
         break
   case "Beşiktaş" :
         alert ("Siz Bir Kartalsınız");
         break
   default: alert ("Başka Bir Takımın Taraftarısınız.");
```

```
   }
</script>
```

DÖNGÜLER

İstenilen bir koşul sağlanana kadar belli bir işlemin tekrar tekrar yapılmasını sağlar. Bu tekrarlı işlemleri döngüler ile birkaç satır kod ile halledilebilir. JavaScript dilinde üç döngü deyimi bulunmaktadır. **`While`**, **`do...while`** ve **`for`**. Şimdi bunları sıra ile inceleyelim.

WHILE DEYİMİ

Bu döngü ile belli işlemleri istenilen bir koşul sağlanıncaya kadar yaptırmak mümkündür. İşlemlerin ne kadar tekrarlanacağı değişkenin alacağı değerlere göre belirlenir. `while` döngüsünde eğer başlangıçtaki koşul sağlanamazsa döngü başlamaz.

```
<script type="text/javascript">
x=-1;
while (x<=3)
{
document.write("Sayı:="+x);
document.write("<br />");
x++;
}
</script>
```

Bu örnekte `x<=3` koşulu sağlanana kadar her seferinde bir arttırılarak döngü devam eder.

Programın çıktısı ise şöyledir:

```
Sayı:=-1
Sayı:=0
Sayı:=1
Sayı:=2
Sayı:=3
```

DO...WHILE DEYİMİ

`while` döngüsünden farkı kontrol döngünün sonuna yerleştirilmiştir. Dolayısıyla kontrol işlemi döngünün sonunda yapılır. Böylece döngü en az bir kez muhakkak çalışır.

Örnek

```
<script type="text/javascript">
x=-1;
do {
document.write("Sayı:="+x);
document.write("<br />");
x++;
}while (x<=3);
</script>
```

Bu örnek de `while` döngüsünde verdiğimiz örnekle aynı çıktıyı verecektir.

FOR DEYİMİ

Belli bir sayıda tekrardan oluşan bir döngü oluşturmak istediğimizde `for` yapısından yaralanırız.

Örnek

```
<script type="text/javascript">
document.writeln("<table border='1' width='400px'>");
document.writeln("<tr><td>sayı</td><td>karesi</td><td> ↓
kübü</td></tr>");
for (var say=1;say<=10;say++){
kare=say*say;
kup=say*say*say;
document.writeln("<tr><td>"+say+"</td><td>"+kare+" ↓
</td><td>"+kup+"</td></tr>");
}
document.writeln("</table>");
</script>
```

Bu örnekte, `for` döngüsünü kullanarak `1`'den `10`'a kadar olan sayıların karesini ve kübünü hesaplatıp bir tabloya yazdırdık. Ayrıca bu örnek JavaScript içinde `html` etiketlerinin kullanımı konusunda da iyi bir alıştırma. Denemenizi tavsiye ederim.

Yaptığımız işin mantığına gelecek olursak; önce say değişkeninin değeri `10`'a eşit oluncaya kadar her seferinde birer birer arttır dedik. Sonra da `say` değişkeninin değeri kullanılarak yapılacak aritmetik işlemleri tanımladık. İşte hepsi, bu!

Ve son olarak örneğimizin ekran çıktısına bakalım:

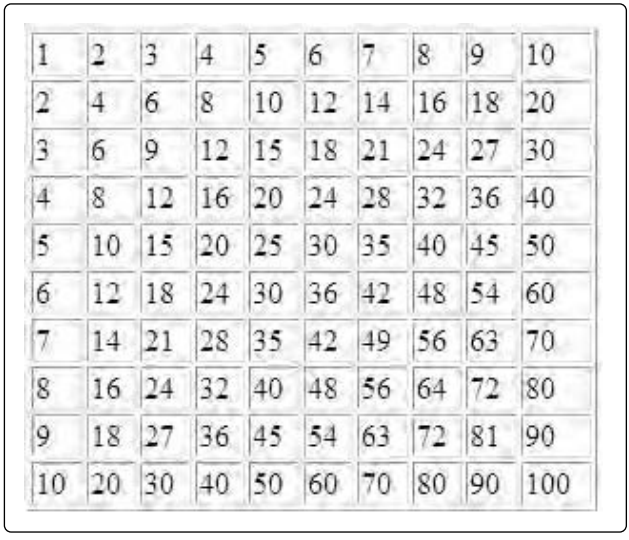
sayı	karesi	kübü
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

Yuvalanmış Döngü

Yuvalanmış döngüden kastımız, bir döngü içerisine yerleştirilmiş bir başka döngünün kullanıldığı yapılardır. İç içe yuvalanmış döngülerde kural ilk önce en içteki döngünün tamamlanmasıdır.

İşte örnek:

```
<script type="text/javascript">
var tablo;
tablo="<table border='1' width='300'>";
for (var say1=1;say1<=10;say1++){
tablo=tablo+"<tr>";
for (var say2=1;say2<=10;say2++){
tablo=tablo+"<td>"+say1*say2+"</td>";
}
tablo=tablo+"</tr>";
}
tablo=tablo+"</table>";
document.write(tablo);
</script>
```



1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Ve bu da ekran çıktısı. Döngümüz tablo içine aldık. Ve iç içe iki `for` döngüsü çalıştırdık.

CONTINUE

`Continue` ile döngü içindeki bir işlem atlanılabilir ve döngü bir sonraki işlemden devam eder.

Örnek

```
<script type="text/javascript">
for(x=1;x<=10;x++){
if(x==5){
alert("Döngü, 5 değeri atlanarak devam edecektir.");
continue;
}
document.write("x=" +x+"<br>");
}
</script>
```

Örneğimizde, döngünün `1`'den başlayıp `10`'a kadar devam etmesi istenmektedir. Fakat döngü esnasında sıra `5`'e geldiğinde döngüde bu değer atlanacak ardından döngü kaldığı yerden `10`'a kadar yani `x<=10` olana kadar devam edecektir.

FONKSİYONLAR

JavaScript dilinde yapılması istenilen birtakım işlerin icra edilebilmesi için kullanılan program parçalarıdır.

Kullanımı

```
function fonksiyonun_adı (parametre1, parametre2)
{
kodlar
}
```

Fonksiyonda parametre olarak, fonksiyonu çağıran program tarafından gönderilen veriler kullanılır. Fakat bazı durumlarda fonksiyonlarda parametre bulunmayabilir. JavaScript ile fonksiyonun sonucunu fonksiyonu çağıran programa göndermek için, bir başka ifade ile değer döndürmek için `return` deyimi kullanılır.

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xlmns="http://www.w3org/1999/xhtml">
<head>
<script type="text/javascript">
function mesaj()
{
alert("Merhaba Dünya!");
}
</script>
</head>
<body>
<form>
<input type="button" value="Tıkla" onclick="mesaj()" />
</form>
</body>
</html>
```

Örneğimizde bir buton oluşturduk ve butona tıklandığında `alert` metodu ile bir mesaj gösteriyoruz. Şimdi bir de `return` ile ilgili örnek yapalım.

```
<head>
<script type="text/javascript">
```

```
function say(a,b)
{
return a*b;
}
</script>
</head>
<body>
<script type="text/javascript">
document.write(say(4,3));
</script>
</body>
```

NESNELER

Konunun başında da açıklamaya çalıştığım gibi JavaScript **Nesneye Yönelik** (*Object-Oriented*) bir dildir. Nesneler, içlerinde değişkenler, fonksiyonlar ve başka bazı unsurlar barındıran yapılardır. Nesneye yönelik programlama terminolojisinde **fonksiyonlar**, **metotlar** olarak adlandırılırlar. Bu bağlamda değişkenler içinde **özellikler** ifadesi kullanılır.

JavaScript dilinde önceden tanımlanmış hazır nesneleri (örn: `document`) kullanabileceğiniz gibi siz de nesne tanımlayabilirsiniz. Nesneler sınıf denilen hiyerarşik bir yapının ürünüdürler. Her nesne ait olduğu sınıfın bir örneğidir ve o sınıfın tüm yapısal özelliklerini barındırır. Fakat bir nesne sadece kendisine ait verileri barındırır.

Bir metod, müstakil bir fonksiyondan farklı olarak ait olduğunu sınıftaki tüm nesneler tarafından kullanılabilirler. Metodlar genellikle parametre alırlar. Fakat parametre almasalar da mutlaka `( )` (parantezlerle) birlikte yazılırlar.

Örnek

```
document.write("Mustafa Aydemir");
```

Bu örnekte `write` metodu bir parametreye (`"Mustafa Aydemir"`) sahiptir.

Evet artık bir nesnenin nasıl oluşturulacağı konusundan bahsedebiliriz. Önce bir fonksiyon ile nesne tipini tanımlamamız gerekiyor.

Örnek

```
function kitaplar (ismi,yayinevi,yazar){
      this.ismi=ismi
      this.yayinevi=yayinevi
      this.yazar=yazar
}
```

Amacımız kitap adında bir nesne oluşturmak. Önce oluşturacağımız nesnenin özelliklerini ortaya çıkaran bir fonksiyon yazdık. Artık sadece kitap adında bir tek nesne değil, bu fonksiyonu kullanan birçok nesne oluşturabiliriz. Nesnemizi oluştururken new operatöründen faydalanıyoruz.

İşte kitap nesnemiz:

```
kitap=new kitaplar ("Web Tasarım","kodlab","Mustafa Aydemir");
```

Evet şimdi de nesnemizi ekrana yazdırmak için nesnemizle ilişkili bir metod tanımlayalım.

```
function bilgiler (){
   var bilgi = this.ismi + "<br />" + this.yayinevi + "<br />"
+ this.yazar;
   document.write (bilgi);
}
```

Bu metodu nesneye eklemek için this.bilgiler=bilgiler ifadesini nesne tanımına eklememiz yeterli olacaktır. Artık kitap adlı nesnenin bilgilerine ulaşmak için kitap.bilgiler() ifadesi yeterlidir. Ve işte çalışmamızın son hali:

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xlmns="http://www.w3org/1999/xhtml">
<head>
</head>
<body>
<script type="text/javascript">
function kitaplar (ismi,yayinevi,yazar){
   this.ismi=ismi
   this.yayinevi=yayinevi
```

```
      this.yazar=yazar
   this.bilgiler=bilgiler
}
function bilgiler (){
   var bilgi = this.ismi + "<br />" + this.yayinevi + "<br />"
+ this.yazar;
   document.write (bilgi);
}

kitap=new kitaplar ("Web Tasarım","kodlab","Mustafa Aydemir");
kitap.bilgiler();
</script>
</body>
</html>
```

Bu örneğin ekran çıktısı aşağıdaki gibi olacaktır.

```
Web Tasarım
kodlab
Mustafa Aydemir
```

JavaScript Hazır Nesneleri

JavaScript ile kullanabileceğiniz hazır nesneleri ve işlevlerini aşağıdaki tablodan inceleyebilirsiniz.

Nesne	İşlevi
anchor	Document nesnesine aittir. Link oluşturmak için çapa.
button	Form nesnesine aittir. Buton
checkbox	Form nesnesine aittir. Onay kutusu
date	Üst düzey bir nesnedir. Zaman ve tarih işlemleri içindir.
document	window nesnesine aittir.
form	Document nesnesine aittir. Form oluşturmak içindir.
hiden	Form nesnesine aittir. Görünmez metin kutusu
history	Document nesnesine aittir. Tarayıcı geçmişidir.l
ink	Document nesnesine aittir. Link oluşturmak içindir
location	Document nesnesine aittir. Aktif URL

math	Üst düzey bir nesnedir. Matematiksel işlemler içindir.
navigator	Tarayıcı sürüm bilgileri
password	Form nesnesine aittir. Şifre metin kutusu
radio	Form nesnesine aittir. Radyo butonu
select	Form nesnesine aittir. Seçim listesi
string	Document nesnesine aittir. Karakter dizisi
submit	Form nesnesine aittir. Gönder butonu
textForm	nesnesine aittir. Metin kutusu
textarea	Form nesnesine aittir. Metin alanı
window	Üst düzey bir nesnedir. Tarayıcı penceresi

Nesnelere Ait Özellikler

JavaScript nesnelerinin özelliklerini ve aldıkları değerleri göstermek için aşağıdaki yapı kullanılır.

```
nesne.özellik= değer
```

Örneğin document nesnesine ait zemin rengini belirlemek için kullanılan `bgColor` özelliğini ve mesela kırmızı renk değeri ile gösterimini yazalım.

```
document.bgColor="red";
```

`String` nesnesinin bir tek `lenght` özelliği mevcuttur. Bu özellik string'in kaç karakterden oluştuğunu belirtir. Diğer nesnelere ait özellikleri ise takibi kolay olduğundan yine tablolar şeklinde hazırladım.

Document Nesnesine Ait Özellikler

Özellik	İşlevi
alinkColor	Link rengi
bgColor	Belgenin zemin rengi
cookie	Çerez bellek

defaultStatus	Durum çubuğu metni
fgColor	Metin rengi
lastModified	Belgede en son değişikliğin tarihi
linkColor	Tıklanmamış link rengi
location	URL
referrer	Tıklanan link URL'si
status	Durum çubuğu metni
title	Belge başlığı
vlinkColor	Tıklanan linklerin rengi

Form Nesnesine Ait Özellikler

Özellik	İşlevi
action	Hedef URL
defaultChecked	Seçili olma özelliği. (Radyo düğmesi ve seçim kutusu için)
defaultSelected	Varsayılan seçim.
defaultValue	Varsayılan değer. (Metin kutusu ve metin alanı için)
checked	Onay durumu. (Radyo düğmesi ve seçim kutusu için)
encoding	MIME formatı
form	Form çerçevesi
index	Seçim listesi
lenght	Liste eleman sayısı.
method	Form gönderme metodu (get veya post)
name	Form adı
selected	Mevcut durum (radyo düğmesi ve seçim kutusu için)
selectedIndex	Seçim listesinde seçili liste elemanı.
target	Hedef form
text	Seçim listesinde seçili liste elemanının metni.
value	Atanmış metin (metin kutusu ve metin alanı için)

Nesnelere Ait Metotlar

Bir hatırlatma olarak söylemek gerekirse, metotları nesnelerin yapabilecekleri işlevler diye tarif edebiliriz. Metotlar nesneler içindir. Bu kısımda, değişik nesnelere ait metotları bulacaksınız.

Document Nesnesine Ait Metotlar

Metod	Açıklama
clear	Pencereyi silmek için kullanılır.
close	Pencereyi kapatmak içindir.
open	Bir nesneyi açmak içindir.
write	Metni pencereye yazdırmak içindir.
writeln	Metni pencereye yazdırmak içindir. Yeni bir satıra geçer.

Form Nesnesine Ait Metotlar

Metod	Açıklama
click	Bir butona tıklama işlemini yönetir.
focus	Nesnenin seçili olmasını sağlar.
blur	Nesnenin focus durumunu sona erdirir.
select	Nesnenin içindeki veriyi seçer.
submit	Formu gönderme olayını yönetir.

Date Nesnesine Ait Metotlar

Metod	Açıklama
GetDate	Ayın günleri (1-31 arası değer alır)
getaDay	Haftanın günleri (1-7 arası değer alır)
getHours	Saati gösterir.
getMinutes	Dakikayı gösterir.
getMonth	Ayı gösterir.
getSeconds	Saniyeyi gösterir.

getTimeZoneoffset	Bölgesel zaman değerini dakika olarak gösterir.
getYear	Yılı gösterir.
setDate	Tarih ayarlama.
setDay	Tarih ayarlama.
setHours	Saat ayarlama.
setMinutes	Dakika ayarlama.
setMonth	Ay ayarlama.
setSeconds	Saniye ayarlama.
toLocaleString	Yerel tarih bilgisini stringe dönüştürür.

Window Nesnesine Ait Metotlar

Metod	Açıklama
clear	Pencereyi siler.
close	Pencereyi kapatır.
open	Bir pencereyi açar.

String Nesnesine Ait Metotlar

Metod	Açıklama
anchor	Bir çapa oluşturur.
big	Metni büyük oluşturur.
blink	Yanar söner metin.
bold	Kalın harfli metin.
fontcolor	Font rengi için.
italics	İtalik yazı.
link	Link oluşturmak içindir.
small	Metni küçültür.
strike	Metne üst çizgi ekler.

write	Belgeye metin eklemek içindir.
writeln	Belgeye metin eklemek içindir. Satırbaşı yapar.

MATH NESNESİNE AİT METOTLAR

Metod	Açıklama
abs	Bir sayının mutlak değeri.
log	Bir sayının logaritması.
cos	Bir sayının kosinüsü.
eval	Bir string içinde matematiksel işlem yapma.
max	Büyük sayı.
min	Küçük sayı.
random	Rastgele sayı.
round	Yuvarlama işlemi.
sin	Bir sayının sinüsü.
sqrt	Bir sayının karekökü.
tan	Bir sayının tanjantı.

OLAY YÖNETİCİLERİ (EVENT HANDLERS)

Bir web sayfasında çalışılırken fare ile bir butona tıklamak, klavyenin bir tuşuna basmak, bir form ile veri göndermek gibi bir çok olay gerçekleştiririz. JavaScript bir olay tarafından yönlendirilen (*event-driven*) tipte programlama yapmaya imkan veren bir dil olduğundan bize bu olayları **Olay Yöneticileri** (*Event Hendler*) ile programlama imkanı verir.

onLoad-onUnLoad

Bir sayfanın yüklenmesi, yani bir sayfanın ziyaret edilmesi load olayıdır. Sayfanın terk edilmesi ise **unload** olayıdır.

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xlmns="http://www.w3org/1999/xhtml">
<head>
<script type="text/javascript">
function gel()
{
alert("Sitemize Hosgeldiniz")
}
function git()
{
alert("Güle güle. Tekrar Bekleriz")
}
</script>
</head>
<body onLoad="gel()" onUnload="git()">
</body>
</html>
```

Bu örnekte onLoad olay yöneticisi ile gel adlı fonksiyonu, onUnload olay yöneticisi ile de git adlı fonksiyonu tetikliyoruz. Ve tabi fonksiyonlar da tanımlanan görevi icra edince ortay bu program çıkmış oluyor. Kişi sayfaya ilk geldiğinde onu bir hoş geldin mesajı ile karşılayıp, sayfadan ayrılırken de bir güle güle mesajı ile uğurluyoruz. Çok kibar bir webmaster'ız gerçekten.

onSubmit Olayı

Submit olayı submit tuşuna tıklandığında gerçekleşir. Biz burada bir fonksiyon yazıp onSubmit ile çalıştıracağız.

Örnek

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xlmns="http://www.w3org/1999/xhtml">
<head>

</head>
<body>
<form name="form"
      method="post"
      action=""
```

```
        onsubmit="alert('Bilgi gönderildi.')">
    <input type="text" name="alan">
    <input type="submit" value="Gönder">
</form>
</body>
</html>
```

onMouseOver Olayı

Farenin bir nesne üzerine gelmesi de JavaScript için bir olaydır.

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xlmns="http://www.w3org/1999/xhtml">
<head>
</head>
<body>
<a href="" onMouseOver="alert('Örnek bir onMouseOver ola-
yı');"> Fare imlecini yazının üstüne konumlandırın.</a>
</body>
</html>
```

Örneğimiz, `onMouseOver` olayını gayet güzel açıklıyor. Örneği denediğinizde yani fare imlecini yazının üzerine getirdiğinizde olay gerçekleşmiş olur. Bu durumda biz olay yöneticisine ne istediğimizi bildiriyoruz. Bizim örneğimize göre fare imleci yazının üstüne gelince, yani olay gerçekleşince bize bir pencere aç diyoruz.

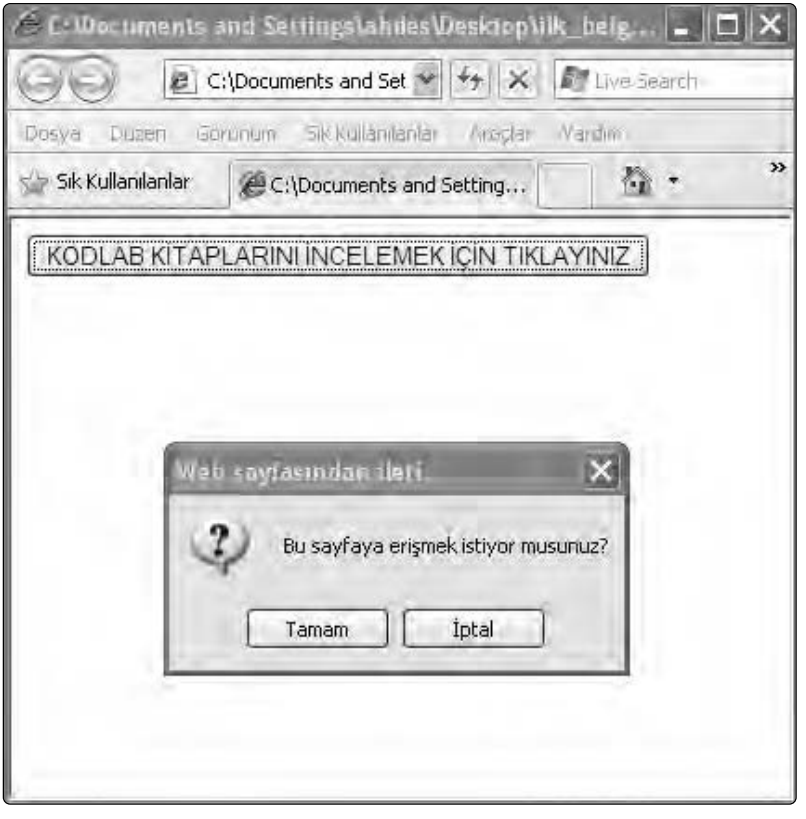
onClick Olayı

Fare ile bir nesneye (örn: buton, onay kutusu, radyo düğmesi... vb) tıklandığında `onClick` olayı gerçekleşmiş olur. Burada dikkat edilecek husus, `onClick` yöneticisine kendisinden beklenen işi ilgili nesnenin HTML etiketleri arasında bildirmenizdir.

Örnek

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xlmns="http://www.w3org/1999/xhtml">
<head>
<script type="text/javascript">
```

```
function soru (){
sor="Bu sayfaya erişmek istiyor musunuz?";
if (confirm(sor)){
location.href="http://www.kodlab.com";
}
}
</script>
</head>
<body>
<form>
<input type="button" value="KODLAB KİTAPLARINI İNCELEMEK
İÇİN TIKLAYINIZ" onClick="soru()">
</form>
</body>
</html>
```



Örneğimizi çalıştırdığınızda karşınıza:

```
KODLAB KİTAPLARINI İNCELEMEK İÇİN TIKLAYINIZ
```

şeklinde bir buton gelecek. Butonu tıklarsanız bir pencere açılır ve size ilgili sayfaya erişmek isteyip istemediğinizi sorar. **OK** derseniz site açılacaktır. Burada `if` kontrolü ile sadece **OK** dediğinizde sayfanın açılmasını sağlamış oluyoruz.

onFocus-onBlur

Focus olayı aktif olma durumunu ifade eder. Bir pencerenin seçili olması ya da bir formda bir text alanının içine metin yazılacağı anda tıklanarak seçilmesi gibi. Örneğimizde tarayıcı penceresi aktifken yani onFocus olayı gerçekleşince sayfanın rengi mavi, sayfa aktif değilken yani onBlur olayı gerçekleşince (mesela bir başka tarayıcı penceresine veya program penceresi tıkladığınızda) kırmızı renkte olacaktır.

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xlmns="http://www.w3org/1999/xhtml">
<head>
<script type="text/javascript">
function ayarla(renk)
{
document.bgColor=renk;
document.body.style.backgroundColor=renk;
}
</script>
</head>
<body onFocus="ayarla('blue')" onBlur="ayarla('red')">
</body>
</html>
```

onChange Olayı

Bir seçim kutusu tıklandığında, bir metin alanına ya da bir metin kutusunda metin girildiğinde bir **değişiklik** (*change*) olayı gerçekleşmiş olur. İşte biz onChange yöneticisi ile bir değişiklik olayı ile yapmak istediğimiz şeyi programlayacağız.

Örnek

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xlmns="http://www.w3org/1999/xhtml">
<head>
<script type="text/javascript">
 function ayarla(renkler) {
         var renk = (renkler.options[renkler.selectedIndex].value);
         if (renk) {
            document.bgColor = renk;
         }
         alert("Arkaplan rengi değiştirildi")
    }
</script>
</head>
<body>
<select name="renkler" onChange="ayarla(this)">
     <option value="white" selected="selected">Beyaz</option>
     <option value="red">Kırmızı</option>
     <option value="blue">Mavi</option>
</select>
</body>
</html>
```

Örneğimizde bir seçim kutucuğundan bir renk seçtiğimizde yani bir **değişiklik** (*change*) olduğunda sayfanın arkaplan rengini değiştiriyoruz.

8

FIREWORKS

Fireworks	379	Maskeler	431
Fireworks Çalışma Alanı	381	Animasyonlar	443
Belgeler ve Belge İşlemleri	393	Dilimler	432
Nesnelerle Çalışmak	400	Rollover'lar	433
Bitmap Grafikler	409	Resimlerle Toplu İşlemler	435
Filtreler	412	Açılır Menü	440
Vektörel Nesneler	415	Slideshow	447
Metinlerle Çalışmak	422	Optimizasyon	454
Renkler	426	Fireworks Tercihleri	456
Sayfalar	428	Uygulamalar	463
Katmanlar	429		

FIREWORKS

Bir web tasarımcı için en gerekli programlardan biri de **Fireworks** programıdır. Adobe firmasına ait bu profesyonel programın deneme sürümünü kitapla beraber ücretsiz olarak verilen DVD'de bulabilirsiniz.

Fireworks çok başarılı bir grafik ve web tasarım editörüdür. Web tasarımcılar için sağladığı pratik çözümler ile ön plana çıkan programlardan birisidir.

Birçok programın yapacağı işi Fireworks ile halledebilirsiniz. Optimum kalite ile düşük boyutlu HTML sayfaları ve kaliteli grafikler elde edebilirsiniz. Şimdi dilerseniz kısa bir Fireworks turu yaparak program hakkında genel bir fikir edinelim.

NELER YAPILIR?

Fireworks ile **bitmap** (*piksel tabanlı*) ve **vektörel** grafikleri tek bir belgede oluşturabilirsiniz. Hazırladığınız grafikleri, grafiklerin boyutunu küçülterek optimize edebilirsiniz. Belgenize son şeklini verdikten sonra

belgenizi HTML dosyası olarak kaydedebilirsiniz. Bu işlemle Fireworks, HTML kodlarını oluşturduğu gibi eğer belgenizde JavaScript gerektiren bir işlem bulunuyorsa (örn: rollover, pou-up menü...vd) gereken JavaScript kodlarını da sizin için oluşturacaktır.

Fireworks ile sadece HTML belgeleri değil aynı zamanda da Photoshop, Flash... vb programların desteklediği dosya biçimlerinde de belgeler oluşturabilirsiniz. Dolayısıyla Fireworks ile başladığınız çalışmaya, sözünü ettiğim diğer programlarla da devam edebilirsiniz.

Bitmap ve Vektörel Grafikler

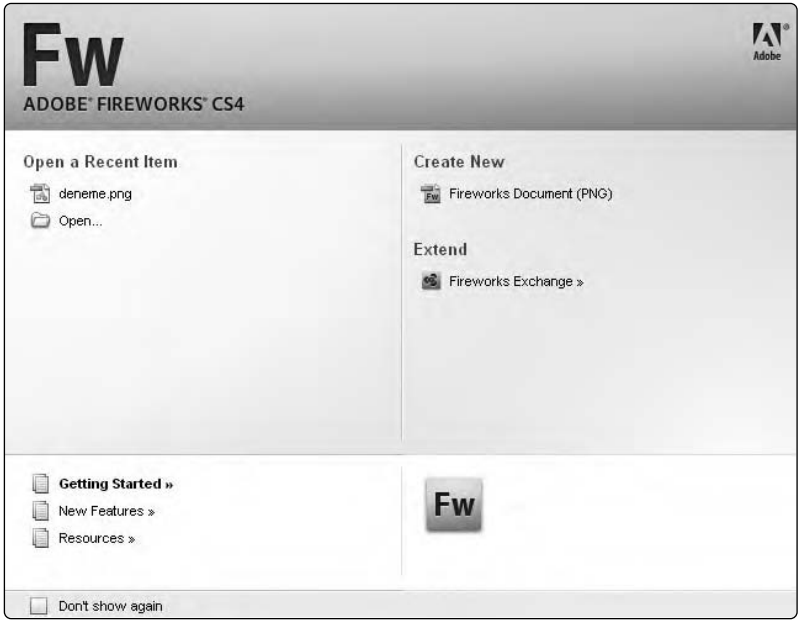
Dilerseniz, **bitmap** ve **vektörel grafikler** demişken öncelikle bunlar üzerinde duralım.

Vektörel grafikler, vektör denilen, konum ve renk bilgisi içeren çizgiler ve eğriler kullanılarak oluşturulurlar. Vektörel grafikler çözünürlükten bağımsızdır. Bu sebeple bir vektörel grafiği yeniden boyutlandırdığınızda ya da yeniden renklendirdiğinizde kalitesinden hiçbir şey kaybetmeyecektir.

Bitmap grafikler ise **piksel** denilen kare şeklindeki noktacıklardan oluşurlar. Yapıları, adeta bir karoya yerleştirilen mozaik parçaları gibi ızgara şeklindedir. Dolayısı ile çözünürlüğe bağımlıdırlar. Yani bir Bitmap grafiğinin boyutlarını değiştirdiğinizde görüntü kalitesi düşecektir.

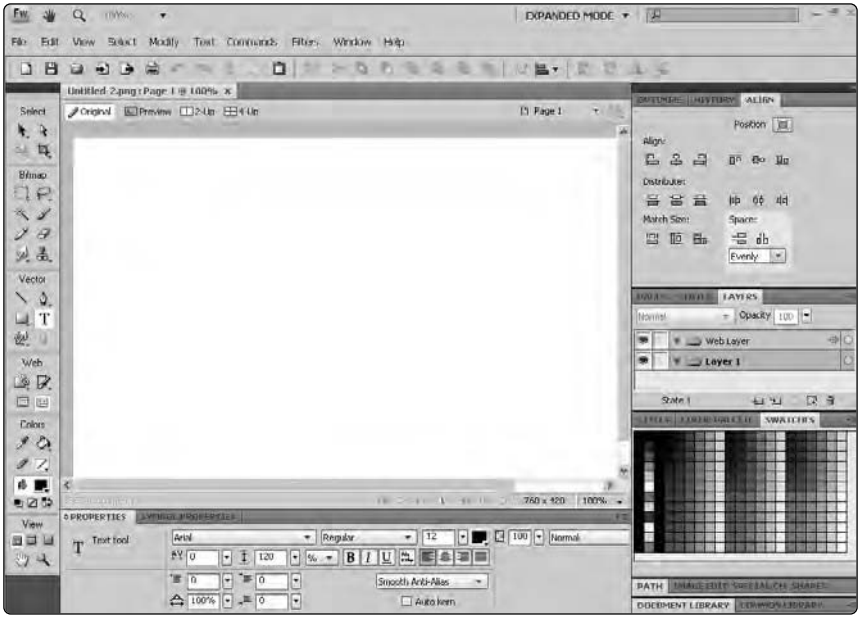
Fireworks Çalışma Alanı

Başlangıç Sayfası



Fireworks'ü açtığınızda sizi resimdeki başlangıç sayfası karşılayacaktır. **Open a Recent Item** kısmı ile son çalıştığınız dosyalara kolayca erişebilirsiniz. Açmak istediğiniz belge listede yer almıyorsa **Open** linkini tıklayarak açabilirsiniz. **Create New** ile de yeni bir belge açabilirsiniz. **Extend** bölümünü ile Fireworks eklentilerine ulaşmanızı sağlayan Adobe sitesinin ilgili sayfasına ulaşabilirsiniz. **Eklentiler** Fireworks'e birtakım yeni özellikler kazandıran program parçalarıdır. Yalnız bunları indirmeden önce Adobe sitesine üye olmanız gerekmektedir. Eğer Fireworks başladığında bu sayfayı görmek istemiyorsanız sayfanın en altındaki **Don't Show Again** kutucuğunu işaretlemelisiniz.

Fireworks Arayüzü



Fireworks CS4 yeni arayüzü ile gerçekten göz dolduruyor. Bu sürümle beraber **Photoshop**, **Dreamweaver**, **Flash** gibi Adobe programları arasında görsel bir ortak yapı tesis edilmiş oldu. Evet Adobe programcıları, üstte menü ve araç çubuğu, solda **Tools** (*Araçlar*) paneli, sağdaki grup halinde gördüğünüz panelleri ve alttaki **Properties** (*Özellikler*) paneli ile tüm Fireworks işlevselliğini büyük bir ustalıkla arayüze yerleştirmişler diyebiliriz. Şimdi Fireworks işlevselliğini oluşturan bu arayüz parçalarına biraz daha yakından bakalım.

Tools (Araçlar) Paneli

Fireworks Tools paneli, nesneleri seçme ve taşıma araçları; bitmap grafikleri ve vektörel grafikleri düzenleme araçları, interaktif web grafik araçları, renk araçları ve görünüm araçlarını içermektedir. Bu araçlarla bir nesneyi seçme ve taşıma, resim kırpma, nesneleri ölçeklendirme, nesneleri boyama, çizim yapma, resimlerin bir bölümünü seçme, resimlere bulanıklık, keskinlik, aydınlatma gibi uygulamalar yapma,

metin oluşturma, nesnelere renk geçişi verme, resimleri büyütme... vb birçok Fireworks işlevine sadece bir tık uzaklıktasınız.



Tools panelindeki bazı araçların sağ alt köşelerinde birer küçük üçgen göreceksiniz. Bunun anlamı, bu araç altında başka alt araçların da olduğudur. Fare ile bu üçgenlere tıkladığınızda ya da fareyi Araç Tuşu üzerinde basılı tutarsanız **Alt Araçlar** menüsü açılacaktır. Kullanmak istediğiniz aracı fare ile bir kez tıklamanız söz konusu aracın kullanılır hale gelmesi için yeterlidir.

Tuşların isimlerinin yanında parantez içinde gördüğünüz harfler o araçla ilgili kısayol tuşlarıdır. Mesela çalışmanız esnasında metin aracını seçmek için **Tools** panelindeki **Text** tuşuna basabileceğiniz gibi, dilerseniz bu işlemi klavyeden **T** harfine basarak da yapabilirsiniz.

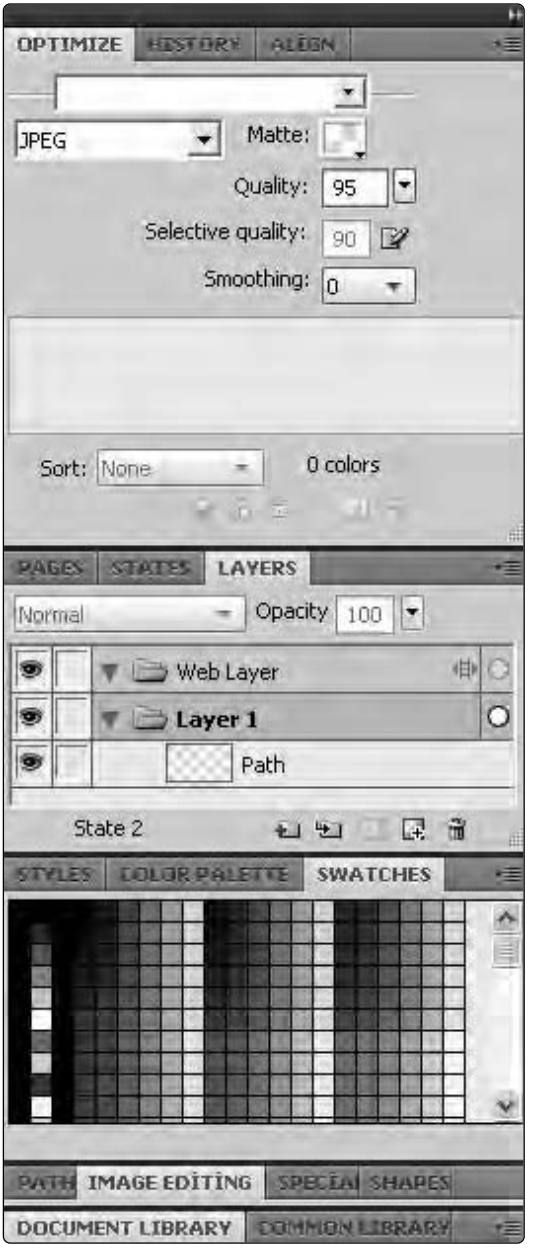
Bir aracın eğer alt araçları varsa onlar da aynı kısayol tuşu ile çağırılırlar. Mesela **Blur** aracının kısa yolu **R** harfidir. Bu aracın altındaki diğer araçların kısayolları da aynıdır. Bu durumda Blur aracının altındaki bir başka aracı seçmek için **R** harfine ardarda basmamız gerekmektedir. Kısayol tuşuna her basışımızda bir alt aracı seçmiş oluruz.

PROPERTIES (ÖZELLİKLER) PANELİ



Properties paneli, araçların ve nesnelerin özelliklerini düzenlememizi sağlayan paneldir ve seçtiğiniz araca göre özellikler değişecektir. Resimde panelin **Text** aracına tıklandıktan sonraki durumunu görüyorsunuz. Panelde şuan, metin oluşturma ve düzenleme işlemi için gereken bir çok özellik mevcuttur. Şimdi **Tools** panelinden mesela **Pen** aracına tıklayın. Paneldeki tüm özellikler **Pen** aracı ile çalışırken gerekecek olan özelliklerle değişecektir.

Panel Grupları



Her biri farklı işlevlere sahip araçlardır. Birbirlerinden bağımsız olarak da hareket ettirilip sayfanın farklı yerlerine fare ile sürüklenerek taşınabilirler. Panellerden dilediğinizi dilediğiniz zaman **window** menüsünden açıp kapatabileceğiniz gibi gerektiğinde simge durumuna da küçültebilirsiniz.

Ayrıca **Window>Hide Panel** menüsü ile tüm panelleri bir seferde kapatabilirsiniz. Her panelde bir **Options** (*Seçenekler*) menüsü bulunmaktadır. Her panelin **Options** menüsünde o panele mahsus işlemlerle ilgili alt menüler yer almaktadır.

Fireworks'te panelleri üç farklı modda görüntüleyebilirsiniz. Belge penceresini sağ üst kısmında bulunan **Expanded Mode** tuşuna tıkladığınızda açılacak menüden tercihinizi belirleyebilirsiniz.

Bunlar:

Expanded Mode: Panelleri geniş haliyle görüntüler.

Iconic Mode: Panelleri simge biçiminde görüntüler.

Iconic Mode With Panel Names: Simgeler ve panel adları şeklinde bir görünüm sağlar.

Şimdi her bir panele tek tek göz atalım.

OPTIMIZE PANELİ

Dosya boyutunu ve dosya türünü ayarlamak içindir.



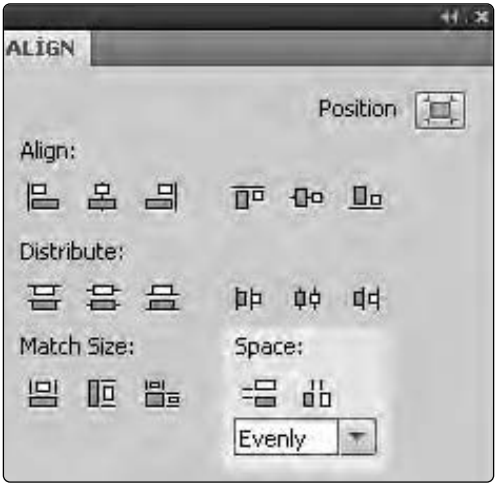
History Paneli

Bu panel son yapılan işlemlerin listesini gösterir. Panelin işlevi, yaptığımız işlemleri gerektiğinde hızlı bir biçimde ileri ve geri alabilmemize olanak tanımasıdır. Ayrıca bir çok eylemi seçip komut olarak kaydederek gerektiğinde yeniden kullanabilirsiniz.



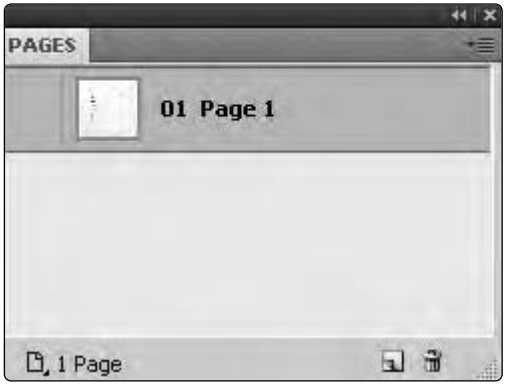
Align Paneli

Tuval üzerindeki nesneleri hizalamak ve dağıtmak için kullanılır.



Pages Paneli

Halihazırda üzerinde çalışılmakta olan belgeye ait sayfaları görüntüler.



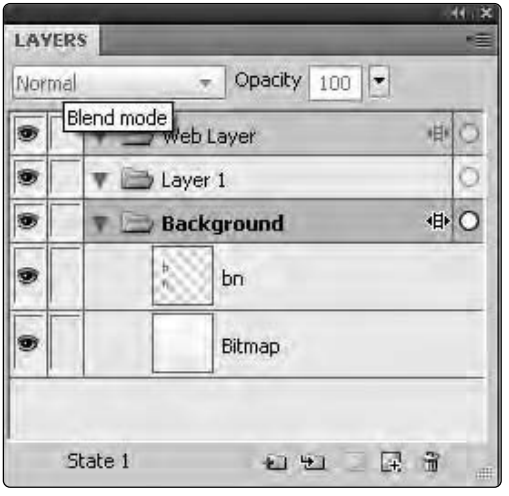
States Paneli

Üzerinde çalışılan belge ile ilgili durumları gösterir. Ayrıca animasyon oluşturmayı sağlayan bazı seçeneklere sahiptir.



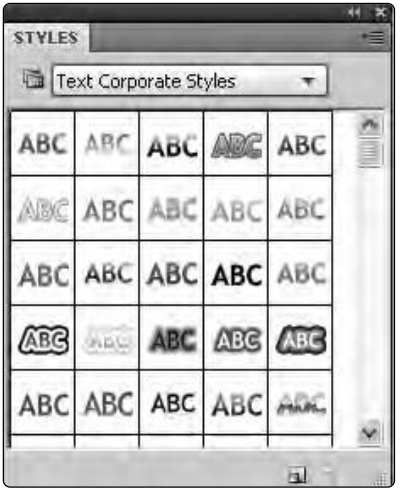
LAYERS PANELİ

Belge üzerinde katman oluşturmaya, mevcut katmanları silmeye ve değiştirmeye yarayan bir paneldir.



STYLES PANELİ

Kullandığınız nesnelerin niteliklerini saklayarak tekrar kullanmanıza olanak tanır. Ayrıca işinizi kolaylaştıracak olan hazır bir çok stil barındırmaktadır.



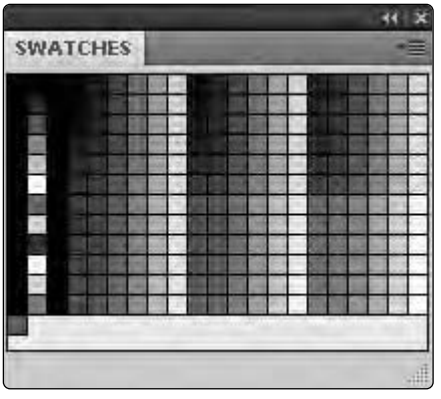
Color Palette Paneli

Bu panel ile renk paletleri oluşturabilir, oluşturduğunuz paletleri değiştirebilir ve nesnelerinizde kullanacağınız yeni renkleri bu şekilde oluşturabilirsiniz.



SWATCHES PANELİ

Geçerli belgenin renk paletini yönetmek içindir.



PATH PANELİ

Tuvalinizde oluşturacağınız yollarla ilgili hızlı erişim komutları içerir.

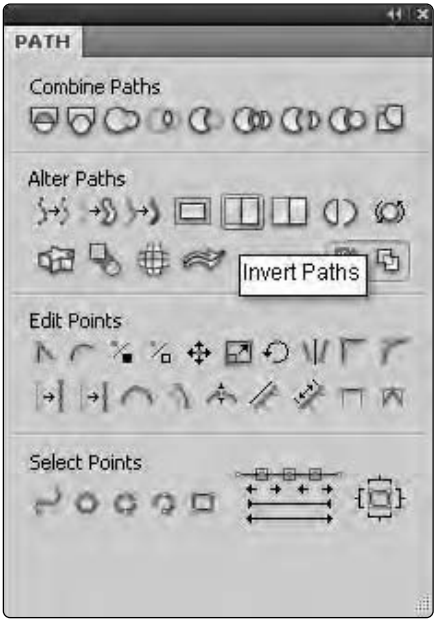
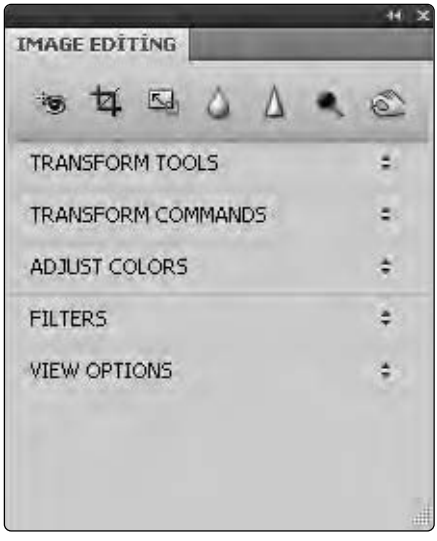


Image Editing Paneli

Bitmap resimleri düzenlemeye yarayan araçları içerir.



Special Characters Paneli

Çalışmalarınızda kullanabileceğiniz özel karakterleri içerir.



Document Library Paneli

Bu panel, grafik, animasyon ve tuş sembolleri içerir. Bunları panelden belgenize sürükleyip bırakarak kolayca kullanabilirsiniz.



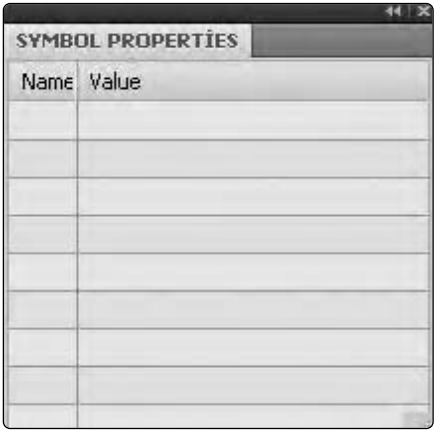
Common Library

Belgelerinizde kullanabileceğiniz, değişik kategorilerde tasnif edilmiş çok sayıda hazır sembol içerir.

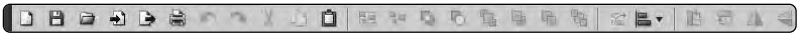


Symbol Properties Paneli

Grafik sembollerini yönetmek içindir. Properties panelinin yanında bulunur.



Tool Bar (Araç Çubuğu)



Araç çubuğu bir çok işlemi (hizalama, döndürme, sıralama, belge açma, kaydetme, kopyalama, yapıştırma, yazdırma) süratle yapmanızı sağlayan tuşları içerir. Araç çubuğunu dilerseniz **Window>Toolbars>Main** menüsünden açıp kapatabilirsiniz.

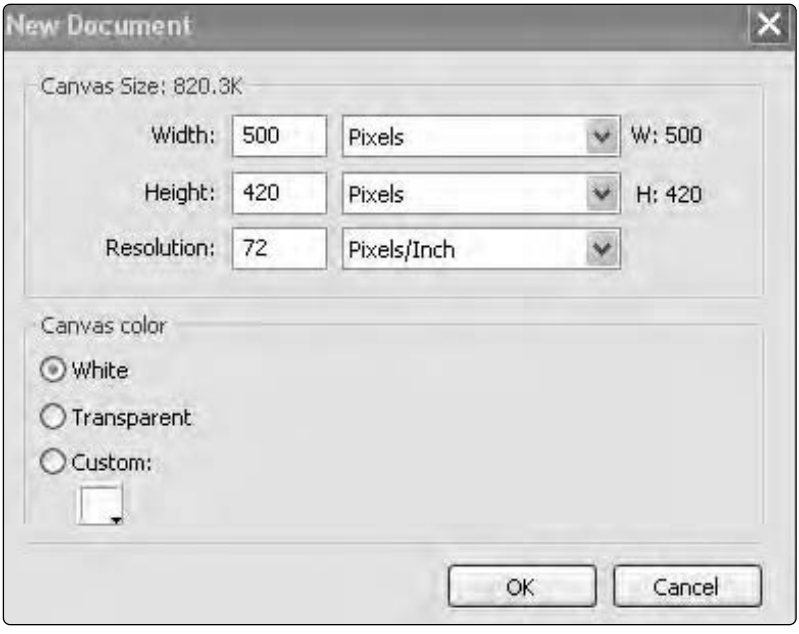
Belgeler ve Belge İşlemleri

Bu başlık altında, yeni belge oluşturma, belge açma, belge ithal etme işlemleri, belge büyütme ve kaydırma, tuval ebatlarını düzenleme, cetvel, kılavuz ve ızgara kullanımı konularına değineceğiz.

Yeni Belge Açma ve Kaydetme

Yeni bir belge oluşturmak için başlangıç sayfasındaki **Fireworks Document** (PNG) linkine tıklamanız yeterlidir. Dilerseniz bu işlemi **File>New** menüsü yardımı ile de yapabilirsiniz. Açılacak olan belge **PNG** formatında olacaktır. Fakat bu Fireworks ile sadece bu formatta çalışa-

bileceğimiz anlamına gelmemektedir. Grafiğinizi oluşturduktan sonra bunu **JPEG**, **GIF** ya da **Animated GIF** olarak kaydedebilir ve ihraç edebilirsiniz.



Ayrıca başka programlarla çalışmak amacıyla grafiğinizi **TIFF**, **PSD** ve **BMP** formatlarında da ihraç edebilirsiniz. Bir çalışmanızı farklı bir formatta ihraç etmeniz, orijinal PNG dosyanızın silinmesine sebep olmayacaktır ve daha sonra kullanabilmeniz için saklanacaktır.

Yeni bir belge açmak istediğinizde karşınıza yukarıdaki resimde gördüğünüz pencere gelecektir. Buradan üzerinde çalışacağınız tuvalin ebatlarını, çözünürlüğünü ve ardından **Canvas Color** kısmından tuvalin rengini belirleyeceksiniz. İsterseniz renk belirlemek yerine standart olan **White** (*Beyaz*) radyo düğmesini seçili bırakabilir ya da **Transparent** (*Saydam*) seçeneğini tercih edebilirsiniz.

Bir belgedeki nesneyi, yeni açacağınız bir belgeye yapıştırmak üzere kopyaladıktan sonra **File>New** dediğinizde, açılacak olan **New Document** penceresindeki tuval boyutları, panoya kopyaladığınız nesnenin

ebatları kadar olacaktır. Daha sonra nesneyi tuval üzerinde farenin **SAĞ** tuşuyla açacağınız kısayol menüsünden **Paste** komutu ile ya da **Edit>Paste** menüsü yardımıyla belgenize yapıştırabilirsiniz.

Çalışmanızı PNG yada PNG haricindeki bir formatta kaydetmek için File>Save As menüsünü kullanmalısınız. Fireworks ile dosyanızı kaydedebileceğiniz dosya formatları, **PNG**, **JPEG**, **AI**, **PSD**, **WBMP**, **TIFF**, **BMP**, **GIF**, **Animated GIF** ve **SWF** formatlarıdır.

Dosya İthal Etme

Bir dosyayı belgenize ithal etmek için **File>Import** menüsünü kullanın. Açılan pencereden dosyayı seçtikten sonra tuval üzerinde beliren üçgen şeklindeki işaretçiyi dosyayı yerleştirmek istediğiniz yere getirip fare ile tıkladığınızda dosya tuvalinize eklenmiş olacaktır.

Eğer dosyayı orijinal boyutlarında değil de istediğiniz bir başka ebatta ithal etmek isterseniz, ithal işaretçisini fareyi basılı tutarak sürükleyin ve istediğiniz ebadı elde ettiğinizde farenin tuşunu bırakın. Böylece dosya sizin istediğiniz ebatlarda ithal edilmiş olacaktır.

Büyütme ve Kaydırma İşlemleri

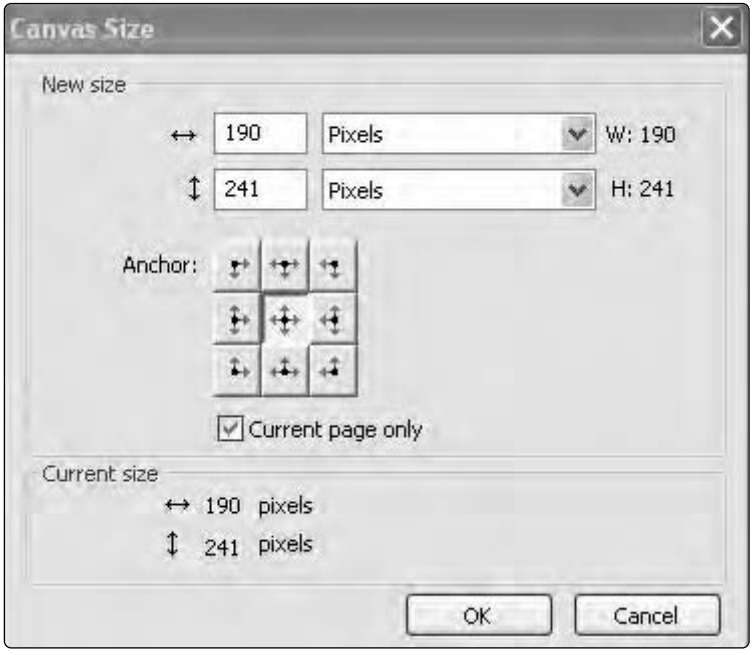
Bir belgeyi büyütmek istediğinizde **Tools** panelindeki **Zoom** aracını seçin ve belge içinde zoom odağı olmasını istediğiniz yere tıklayın. Her tıklama yapışınızda belge, programdaki bir sonraki büyüme oranında yakınlaşacaktır. Bu işlemi, tuvalin sağ alt kısmında bulunan **Set Magnification** açılır menüsünden istediğiniz büyüme oranını seçerek de yapabilirsiniz.

Bu işi yapmanın üçüncü yolu ise **View>Magnification** menüsüne tıkladığınızda açılacak olan pencereden yapmaktır. Fireworks ile bir işlemi yapmanın genelde birden çok yöntemi mevcuttur. Siz bunlardan dilediğinizi kullanabilirsiniz.

Belgeyi küçültmek istediğinizde ise yine **View>Magnification** ya da **Set Magnification** açılır menülerinden istediğiniz oranı seçerek belgenizi küçültebilirsiniz. Ya da bir üçüncü yol olarak zoom aracı seçili iken aynı anda **ALT** tuşuna da basılı tutup fare ile belgeye tıklayabilirsiniz.

Belgeyi tuval üzerinde kaydırmak için **Tools** panelindeki **Hand Tool** aracını seçtikten sonra fareye basılı tutarak belgeyi istediğiniz yöne kaydırabilirsiniz.

TUVAL (CANVAS) DEĞİŞİKLİKLERİ



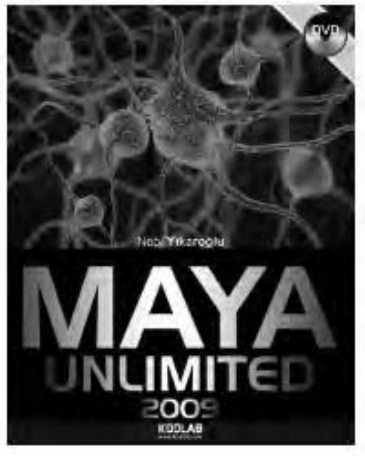
Üzerinde çalıştığınız tuvalin ebatlarını değiştirmek istediğinizde **Modify>Canvas>Canvas Size** menüsünü kullanmalısınız. Menüye tıkladığınızda resimde gördüğünüz pencere açılacaktır. Buradan tuvalin yeni `genişlik` ve `yükseklik` değerlerini belirleyebilirsiniz.

Pencerede gördüğünüz **Anchor** tuşlarını kullanarak değişikliğin tuvalin neresine uygulanacağını ayarlayabilirsiniz. Resimde görüldüğü gibi ortadaki anchor seçili olduğunda değişiklik tuvalin tüm kenarlarına uygulanacaktır. Bu işlemi belgedeki nesne seçili değilken **Properties** panelindeki **Canvas Size**... tuşu ile de yapabilirsiniz.

Tuvalin rengini değiştirmek istediğinizde ise **Modify>Canvas>Canvas Color** menüsünü kullanacaksınız. Yine bu işlemi de dilerseniz belgede-

ki nesne seçili değilken **Properties** panelindeki **Canvas** renk paletini tıklayarak da yapabilirsiniz.

Tuvalinizi döndürmeniz gerektiğinde eğer saat yönünde `180 derece` döndürmek istiyorsanız **Modify>Canvas>Rotate 180º** menüsünü, saat yönünde `90 derece` döndürmek istiyorsanız **Modify>Canvas>Rotate 90ºCW** menüsünü ve saat yönünün tersine `90 derece` döndürmek istiyorsanız **Modify>Canvas>Rotate 90ºCCW** menüsünü kullanmalısınız.



Tuvalinizin ebatları nesnenizin ebatlarından farklı ise ve siz tuvalinizi kırpmak istiyorsanız bu işlemi belgedeki nesne seçili değilken **Properties** panelindeki **Fit Canvas** tuşu ile ya da **Modify>Canvas>Fit Canvas** menüsü aracılığı ile yapabilirsiniz.

Cetvel, Kılavuz ve Izgara Kullanımı

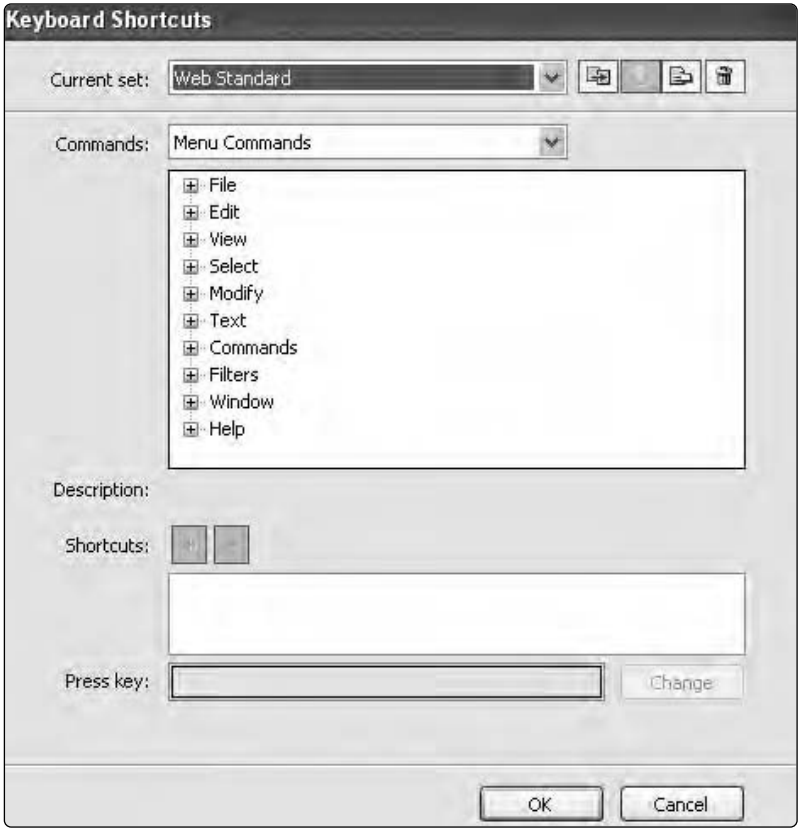
Cetvel kullanmak, çalışmanızı ölçme ve hizalama konusunda işinizi kolaylaştıracaktır. Fireworks'te cetveller piksel cinsinden ölçüm yaparlar. Cetvelleri kullanabilmek için **View>Rulers** menüsünü tıklamanız yeterlidir.

Kılavuzlar ise, nesneleri tuval üzerinde göz kararı değil de net bir şekilde yerleştirmenizi sağlarlar. Kılavuzlar, cetveller üzerinde iken, farenin **SOL** tuşunu basılı tutarak, tuvaliniz üzerinde, gereken yere sürükleyip bırakılarak çizilirler.

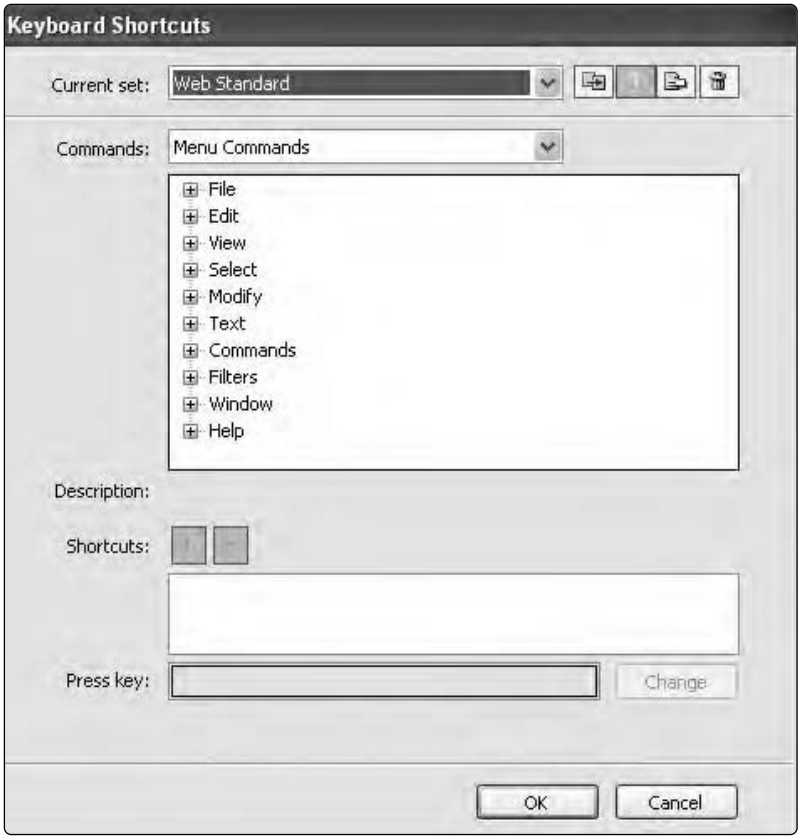
Nesnelerinizi, bu kılavuzlara tutturabilirsiniz. Bunun için **View>Guides>Snap to Guides** menüsünün seçili olması gerekir. Dilerseniz kılavuzları kilitleyebilirsiniz. Bunun için **View>Guides>Lock Guides** menüsünü kullanmalısınız. Kılavuzlar arasındaki mesafeyi öğrenmek için **SHIFT** tuşunu basılı tutup fare ile tuvaliniz üzerinde gezindiğinizde Fireworks imlecin çevresindeki kılavuz çizgilerinin mesafesini piksel olarak gösterecektir.

Izgaralar, nesneleri konumlandırma konusunda en önemli yardımcılarınızdan biridir. Izgaraları görüntülemek içinse **View>Grid>Show Grid** menüsünün seçili olması gerekir. Nesnelerinizi ızgaralara tutturabilmek içinse **View>Grid>Snap to Grid** menüsünü kullanın.

KLAVYE KISAYOLLARI



Fireworks komutlarının çoğuna kısayol tuşlarıyla ulaşmak mümkündür. Kısayol kullanımı, birçok işlemi daha hızlı bir biçimde yapmanıza olanak tanır. Ayrıca başka bir programdan (örn: Photoshop) alışkın olduğunuz kısayol kullanımlarını Fireworks ile de kullanabilirsiniz.



Bu işlem için; **Edit>Keyboard Shortcuts** menüsünü tıkladığınızda açılan, resimde gördüğünüz pencereden yararlanacağız. Penceredeki **Current set** açılır kutucuğunda, kısayollarını kullanmaya alışık olduğunuz programı seçerek kullanabilirsiniz.

Resimdeki pencerede **Current set** açılır kutucuğunda gördüğünüz kısayol setleri değiştirilemez. Fakat dilerseniz kendinize özel bir kısayol seti oluşturabilirsiniz.

Bunun için **Current set** açılır kutucuğunda Fireworks seçiliyken hemen sağdaki **Duplicate Set** tuşuna tıklayın. Açılacak pencereyi **OK** tuşu ile geçtiğinizde **Fireworks copy** adında bir kısayol seti oluşturmuş olacaksınız. Oluşturduğunuz kısayol setini artık **Current set** açılır penceresinde görebilirsiniz.

Kısayol setinizi oluşturdunuz fakat henüz herhangi bir kısayol kombinasyonu eklemediniz. Bunun için önce **Commands** açılır kutucuğunda gördüğünüz seçeneklerden birini tercih etmelisiniz.

Bu seçenekler ve anlamları şöyledir:

Menu Commands: Menü çubuğundaki komutlar için kısayol oluşturmanıza olanak tanır.

Tools: **Tools** panelindeki araçlar için kısayol oluşturmanıza olanak tanır.

Miscellaneous: Belirlenmiş işlemler için (örn: animasyon oynatma, Bitmap düzenleme... vb) kısayol oluşturmanıza olanak tanır.

Seçiminizi yaptıktan sonra aşağıdaki kutudan kısayol oluşturacağınız komutu seçin ve **Shortcut** kısmındaki **Add a new shortcut** (**+**) tuşuna tıklayın. Farenizin imleci alttaki **Pres Key** kutucuğunun içinde olacaktır. Şimdi bu kutucuğa istediğiniz tuş kombinasyonunu girin ve işlemi **OK** tuşu ile tamamlayın. Artık kısayol atadığınız komut, oluşturduğunuz tuş kombinasyonu ile açılacaktır.

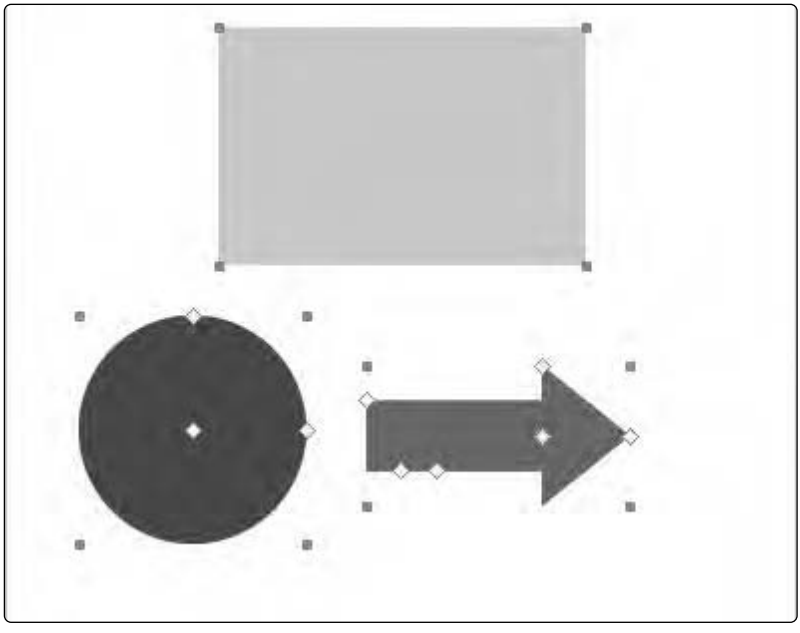
NESNELERLE ÇALIŞMAK

Bu bölümde nesneleri seçme, seçimi değiştirme, seçili nesneleri düzenleme, klonlama, dönüştürme, istifleme, hizalama... vb konuları inceleyeceğiz.

NESNELERİ SEÇME

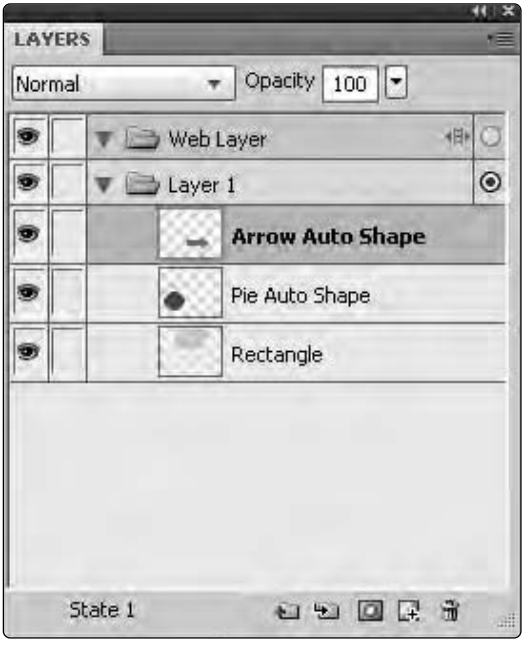
Bir nesne üzerinde herhangi bir işlem gerçekleştirebilmek için öncelikle nesnenin seçilmesi gerekir. Şimdi seçme işlemini hangi yöntemlerle yapabileceğimizi inceleyelim.

İlk olarak bir nesneyi **pointer** aracı ile seçmek için nesneyi tıklamanız yeterlidir. Nesne seçildiğinde nesnenin etrafında seçimin yapıldığı anlamına gelen mavi renkli tutamaçlar belirecektir.

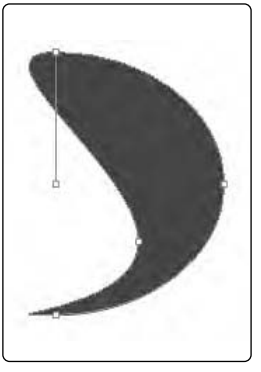


Pointer aracı ile farenin **SOL** tuşuna basılı tutup sürükleyerek tuvaldeki tüm nesneleri kapsayan bir seçim yaptığınızda tüm nesneler tek seferde seçili hale gelmiş olacaklardır.

Pointer aracı ile bir nesneyi seçtikten sonra, seçime bir başka nesneyi de dahil edebilmek için ilk nesne seçiliyken **SHIFT** tuşuna basılı tutarak diğer nesneye tıklamanız yeterlidir.



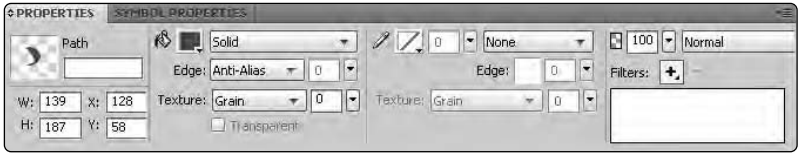
Tüm nesneleri pointer aracı ile tıklayarak **Layer** panelinden de seçebileceğinizi belirteyim. Panelde katman grupları klasörlerde bulunurlar. Klasörlerin yanındaki ok işaretlerine tıklayarak katmanları açık hale getirebilirsiniz. **Layer** panelinden seçtiğiniz bir nesne tuval üzerinde de seçili hale gelecektir.



Subselection aracı ile seçim yapmak için nesneye tıkladığınızda nesnenin çevresinde içi boş küçük karelerden oluşan tutamaçlar belirecektir. Bu tutamaçların üzerine fare imlecini getirirseniz fare imlecinin şekli üçgene dönüşecektir. Farenin **SOL** tuşuna basılı tutarak noktaları sürüklediğinizde nesnenin şeklini istediğiniz gibi değiştirebilirsiniz.

Select Behind aracı ile birçok nesne arasında görünmeyen ya da alt kısımda kalan nesneleri seçebilirsiniz. İlk tıklamanızda en üstteki nesne seçilecektir. Bir alttaki nesneyi seçebilmek için en üstteki nesneye fare

ile ikinci kez tıklamalısınız. Seçili olan nesneyi bulunduğu konumdan almak için fareyi basılı tutarak sürükleyin. Bu yöntemle her tıklayışınızda bir alttaki nesneyi seçebilirsiniz. En alttaki nesne seçil iken bir kez daha tıkladığınızda tekrar en üstteki nesne seçili duruma gelecektir.



Siz bu seçimleri yaparken alttaki **Properties** panelinde yaptığınız seçimlerle ilgili bilgiler ve özelikler görüntülenecektir.



Tools panelindeki **Bitmap** kısmında bulunan **Marquee**, **Oval Marquee**, **Lasso**, **Polygon Lasso** ve **Magic Wand** araçları ile Bitmap resimlerindeki istediğiniz alanları düzenlemek üzere seçebilirsiniz. Bu araçlardan birisi ile seçim yaptığınızda seçime ek olarak bir başka seçim daha yapmak isterseniz araç seçili iken **SHIFT** tuşuna basılı tutarak yeni seçiminizi yapabilirsiniz.

Seçimlerinizi yaparken **Properties** panelindeki **Edge** açılır menüsünden seçimlerinize üç farklı kenar seçeneği uygulayabilirsiniz. Bunlardan **Hard**, keskin kenarlı seçimler oluşturur. **Anti-alias**, çentikli kenarların oluşmasını önler. **Feather** ise seçim alanının kenarlarını yumuşatır. Yumuşatma işlemini **Select>Feather** menüsünden de yapabilirsiniz.

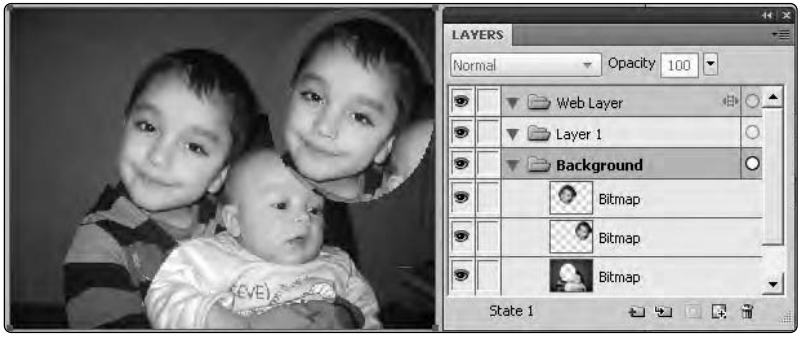


Yapmış olduğunuz seçimi daha sonra kullanılmak üzere kaydedebilirsiniz. Bunun için ilk seçiminizi yaptıktan sonra **Select>Save Bitmap Selection** menüsünü kullanın. Ardından, açılan pencereden Name kutucuğuna bir isim girerek **OK** ile seçiminizi kaydedin. İkinci bir seçim daha yaptığınızda da yine aynı aşamaları tekrar ederek resimde gördüğünüz pencereye tekrar geldiğinizde, **Selection** kısmından ilk kaydınızın adını seçin. **Operation** kımından **Add to selection** radyo düğmesini işaretleyip **OK** dediğinizde ikinci seçiminiz de kaydedilerek birinci seçiminize eklenecektir. Kaydınızı kullanmanız gerektiğinde **Select>Restoer Selection** menüsünden yararlanabilirsiniz.

Seçiminizi yaptıktan sonra bunu nesne üzerinde başka bir alana kaydırmanız gerektiğinde fare ile tutup sürüklemeniz yeterlidir. Nesnenin seçtiğiniz kısmını taşımak içinse seçim için kullandığınız Bitmap aracını

seçim alanında iken **CTRL** tuşuna basılı tutup ya da **Pointer** aracını tıklayıp fareyi seçili alan üzerinde basılı tutarak seçili kısmı istediğiniz gibi sürükleyebilirsiniz.

Nesneleri Kesme-Kopyalama



Bir nesne üzerinde yapmış olduğunuz seçimi keserek ve kopyalayarak yeni bir Bitmap katmanı oluşturabilirsiniz. Örneğin **Oval Marquee** aracı ile seçiminizi yaptıktan sonra **Edit>Insert>Bitmap via Cut** menüsünü tıklayın. Böylece yaptığınız seçim kesilerek yeni bir Bitmap nesne oluşturulmuş oldu. Nesneyi **Layers** panelinde görebilirsiniz.

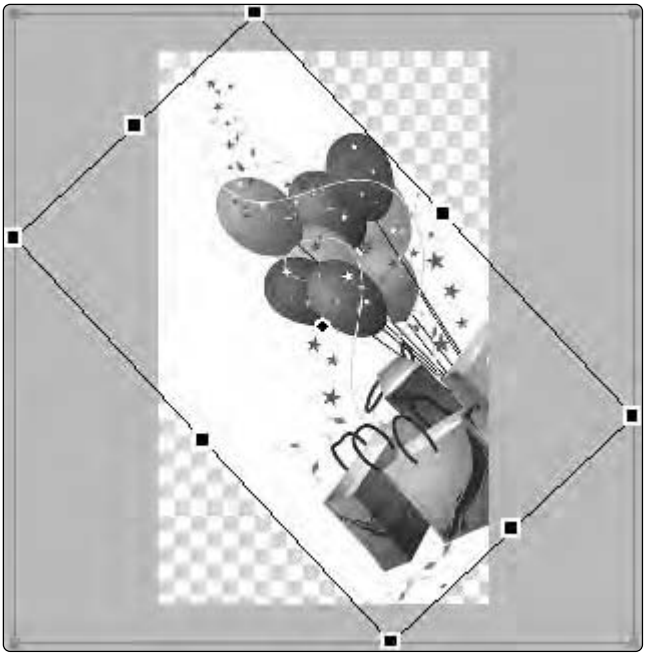
Ardından **Edit>Insert>Bitmap via Copy** menüsünü tıklayarak kopyalayın. Artık **Pointer** aracını kullanarak yeni nesneyi tuval üzerinde istediğiniz konuma sürükleyerek taşıyabilirsiniz.

Bir nesne üzerinde yapmış olduğunuz seçimi keserek veya kopyalayarak bir başka uygulamada kullanabilmek için Edit menüsünde bulunan **Cut** (**CTRL+X**) komutu ile kesin ya da **Copy** (**CTRL+C**) komutuyla kopyalayın. Yeni bir belge açtığınızda yeni belgenin ebatları kopyaladığınız nesne ebatlarında olacaktır. Son olarak **Edit>Paste** (**CTRL+V**) komutuyla yapıştırın. Dilerseniz bu işlemi araç çubuğundaki (*Toolbars*) tuşlar yardımıyla da yapabilirsiniz.

Bir nesneyi çoğaltmanın bir diğer yolu da onu klonlamaktır. Bu işlem için **Edit>Clone** menüsünü kullanmalısınız. Oluşan kopya orijinal nesnenin üzerinde oluşturulur. Öğrendiğiniz yöntemlerle bu kopyayı farklı bir belgeye taşıyabilir ya da tuval içinde farklı bir konumda kullanabilirsiniz.

Seçili bir nesneyi silmek için klavyeden **DELETE** tuşuna basmalısınız. Bu işlemi **Edit>Clear** menüsü ile de yapabilirsiniz.

Nesneleri Döndürme



Bir nesneyi döndürmek için **Pointer** aracı seçili iken, **Scale**, **Scew**, **Distort** veya **9-slice scaling** araçlarından birisini seçtiğinizde nesnenin etrafında küçük kare şeklinde tutamaçlar belirecektir. Fare imlecini tutamaçların yakınına getirdiğinizde döndürme işareti belirecektir. Bu durumda fareyi basılı tutarak nesneyi istediğiniz yönde döndürebilirsiniz.

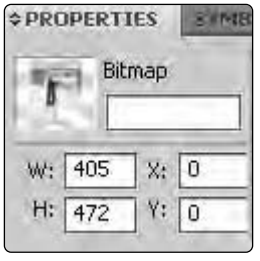
Nesneyi döndürme işlemini, dilerseniz tuvali nesne ile birlikte döndürerek de yapabilirsiniz. Bunun için **Modify>Canvas>Rotate90º** ya da **180º** derece komutlarını kullanabilirsiniz. Aynı işlemi **Toolbar**'daki **Rotate** tuşları yardımıyla da uygulayabilirsiniz.

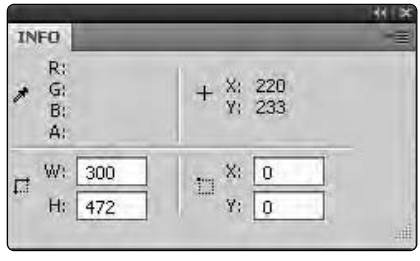
Bir nesneyi dikey ve yatak eksende ters çevirmek için **Modify>Trans-**

form>Flip Horizontal ya da **Vertical** menülerini kullanabilirsiniz. Aynı işlemi **Toolbar**'daki **Flip** tuşlarını kullanarak da yapabilirsiniz.

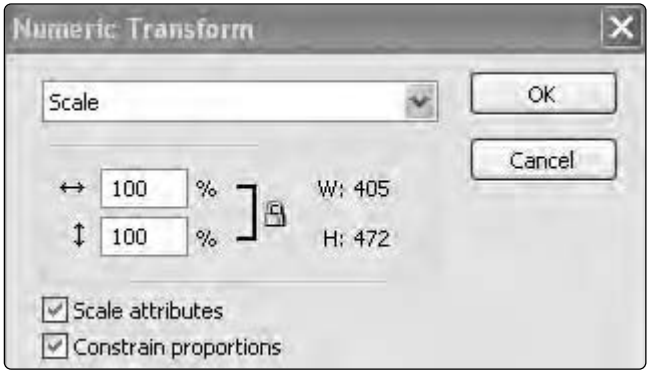
Nesneleri Ölçeklendirme

Bir nesneyi **Scale** aracını ya da **9-slice scaling** aracını kullanarak ölçeklendirebilirsiniz. Nesnede bir bozulma ya da çarpıklık olmaması için bu işlemi **9-slice scaling** aracı ile yapmalısınız.





Ölçeklendirmenin bir diğer yoluna da değinelim. **Pointer** aracı ile nesneyi seçtiğinizde **Properties** panelinde ve **Info** panelinde nesneye ait ebatlar görüntülenecektir. Genişliği değiştirmek için **W** kutucuğundaki rakamsal değeri, yüksekliği değiştirmek için de **H** kutucuğundaki rakamsal değeri değiştirmelisiniz. Sözünü ettiğim Info panelini **Window>Info** menüsünden açabilirsiniz.



Daha önce de söylediğim gibi, Fireworks'te bir işlemi gerçekleştirmenin genelde birçok yolu vardır. Zaten bu durumla şu ana kadar bir çok kez karşılaştınız. İşte o durumlardan biri daha geliyor.

Şimdi **Modify>Transform>Numeric Transform** menüsüne tıklayın. **Numeric Transform** penceresi açılacaktır. Bu penceredeki açılır kutuda bulunan **Scale** seçeneği ile bir nesneyi ölçeklendirebilir, **Resize** seçeneği ile nesneyi yeniden boyutlandırabilir, **Rotate** seçeneği ile de nesneyi döndürebilirsiniz.

Nesneleri Gruplama

Fireworks ile birçok nesneyi gruplayarak, grupladığınız nesneleri sanki tek bir nesneymiş gibi kullanabilirsiniz. Bu işlemi gerçekleştirmek için önce Pointer arası ile tüm nesneleri seçin. Ardından **Modify>Group** menüsüne tıklayın. **Layer** paneline baktığınızda tüm nesnelerin tek bir nesne olarak gruplandığını görebilirsiniz.

Grubu çözmek için **Modify>Ungroup** menüsünü, grubu çözmeden grup içindeki bir nesneyi seçebilmek için de **Subselection** aracını kullanabilirsiniz.

Nesneleri İstifleme

Tuvalinizde birçok nesneyi üst üste istiflediğinizi var sayalım. Bu durumda nesnelerin yerlerini değiştirmek istediğinizde, mesela üst üste yer alan üç nesneniz varsa ve siz ortadaki nesneyi en üste çıkarmak istiyorsanız ne yapabilirsiniz? Bu işlem için **Modify>Arrange** menüsünün altın yer alan alt menülerden yararlanacağız.

Buradaki alt menülerden **Bring to Front** ile bir nesneyi istifin en üstüne çıkarabiliriz. **Send to Back** ile bir nesneyi istifte alta gönderebiliriz. **Bring Forward** ile bir nesneyi bir basamak üste çıkarır, **Send Backward** ile de bir nesneyi bir basamak alta göndeririz.

Tahmin edeceğiniz gibi Fireworks ile bu işlemleri yapmanın da başka yolu var. **Layer** panelinde istifteki nesneleri bir katman klasörünün altında göreceksiniz. Buradan her bir nesneyi fare ile tutup ortaya, en alta ya da en üste sürükleyerek taşıyabilirsiniz.

Nesneleri Hizalama



Tuvalinizdeki nesneleri hizalama işlemini **Modify>Align** menüsünün alt menüleri ile ya da **Align** paneli ile yapabilirsiniz. Nesneleri yatay, dikey, sağa, sola, ortaya alta ve üste hizalamanız gerektiğinde Fireworks'ün bahsettiğim hizalama özellikleri ile kolaylıkla yapabilirsiniz.

Bitmap Grafikler

Piksel denilen kare şeklindeki noktacıklardan oluşan **Bitmap grafikler**, adeta bir karodaki mozaik parçaları gibi ızgara şeklinde yerleştirilmişlerdir. Bu bölümde Fireworks ile Bitmap nesnesi oluşturma, bitmap nesnelerde renkler ve renk ayarları... vb konular üzerinde duracağız.

Bitmap Nesnesi Oluşturma

Boş bir Bitmap nesnesi oluşturmak için **Edit>Insert>Empty Bitmap** menüsüne tıklayın. Bir vektörel nesneyi Bitmap nesnesine dönüştürmek içinse **Modify>Flatten Selection** menüsünü tıklamalısınız. Vektörel nesneler Bitmap grafiklerine dönüştürülebilirler fakat Bitmap grafikleri vektörel nesnelere dönüştürülemezler.

Bitmap Nesnesi Çizme

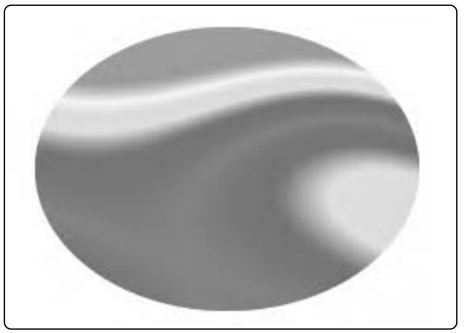
Bir Bitmap nesnesi çizmek için **Pencil** aracını kullanırız. Nesnenizi çizerken size yardımcı olacak **Properties** özelliklerinden **Anti-ali-**

ased ile çizgilerin kenarlarını yumuşatabilirsiniz. **Preserve Transparency** ile de grafiğin saydam kısımlarında çizim yapılması engellenebilir. **Pencil** aracı ile çizim yaparken düz çizgiler çizebilmek için **SHIFT** tuşuna basılı tutmanız gerekir.

BOYAMA

Serbest boyama yapabilmek için **Brush** aracını, dolgu rengi uygulamak içinse **Paint Bucket** aracını kullanın.

GRADIENT



Gradient aracı, nesnelere renk geçişleri uygulamak için kullanılır. Aracı seçtikten sonra **Properties** panelindeki renk kutusundan renk seçebilir ve renk geçişi ayarlarını ön izleyerek istediğiniz şekle gelene kadar değiştirebilirsiniz.

RENK ÖRNEKLEME

Eyedropper aracı çizgi ve dolgu rengi örneklemenizi sağlayan bir araçtır. Böylece bir rengi tekrar kullanmak çok kolay ve hızlı uygulanabilen bir işleme dönüşmüş olur. Şimdi yeni bir belge açın ve tuvale **Rectangle** aracı ile iki tane dikdörtgen çizin.

Dikdörtgenlerden bir tanesini istediğiniz her hangi bir renk ile doldurun. Ardından **Eyedropper** aracı ile renge tıklayın. Böylece rengi örneklemiş oldunuz. Şimdi **Paint Buket** aracı ile ikinci dikdörtgene tıkladığınızda bu dikdörtgenin rengi örneklediğiniz ilk dikdörtgenin rengi ile doldurulacaktır.

Klonlama

Rubber Stamp  aracı ile resmin bir bölgesini bir başka bölgesine klonlamak mümkündür. Bu sayede resmin bir yerindeki kusurları rahatlıkla kapatmış olursunuz. **Tools** panelinden **Rubber Stamp** aracını seçtikten sonra resmin örnekleme yapacağınız yerine tıklayın. Daha sonra resmin düzelteceğiniz kısmına tıklayarak örneklemeyi buraya klonlayın. Yeni bir örnek almanız gerektiğinde **ALT** tuşuna basılı tutarak örnek alacağınız yere tıklamalayın.

Bulanıklık ve Keskinlik Verme

Blur  aracı uygulandığı alanı bulanıklaştırır. **Sharpen**  aracı ise uygulandığı alanı keskinleştirir. **Blur** ve **Sharpen** araçları için **Properties** panelinde yer alan özelliklerden **Size**, fırçanın boyutunu ayarlamak içindir. Edge, fırça yumuşaklığını, **Shape** ise fırçanın yuvarlak mı kare şeklinde mi olacağını belirler. **Intensity Blur** aracında bulanıklaştırma oranını, **Sharpen** aracı içinse keskinleştirme oranını gösterir.

Renkleri Katıştırma

Smudge  aracı renkleri katıştırmak için kullanılır. Böylece Bitmap üzerinde farklı efektler oluşturmak mümkündür. **Smudge** aracı için **Properties** panelinde yer alan özelliklerden Pressure, aracın uygulanma şiddetini belirlemek içindir. **Smudge color** kısmı ise, katıştırmada kullanılacak renkleri seçmemize olanak tanır.

Eğer bu kutucuğu seçili hale getirmezseniz araç, imlecin altındaki renkleri katıştıracaktır. **Use entire document** ise katıştırma işleminin tüm katmanlardaki tüm nesneler dikkate alınarak yapılmasını sağlamak içindir. Bu kutucuğu seçili hale getirmezseniz **Smudge** aracı yalnız aktif nesnenin rengini katıştırır.

Aydınlatma ve Karartma

Dodge  aracı, uygulandığı yerleri aydınlatır. **Burn**  aracı ise tam tersine uygulandığı yerleri koyulaştırır. Bu araçlar için **Properties** panelinde göreceğiniz **Exposure** özelliği, resme uygulanacak aydınlatma ve karartma seviyelerinin tespiti için kullanılır. Bu özelliğe `1` ile `100` arasında bir değer atanabilir.

Renk Kaplama

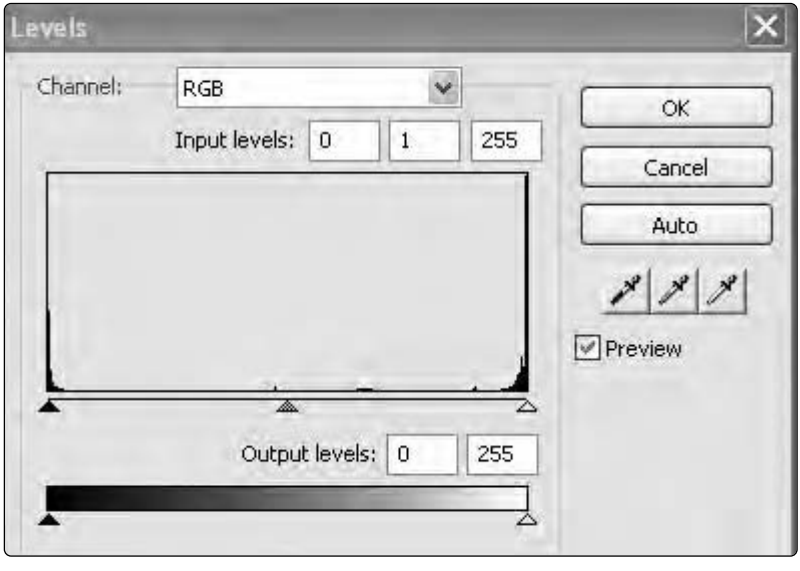
Bir başka önemli araç da, bir rengin üzerini başka bir renkle kaplamaya yarayan **Replace Color** aracıdır. Bu araçlar için **Properties** panelinde göreceğiniz **Tolerance** özelliği, eski rengin yerini alacak renge yakın renklerin değiştirilmesi içindir. `0` ile `100` arası bir değer alır. Değer arttıkça daha çok yakın renk değiştirilir. **Strength** özelliği ise renk değişim miktarını belirler.

Kırpma İşlemi

Kırpma işlemi **Crop** aracı ile gerçekleştirilir. Böylece resimdeki istenmeyen kısımlardan kurtulmuş olursunuz. **Crop** aracı ile çizdiğiniz çerçeve içindeki alan, kalmasını istediğiniz alanı ifade eder. Kırpılacak kısım çerçeve dışında kalan alandır. Çerçeveyi belirledikten sonra seçtiğiniz alanda fare ile çift tıkladığınızda ya da **ENTER** tuşuna bastığınızda kırpma işlemi gerçekleşecektir. Fazlalıklar tuvalle birlikte kırpılacaktır. Dilerseniz kırpma işlemini, seçimi **Marquee** aracı ile yaptıktan sonra **Edit>Crop Document** menüsünü tıklayarak da yapabilirsiniz.

Filtreler

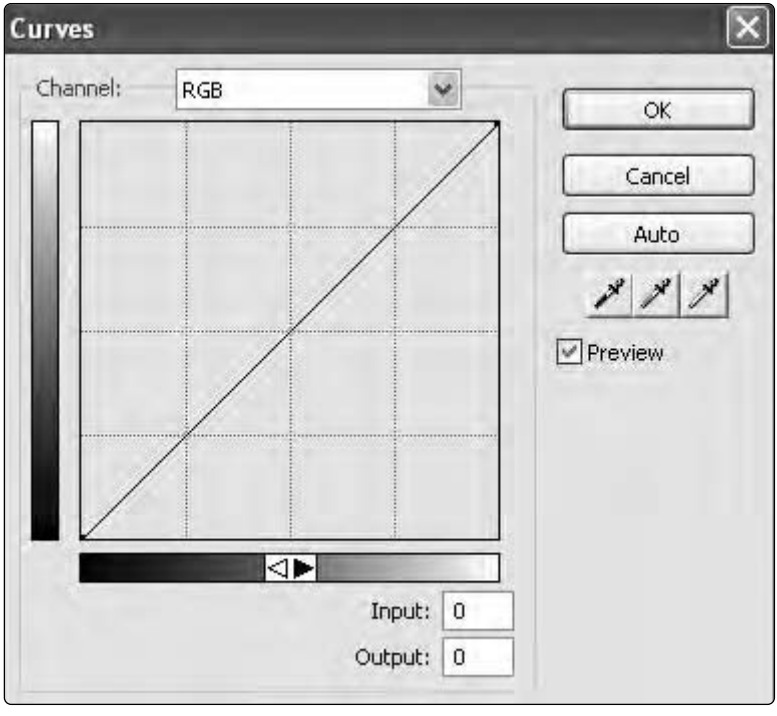
Levels Özelliği ile Renk Ayarları



Bu özellik resimde daha keskin bir görüntü elde etmek için kullanılır. Bu işlemi piksellerdeki renkleri tekrar dağıtarak yapar. Bir Bitmap üzerindeki piksellerin dağılım durumunu görmek için **Filters>Adjust Color>Levels** menüsünden ya da **Properties** panelindeki **Filters** (**+**) tuşu ile açılan menüden ulaşacağınız **Level** penceresinden görebilirsiniz. Histogramın altındaki kaydırma tutamaçlarını kullanarak renk dağılımını değiştirebilirsiniz. Pencerenin en altında yer alan ve açıktan koyuya doğru giden **Output levels** kısmındaki tutamaçlar yardımıyla da resmin kontrastını değiştirebilirsiniz.

Bu işlemleri sizin yerinize Fireworks'ün otomatik olarak yapmasını isterseniz, **Filters>Adjust Color>Auto Levels** menüsünden yapabilirsiniz. **Auto Levels** menüsüne, **Properties** panelindeki **Filters** (**+**) tuşu ile açılan menüden de ulaşabilirsiniz.

Curves Özelliği ile Renk Ayarları

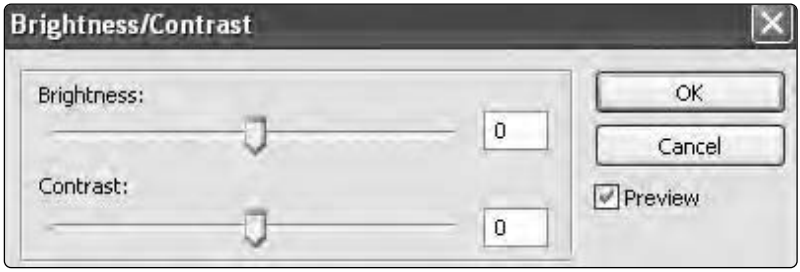


Levels özelliğine benzer. Fakat renk ayarını eğrilerle yaptığından renk tonlarına daha fazla kontrol imkanı sağlar. Özelliğe **Filters>Adjust Color>Curves** menüsünden ya da **Properties** panelindeki **Filters** (**+**) tuşu ile açılan menüden ulaşabilirsiniz. Açılan pencerede gördüğünüz düz çapraz çizgi ve `0` değerini gösteren input ve output kutucukları resme henüz herhangi bir müdahalenin olmadığı anlamına gelir.

Şimdi çapraz çizginin üzerindeki noktaya tıklayıp sürükleyerek istenilen değişikliği yapabilirsiniz. Çizgi üzerindeki bir başka yere tıklayarak bir başka ayar noktası oluşturabilir ve böylece daha hassas ayar yapma imkanına kavuşabilirsiniz. Oluşturacağınız her noktanın input ve output değerleri farklı olacaktır.

Noktaları sürüklerken kutucuklardaki değişimi gözlemleyebilirsiniz. Eğri üzerinde oluşturduğunuz noktaları silmek için fareye basılı tutarak pencere dışına sürükleyin. Pencerede gördüğünüz **Auto** tuşu ile de dilerseniz ayarlamayı otomatik olarak Fireworks'e yaptırabilirsiniz. **Properties** panelinde biraz evvel kullandığımız Filters kısmında yaptığınız işlemlerin listesini görebilirsiniz. Buradan bir işlemi seçerek **–** tuşuna tıkladığınızda yaptığınız işlem iptal edilecektir.

Brightness (Parlaklik) ve Contrast (Karşıtlik) Ayarları



Eğer resminiz çok karanlık ya da çok parlak ise parlaklık ve karşıtlık ayarlarını değiştirerek daha güzel bir görüntü elde edebilirsiniz. Bu işlemleri **Filters>Adjust Color>Brightness-Contrast** menüsü yardımıyla açılan penceren gerçekleştirebilirsiniz. Ayrıca pencereye **Properties** panelindeki **Filters** (**+**) tuşu ile açılan menüden de ulaşabilirsiniz. Pencerede gördüğünüz kaydırma tutamaçları ile parlaklık ve karşıtlık

değerlerini değiştirebilir, pencerede yer alan **Preview** kutucuğu işaretli ise yaptığınız değişiklikleri tuval üzerinde görebilirsiniz. Kutucuğu boşaltırsanız resmin ilk halini görürsünüz.

Renk Değiştirme

Bitmap ve vektörel grafiklerin renklerini değiştirmenin bir yolu da **Color fill** özelliğidir. Bu özelliğe **Properties** panelindeki **Filters** (+) tuşu ile açılan menüden ulaşabilirsiniz. Bu özellik ile dilerseniz resimdeki tüm pikselleri belirlediğiniz renkle doldurabileceğiniz gibi, dilerseniz de açılan **Blend Modes** kutucuğundan yaralanarak renk karışım modlarını kullanarak değişik etkiler elde edebilirsiniz.

Renkleri Tersine Çevirme

Resminizdeki tüm renkleri tersine çevirmek için **Filters>Adjust Color>Invert** menüsünü ya da **Properties** panelindeki **Filters** (+) tuşu ile açılan menüyü kullanabilirsiniz.

Renkleri Bulanıklaştırma

Renkleri bulanıklaştırmak için **Blur** filtresini kullanacağız. **Blur** filtresi, **Filters** menüsü altında yer alır ve birçok farklı Blur uygulaması içerir. Bunlardan **Blur**, netliği biraz yumuşatırken**, Blur More**, **Blur** seçeneğinden daha fazla bir bulanıklık temin eder. Gaussian Blur sisli bir görüntü oluşturur. **Motion Blur** resme hareket hissi katar. **Radial Blur** resmi büker. **Zoom Blur** ise resme hız ve hareket etkisi verir.

Renkleri Keskinleştirme

Sharpen filtresi, **Blur** filtresinin tam tersi bir işleve sahiptir. Filters menüsü altında yer alır. Resmi keskinleştirmek için kullanılır. Üç değişik uygulaması mevcuttur. Bunlardan **Sharpen**, resimdeki bulanıklıkları giderir. **Sharpen More**, daha keskin bir görüntü oluşturur. **Unsharp Mask** ise sahip olduğu pencere ile daha detaylı ayarlama yapma imkanı verir.

Vektörel Nesneler

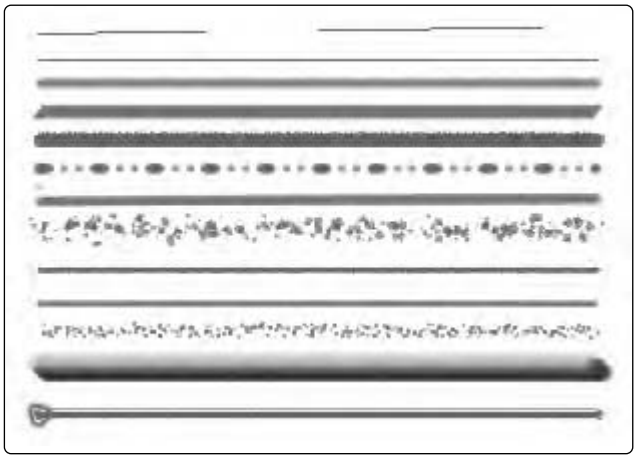
Bölümün başında vektörel grafikler için, **vektör** denilen, konum ve renk bilgisi içeren çizgiler ve eğriler kullanılarak oluşturulurlar demiş-

tik. Bu çizgiler yanyana gelerek resmi oluştururlar. Grafiği oluşturan bu çizgiler nesne olarak tanımlanır.

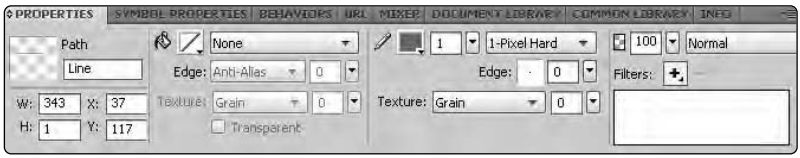
VEKTÖREL NESNE ÇİZME

Fireworks ile vektörel nesneler oluşturma noktasında bir çok özellik ve araç kullanılmak üzere sizi beklemekte. Fireworks'te serbest çizim araçlarını kullanarak karmaşık şekiller oluşturabilirsiniz. İhtiyacınız olan kare, daire üçgen... vb şekillerse Fireworks bu konuda sizi yormaz ve size hazır olarak bir çok şekil sunar. Şimdi bunları detaylı olarak inceleyelim.

ÇİZGİLER OLUŞTURMA



Çizgi çizmek için **Line** aracını kullanacağız. Aracı seçtikten sonra **Properties** penceresi yardımıyla değişik özelliklerde ve renklerde çizgiler oluşturabilirsiniz.

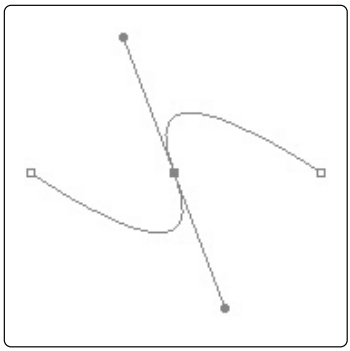


Properties penceresindeki renk kutusundan çizginizin rengini belirleyebilirsiniz. **Tip size** kutucuğundan çizginin kalınlığını ayarlayabilirsiniz. **Stroke category** tuşuna tıkladığınızda açılan menüden çizgi tipini,

Texture kısmından ise çizgiye uygulanacak dokuyu seçebilirsiniz. Blend Mode ile uygulanabilecek karışım modlarını belirleyebilirsiniz. Bu arada, düz çizgi çizebilmek için **SHIFT** tuşunu basılı tutmanız gerektiğini de hatırlatayım.

Serbest Çizimler

Fireworks ile serbest çizimler yaparak istediğimiz grafikleri oluşturabiliriz. Bunun için yine vektör araçlarının bazılarını kullanacağız. İşte o araçlar;



Pen Aracı  ile düz ya da eğri çizgiler oluşturacağız. Bunu yapabilmek için önce tuvalde **Pen** aracı ile bir noktaya tıklayın. Daha sonra bir başka noktaya daha tıkladığınızda bu iki nokta arasında noktaları birbirine bağlayan bir vektör yolu çizilir. Aracı denediğinizde ikinci noktayı tıkladıktan sonra da aracın fareyi takip eden bir vektör yolu çizdiğini göreceksiniz.

Siz bu noktada çalışmanızı sonlandırmak istiyorsanız son oluşturduğunuz noktaya çift tıklamalısınız. Bir başka sonlandırma şekli de **Pen** aracını tekrar ilk noktaya getirip aracın yanında içi boş küçük bir daire belirdiğinde ilk noktayı fare ile tıklamaktır. Eğer fareyi basılı tutarak sürüklerseniz bu sefer düz bir çizgi değil de bir eğri elde edersiniz. Bir vektör çalışmanızda değişiklik yaparken vektör yollarındaki noktaları sürükleyerek şekli yeniden tasarlayabilirsiniz.

Rectangle ve Ellipse Araçları ile Geometrik Şekiller

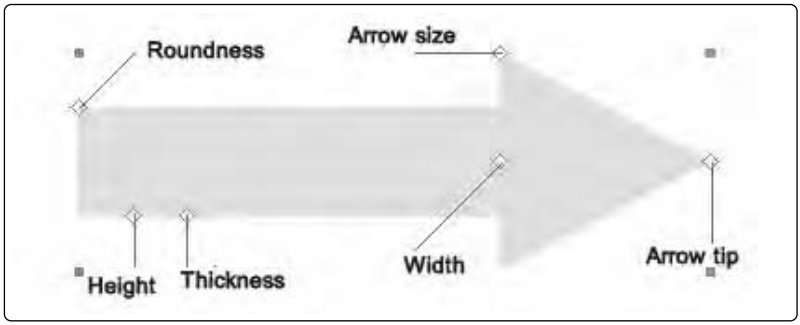
Fireworks'te dikdörgen ve kare çizmek için **Rectangle** aracından yararlanıyoruz. **Rectangle** aracını seçip fare ile tuvalde sürükleyerek istediğiniz büyüklükte bir dikdörtgen çizebilirsiniz. Kare çizmek içinse **SHIFT** tuşunu basılı tutmalısınız. Bir elips çizmek istediğiniz de ise **Ellipse** aracını kullanın. **Ellipse** aracı ile çizim yaparken **SHIFT** tuşunu basılı tutarsanız bir daire çizmiş olursunuz.

Eşkenar çokgen çizmek istediğinizde ise **Polygon** aracını kullanabilirsiniz. Çizdiğiniz şekillerin kenar ve dolgu renklerini, belirlemek ve diğer özelliklere bakmak için **Properties** panelini kullanın.

OTOMATİK ŞEKİLLER

Otomatik şekiller ile Fireworks'te birçok vektörel nesne oluşturma çalışmalarınızı hızlı bir biçimde yapabilirsiniz. Otomatik şekiller **Tools** panelinde **Rectangle** aracının altında yer alırlar. **Rectangle** aracına tıkladığınızda açılan menüden istediğiniz otomatik şekli seçerek kullanabilirsiniz.

Otomatik nesneler üzerinde küçük kareler şeklinde kontrol noktaları bulunmaktadır. Bu kontrol noktalarının üzerlerine fare ile geldiğinizde, her bir kontrol noktasının işlevi ile ilgili olarak bir açıklama görüntülenir. Kontrol noktalarını fare ile tutup sürükleyerek şekilleri istediğiniz gibi değiştirebilirsiniz. Mesela bu kontrol noktalarını kullanarak nesneleri küçültüp büyütebilir, dik olan köşelerini yuvarlayabilirsiniz. Şimdi otomatik şekilleri tek tek ele almaya başlayabiliriz.



Arrow otomatik şekli ok şekilleri çizmek içindir. **Arrow** aracında altı adet kontrol noktası bulunmaktadır.

Arrow tip: Okun baş kısmının sivriliğini ayarlar. Kontrol noktasını sağa ya da sola sürükleyerek gereken ayarlamayı yapabilirsiniz.

Arrow size: Okun baş kısmının genişliğini ve eğimini ayarlamak içindir.

Width: Okun gövdesinin uzunluğunu ayarlamak için kullanılır.

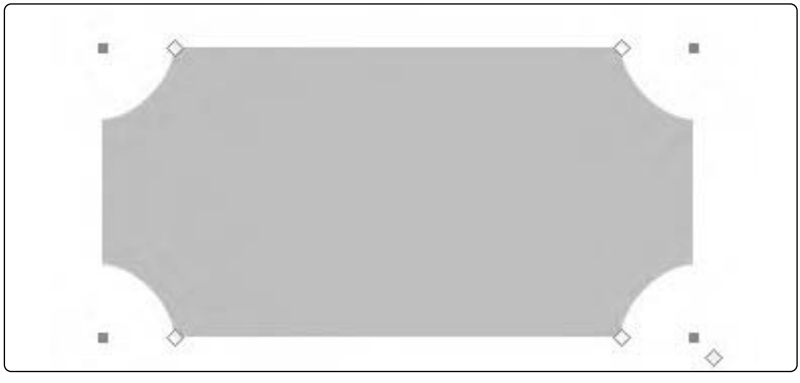
Thickness: Okun kuyruk kısmının kalınlığını ayarlamak içindir.

Height: Bu kontrol noktasını sağa sola hareket ettirerek okun gövde kısmının uzunluğunu ayarlayabilirsiniz. Kontrol noktasını aşağı yukarı hareket ettirerek de okun kuyruk kısmında kıvrım oluşturabilirsiniz.

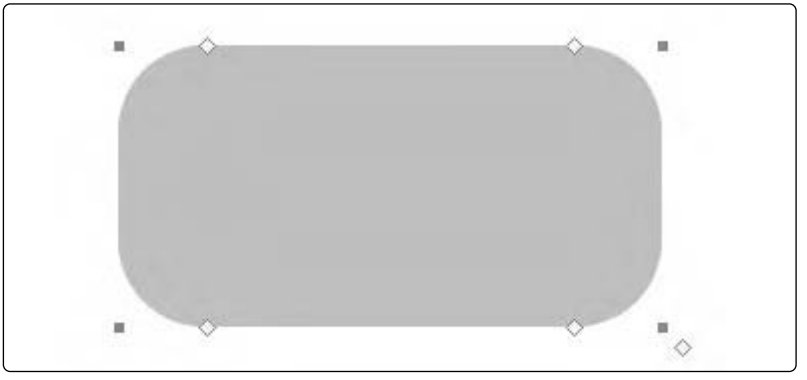
Roundness: Okun arka üst köşesini yuvarlamak içindir.



Beveled Rectangle otomatik şekli, köşeleri kırpılmış dikdörtgen oluşturmak için kullanılır. Kontrol noktalarını kullanarak köşelerin eğimini ayarlayabilirsiniz.



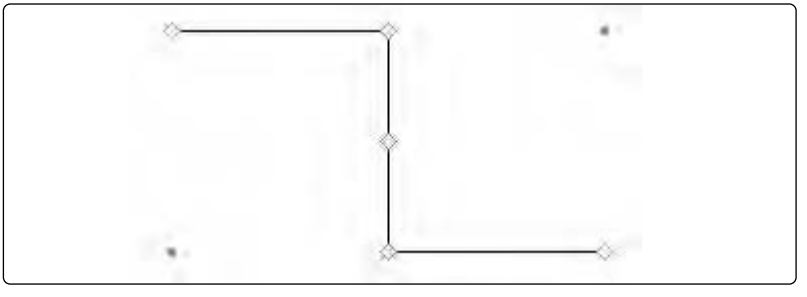
Chamfer Rectangle otomatik şekli, içe doğru yuvarlatılmış köşelere sahip dikdörtgen çizmek için kullanılır.



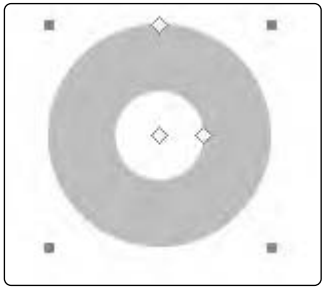
Rounded Rectangle otomatik şekli, yuvarlatılmış köşelere sahip dikdörtgen çizmek için kullanılır.

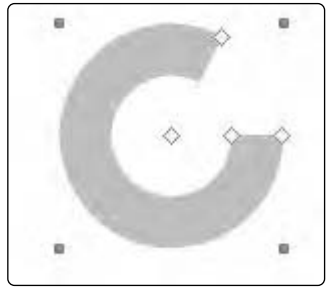
Beveled Rectangle, **Chamfer Rectangle** ve **Rounded Rectangle** otomatik şeklilerinde diğer köşelere dokunmadan sadece bir köşenin şeklini değiştirmek için kontrol noktası ile birlikte **ALT** tuşunu da basılı tutmak gerekir.

Her üç şekilde de kontrol noktalarını sağa ya da sola sürükleyerek istediğiniz değişiklikleri yapabilirsiniz.

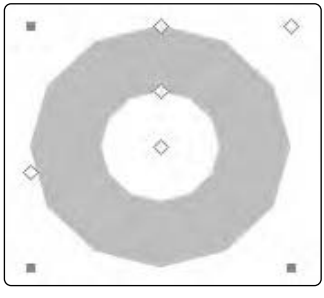


Connector Line otomatik Resim301 otomatik şekli üç kısımdan oluşan bir birleştirme şeklidir. Bir çok farklı amaçla hazırlanan şemalarda kullanılabilir. Beş adet kontrol noktasına sahiptir.

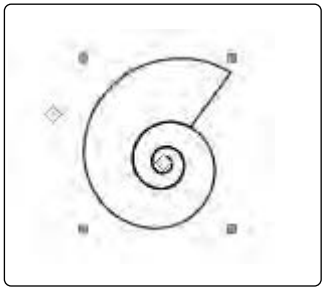




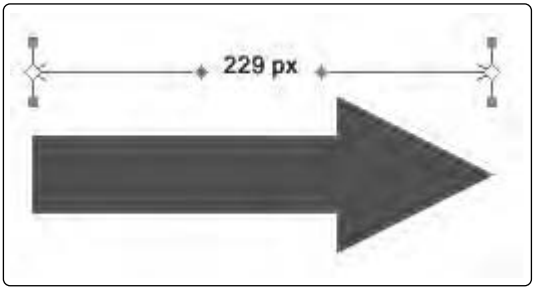
Doughnut otomatik şekli, ortası boş tam ve eksiltmeli daireler çizer. Kontrol noktaları ile orta boşluğu küçültebilir ya da büyütebilirsiniz. Ayrıca şekli parçalara da ayırabilirsiniz. Mesela bu şekil yardımıyla bir CD tasarlayabilirsiniz.



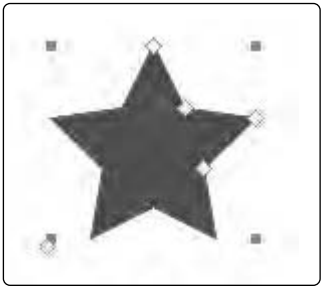
Smart Polygon otomatik şekli, içi dolu ya da boş çokgenler çizer.



Spiral otomatik şekli, **spiral** (*sarmal*) şeklinde şekiller çizer.



Measure Tool otomatik şeklini nesnelerin ölçülerini göstermek için kullanabilirsiniz.



Star otomatik şekli ile yıldız şeklinde nesneler oluşturabilirsiniz. Star otomatik şekli beş kontrol noktası olan bir şekildir.

METİNLERLE ÇALIŞMAK

Fireworks ile metin oluşturabilmek için **Text** aracını kullanacağız. Metninizi girdiğinizde **Properties** panelinde **Text** aracı ile ilgili özellikler görünecektir. Bu özellikleri kullanarak metninizi şekillendirebilirsiniz.

TEXT ARACININ ÖZELLİKLERİ



Text: Bu metin kutusuyla girdiğiniz metni isimlendirebilirsiniz.

Font Family: Oluşturduğunuz metnin yazı tipini seçmeniz içindir.

Font Style: Yazı tipinin stilini bu açılır menüden seçeceğiz. Seçeneklerden **Bold**, metnin kalın olmasını sağlar. **Black**, bold tipinden daha kalın bir yazı stilidir. **Italic** metnin eğik olması içindir. **Bold italic**, metnin hem kalın hem de eğik olmasını temin eder. **Regular** ise yazı tipinin orijinal halidir.

Size: Yazı tipinin ebatlarını ayarlayabilirsiniz.

Tracking: Metindeki karakterler arasındaki mesafeyi ayarlayabilirsiniz.

Pragraph indent: Oluşturacağınız paragraflara girinti ekleyebilirsiniz.

Horizontal scale: Metnin yatay ebatlarını değiştirebilirsiniz.

Leading: Satır aralarındaki boşluğu ayarlayabilirsiniz.

Leading unit: Satır aralarındaki boşluğun yüzdelik değer olarak mı piksel cinsinden mi ayarlanacağını belirleyebilirsiniz.

Space precceding paragraph: Oluşturduğunuz paragrafın kendinden önce gelen paragrafla arasında ne kadar mesafe olacağını ayarlayabilirsiniz.

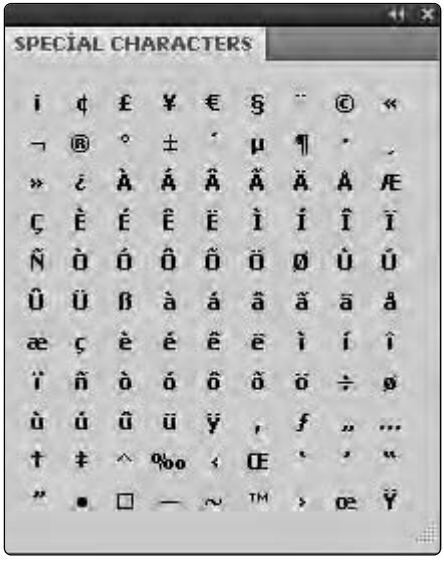
Space after paragraph: Oluşturduğunuz paragrafın kendinden sonra gelen paragrafla arasında ne kadar mesafe olacağını ayarlayabilirsiniz.

Anti-aliasing level: Harflerin kenarlarının yumuşaklık değerini belirlemek içindir. Burada **No Anti-Alias** seçeneğini kullanırsanız harflerin kenarları tırtıklı bir biçimde görünecektir.

Set text orientation: Yazı yönünün soldan sağı doğru mu yoksa yukarıdan aşağıya doğru mu olacağını belirlemek içindir.

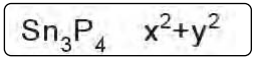
Oluşturduğunuz metni istediğiniz bir yöne kaydırmak için **Pointer** aracını yada ok tuşlarını kullanabilirsiniz. Bu işlem için ok tuşlarını kullandığınızda, ok tuşlarına her basışınızda metin `1 piksel` kayacaktır. Eğer ok tuşu ile birlikte **`SHIFT`** tuşunu da basılı tutarsanız metin `10 piksel` kayar.

Özel Karakterler Kullanma



Metninize özel karakterler eklemek isterseniz, resimde gördüğünüz **Special Characters** panelini kullanabilirsiniz.

Metni Farklı Hatlara Yerleştirme



Bir metnin bazı kısımlarını farklı bir çizgi üzerinde konumlandırmak isteyebilirsiniz. Örneğin bir kimyasal formül yazarken ya da bir matematik işlemini gösterirken bu özelliğe ihtiyacınız olacaktır. Bunun için, farklı bir hatta alacağınız metni seçtikten sonra **Properties** panelinde belirecek olan **Baseline Shift** kutucuğuna istediğiniz değeri girmeli ya da kutucuğun yanındaki kaydırma çubuğunu kullanmalısınız.

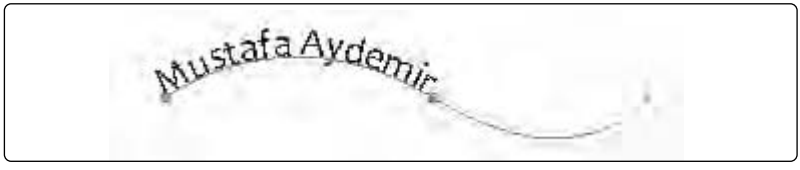
Metinleri Vektörel Yollara İliştirme



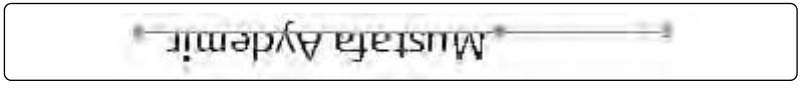
Metinlerinizi vektörel yollara iliştirerek farklı formlara sokabilirsiniz. Logo tasarımı veya bir reklam banner'ı gibi işlerinizde sıkça kullanabileceğiniz bir özelliktir. Mesela resimde gördüğünüz gibi metninizi bir dairenin vektörel yoluna iliştirerek güzel bir görsel format yaka-

layabilirsiniz. Bu işlem için önce **Ellipse** aracı ile ve **SHIFT** tuşuna basılı tutarak bir daire çizin. Ardından **Text** aracını seçip tuvalin herhangi bir yerine metninizi yazın. Şimdi hem yazıyı hem de daireyi birlikte seçin. Ve son olarak **Text>Attach to Path** menüsüne tıklayın. Metniniz daireye iliştirilecektir.

Metnin daire çevresindeki konumunu değiştirmek için **Pointer** aracıyla metnin baş tarafını çift tıklayın ya da **Text** aracını seçip imleci metnin baş tarafına konumlandırın. Artık klavyeden **Backspace** tuşuna basarak metninizi daire etrafında kaydırabilirsiniz. Gerektiğinde metninizi yoldan ayırmak için **Text>Detach from Path** menüsünü kullanabilirsiniz.



Bu tarz bir uygulamayı sadece daireler ile değil tüm vektör yollarıyla yapabilirsiniz. Mesela **Pen** aracı ile serbest olarak çizdiğiniz bir yola resimde gördüğünüz gibi metninizi iliştirebilirsiniz.



Yoldaki bir metni resimdeki gibi ters çevirmek için **Text>Reverse Direction** menüsüne tıklamanız yeterli olacaktır.



Metni yolun içine iliştirmek içinse **Text>Attach to Path** menüsünü kullanmalısınız.



Metinlerinizle orijinal çalışmalar yapabilmenizi kolaylaştıran bir uygulama da metinleri yola dönüştürmektir. Bunun için **Text>Convert to**

Path menüsünü kullanmalısınız. Ardından **Subselection** aracını seçerek metindeki her harfi bağımsız olarak düzenleyebilirsiniz.



Metinlerinizi biçimlendirmek için Fireworks'ün size sunduğu imkanlardan biri de **Styles** panelidir. Bu panelde resimdeki gibi metinler üretebileceğiniz birçok hazır stil, kullanılmak üzere sizi beklemektedir.

RENKLER

Fireworks ile renk işlemleri ve renk kontrolü için öncelikle Tools panelindeki **Colors** bölümünde yer alan araçlara bir göz atalım. **Stroke Color** nesnelere kenar rengi, **Fill Color** ise dolgu rengi uygulamak için kullanılan araçlardır.

Set default Stroke/Fill Colors: Nesnenize bir dolgu rengi ya da kenar rengi uyguladıktan sonra eğer varsayılan renklere dönmek isterseniz kullanmanız gerek araç olmaktadır. **Stroke Color** için varsayılan renk `siyah`, **Fill Color** için varsayılan renk ise `beyaz`'dır.

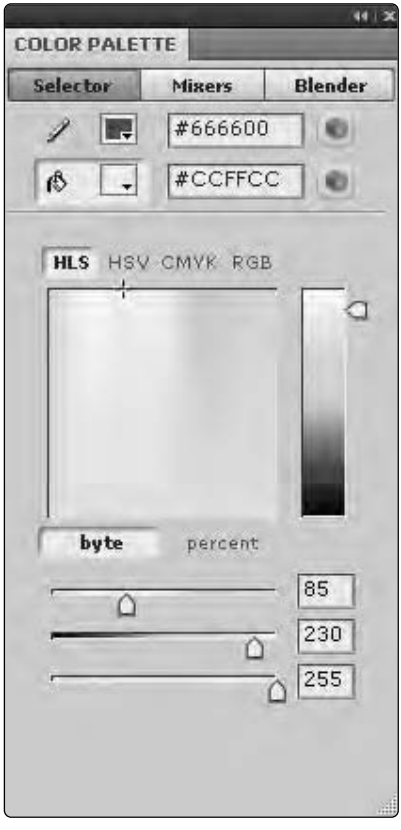
Swap Stroke/Fill Colors: Stroke Color ile Fill Color araçlarında bulunan renkleri yer değiştirmek için kullanılır.

No Stroke or Fill: Nesnenize bir dolgu rengi ya da kenar rengi uyguladıktan sonra eğer bunları silmek isterseniz bu aracı kullanmalısınız.

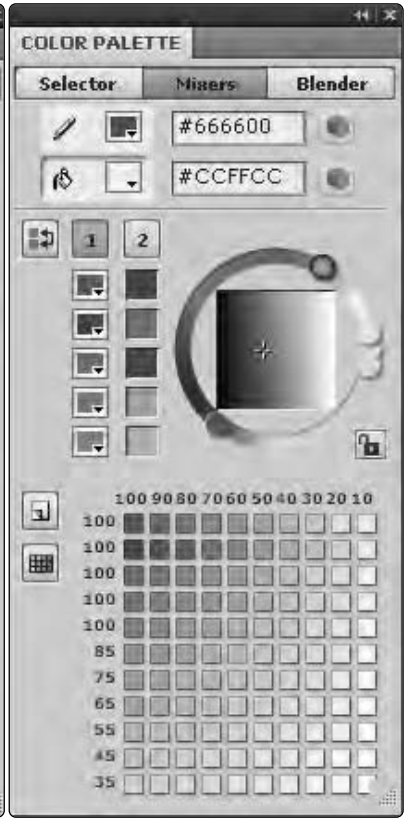
Color Palette Paneli

Selector Sekmesi: Selector sekmesi yardımı ile **HSV**, **HLS**, **RGB, CMYK** renk modellerinden birini seçebilirsiniz. Penceredeki renk tayfını kullanarak da istediğiniz rengi üretebilirsiniz.

Mixers Sekmesi: Bu sekme yardımıyla web projelerinizde kullanabileceğiniz birbiriyle uyumlu renk grupları oluşturabilir ve bu renkleri sekmedeki **export tuşları** ile kaydedebilirsiniz.

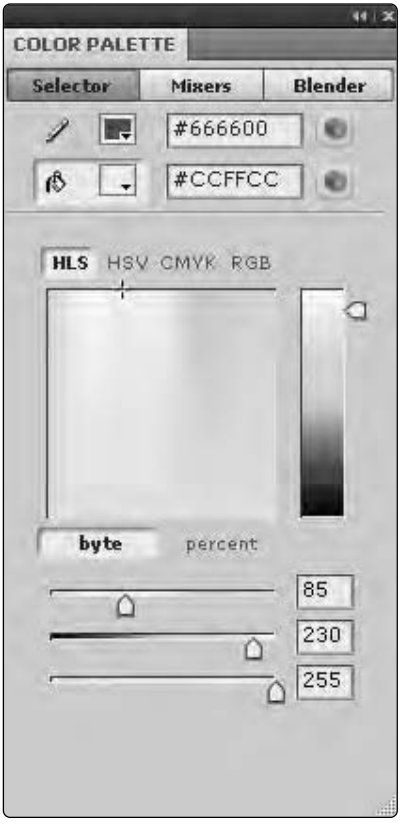


Selector Sekmesi



Mixers Sekmesi

Blender Sekmesi: Bu sekme sayesinde belirleyeceğiniz iki renk arasındaki en fazla 36 adet rengi görebilirsiniz.



Blender Sekmesi

SAYFALAR

Fireworks CS4 ile bir **PNG** dosyasının içinde birden fazla sayfa oluşturabilirsiniz. Üstelik her bir sayfayı diğerlerinden bağımsız olarak düzenleyebilirsiniz. Panelin sol alt kısmında oluşturduğunuz sayfaların sayısını görebilirsiniz. Yine panelin alt kısmındaki **New Duplicate>Page** tuşu ile yeni bir sayfa oluşturabilirsiniz. Mevcut bir sayfayı bu tuşun üzerine sürükleyip bırakarak o sayfayı çoğaltabilirsiniz. **Delete Page** tuşu ile de seçili olan sayfayı silebilirsiniz.

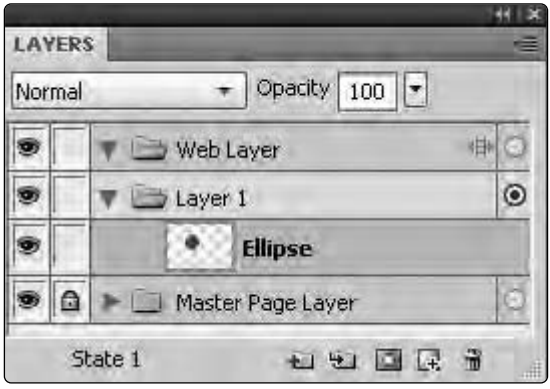


Sayfalardan birini anasayfa yapmak isterseniz panelin sağ üst köşesindeki **Options** menüsünden **Set As Master Page** linkini tıklayın. Bu işlemi uyguladığınızda anasayfa diğer sayfalara da katman olarak eklenir. Bir sayfayı anasayfaya bağlamak içinse yine Option panelinde yer alan **Link to Master Page** linkine tıklayın. Bunu yaptığınızda bağladığınız sayfa bağımsızlığını kaybedecek ve anasayfaya ait tuval rengi, tuval boyutu gibi özellikleri alacaktır. Link verdiğiniz sayfada, mesela tuval rengini değiştirirseniz bu değişiklik bu sayfa ile anasayfa arasındaki linkin kopmasına sebep olur.

Oluşturduğunuz sayfaları dilerseniz HTML belgesi olarak ihraç edebilirsiniz. İhraç işleminden önce sayfayı önizlemek isterseniz **File>Preview in Browser >Preview All Pages** menüsünü tıklayın. İhraç işlemi için ise File>Export menüsüne tıklayın. Açılan pencerede, **Export** açılır kutusundan **HTML and Images** seçeneğini seçin. HTML açılır kutusunda ise **Export HTML File** seçili olmalıdır. Artık sayfanızı HTML belgesi olarak ihraç etmek üzere **Kaydet** tuşuna basabilirsiniz.

KATMANLAR

Fireworks'te kendi oluşturduğunuz ya da ithal ettiğiniz tüm nesneler katmanlara yerleştirilirler. **Katmanlar** birden farzla nesne içerebilirler. Katmanların bize olan yararı, nesneleri dolayısıyla çalışmamızı organize bir biçimde düzenli olarak yapabilmemizi sağlamalarıdır.



Fireworks'te katmanları **Layers Paneli** ile organize ederiz. Resimde gördüğünüz gibi en üstte web katmanı yer alır. Bu katman web nesneleri içindir ve silinemez. Her bir katmanın ve nesnenin solunda yer alan küçük üçgen işareti ile katmanları açıp kapatabilirsiniz. Katmanı kapattığınızda içindeki nesneleri göremezsiniz. Küçük üçgenin bir solundaki boş alana tıklarsanız bir kilit işareti belirecektir. Bu işaret o katmanın kilitlendiği anlamına gelir. Kilitlenen katman üzerinde her hangi bir işlem yapılamaz. Bu sayede sayfanızda çalışırken kazara bu katmana ait bir nesnenin ayarının bozulması gibi ihtimaller devre dışı kalmış olur.

Layers panelinin alt kısmındaki **New>Duplicate Layer** tuşuna tıklayarak yeni bir katman oluşturabilirsiniz. Bir katmanı bu tuşun üzerine sürükleyip bırakırsanız bu katmanın bir kopyasını çıkarmış olursunuz. Bu yeni katman kopyaladığınız katmandaki nesneleri de içerir. Bir katmanın altında yer alacak bir alt katman oluşturmak istediğinizde ise **New Sub Layer** tuşuna tıklamanız yeterlidir.

Katmanın içinde boş bir Bitmap nesnesi oluşturmak için New Bitmap Image tuşuna tıklayın. Bir katmanı silmeniz gerektiğinde ise silinecek katman seçili durumdayken **Delete Layer** tuşuna tıklayın. Bu işlemleri, **Layers** panelinin sağ üst köşesinde yer alan **Options** menüsü ile de gerçekleştirebilirsiniz.

Maskeler

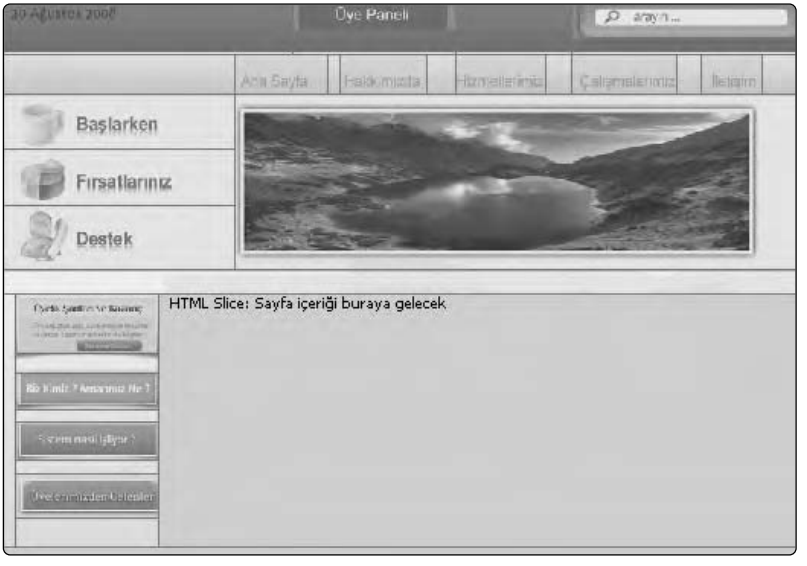




Maskeleri, resmin bir bölümünü gizleyip bir bölümünü göstermek için kullanacağız. Kullanacağınız şekillerle çok estetik maskeler elde edebilirsiniz. Bu işlem için resminizin üzerinde bir nesne (örneğin bir yıldız) oluşturup bunu **Edit>Cut** menüsü ile kesin. Ardından resminizi seçip **Modify>Mask>Paste as Mark** menüsü ile maske olarak yapıştırın. Gerek vektörel maskeler gerekse bitmap maskeler aynı yöntemle, yani **Paste as Mark** komutu ile oluşturulur.

Maskeleme işlemini geçici olarak devre dışı bırakmak için M**odify>Mask>Disable Mask** menüsünü, daha sonra tekrar etkin olması içinse **Modify>Mask>Enable Mask** menüsünü kullanın. Bir maskeyi tamamen silmek için de **Modify>Mask>Delete Mask** menüsüne tıklayın.

DİLİMLER



Dilimleme (*slices*) kavramını bir grafiği küçük dilimlere bölme işlemi olarak tarif edebiliriz. Dilimlerin her biri ayrı bir resim olarak ihraç edilir. Bir HTML sayfa ile tekrar bir arada görüntülenir. Bu işlemin avantajı sayfaların daha hızlı açılmasını sağlamasıdır. Dilimlere link atayabildiğiniz gibi animasyon ve rollover'da uygulayabilirsiniz.

Dilimleme işlemi için iki araçtan yararlanırız. Bunlardan **Slice tool** aracı dikdörtgen dilimleme yaparken **Polygon Slice tool** aracı ise dairesel dilimlemeler yapar.

Sayfanızı ihraç ettikten sonra oluşacak html sayfa, sadece resimlerden değil aynı zamanda da metinlerden oluşur. Bu sebeple resimdeki gibi bir web sayfasını dilimlerken sayfada metinlerin geleceği kısmı da **Slice tool** aracı ile dilimleyip bırakırsak bu kısım da **image olarak** dilimlenecektir. Bu kısmın metin girilecek HTML dilimi olarak ihraç edilebilmesi için, dilimi oluşturduktan sonra **Properties** panelindeki **Type** açılır menüsünden **HTML** seçeneğini seçmeliyiz. Ardından da altındaki **HTML Edit** tuşuna tıklayarak burasının metin alanı olduğunu belirten bir metin girebilirsiniz. Bu şekilde işlemleri bitirip dilimlerinizi ih-

raç ettiğinizde bu bölüm bir paragraf olarak HTML belgesinde yer alacaktır.

Fireworks size dilimlere link atama imkanı da vermektedir. Şimdi temamızdaki menü dilimlerine nasıl link atayacağımızı görelim. Mesela temamızda sol üst kısımda bulunan "Başlarken" kısmına link verelim. Böylece resimde gördüğünüz anasayfamızda, "Başlarken" linkine tıklandığında birtakım bilgilerin yer aldığı **baslarken.html** adlı sayfaya gideceğiz. Bu işlem için **Properties** panelindeki **Link** kısmına, linkin gideceği sayfayı (*baslarken.html*) yazıyoruz. **Target** kısmını da `_self` olarak seçip sayfanın aynı pencerede açılmasını sağlıyoruz.

ROLLOVER'LAR

Rollover kavramını, bir nesnenin üzerine fare ile gelindiğinde nesnenin yerini bir başka nesneye bırakması şeklinde tanımlayabiliriz. Fireworks'te her bir rollover bir davranıştır ve davranışlar durumların (*state*) bir araya gelmesinden oluşur.

Fireworks ile rollover oluştururken **Behaviors** panelinden yaralanacağız. Ama önce bir hotspot ya da bir dilim oluşturmak gerekmekte. Uygulayabileceğimiz davranışların listesine **Behaviors** panelindeki **Add behavior** (**+**) tuşu ile ulaşıyoruz.

Bu menüde göreceğiniz rollover türlerini sayalım:

Simple Rollover: Standart bir rollover davranışı oluşturmak içindir.

Swap image: Bir web sayfası üzerinde farklı konumlardaki nesnelerle rollover davranışı oluşturmak içindir.

Set Nav Bar Image: Bir **navigasyon** (*gezinti*) çubuğu davranışı oluşturur.

Set Pop-up Menu: Pop-up menüler oluşturmak içindir.

Set Text of Status Bar: Tarayıcının durum çubuğunda ileti göstermek içindir.

Şimdi üzerine gelince renk değiştiren bir buton yapmak istediğimizi var sayalım. Bunun için ilk önce butonumuzu Fireworks ile açalım ve ardından da **State** panelini açalım.

Panelin altındaki **New>Duplicate State** tuşu ile bir state oluşturalım. Artık panelde iki state var. **İlk state** belgemizi açmakla otomatik olarak Fireworks tarafından oluşturulur.

Birinci state seçili iken butonumuzu tek parça olarak dilimleyelim. Bu oluşturacağımız rollover davranışının **ilk durumu** (*birinci state*) olmuş oldu. Ardından **ikinci state**'i seçelim. **İkinci state**'i seçince butonumuzun yeri dilimde boş olarak görünecektir. Bu aşamada butonun üzerine fare ile gelindiğinde gösterilecek olan farklı bir renkteki ikinci butonumuzu ithal edelim ve bu boş dilimin içine ekleyelim.

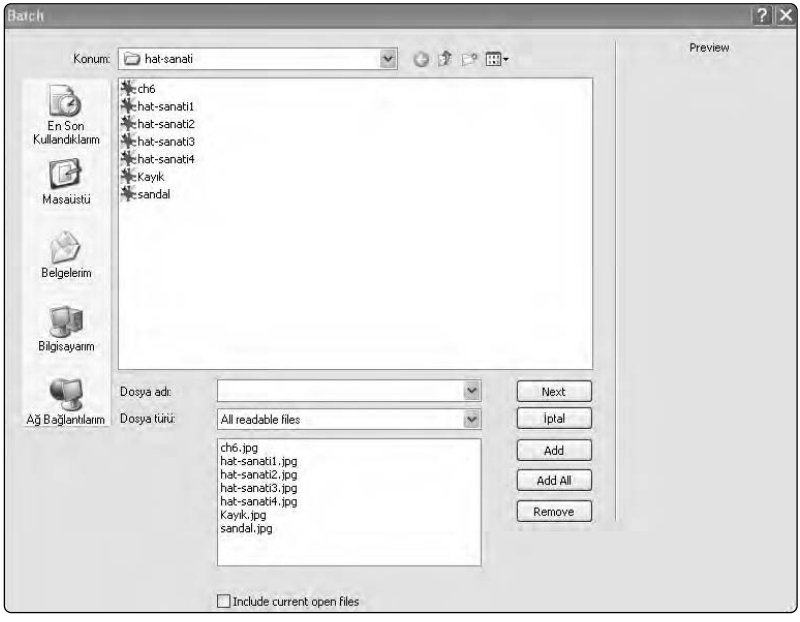
Şimdi **Behaviors** panelindeki **Add behavior** (**+**) tuşu ile açılan menüden **Simple Rollover** davranışını seçelim. İşte hepsi bu kadar! Davranışın çalışıp çalışmadığını test etmek için **File>Preview in Browser** menüsünden tarayıcıyı açalım ve fare ile butonun üzerine gelerek test edelim. Artık rollover'ı HTML olarak ihraç edebiliriz.

File>Export menüsünü tıkladığınızda açılan pencerede, **Export** kutucuğunda **HTML and Images** seçeneğini, **HTML** kutucuğunda ise **Export HTML File** seçeneğini seçin. **Kaydet** tuşu ile rollover'i kaydedin.

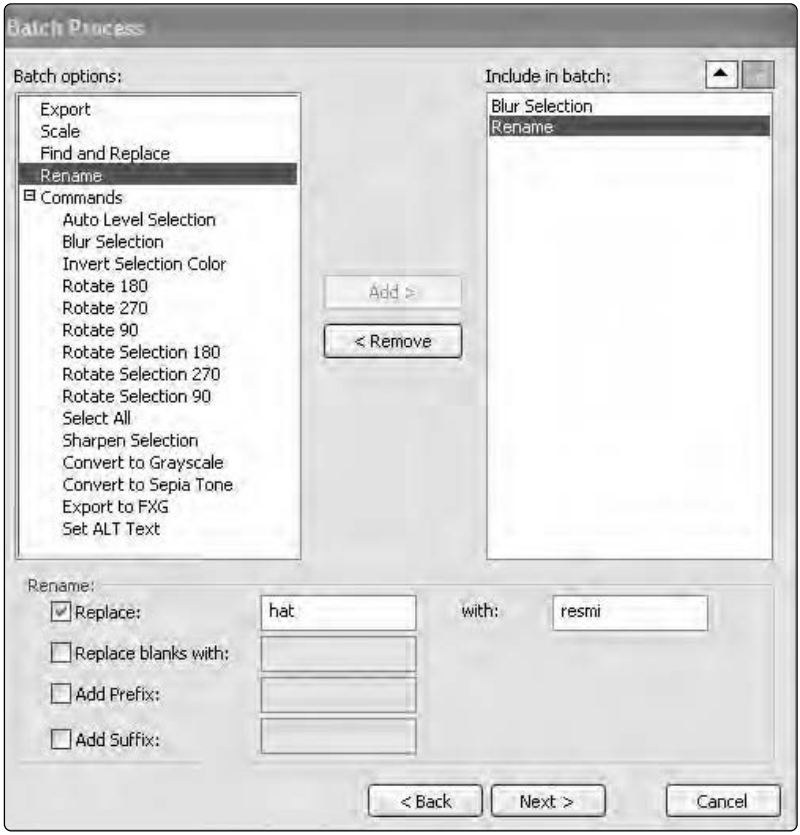
Rollover'i ihraç ettiğimiz klasöre bakarsanız bir adet HTML dosyası ve iki adet buton resmini göreceksiniz. HTML belgesini açtığınızda ise içinde HTML kodları ile beraber rollover için gereken JacaScript kodlarını da bulacaksınız. Bir başka ifade ile söylersek, Fireworks HTML dosyasını ihraç ederken gerekli JavaScript kodlarını da belgeye dahil eder.

Resimlerle Toplu İşlemler

Elimizde düzenlememiz gereken çok sayıda resim olduğunu ve bunların dosya türlerini, ebatlarını... vb özelliklerini değiştirmemiz gerektiğini varsayalım. Tüm resimlerle tek tek uğraşmak bir hayli vakit alıcı bir işlem olacaktır. Fakat Fireworks ile bu işlemleri topluca yaparak ciddi anlamda zamandan kazanmış olursunuz.



İlk olarak **File>Batch Process** menüsü yardımıyla Batch penceresini açılım. Pencereden düzenlememiz gereken resimleri seçip **Add** tuşu ile ekleyelim. Eğer açtığımız klasördeki tüm resimleri alacaksak **Add All** tuşu ile hepsini bir seferde seçebiliriz. Ardından **Next** tuşu ile yolumuza devam ediyoruz.

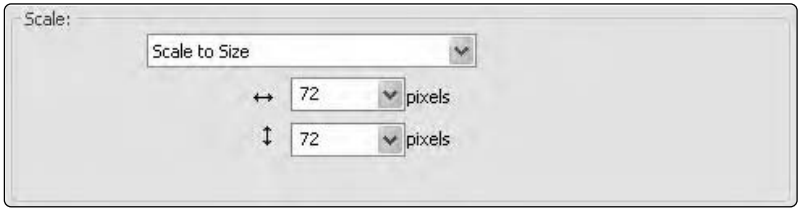


Bu sefer de resimdeki pencere ile karşılaşacağız. Pencerede sol kısımda kullanabileceğimiz özelliklerin listesini görmektesiniz. Biz resimlerimize hangi işlemleri yapacak isek o işlemler ile ilgili olanları seçip **Add** tuşu ila sağ tarafa geçireceğiz. Bu şekilde bir özellik seçtiğinizde pencerenin alt kısmında o özellik ile ilgili ayarlar belirecektir. Buradan gerekli düzenlemeleri yapabilirsiniz.

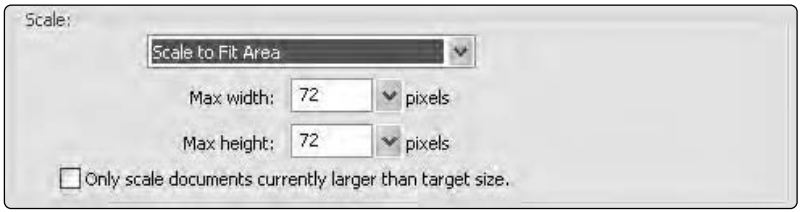
Şimdi bu özelliklerin detaylarına bakalım

Export: Bu özellik, resimlerinizin dosya türünü değiştirebilmeniz içindir. Bu şekilde resimlerinizi bir başka resim formatına çevirebilirsiniz.

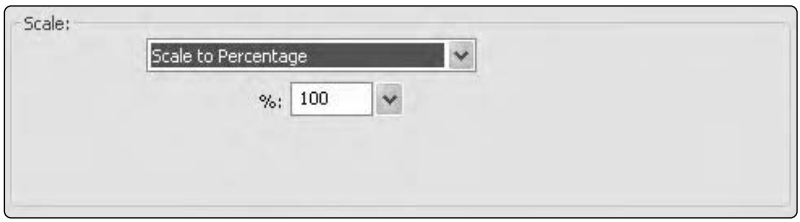
Scale: Bu özellik resimlerinizin boyutunu ayarlamak içindir.



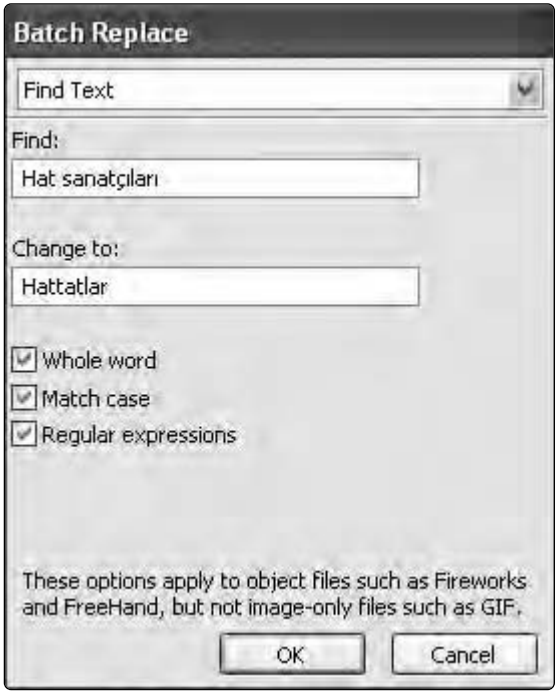
Scale özelliğinin ilk alt özelliği olan **Scale to Size** özelliğini seçerek resimlerinizin yeni ebatlarını belirleyebilirsiniz. Fakat bu özellikte resimler orantılı olarak değil sizin verdiğiniz değerlerle ayarlanır.



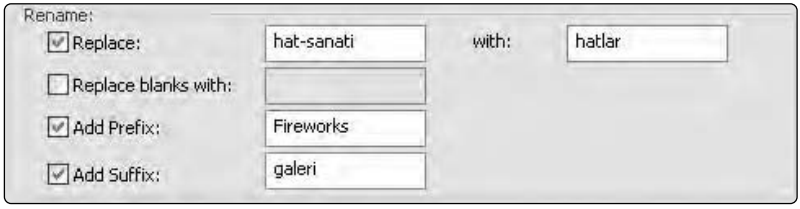
Eğer **Scale to Fit Area** özelliğini seçerseniz, aşağıdaki resimler orantılı olacak ölçeklendirilecektir.



Eğer **Scale to Percentage** özelliğini seçerseniz resimleriniz yüzdelik (%) oranlarda ölçeklendirilir.

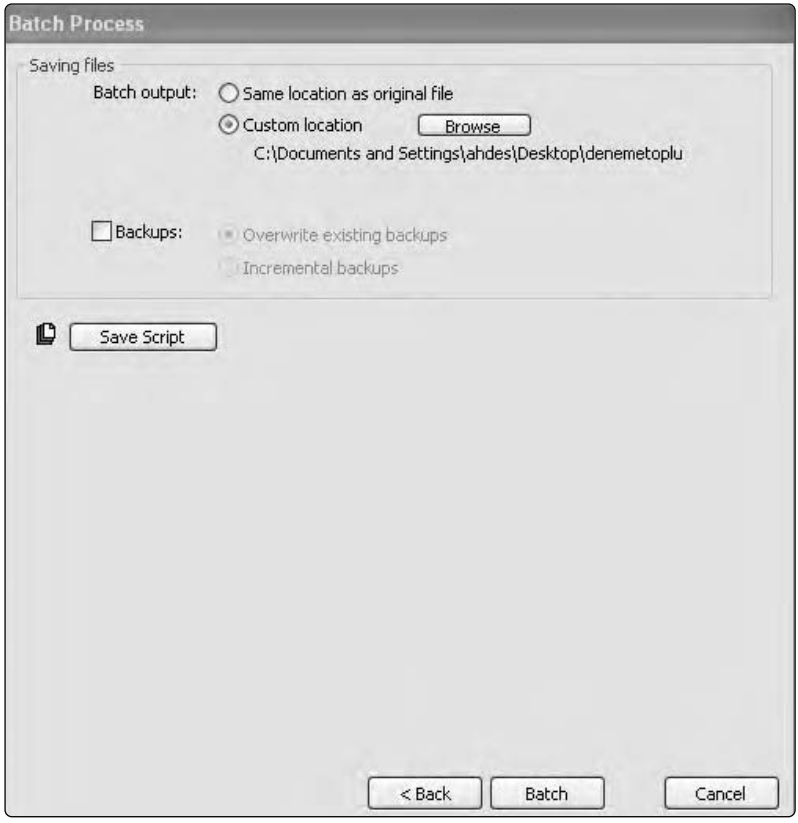


Find And Replace: Bu özelliği sadece Fireworks PNG dosyaları için kullanabileceğiz. PNG'lerin içindeki fontları, vektörel grafikleri, URL'leri, metinleri ve renkleri bul-değiştir mantığı ile değiştirebiliriz.

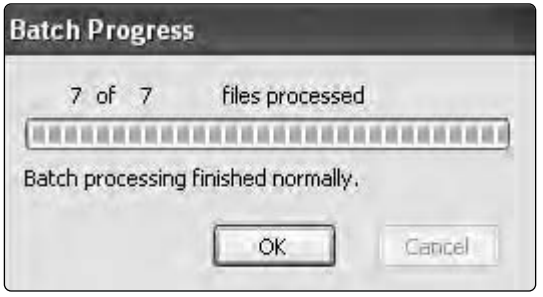


Rename: Bu özellik ile resimlerinizin adlarını toplu bir şekilde değiştirebilirsiniz. **Replace blanks with** seçeneği ile resim adlarında yer alan boşlukları kelimelerle ya da sayılarla değiştirebilirsiniz. **Add Prefix**, seçeneği ile de resimlerinizin adlarının başına bir kelime ekleyebilirsiniz. **Add Suffix** seçeneği ile ise resimlerinizin adlarının sonuna bir kelime ekleyebilirsiniz.

Commands: Command kategorisindeki özellikleri (örn: resmi keskinleştirme ya da bulanıklaştırma, resmi döndürme... vd) toplu olarak uygulamak içindir.

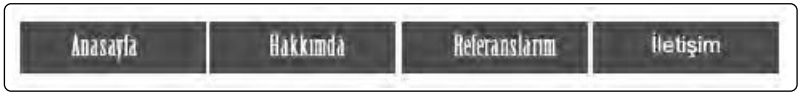


İstediğimiz özellikleri sağ tarafa geçirdikten sonra **Next** tuşu ile devam ettiğimizde resimdeki pencere ile karşılaşacağız. Pencerenin **Batch Output** kısmında gördüğünüz ilk radyo düğmesi resimlerinizi yine mevcut oldukları klasöre kaydetmeniz içindir. İkinci radyo düğmesini seçerseniz başka bir klasör belirleyebilirsiniz. Alttaki **Backup** kısmı ile de eğer resimlerinizin yeni halleri, mevcut oldukları klasöre yazdırılacaksa resimlerin üstüne yazdırılarak orijinallerinin kaybedilmesi ihtimaline karşılık resimlerinizi yedeklemenizi sağlar. Evet artık tek işimiz en alttaki **Batch** tuşuna tıklamak.

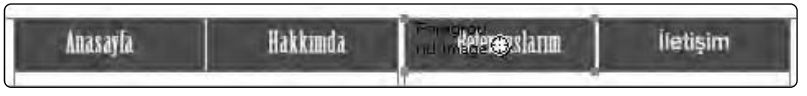


Böylece tüm resimlerinizde belli işlerinizi tek seferde halletmiş oldunuz.

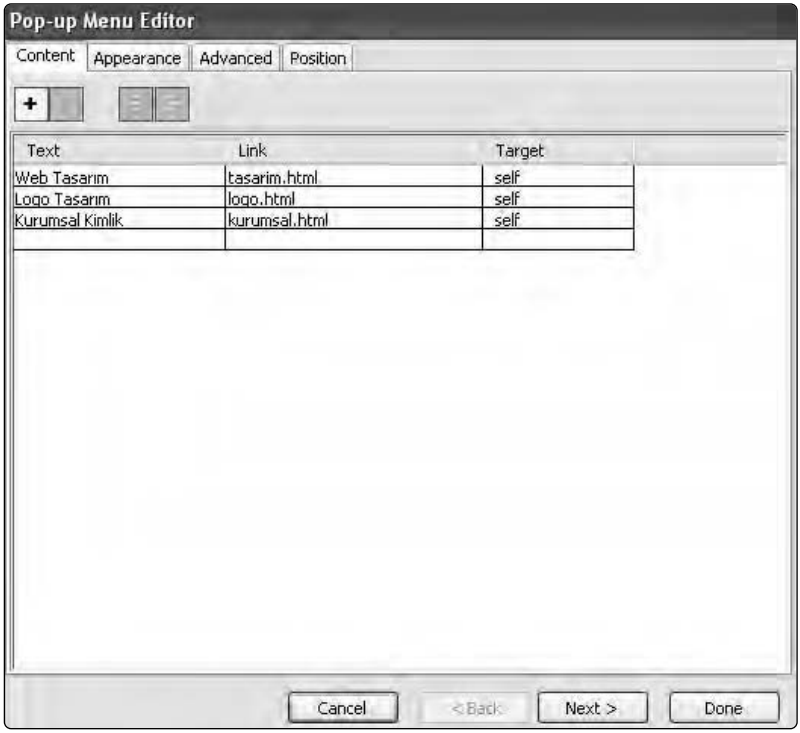
Açılır Menü



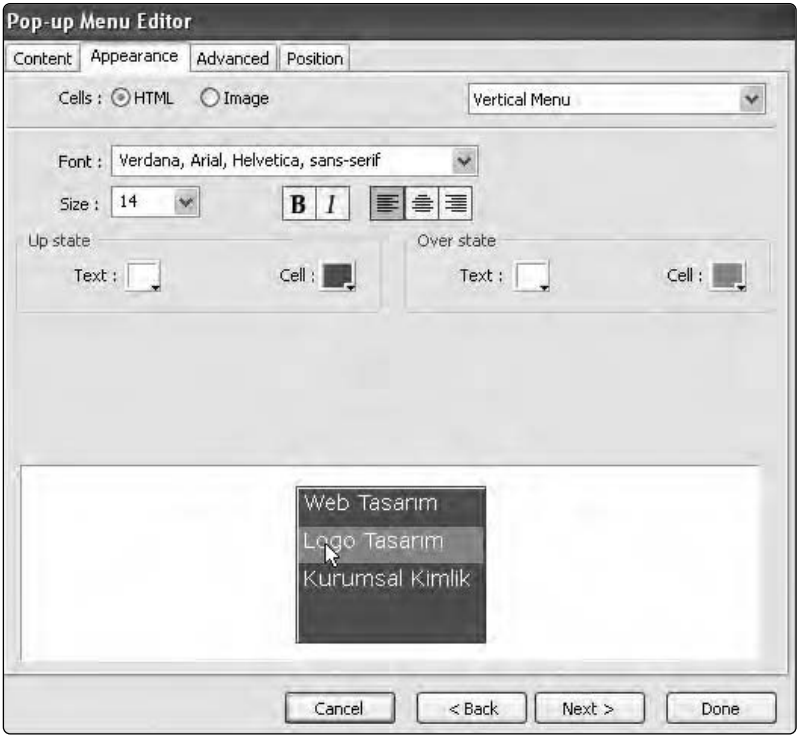
Şimdi de birlikte bir **açılır menü** (*Drop-Down*) yapalım. Öncelikle `500*50` piksel ebatlarında yeni bir belge açalım ve **Rectangle Tool** aracı ile tuvalimize bir adet dikdörtgen çizip kırmızı ile dolduralım. Ardından dikdörtgenimizi **Edit>Duplicate** menüsü ile üç kez çoğaltalım. Ve dikdörtgenlerin üstüne menü adlarını yazalım.



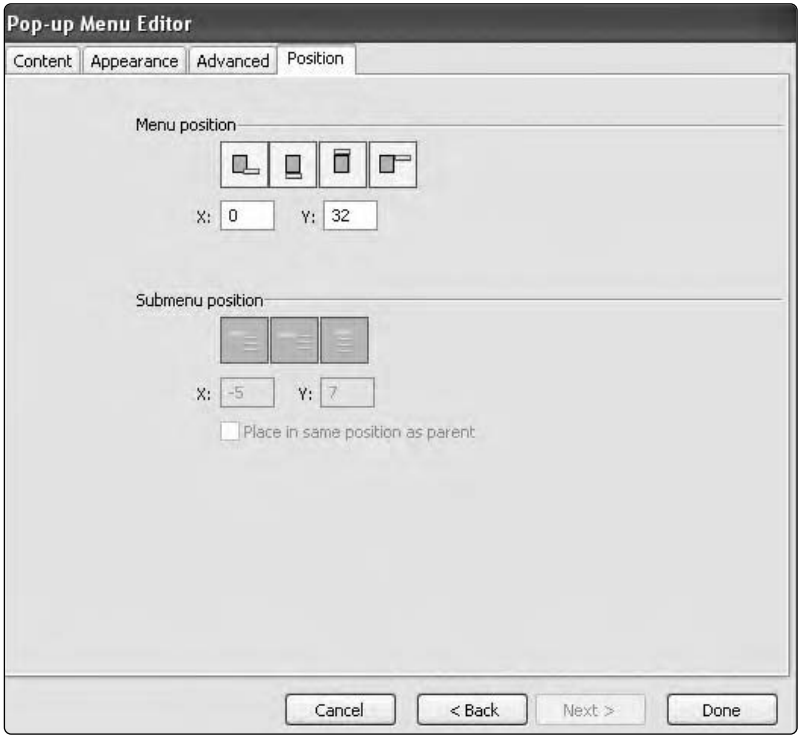
Şimdi de her bir menüyü **Slice** aracı ile dilimleyelim. Şimdi açılır menü ekleyeceğimiz menü dilimini seçelim ve **Modify>Pop-up Menu>Add Pop-up Menu** menüsü ile **Pop-up Menu Editör** penceresini açalım.


Pop-up Menu Editor
Content
Appearance
Advanced
Position
Text
Link
Target
Web Tasarım
tasarim.html
self
Logo Tasarım
logo.html
self
Kurumsal Kimlik
kurumsal.html
self
Cancel
< Back
Next >
Done

Editörde ilk olarak **Content** sekmesinden alt menüleri oluşturuyoruz.



Bir sonraki **Appearance** sekmesinden alt menülerin görünümünü ayarlayacağız. Burada yaptığınız ayarları pencerenin alt kısmındaki ön izleme alanından görebilirsiniz.



Son olarak **Position** sekmesinden alt menünün pozisyonunu seçiyoruz ve **Done** tuşu ile işlemi tamamlıyoruz. Artık menümüzü **File>Export** ile HTML olarak ihraç edebilirsiniz.

ANİMASYONLAR

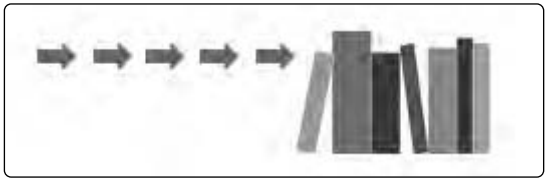
Fireworks bize hareketli GIF resimleri üretme imkanı vermektedir. Üstelik bu resimleri optimize ederek çok düşük boyutlu GIF'ler elde edebiliriz.

Şimdi gelin birlikte bir **GIF banner** yapalım. Öncelikle `234*64` ebatlarında yeni bir belge açalım ve **banner.png** adı ile kaydedelim. Resim-

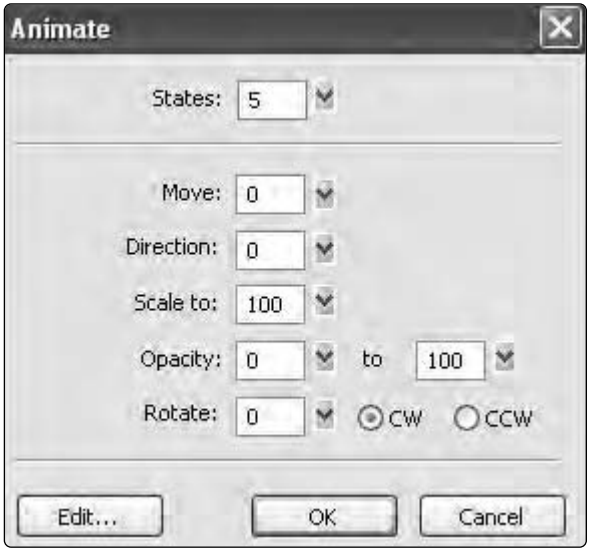
de gördüğünüz gibi banner'in sağ kısmına bir resim yerleştirdik. Ardından Banner'ın üst kısmına resmimizden ayrı bir katmanda art arda ok nesneleri yaptık.



Bir sonraki hamle olarak animasyonumuzun hareketli kısmını oluşturacak olan bu okların yer aldığı katmanımız seçili iken **State** panelinin alt kısmındaki **Distribute to States** tuşu ile nesneleri durumlara dağıttık. Bu işlemin amacı, animasyonda yer alacak her durum için ayrı ayrı uğraşmaktan kurtulmaktır.



Ardından **Layer panelinde** animasyonumuzun sabit kısmını oluşturan kitap resminin olduğu katman seçili iken, panelin sağ üst kısmındaki **Options** menüsünden **Share Layer to Statates** alt menüsünü seçili hale getiriyoruz. Buradaki amacımız da animasyonda sabit olan kitap resminin her ok nesnesi ile birlikte gösterilmesini sağlamaktır.



Şimdi de banner'da, solda gördüğünüz "web tasarım rehberi" ibaresini yine animasyonlu olarak oluşturacağız. Bunun için **State** panelinde **New>Duplicate State** tuşu ile yeni bir durum oluşturuyoruz. Ardından banner'a text aracı ile metnimizi giriyoruz. Metni yanıp sönen bir animasyon şeklinde yapacağız. Bunun için **Modify>Animation>Animate Selection** menüsünü tıklıyoruz. Açılan pencereden bu işlemin beş adet duruma dağılmasını istediğimiz için **State** kutucuğuna `5` rakamını giriyoruz.

Metnin konumunun sabit olmasını istediğimizden **Move** ve **Direction** kutucuklarına `0` değerini giriyoruz. **Opacity** değerlerini de `0` to `100` olarak giriyoruz. Bundan maksadımız da metin önce görünmeyip beş farklı durumda yavaş yavaş görünür olmasıdır.



Artık çalışma sayfamızın alt tarafındaki **VCR** kısmından **Play** tuşuna basarak animasyonumuzu test edebiliriz. Banner gayet güzel oldu! Ban-

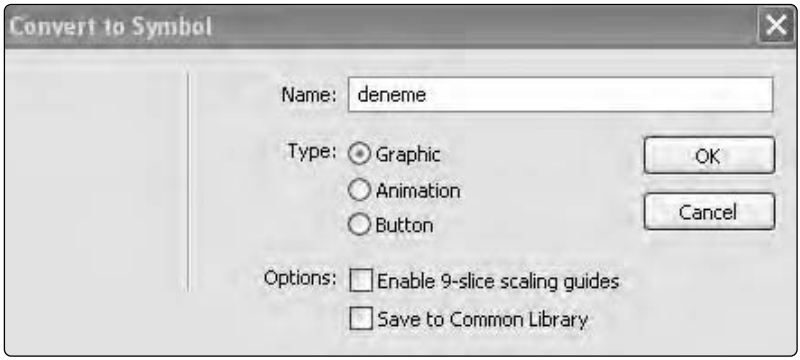
ner'ı bu haliyle DVD içinde **Uygulamalar> Fireworks>** içinde (*banner1.gif*) bulabilirsiniz.

Banner'ı denerseniz, animasyonun çok hızlı olduğunu göreceksiniz. Çünkü durumların varsayılan zaman değeri olan 7 değerini değiştirmemiştik. Bu değer saniyenin 100'de 7'sini ifade etmektedir. Şimdi bu rakamı arttırarak animasyonumuzun hızını düşürelim ve böylece animasyonumuzu daha rahat algılanabilir bir hale getirmiş olalım.

Bu işlem için **State** paneli sağ üst köşedeki **Option** menüsünü açalım ve menüden **Properties** komutunu seçelim. Durumların varsayılan değeri olan 7 değerini 15 yapalım ve animasyonumuzu tekrar izleyelim. Son hamle olarak bannerimizi **File>Save as** komutunu kullanarak kaydedelim.

TWEEN ANIMASYON

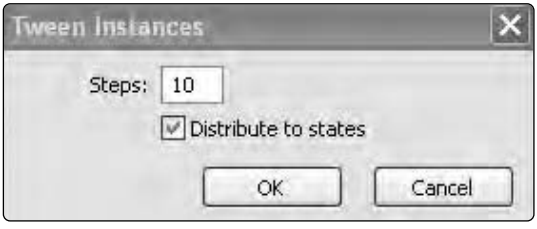
Tweening tekniği, iki nesne arasında birçok farklı durumlar oluşturarak akıcı bir hareket elde edilmesini sağlar. Önceki animasyon örneğimizde, animasyondaki tüm hareketli nesneleri kendimiz oluşturduğumuz için bu tekniğe gerek yoktu. Şimdi bir uygulama yapalım ve aradaki farkı daha net görmüş olalım.



Öncelikle **Star** aracı ile bir yıldız oluşturalım. Yıldızı istediğiniz bir renkle doldurabilirsiniz. Bu nesneyi öncelikle bir sembole dönüştüreceğiz. Bunu yapmaktaki maksadımız vektörel grafiğimizi bir resme dönüştürmektir. Bu işlem için **Modify>Symbol>Convert to Symbol** me-

nüsünü tıklıyoruz. Açılan pencerede sembole bir isim veriyoruz. **Type** olarak ise **Graphic** radyo düğmesini tıklayarak onaylıyoruz.

Şimdiki aşama ise bu sembolün bir kopyasını sembolün tam üstünde elde etmek. Bunun için **Edit>Clone** menüsünü kullanıyoruz. Şimdi ikinci yıldızı seçip `180 derece` döndürelim. Bu işlemi de **Modify>Transform>Rotate180º** komutu ile yapıyoruz.



Şimdi her iki sembolü de seçili duruma getirip **Modify>Symbol>Tween Instances**... komutunu tıklayalım. Bu şekilde programa iki sembolümüzün arasını doldurmasını söylemiş olduk. Bu menüyü tıkladığımızda karşımıza resimdeki pencere gelecektir.

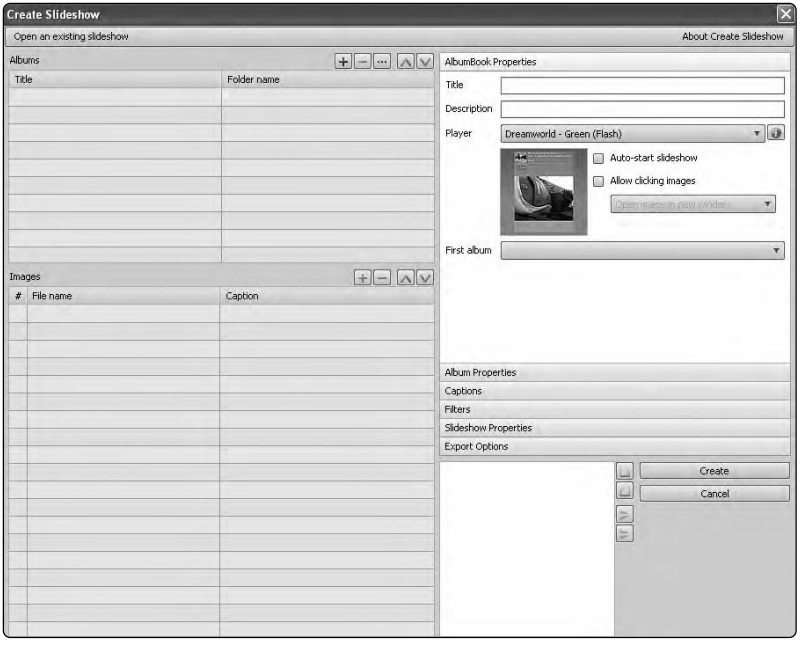
Penceredeki **Steps** kutucuğuna gireceğimiz değer iki sembol arasında kaç adım olacağını belirlemek içindir. Bu değeri biz `20` olarak seçelim. Yani ilk sembolümüzle ikinci sembolümüz arasındaki `180 derece`'lik dönüş `20` adımda gerçekleştirilecektir. **Distribute to States** kutucuğunun da seçili olmasına dikkat ediyoruz. Bu sayede sembollerimiz durumlara dağıtılacaktır.

Adım sayısı olarak `20` seçtiğimiz için bu kutucuğu onayladığımızda **State** paneline bakarak otomatik olarak oluşturulmuş `20` durumu, iki sembolümüzle beraber panelde görebilirsiniz. Artık animasyonumuzu Animated GIF formatında kaydedebiliriz.

SLIDESHOW

Fireworks size birkaç adımda profesyonel kalitede slayt gösterisi hazırlama olanağı sunmaktadır. Üstelik HTML olarak kaydederek slayt gösterinizi web sitenizde de yayınlayabilirsiniz.

O zaman hemen başlayalım ve ilk olarak slayt gösterimizde kullanılmak üzere iki adet albüm klasörü hazırlayıp resimlerimizi içine kopyalayalım. İsterseniz bu ilk denemede **DVD>Uygulamalar>Fireworks** içindeki albümleri de kullanabilirsiniz.

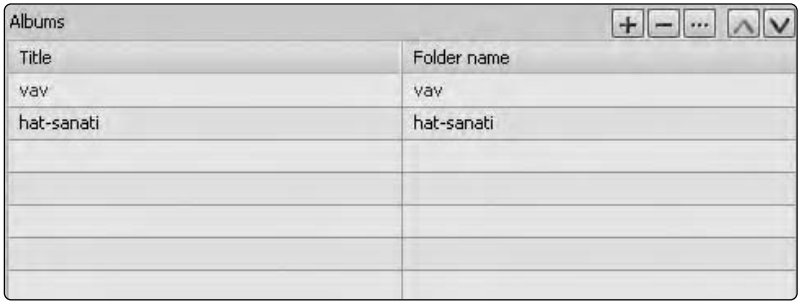


Albümlerimiz hazır olduğuna göre ilk olarak **Commands>Create Slideshow** menüsünü kullanarak **Slide Show** panelini açalım ve **Add an Album** + tuşuna tıklayalım.

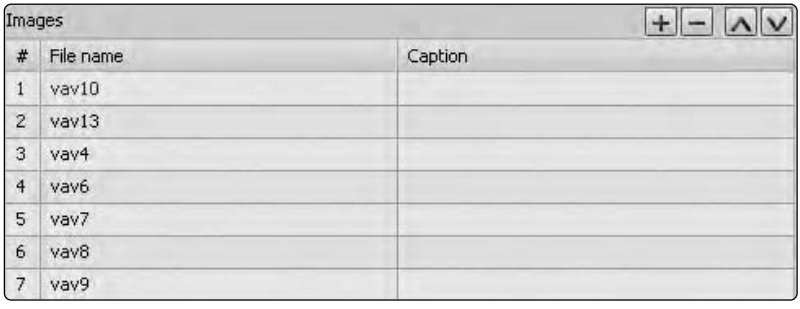


Açılan **Files to Process** penceresinden üç noktalı tuşa tıklayarak albüm klasörümüzü ve klasörümüzün içimdeki tüm resimleri **SHIFT** tuşu ile seçili hale getirip programa gösterelim. Bu işlemi bir kez daha tekrarla-

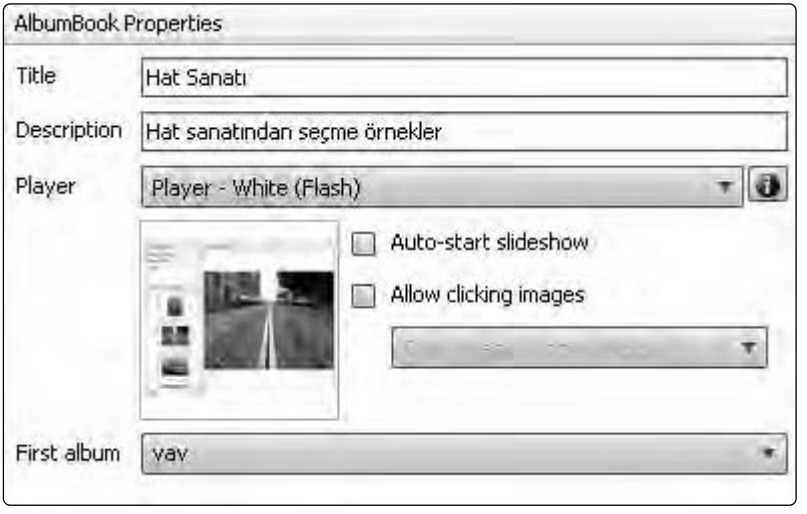
yarak ikinci albüm klasörümüzü ve içindeki resimleri de programa gösterelim.



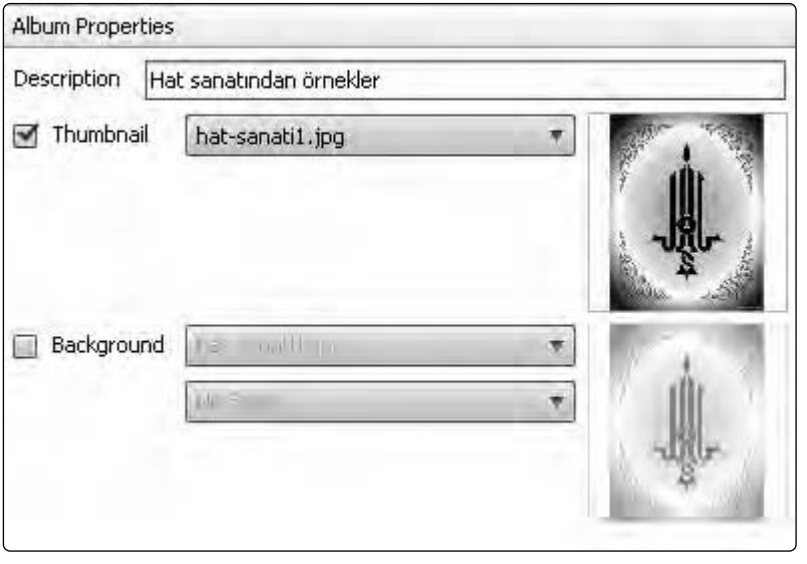
Artık albümlerimiz **Slideshow** sayfasında listelenecektir. Şimdi ilk albümü tıklayarak seçin. Albümü seçtiğinizde alttaki **images** kısmında albüm içindeki resimler listelenecektir.



Buradan **Move Selected Image(s) up** ve **Move Selected Image(s) down** tuşları ile listedeki resimleri aşağı yukarı kaydırarak resimlerin sırasını değiştirebilirsiniz. Ayrıca resimlerin üzerine tıklayarak adlarını da değiştirebilirsiniz. Resim listesinin yan sütunundaki **Caption** kısmından resimlere açıklama girebilirsiniz. Resimler seçili iken **Slideshow** penceresinin sağ alt kısmından seçili olan resmin ön izlemesini görebilirsiniz. Ön izleme resminin yanındaki **Rotate** tuşları ile de gerektiğinde resimleri döndürebilirsiniz.

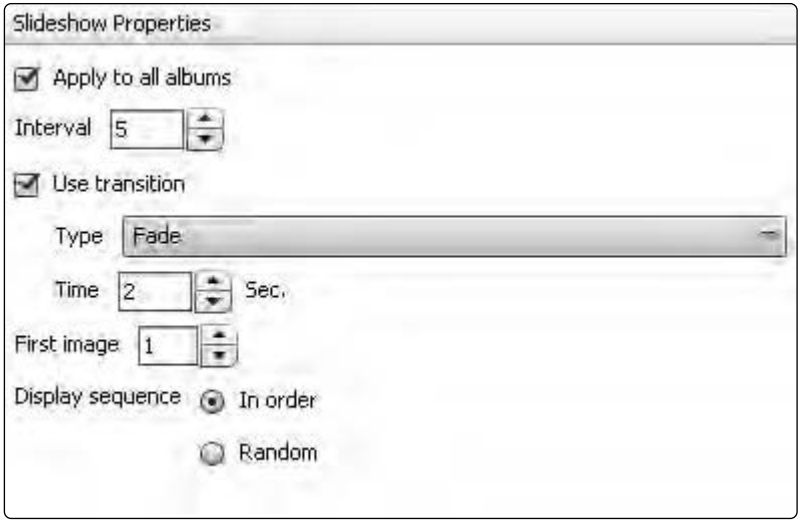


Şimdi, **Slideshow** penceresindeki **AlbumBook Proprties** panelini düzenleyelim. Bu bölüm tüm albümler için geçerli olan özelliklerle ilgilidir. **Title** kutucuğuna albüm kitaplığının adını, **Description** kutucuğuna da albüm kitaplığımız için bir açıklama girelim. **Player** kısmından ise albümümüzün oynatıcısını seçelim.



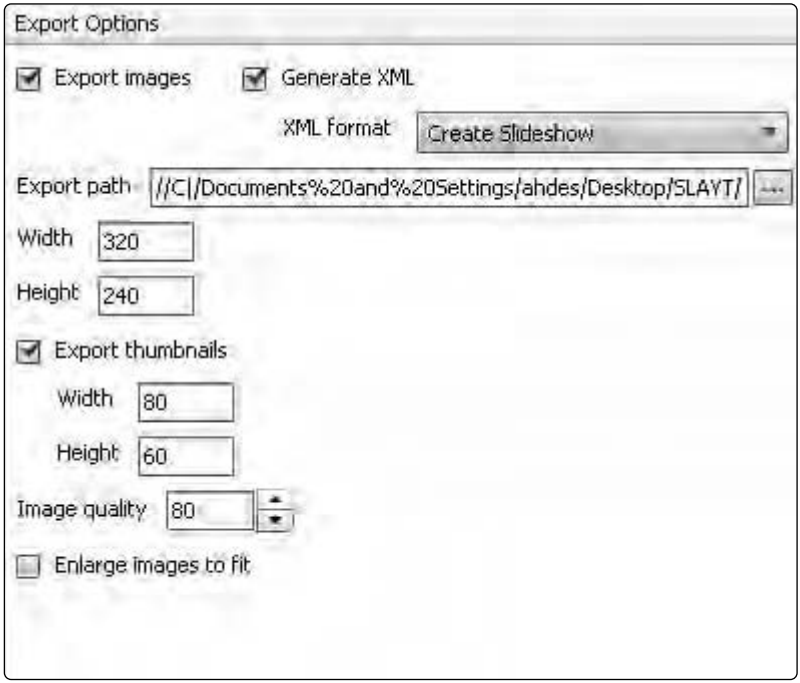
Yolumuza **Album Properties** paneli ile devam ediyoruz. Buradaki ayarlar sadece **Album Properties** panelinden **First album** olarak seçtiğiniz albüme uygulanır. **Description** kutucuğuna albüm ile ilgili bir açıklama girelim. Albümdeki resimlerin ön izlemelerini görebilmek için **Thumbnail** kutucuğunu seçili hale getirelim.

Diğer panelimiz olan **Captions** paneli, resimlerle ilgili açıklama bilgilerini girmek için kullanılır. **Filters** paneli ise resimlere filtre uygulamak içindir. Biz yolumuza **Slideshow Properties** paneli ile devam edelim.



Slideshow Properties paneli slayt gösterisine ait birtakım özellikleri düzenlemek içindir. Burada yapacağınız ayarlamaların tüm albüm kitaplığında geçerli olabilmesi için **Apply to All Album** kutucuğunu etkin hale getirmelisiniz. **Interval** kutucuğundan slayt gösterisindeki resimlerin gösterim süresini ayarlayabilirsiniz.

Use transition kutucuğunu tıklayarak alttaki **Type** açılır menüsünün aktif hale gelmesini sağlayın ve **Type** menüsünden slayt gösterisindeki resimlere uygulayacağınız geçiş efektini seçin. **Time** kutucuğu ile de geçiş süresini belirleyin.



Son panelimiz olan **Export Options Paneli** ile işlemlerimizi tamamlayacağız. Bu panel slayt gösterimizin ihracı ile ilgili ayarları barındırır. **Export Images** kutucuğu resimlerin ihracı için seçili olmalıdır. Generate XML kutucuğunu seçili hale getirerek de programın bir XML dosyası üretmesini sağlamış olursunuz. **Export path** kısmından slayt gösterisinin dosyalarının nereye ihraç edileceğini seçiyoruz. **Width** ve **Height** kutucukları ile slayt gösterisindeki resimlerin ebatlarını belirliyoruz.

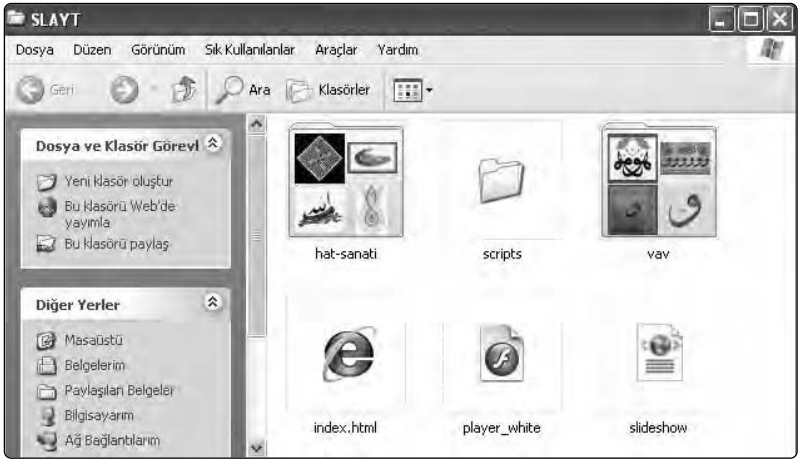
Ön izleme resimlerinin de ihraç edilebilmesi için **Export Thumbnails** kutucuğunu işaretleyip alttaki **Width** ve **Height** kutucukları ile ön izleme resimlerinin ebatlarını belirliyoruz. **Image Quality** kutucuğundan da ihraç edilecek resimlerin kalitesini belirliyoruz. Belirleyebileceğimiz azami değer `100` olmaktadır.

Enlarge Images to Fit kutucuğu tıkladığınızda, slayt gösterisinde kullanılan resimler, gösteri için belirlediğiniz ebatlardan küçükse, program

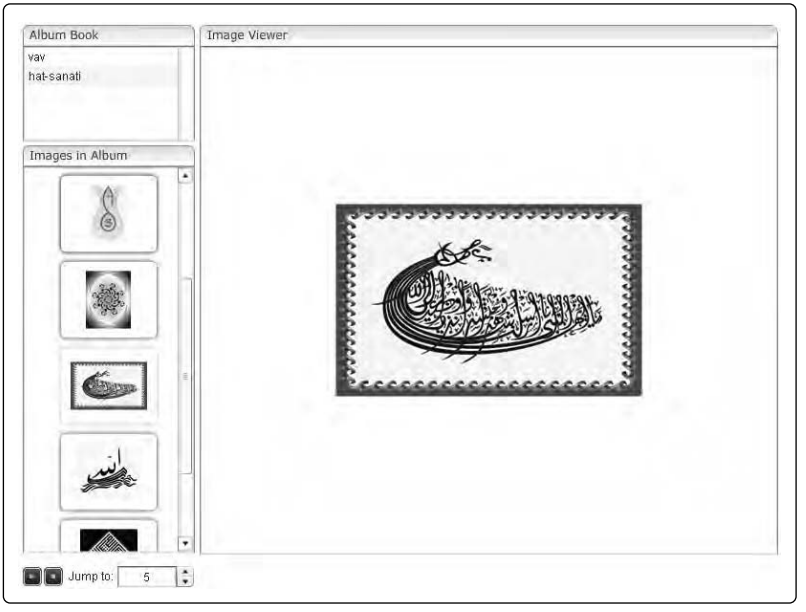
o resimleri belirlediğiniz ebatlara büyütecektir. Onayı kaldırırsanız söz konusu resimler orijinal ebatlarında kalacaklardır.



Artık işin sonuna geldik. Tüm ayarlarımız tamam ve slayt gösterimizin oluşturulmaya başlanması için yapmamız gereken **Create** tuşuna tıklamak. Tuşa tıkladığımızda işlem resimdeki gibi devam eder. Birkaç saniye sonra program işini bitirdiğinde **Done** tuşu ile işlemi sonlandırın.



Şimdi bilgisayarınızda slayt gösterisinin oluşturulması için programa gösterdiğiniz klasörü açın. Resimde gördüğünüz gibi gereken tüm dosyalar Fireworks tarafından oluşturulmuş durumda olacaktır. Şimdi **index.html** dosyasını çift tıklayarak açın.



İşte slayt gösterimiz karşınızda. Şimdi sayfanın sol alt kısmındaki **Play** tuşuna tıklayıp arkanıza yaslanın ve slayt gösterisinin keyfini çıkarın!

OPTİMİZASYON

Sitenizde kullanacağınız resimlerin mümkün oldu kadar küçük boyutlu ama aynı zamanda da kaliteli olması gerekir. Böylece sitenizdeki resimler hem göze hoş gelen bir yapıda olacak hem de sitemiz daha kısa sürede yüklenecektir. Ve siz de kaliteli görselleri olan ve aynı zamanda da hızlı açılan bir siteye sahip olmuş olacaksınız.



Fireworks bu konuda da bize yardımcı olmaktadır. Fireworks ile optimizasyon işlemlerini **Optimize** paneli ile yapıyoruz.

Şimdi panelimizi daha yakından tanıyalım.

Saved settings: Varsayılan optimizasyon seçeneklerinden birini seçmenizi sağlayan bir açılır menüdür. Dilerseniz resminiz için buradaki ayarlardan birini seçebilirsiniz. Bir seçim yaptığınızda, panelde o seçime ait varsayılan değerler görüntülenecektir.

Export file format: İhraç edilecek dosyanın formatını seçmek içindir.

Indexed palette: İstediğiniz renk paletini açılır menüden seçebilirsiniz.

Matte: Arkaplan renk uyumu ayarı içindir.

Colors: Resimde kullanılacak renk adedini belirlemek içindir.

Dither: GIF'lerdeki cansız renkleri kaldırır.

Choose type of Transparecy: Resme uygulanacak saydamlık tipini seçmek içindir.

Swatches (Renk Paleti): Resimdeki mevcut renkleri gösteren renk paletidir.

Sort: Paletteki renkleri kullanım sıklıklarına ya da parlaklılıklarına göre sıralamak içindir.

Add color to trancparecy: Renk paletinden seçtiğiniz rengi saydamlaştırır. Tuşa tıkladıktan sonra beliren damlalık ile saydamlaştırmak istediğiniz renge tıklayın.

Remove color transparency: Saydamlaştırdığınız rengi eski haline çevirmek içindir.

Select transparent color: Sadece tek bir renge uygulanabilen saydamlaştırma işlemidir.

Edit color: Bu tuş, paletten seçtiğiniz rengi düzenlemek için kullanılır.

Sanap to web safe: Paletten seçilen rengi web için uygun hale getiren tuştur.

Lock color: Seçilen rengi kilitlemek için kullanılır.

Add color: Palete yeni renkler eklemek içindir.

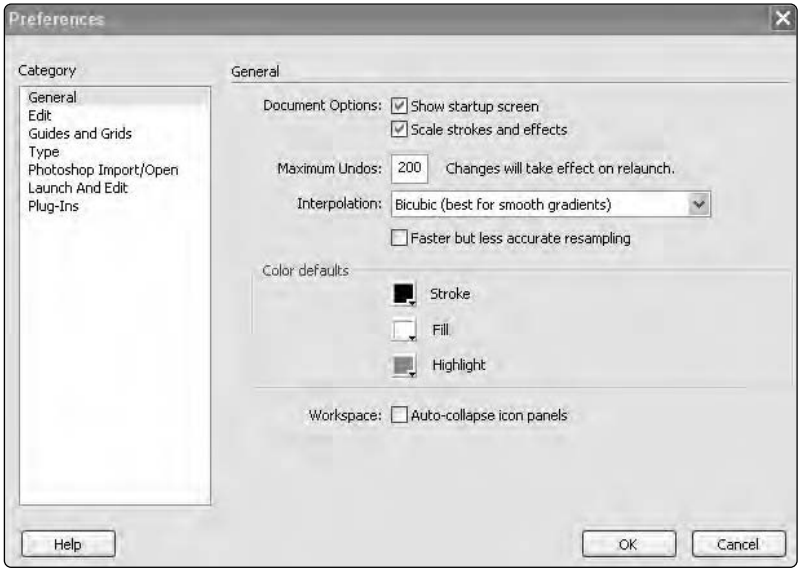
Delete color: Seçilen rengi silmek için kullanılır.

FIREWORKS TERCİHLERİ

Edit>Preferences menüsü altında yer alan bu pencereyi kullanarak Fireworks ile ilgili tercihlerimizi ayarlayacağız.

Şimdi penceredeki tüm kategorileri tek tek inceleyelim.

General Kategorisi



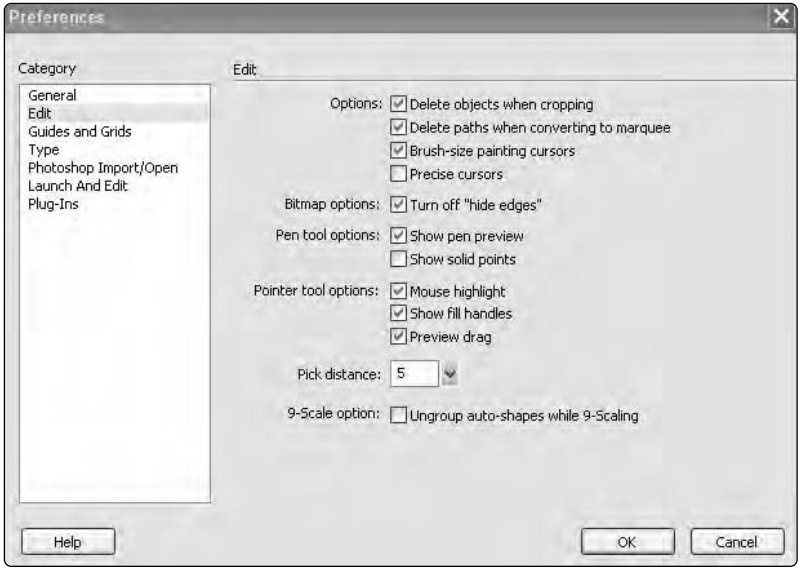
Document Options: Bu kısımda iki adet onay kutucuğu bulunmaktadır. Bunlardan **Show Startup Screen**, Fireworks çalıştırıldığında başlangıç sayfasının görüntülenip görüntülenmeyeceğini belirlemek içindir. **Scale Strokes and Effects** ise, nesneler ölçeklendirilirken dolgu renkleriyle beraber kenar çizgilerinin de ölçeklendirilmesi içindir.

Maximum Undos: Yaptığınız işlemlerin kaç adımının ileri ve geri alınabileceğini belirlemek içindir. Varsayılan değer `200` olup bunu `1009`'a kadar arttırtabilirsiniz. Fakat bu durumda Fireworks'un ihtiyaç duyacağı bellek miktarı da artmış olacaktır.

Interpolation: Bu kısımdan Fireworks'ün resimleri ölçeklendirme işleminde kullanacağı yöntemi seçebilirsiniz. Bunlardan **Bicubic**, varayılan tercih olup en keskin ölçeklendirmeyi ve en yüksek kaliteyi sağlayan yöntemdir. **Bilinear**, **Bicubic** kadar olmamakla beraber bu seçenek ile de keskin sonuçlar elde edebilirsiniz. **Soft** seçeneği keskin detayları bulanıklaştırır. **Nearest Neighbour**, bulanıklık etkisi oluşturmadan tırtıklı kenarlar ve keskin kontrastlar oluşturmak içindir.

Color Defaults: Dış hat rengi, dolgu rengi ve vurgular için renk tercihlerini buradan yapabilirsiniz. Buradaki varsayılan dış hat ve dolgu rengini değiştirdiğinizde, **Tools** panelinde bulunan renk kutularındaki renklerin ayarladığınız renklerle değişmesi için, **Tools** panelinin **Colors** kısmından **Set Defaults Stroke>Fill Colors** tuşuna tıklamanız yeterli olacaktır.

EDIT KATEGORİSİ



Delete object when cropping: Kırpma işlemi esnasında kırpacağınız alan dışında kalan yerlerin kalıcı olarak silinmesini sağlamak içindir.

Delete Paths when converting to marquee: Seçim çerçevesine dönüştürülen yolları kalıcı olarak silmek içindir.

Brush-size painting cursors: Boyama ve silme araçlarına ait **imleçlerin** (*cursor*) boyutlarını ve şekillerini ayarlamak içindir. Varsayılan olarak **+** (artı) işareti kullanılır.

Precise cursors: Boyama ve silme araçlarına ait imleçleri **+** (artı) işaretine dönüştürür.

Show pen preview: Pen aracı ile çalışırken bir sonraki yol için ön izleme oluşturur. Kutucuk seçili olmazsa ön izleme çizgilerini göremezsiniz.

Show solid point: Pen aracı ile çalışırken seçili noktalar içi boş, seçili olmayan noktalar ise dolu olarak görüntülenir.

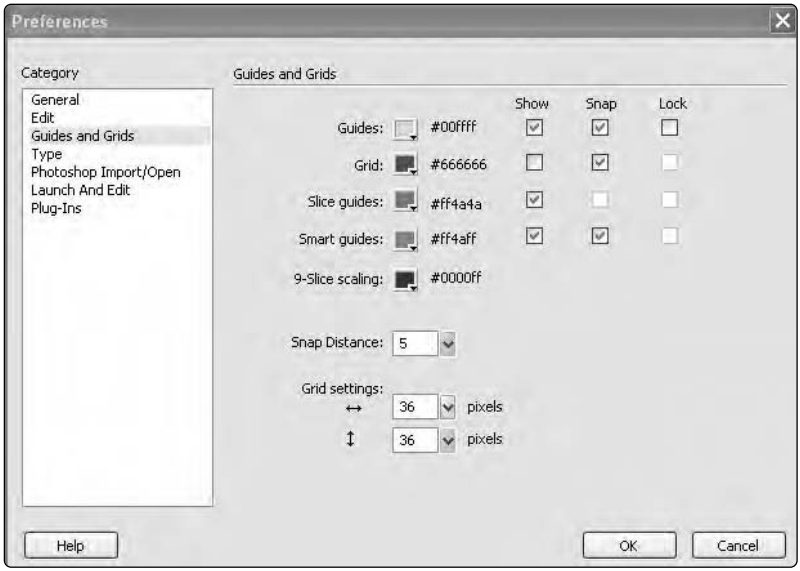
Mouse higlight: Fare ile üzerine geldiğiniz nesnenin tıklanması halinde seçili hale gelecek alanı gösterir.

Show fill handless: Dolgular ile çalışırken dolguyu düzenlemenizi kolaylaştıran tutamaçların gösterilmesi içindir.

Preview drag: Sürüklediğiniz nesnenin yeni konumunun ön izlemesini görebilmeniz içindir.

Pick distance: Bir nesneyi seçebilmek için, fare imlecinin nesneye ne kadar yakın olması gerektiğini belirlemek içindir. `1–10` arası değer alır.

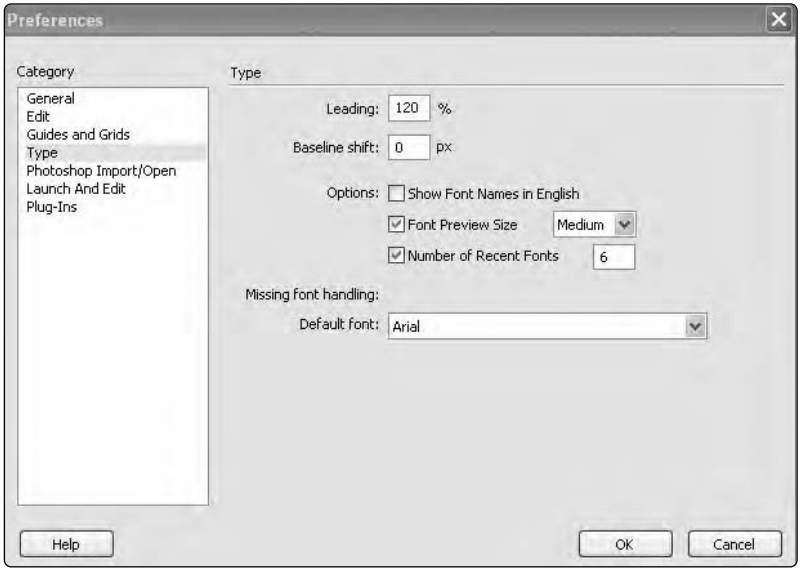
Guides And Grids Kategorisi



Kılavuzların ve ızgaraların renklerini ayarlamak için kullanacağımız kategoridir. Ayrıca **Snap** onay kutucukları ile kılavuzların ve ızgaraların

nesnelere tutturulup tutturulmayacağını, **Show** onay kutucukları ile kılavuzların ve ızgaraların gösterilip gösterilmeyeceğini, **Lock** onay kutucukları ile de kılavuzların kilitlenip kilitlenmeyeceğini belirleyebilirsiniz.

Type Kategorisi



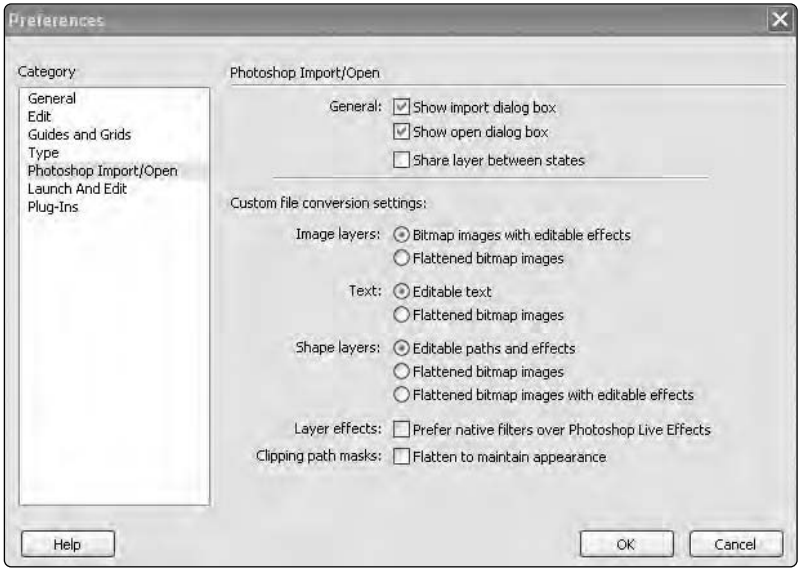
Leading: Satırlar arası boşlukları ifade eder.

Baseline shift: Satır çizgilerini piksel cinsinden belirlemek içindir.

Font Preview Size: Properties panelindeki font tipi seçmek için kullanılan açılır pencerede göreceğiniz font tipi ön izlemesinin, boyutunu belirlemek için kullanılır. **Small**, **Medium** ve **Large** olmak üzere üç seçenek barındırır.

Default Font: Fireworks için varsayılan fonttur. Bilgisayarınızda olmayan bir font kullanan bir belge açtığınızda, burada seçeceğiniz font, o fontun yerini alacaktır.

PHOTOSHOP IMPORT/OPEN KATEGORİSİ



Show Import Dialog Box: Bir PSD dosyasını Fireworks'e ithal edeceğiniz zaman Options penceresinin açılıp açılmayacağını belirlemek içindir.

Show Open Dialog Box: Bir PSD dosyasını Fireworks ile açacağınız zaman Options penceresinin açılıp açılmayacağını belirlemek içindir.

Share Layer Between States: Açtığınız PSD dosyasındaki her katmanın, durumlarda paylaşılmasını isteyip istemediğinizi belirlemek içindir.

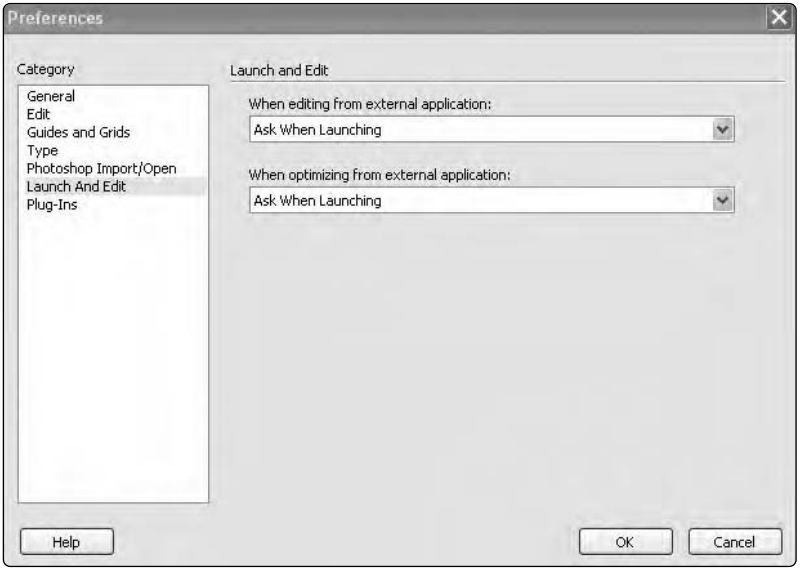
Image layers: İthal ettiğiniz Bitmap resimlerindeki efektlerin Fireworks'te de düzenlenebilir olmasını isteseniz **Bitmap Images With Editable Effects** radyo düğmesi seçili durumda olmalıdır. Düzenlenebilir olmaması içinse **Flattened Bitmap Images** radyo düğmesini seçin.

Text: Dosyadaki yazıların düzenlenebilir olmasını sağlamak için **Editable Text** radyo düğmesi seçili durumda olmalıdır. Düzenlenebilir olmaması içinse **Flattened Bitmap Images** radyo düğmesini seçin.

Shape layers: Yolların ve efektlerin düzenlenebilir olması için Editable Paths and Effects radyo düğmesi seçili durumda olmalıdır. Düzenlenebi-

lir olmamaları içinse **Flattened Bitmap Images** radyo düğmesini seçin. Sadece filtre özelliklerinin düzenlenebilir olmasını istediğinizde ise **Flattened Bitmap Images With Editable Effects** radyo düğmesini seçin.

Lunch And Edit Kategorisi

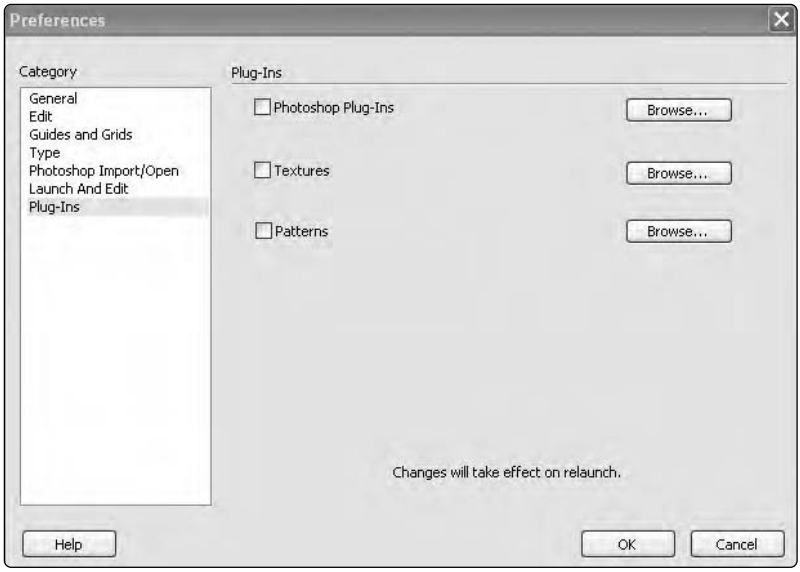


Bu kategori, harici uygulamalarda düzenlenmiş grafikleri düzenleme konusunda bazı tercihleri içerir.

When editing from external application: Harici uygulamalarla düzenlenmiş PNG dosyalarını açarken, Fireworks PNG dosyası açılıp açılmayacağını belirlemek içindir.

When optimizing from external application: Harici uygulamalarla optimize edilmiş PNG dosyalarını açarken, Fireworks PNG dosyası açılıp açılmayacağını belirlemek içindir.

Plug-ins Kategorisi



Bu kategoriden Fireworks'e, **Photoshop** eklentileri, dokular ve desenler ile yeni özellikler ekleyebilirsiniz. Bunun için **Browse**... tuşlarına tıklamalı ve açılan pencereden ilgili dosyanın konumunu göstermelisiniz. Eklediklerinizin kullanılabilir olmaları için Fireworks'ü yeniden başlatmalısınız.

Tüm yaptığınız ayarlamalardan sonra Fireworks varsayılan tercihlerine geri dönmek isterseniz tercihler dosyasını (*userPreferences.txt*) Fireworks kapalıyken silmelisiniz. Bu dosyanın konumu; **Program Files>Adobe>Adobe Fireworks CS4>Configuration> Installer>userPreferences.txt** şeklindedir. Dosyayı sildikten sonra Fireworks'u yeniden başlattığınızda ayarlar ilk haline dönecektir.

Uygulamalar

Dilerseniz şimdi de öğrendiklerimizi pekiştirmek için birkaç örnek uygulama yapalım.

Ve işte başlıyoruz...

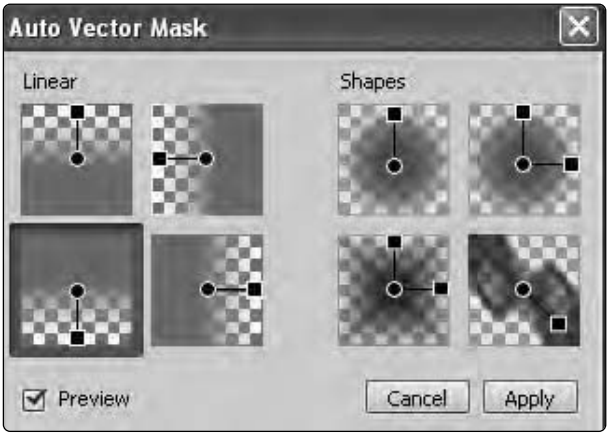
YAZIYA YANSIMA EFEKTİ VERME



İlk olarak yeni bir belge açın ve **Text** aracını kullanarak tuvale metninizi yazın. Ardından metne istediğiniz font büyüklüğünü, font tipini ve font rengini uygulayın. Belgeyi kaydedip **Edit> Clone** ya da **Edit >Dublicate** menülerinden herhangi biri yardımı ile metni çoğaltın.



Ardından ikinci kopyayı **Modify>Transform>Flip Vertical** menüsü yardımıyla ya da **Tool Bar**'daki **Flip Vertical** tuşu ile ters çevirin.



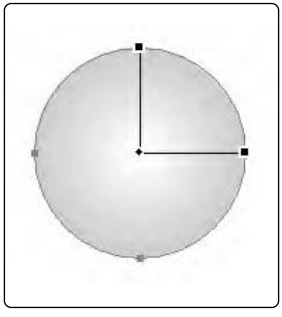
Yine ikinci metin seçili iken **Modify>Flatten Selection** menüsüne tıklayın. Bu işlemle ikinci metin nesnesi artık bir Bitmap resim oldu. Ardından **Command>Creative>Auto vector Mask** menüsüne tıklıyın.

Açılan pencereden soldaki **Liner** grubundan üstü normal altı saydam olan örneği seçin ve **Apply** tuşu ile uygulayın.

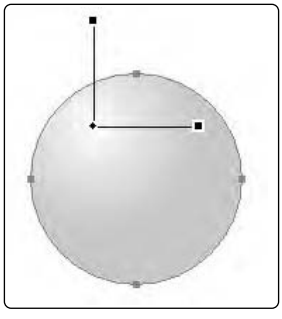


Sayfanın ortasında gördüğünüz çubuğu döndürerek ya da uzatıp kısaltarak saydamlık oranını ve saydamlık yönünü istediğiniz gibi değiştirebilirsiniz.

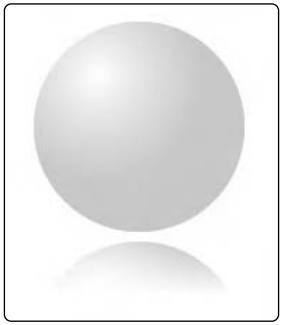
KÜRE YAPIMI



Önce **Ellipse** aracı ile bir daire çizin. Ardından **Properties** panelindeki **Fill category** açılır menüsünden **Gradient>Elipse**'i seçin.



Şimdi ışık efektini daha gerçekçi yapma ve dairenin bir küre gibi görünmesi için çubukları dairenin ortasından sol üst tarafa alın. Evet şimdi daire küreye benzedi.



Dilerseniz bir de kürenin altına yansıma efekti ekleyelim. Bunun için küreyi **Edit >Dublicate** ile çoğaltın.

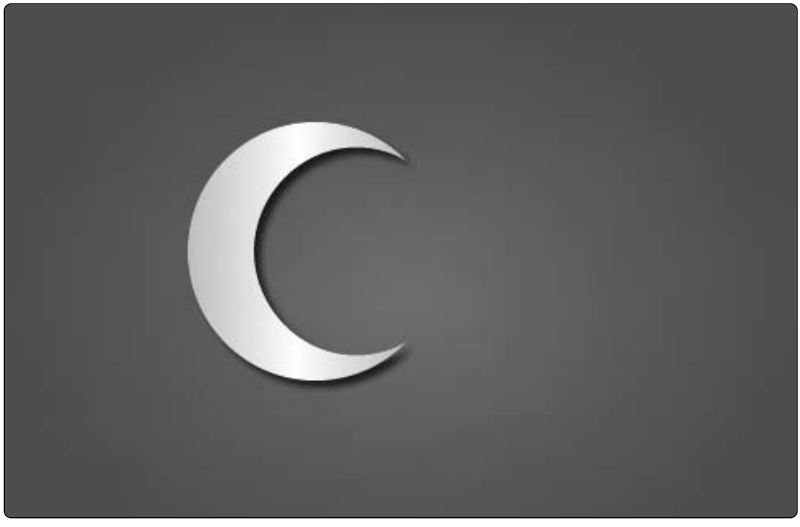
İkinci küre seçili iken **Properties** panelindeki **Fill category** açılır menüsünden yine **Gradient>Elipse**'i seçin. Ve çubuğun boyunu kısaltarak gölgeyi kısaltıp resimdeki hale getirin.

Türk Bayrağı Yapımı



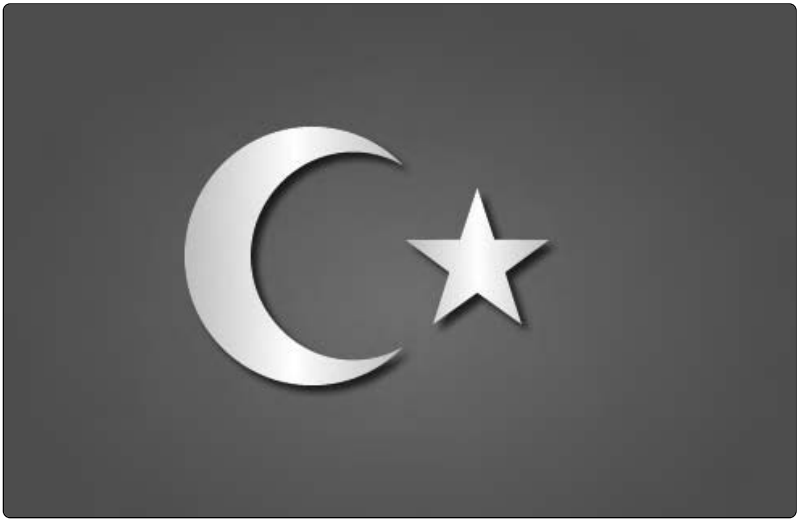
Bayrağımızı yapabilmek için öncellikle `500*320` ebatlarında yeni bir belge açalım ve tam belgemizin ebatlarında bir dikdörtgen çizelim. Dikdörtgenimizi koyu kırmızı renk ile dolduralım.

Ardından **Properties** panelindeki **Fill category** açılır menüsünden **Gradient>Elipse**'i seçelim. **Fill category** açılır menüsünün solundaki renk kutucuğuna tıklayarak gradient penceresini açıp sol başta açık kırmızı sağ başta ise koyu kırmızı renkte olacak şekilde ayarlayalım.



Sırada "ay" yapımı var. Bunun için Fireworks'ün bize sunduğu kolaylıktan yaralanıyoruz ve **Shapes** panelindeki ay şeklini fare ile tutup bayrağımıza sürüklüyoruz. Buradaki ay bize göre ters olduğundan **Modify >Transform> Rotate 180º** menüsü ile döndürüyoruz. Şimdi aya gradient efekti vereceğiz.

Bunun için **Fill category** açılır menüsünden **Gradient>Bar**'ı seçelim. **Gradient** efektinin daha etkili olabilmesi için renk kutucuğuna tıklayarak gradient penceresini açıp sol başta beyaz sağ başta ise açık gri renkte olacak şekilde ayarlayalım. Ay için yapacağımız son işlem gölge efekti vermek. Hemen **Properties** panelindeki **Filters** (**+**) tuşuna tıklayalım ve açılan menüden **Show and Glow>Drop Shadow**'u seçelim. Artık yıldıza geçebiliriz.



Tools panelinden **Star** aracı ile yıldızımızı çizelim. Ay için yaptığımızın tam olarak aynısını yaparak yıldıza da gradient efekti verelim. Ardından yine aynı yöntemle gölge efekti verelim. Son olarak ay ve yıldız nesnelerini üzerlerindeki sarı tutamaçlar yardımı ile biraz inceltelim.

Ve işte bayrağımız...

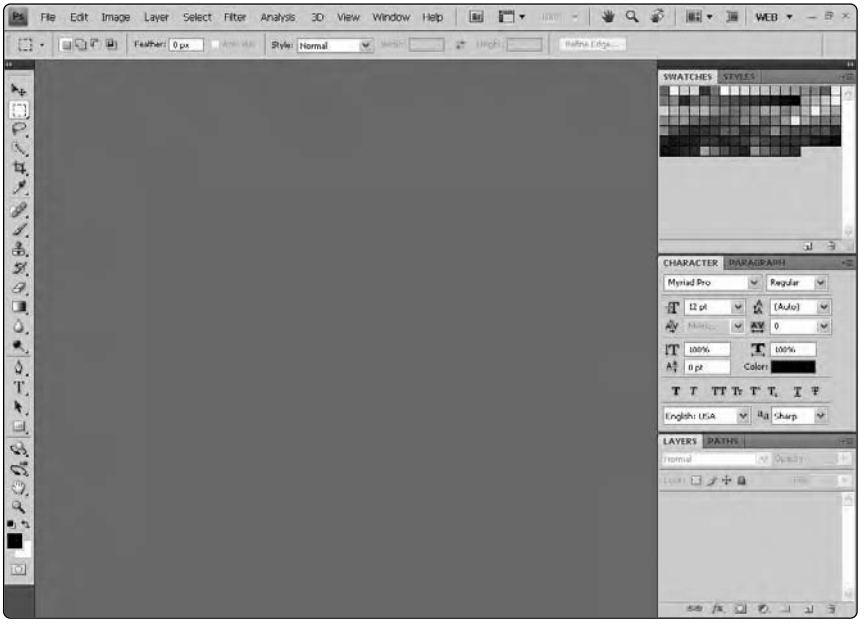
9

PHOTOSHOP

Photoshop Çalışma Alanı	470	Filtreler	509
Zevkler ve Renkler	476	Çizim	516
Seçim İşlemleri	482	Metinler	519
Temel Rötuş İşlemleri	486	Web Sayfaları	524
Katmanlar	492	Photoshop CS4 Automate	528
Fırçalar	498	Uygulamalar	534

Photoshop, bir fotoğraf ve grafik düzenleme editörüdür. Görsellerinizi renk, kontrast, ebat... vb birçok açıdan en ince detaylarına kadar işleyebileceğiniz üstün yeteneklere sahip bir programdır. Bir web master olarak hazırlayacağınız sitelerde, banner, logo... vb çalışmalarınızda ve yapacağınız daha pek çok görsel uygulamada en önemli yardımcılarınızdan biri olacaktır.

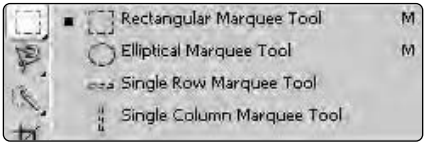
PHOTOSHOP ÇALIŞMA ALANI



Photoshop bitmap resimlerinizi ve vektörel grafiklerinizi profesyonelce ve hızlı bir biçimde düzenlemenizi sağlayan menülere, araçlara ve panellere sahiptir. Photoshop penceresinin en üst kısmında **menüler**, menülerin altında ise **Options** paneli yer almaktadır. Çalışma alanının en solunda resimde gördüğünüz gibi **Tools** paneli, sağ kısımda ise **Character**, **Paragraph**, **Layers**, **Brushes**... vd Photoshop panellerini bulunur.

PHOTOSHOP ARAÇLARI (TOOLS)

Tools (*Araçlar*) panelini sol üst kısmında bulunan küçük çift ok işaretine tıklayarak çift ya da tek sütunlu hale getirebilirsiniz. **Tools** panelindeki bazı araçların sağ alt köşelerindeki küçük üçgenler bu araç altında başka alt araçların da olduğu anlamına gelir.



Fare ile bu üçgenlere tıkladığınızda ya da fareyi araç tuşu üzerinde basılı tuttuğunuzda alt araçlar menüsü açılacaktır. Kullanmak istediğiniz aracı fare ile bir kez tıklamanız söz konusu aracın kullanılır hale gelmesi için yeterlidir. Tuşların isimlerinin yanında parantez içinde gördüğünüz harfler, o araçla ilgili kısayol tuşlarıdır.

HAZIR AYARLAR

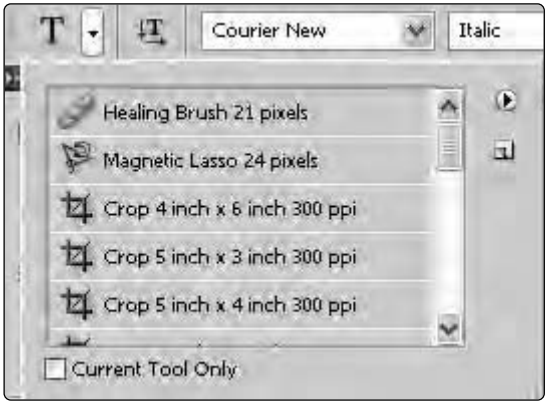


Tüm Photoshop araçları kendilerine has bir takım ayarlar içerirler. **Tools** panelinde bir aracı tıkladığınızda üstteki **Options panelinde o araçla** ilgili ayarlar belirecektir. Bu ayarlar yardımıyla aracınızı çalışmanızın gerektirdiği değerlerde ayarlayabilirsiniz.

Options panelinin en solundaki araç sembolünün yanında bulunan küçük ok işaretine tıkladığınızda açılacak pencereden Photoshop'ta sık kullanılan bazı araç ayarlarına ulaşabilirsiniz. İsterseniz siz de sık kullandığınız bir araç ayarını hazır ayar olarak kaydedebilirsiniz.



Örneğin, metin oluşturmak için kullandığımız **Type Tool** aracını sıklıkla, **`Courier New`** fontu ile, kırmızı renkte, `11 punto` büyüklüğüyle, *italik* ve **kalın yazı** olarak kullandığınızı var sayalım. Her seferinde aracı bu ayarlara getirmeye uğraşmak yerine hazır ayar olarak kaydetmek size zaman kazandıracaktır.



Bunun için **Tools** panelinde araç seçili iken **Options** panelinden istediğiniz ayarları oluşturun. Ardından **Options** panelindeki seçtiğiniz araç resminin yanındaki ok işareti ile resimde gördüğünüz hazır ayar penceresini açın. Pencerenin sağ kısmında bulunan yeni hazır ayar oluşturma tuşuna tıklayın.



Açılan pencereden ayarınıza bir ad verebilirsiniz. **OK** tuşu ile işlemi sonlandırdığınızda artık hazır ayar açılır penceresinde sizin ayarınız da yer alacaktır. Gerektiğinde bu aracı hazır ayar pencereden seçerek, hazırlamış olduğunuz ayarlarla kullanabilirsiniz.

Görüntüyü Sürükleme ve Büyütme

Sürükleme işlemi için **Tools** panelinden **Hand Toll** aracını kullanacağız. Araç seçiliyken fareye basılı tutarak görüntüyü istediğiniz yöne sürükleyebilirsiniz.

Büyütme işlemi içinse **Zoom Tool** aracını kullanacağız. Bu aracı seçtiğinizde üstteki **Options** panelinde **yakınlaştırma** ve **uzaklaştırma** tuşları görüntülenecektir. Bu tuşlar yardımıyla görüntünün istediğiniz yerine tıklayarak görüntüyü uzaklaştırabilir ya da yakınlaştırabilirsiniz. Bu işlemi, görüntünün istediğiniz bir bölgesini fareyi basılı tutup bir seçim alanı oluşturarak da yapabilirsiniz.

Durum Çubuğu Kullanımı



Belgenizin alt kısmında yer alır. Resimde gördüğünüz gibi belgenin yakınlaştırma oranını ve belge ile ilgili bir takım bilgileri gösterir. Durum çubuğunun sağındaki ok şeklindeki tuşa tıklayarak açılan pencereden durum çubuğunda görüntülenmesini istediğiniz bilgiyi seçebilirsiniz.

Cetvel, Izgara ve Kılavuz Kullanımı

Çalışmalarınızı daha hassas bir biçimde konumlandırmanızı kolaylaştıracak olan cetvelleri, **Wiew>Rulers** menüsünden açabilirsiniz.



Tools panelindeki **Ruler Tool** aracı ise **Eyedropper Tool** aracının altında yer almakta ve çalışma alanındaki herhangi iki nokta arasındaki mesafeyi hesaplamanızı sağlamaktadır. Aracı seçip ölçmek istediğiniz ilk noktadan son noktaya doğru sürükleyin. Üstteki **Options** panelinde iki mesafe arası ile ilgili değerleri görebilirsiniz. İki mesafe arasında çizilen çizginin üzerine geldiğinizde **+** şeklindeki imleç ok işaretine dönüşecektir. Bu durumda çizgiyi istediğiniz bir başka konuma sürükleyebilirsiniz. Çizginiz her iki başında bulunan **+** işaretlerinden tutarak da ölçmek istediğiniz mesafeyi kısaltabilir ya da uzatabilirsiniz.

Konumlandırma işlemlerinde bir diğer yardımcınız da ızgaralar ve kılavuzlardır. Izgaraları kullanmak için **Wiew>Show>Grid** kılavuzlar içinse **Wiew>Show>Guides** ile **Wiew>Show>Smart Guides** menülerinden yararlanabilirsiniz.



Cetvelleri fare ile tutup belgenize sürükleyerek istediğiniz konumda bir kılavuz oluşturarak daha hassas konumlandırma imkanına kavuşmuş olursunuz.

Oluşturduğunuz kılavuzları sabitlemek isterseniz **Wiew>Lock Guides** menüsünü, tüm kılavuzları tek seferde kaldırmak istediğinizde ise **Wiew>Clear Guides** menüsünü kullanmalısınız.

Yapılan İşlemleri Geri Almak

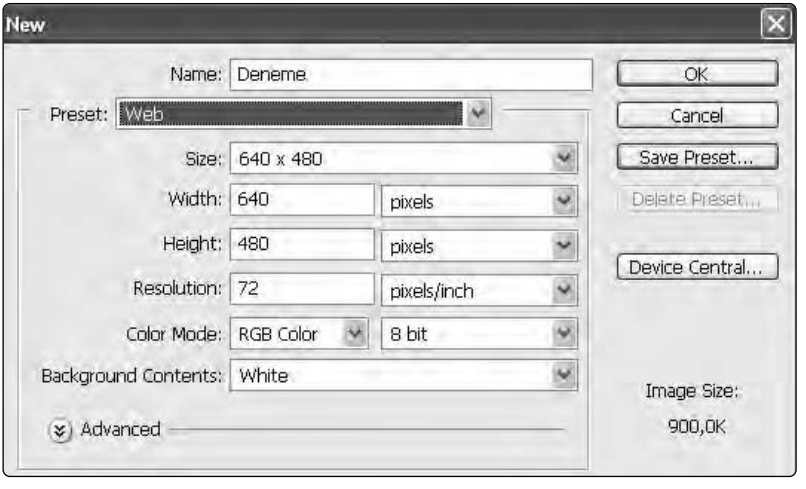
Photoshop'ta bir işlemi geri almak için **Edit>Undo** menüsü ya da **CTRL+Z** kısayol kombinasyonu kullanılır. Fakat bu şekilde sadece bir eylem (varsayılan olarak belleğe aşırı yük bindirmemesi için) geri alınabilir. Daha fazlası için **History** panelinden yararlanacaksınız.



History panelinde yaptığınız son 20 işlemi görebilirsiniz. Dönmek istediğiniz adımı tıkladığınızda o adıma dönersiniz ve çalışmaya döndüğünüz adımdan devam edebilirsiniz. Fakat bu takdirde, yani döndüğünüz noktada yeni bir işlem yaptığınızda geri döndüğünüz noktadan sonraki eski eylemleri artık unutabilirsiniz!

Olur da **History** panelindeki eski işlem adımlarını geri getirmek istersenız **Edit>Undo** menüsü kullanın.

Yeni Belge Açma ve Kaydetme



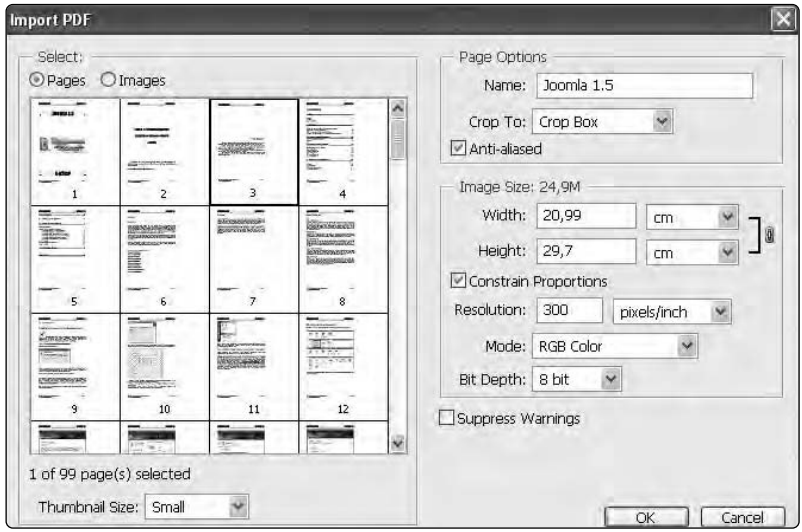
Yeni bir belge oluşturmak için **File>New** menüsünden yararlanacağız.

Açılacak olan **New** penceresi yardımıyla belgemizin adını, ebatlarını, renk modunu, arka plan rengini ve çözünürlülüğünü seçerek **OK** tuşu ile belgemizi açmış olacağız. Bir belgedeki nesneyi, yeni açacağınız bir belgeye yapıştırmak üzere **Edit>Copy** menüsü ile kopyaladıktan sonra **File>New** dediğinizde, açılacak olan **New** penceresindeki tuval boyutları, panoya kopyaladığınız nesnenin ebatları kadar olacaktır. Daha sonra nesneyi *Edit>Paste* menüsü yardımıyla belgenize yapıştırabilirsiniz.

Çalışmanızı kaydetmek için **File>Save** menüsünü kullanmalısınız. Açılan penceredeki **Format** açılır kutucuğundan dosyayı kaydedeceğiniz formatı seçebilirsiniz. Photoshop'un işlenebilir formatı **PSD**'dir. Bu formatta kaydettiğiniz dosyalarınızı daha sonra tekrar açıp çalışmanıza kaldığınız yerden devam edebilir ya da değişiklikler yapabilirsiniz.

Kaydettiğiniz dosyayı bir başka formatta bir kez daha kaydetmek için ise **File>Save As** menüsünü kullanabilir ve açılan pencereden farklı kaydetme işlemi için kullanacağınız dosya formatını belirleyebilirsiniz.

Mevcut bir dosyayı açmak için **File>Open**, son kullandığınız belgeleri açmak içinse **File>Open Recent** menülerini kullanabilirsiniz. Gerçek formatına uygun olmayan bir formatta kaydedilmiş belgeleri düzgün olarak açabilmek için **File>Open As** menüsünü ve açılan pencereden dosyanın formatını seçin.



File>Open menüsü yardımıyla bir PDF dosyasını açmak isterseniz karşınıza resimdeki pencere gelecektir. Pencereden PDF dosyasındaki sayfaları mı resimleri mi açmak istediğinizi pencerenin üst kısmındaki **Select** bölümünde bulunan **Pages** ve **Images** radyo düğmeleri yardımıyla belirleyebilirsiniz. Dilerseniz birden fazla sayfa seçmeniz de mümkündür. Seçtiğiniz her sayfa ya da resim ayrı bir belge olarak açılacaktır.

Zevkler ve Renkler

Zevkler kişiden kişiye değişen bir konu olduğundan kitabımızın kapsama alanı dışındadır. Fakat renkler her Photoshop kullanıcısının temel uğraşılarından biridir. Tabi renklerle uğraşmak çok zevkli bir konu olduğundan başlığımızı zevkler ve renkler şeklinde attık. Yani şunu demek istiyoruz ki Photoshop'ta başarılı olmak bir bakıma renklerle uğraşmaktan zevk almaya bağlı. O zaman hiç vakit kaybetmeden zevkli renklerle haşır neşir olmaya başlayalım.

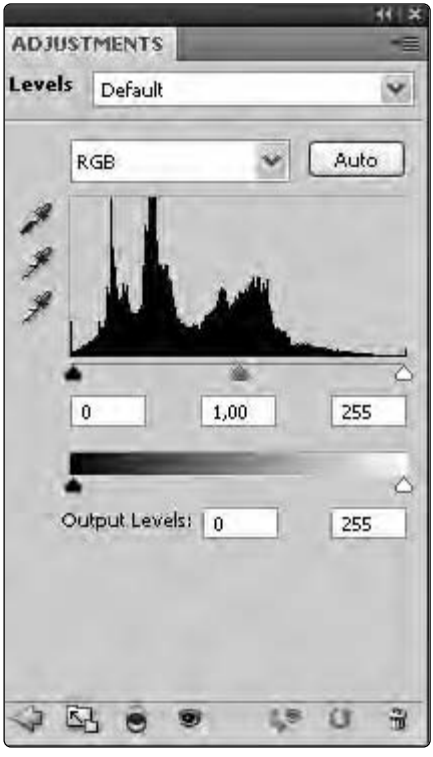
Tabi renklerle kabaca uğraşıyorsanız bunu bazı ufak tefek programlarla da yapabilirsiniz. Ama renklerle profesyonelce uğraşıyorsanız hiçbir program Photoshop'un eline su dökemez. Peki, Photoshop bunu nasıl başarıyor? Tabi ki sahip olduğu, çok hassas ayarlamalar yapabilmemizi sağlayan araçlarıyla, panelleriyle, menüleriyle ve daha bir çok gelişmiş özelliğiyle...



İşte tam da bu noktada işe Histogramlar ile başlayabiliriz. **Histogram**, resminizdeki piksellerin adedini renk yoğunlukları bazında grafiksel olarak gösterir. Histogram üzerinde soldan sağa doğru gölgeler orta tonlar ve vurgular dağılır. Resimlerinizdeki ton ve renk bilgilerini daha detaylı görebilmek için resimde gördüğünüz **Histogram** panelinden yararlanabilirsiniz.



Resminizdeki renkleri ve tonları ayarlamak içinse **Adjustments** panelini kullanabilirsiniz. Paneldeki bir seçeneği seçip bir ayarlama yaptığınızda ayrı bir ayarlama katmanı oluşturulur. Bu katmanı **Layers** panelinden görebilirsiniz. Panelde, yukarıdaki resimde gördüğünüz gibi fotoğraflarınıza uygulayabileceğiniz siyah-beyaz, poz ayarı, ton ve doygunluk ayarı... vb ayarlar mevcuttur.

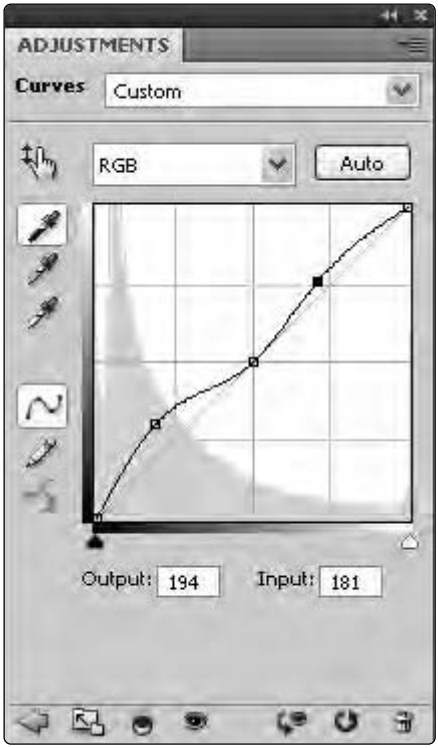


Bunlardan birini seçip uyguladığınızda resminize bu ayarla ilgili hazır değerler uygulanacaktır. Fakat panel yardımıyla bu değerleri istediğiniz şekilde değiştirebilirsiniz.

Örneğin yandaki resimde gördüğünüz **Levels** ayarı ile ikinci resmi, resmin gölge, ton ve vurgu ayarlarıyla oynayarak daha aydınlık bir resim haline getirmiş olduk. Resmin orijinal haline bakarak Photoshop'un sağlamış olduğu hassas ayarlama olanaklarının oluşturduğu farkı daha iyi anlayabiliriz. Buradaki ayarların hangi hassasiyette olacağı konusunda ise bol bol alıştırma yaparak zamanla tecrübe kazanabilirsiniz.







Bu ayarlamaları **Curves** ayarını kullanarak da yapabilirsiniz. **Curves** histogramındaki eğri üzerinde 14 adede kadar fare ile tıklayarak kontrol noktası oluşturabilir ve noktaları kullanarak ayarlamalarınızı hassas bir biçimde yapabilirsiniz. Bir kontrol noktasını silmeniz gerekirse noktayı tutup histogramın dışına sürüklemeniz yeterlidir.



Resminizi siyah beyaz yapmak isterseniz yine **Adjustments** paneli yardımıyla bunu kolaylıkla yapabilirsiniz. Bunun için **Black&White** ayarını kullanabilirsiniz. **Auto** tuşunu kullanarak Photoshop'un sizin için resminize göre en uygun tonları uygulayarak resminizi siyah-beyaz hale getirmesini sağlayabilirsiniz. Ayrıca dilerseniz renk sürgülerini kullanarak ayarları istediğiniz gibi özelleştirebilirsiniz. Ayrıca bu panel yardımıyla **parlaklık ve kontrast** (*Brightness & Contrast*), **renk dengesi** (*Color Balance*), **filtre uygulama** (*Photo Fitler*)... vd birçok renk ayarını düzenleyebilirsiniz.

RENK SEÇME

Photoshop çalışmanızda yapacağınız seçimleri boyamak istediğinizde ön plan rengini, çalışmanızın sileceğiniz alanlarını doldurmak içinse arka plan rengini kullanmalısınız.

TOOLS PANELİNDEN RENK SEÇME



Tools panelinden ön plan rengi seçmek için resimde gördüğünüz, önde yer alan renk kutucuğunu tıklayın. Açılan renk seçim penceresinden istediğiniz rengi seçin. Yine aynı şekilde arkada yer alan renk kutucuğuna tıklayarak da arka plan rengini seçebilirsiniz. Arka plan rengi ile ön plan rengini birbirleriyle değiştirmek için de resimde gördüğünüz iki ucunda ok işareti olan simgeyi tıklamanız yeterlidir. Varsayılan renklere dönmek gerektiğinde ise resmin sol üst kısmında gördüğünüz küçük renk kutucukları simgesine tıklayabilirsiniz.

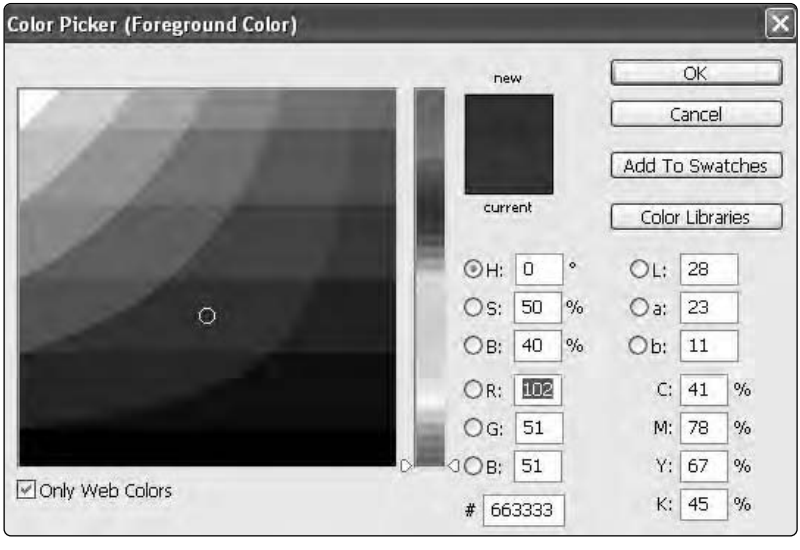
DAMLALIK ARACI İLE RENK SEÇME

Damlalık aracı ile renk seçmek için öncelikle **Eyedropper Tool** aracını seçin. Damlalık aracının örnekleme boyutunu belirlemek için araç seçiliyken **Options** penelinden **Sample Size** açılır kutucuğundan yararlanın. Buradaki ayarlardan **Point sample**, damlalık aracıyla tıklayacağınız pikselin renk değerini alacaktır.

Kutucukta gördüğünüz `3x3 Average`, `5x5 Average`, `11x11 Average`, `31x31 Average`, `51x51 Average`, `101x101 Average` seçenekleri ise, damlalıkla seçim yapacağınız yerdeki belirtilen sayıdaki pikselin ortalama değerini almak içindir. Ardından **Sample** açılır kutucuğundan renk örneğinin tüm katmanlardan mı (*All Layers*) yoksa sadece etkin katmandan mı (*Current Layer*) alınacağını belirleyin.

Artık resim üzerinde istediğiniz bölgeye damlalıkla tıklayarak renginizi seçebilirsiniz. Seçimi arka plan rengi için yapıyorsanız **ALT** tuşunu da basılı tutmalısınız.

RENK SEÇİM PENCERESİ



Renk seçim penceresi, renkler üzerinde ayrıntılı kontrol imkanı vermektedir. Bize seçtiğimiz rengin **HSB**, **RGB**, **Lab**, **CMYK** ve **onaltılık** sayı-

sal değerlerini aynı anda görme ve değiştirme imkanı sunmaktadır. Pencerenin sol alt kısmındaki **Only Web Colors** kutucuğunu işaretleyerek gerektiğinde sadece Web ile uyumlu `216` renk ile de çalışabilirsiniz. Bir renk seçtiğinizde bu renk web ile uyumlu değilse pencerede bir küp simgesi  belirir. Bu küp resmine tıklayarak seçtiğiniz renge en yakın web uyumlu rengi seçebilirsiniz.

Seçim İşlemleri

Bir işlemi çalışmanızın tamamına, belli bir bölümüne ya da belli bir katmana uygulamak için çeşitli seçimler yapmanız gerekir. Bir katmandaki tüm pikselleri seçmek için **Select>All** menüsünü, seçimi kaldırmak içinse **Select>Deselect** menüsünü kullanabilirsiniz. Son yaptığınız seçim işlemini bir kez daha yapmanız gerektiğinde **Select>Reselect** menüsünden yararlanın.

Mevcut tüm katmanları seçmek için **Select>All Layers** menüsünden, seçimi kaldırmak içinse **Select>Deselect Layers** menüsünden yararlanın.

Marquee Araçları ile Seçim

Marquee Araçları'ndan **Rectangular Marquee Tool**  aracı dikdörtgen şeklinde seçimler yapmanızı sağlayan araçtır. Bu aracı **`SHIFT`** tuşu ile beraber kullanarak kare şeklinde seçimler yapabilirsiniz.

Eliptical Marquee Tool  aracı elips şeklinde seçimler yapmanızı sağlar. Bu aracı **`SHIFT`** tuşu ile beraber kullanarak daire şeklinde seçimler yapabilirsiniz.

Single Row Marquee Tool  ve **Single Column Marquee Tool**  araçları ise `1 piksel` olarak satır veya sütun seçer.

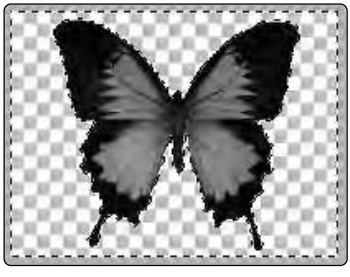
Losso Tool (*kement*) aracı  belli bir alanda serbest seçimler yapabilmek içindir. **Polygonal Losso Tool** aracı  tıklamalar ile oluşturacağınız noktalar yardımıyla daha hassas seçimler yapmanıza olanak tanır. **Magnetic Losso Tool** aracı  ise özellikle seçim yapacağınız alanın çevresinde kontrast yeterli ise daha hızlı seçim yapmanızı sağlayacaktır.

Quick Selection Tool aracı  resimdeki kenarı bulur ve hızlı bir seçim imkanı sağlar. **Magic Wand Tool**  aracı ise piksellerdeki renk

benzerliklerini baz alarak seçim yapar. **Options** panelinden **Tolerance** değerini piksel cinsinden `0–255` arası bir değer girebilirsiniz. Değer **azaldıkça** araç ile tıkladığınız renge en yakın renkler seçilirken, değeri arttırdıkça daha çok benzer renk seçilecektir.

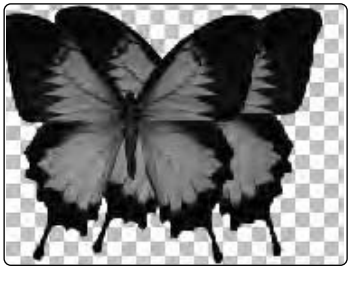
Yukarıda değindiğim tüm Lasso araçları, **Quick Selection** ve **Magic Wand** araçları ile beraber **SHIFT** tuşuna basılı tutarak seçimlerinize ekleme yapabilirsiniz. Burada hatırlatmak istediğim bir husus da bir seçim yaptıktan sonra yapacağınız işlemlerin tüm çalışmanızı değil sadece seçiminizi etkileyeceği hususudur.

Seçimi Taşımak



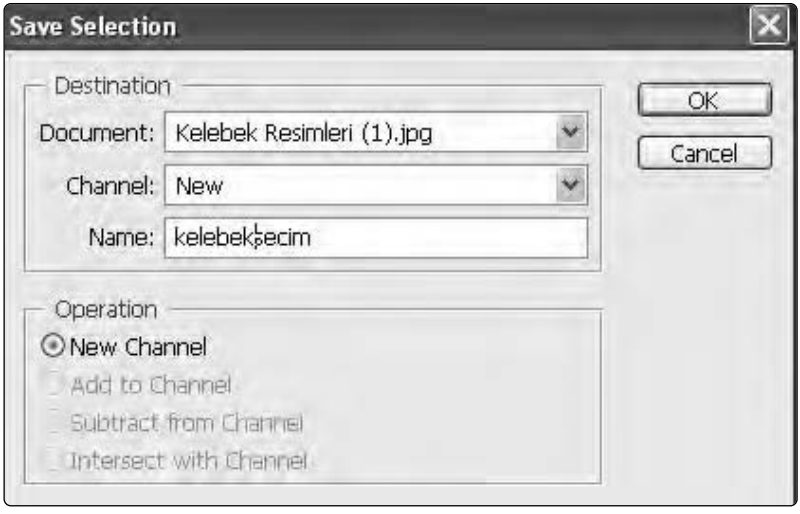


Yapmış olduğunuz seçimi başka bir çalışmanıza taşımak için **Move Tool** aracından yararlanabilirsiniz. Seçiminizi yaptıktan sonra seçili kısmı bu araçla sürükleyip diğer çalışmanıza ekleyebilirsiniz.

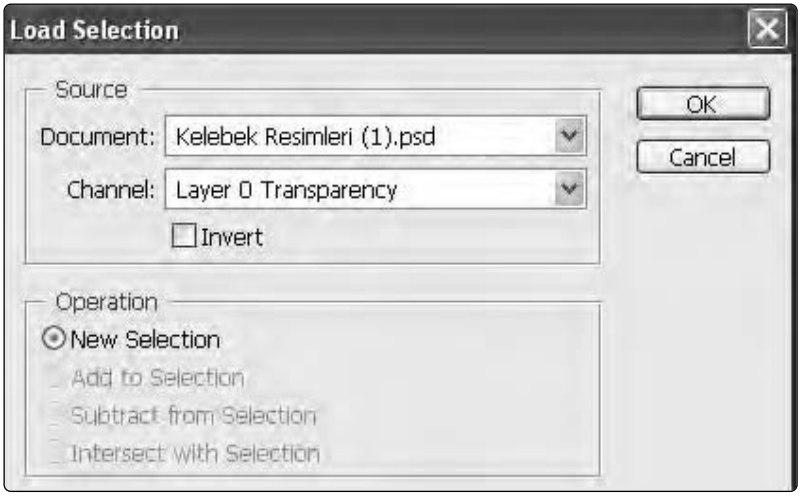


Seçimi taşırken aynı zamanda da seçimin bir kopyasını oluşturmak için **Move Tool** aracı ile aynı alda **ALT** tuşunu da basılı tutmalısınız. Çift ok işareti gördüğünde, fare ile sürüklediğinizde seçimin kopyası oluşturulmuş olacaktır. Bu işlemleri **Edit** menüsündeki kopyalama ve yapıştırma komutları ile de yapabilirsiniz.

Seçimi Kaydetmek



Bir seçimi ilerde kullanmak için kaydedebilirsiniz. Bunun için **Select>Save Selection** menüsünden yararlanabilirsiniz. Açılan pencereden seçiminize bir ad vererek kaydedin. Dosyanızı PSD formatında kaydedip kapatın.



Dosyayı tekrar açın ve **Select>Load Selection** komutu ile açılan pencereyi **OK** ile geçtiğinizde seçiminiz tam da kaydettiğiniz şekilde karşınızda olduğunu göreceksiniz.

MASKELER

Photoshop'ta maskeler, çalışmanızın bir bölümünü yapacağınız işlemlerden etkilenmeyecek şekilde korumaya almak için kullanılır. Örneğin yaptığınız bir seçimi baz alan bir maske oluşturduğunuzda seçiminizin dışında kalan alan maskelenir ve düzenlenemez hale gelir.

Öncelikle çalışmanızın değiştirilecek kısmını seçin ve ardından **Quick Mask** (*hızlı maske*) tuşuna tıklayın. Böylece seçim dışı olan alanlar kırmızı renkte görünür. Yaptığınız seçimi genişletmek istediğinizde herhangi bir fırça aracı ile söz konusu alanı ön plan rengi olarak siyah renk seçili iken boyayın. Seçilmiş bazı alanları seçimden çıkarmak için de yine söz konusu alanı herhangi bir fırça aracı ile ön plan rengi olarak beyaz renk seçili iken boyayın.



Quick Mask (*hızlı maske*) tuşuna çift tıklayarak açılan pencereden, maskenin rengini, şeffaflık derecesini ve maske uygulandığında seçili alanın mı maskelenen alanın mı boyanacağını seçebilirsiniz.

KANALLAR



Photoshop'ta, hızlı maskeler ya da kalıcı maskeler alfa kanalları denilen gri tonlu kanallar olarak saklanırlar. Kanallar resmin renk bilgilerini saklamak için de kullanılır. Alfa kanallarını **Channels** panelinden yönetebilirsiniz.

TEMEL RÖTUŞ İŞLEMLERİ

Resimlerinizde bir takım düzeltmeler ve rötuşlamalar yaparken en çok kullanacağınız araçlardan biri **Clone Stamp Tool** aracıdır. Bu araç sayesinde resmin bir yerindeki pikselleri resmin başka yerindeki piksellerle değiştirebilirsiniz.





Öncelikle **Clone Stamp Tool** aracı seçin ve klonlama yapacağınız yerdeki piksellerden örnekleme yapmak için ilgili bölgeye **ALT** tuşuna basılı tutarak tıklayın. Böylece örneklemeyi yapmış oldunuz. Şimdi resmin aldığınız örnekleme ile değiştireceğiniz kısmına gidin ve fareye basılı tutarak imleci sürükleyin.

Gördüğünüz gibi geçtiğiniz bölgelerdeki pikseller ilk örnekleme noktasından başlayarak değiştirilecektir. Eğer klonlama işlemini durdurup sonra tekrar devam ettiğinizde, işlemin ilk örnekleme noktasından başlayarak devam etmesini isterseniz üstteki **Options** panelinde **Aligned** kutucuğunu boşaltın.

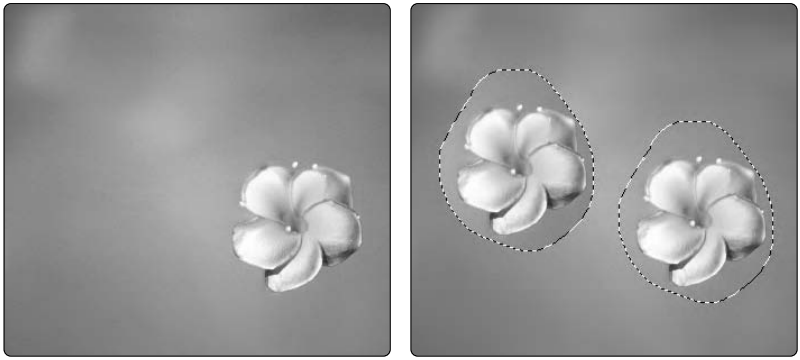
Bizim örneğimizde kangurunun gövdesine geldiğimizde mesela fareyi bırakıp sonra tekrar klonlamaya devam ettiğimizde kangurunun gövdesinde iki tane kulak olmaması için **Aligned** kutucuğunu işaretli olması gerekecektir.

Rötuşlama işlemlerinde kullanabileceğimiz bir başka araç ise **Spot Healing Brush Tool** aracıdır. Bu araçla da fotoğraflardaki hataları hızlı bir şekilde düzeltebilirsiniz. **Clone Stamp Tool** aracından farkı, **Clo-**

ne Stamp Tool aracı gibi sadece pikselleri klonlamakla kalmaz aynı zamanda da ışık, gölge, şeffaflık gibi değerleri de örneklemeye göre değiştirir. Ayrıca aracı kullanmak için bir örnekleme noktası seçmeye de gerek yoktur. Rötuşlanan alanın çevresindeki piksellerden otomatik olarak örnekleme yapar.

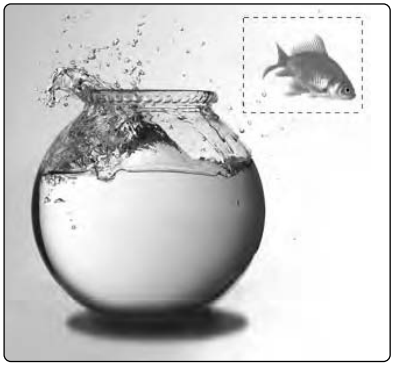


Çok işimize yarayacak bir diğer araçta **Healing Brush Tool** aracıdır. Farklı doku ve aydınlatma değerlerine sahip alanlarda daha doğal görünümlü düzeltmeler yapmanıza olanak tanır. Bu araçla da önce **ALT** tuşuna basılı tutarak bir örnekleme noktası belirliyoruz. Ve örneğimizdeki gibi ayak izlerini yok etmek için imlecimizi bu alanlarda sürüklüyoruz.



Bir diğer önemli rötuş aracı da **Patch Tool** aracıdır. Bu araç ile, önce değiştireceğiniz alanı seçin. Ardından seçimi hedef alana sürükleyip bırakın. Böylece seçtiğiniz alan hedef alana göre değişecektir. Biz örneğimizde çiçeğin sol tarafındaki boş alana yaklaşık olarak çiçeğin boyutlarında bir seçim alanı oluşturduk. Ardından seçimimizi sürükleyerek çiçeğimizin üzerine getirdik. Fareyi bıraktığımızda ikinci çiçeğimiz seçim alanımızda yerini almış oldu.

Kırpma İşlemi





Kırpma işlemini, görüntünün bir kısmını alıp diğer kısımlarını silmek şeklinde tanımlayabiliriz. Bu işlem için **Tools** panelinden **Crop Tool** aracını kullanacağız. Öncelikle kırpacağınız alanı imleci sürükleyerek seçin.

Ardından seçim çerçevesi içinde çift tıklayarak ya da **ENTER** tuşuna basarak kırpma işlemini gerçekleştirin. Seçili alanı iptal etmek istediğinizde ise **ESC** tuşuna basmanız yeterli olacaktır.

Perspektif Dönüştürme





Kırpma işlemini gerçekleştireceğiniz resimdeki perspektifi dönüştürebilirsiniz. Resimdeki pencerenin üst kenarları alt kenarlara göre birbirlerine daha yakın görünmektedir. İlk olarak resim üzerinde kırpma aracımızla bir dikdörtgen alan seçiyoruz. Mesela yan çerçevelerden birini seçebiliriz. Bu aşamada böyle bir seçim yapmamız Photoshop'un perspektif dönüştürmesi yapabilmesi için gereklidir.

Ardından kırpma aracı seçili iken üstteki **Options** panelinden **Perspective** kutucuğunu işaretleyin. Artık seçim çerçevesini kırpacağınız tüm alanı kapsayacak şekilde tutamaçlar yardımıyla genişletebilirsiniz. Son olarak **ENTER** tuşuna basın. Sonuç yukarıdaki ikinci resimdeki gibi olacaktır.

DÖNDÜRME İŞLEMLERİ

Tuvali döndürmek için **Image>Image Rotation** menüsünün alt menülerini kullanabilirsiniz. Bunlardan **180°** görüntüyü `180 derece`, **90° CW** saat yönünte `90 derece`, **CCW 90°** ise saat yönünün tersine `90 derece` döndürür.



Arbitrary... alt menüsü ise tuvali serbest olarak döndürmenize olanak tanır. Bu alt menüyü seçtiğinizde açılacak olan **Rotate Canvas** penceresinden `-359,99` ile **359,99** arasında bir açı değeri girerek tuvali istediğiniz açıda ve **CW** ya da **CCW** radyo düğmelerinden biri seçerek de tuvali saat yönünde ya da saat yönünün tersine döndürebilirsiniz.



Tuvali değil de katmanları döndürmek istediğinizde **Edit>Free Transform** ya da **Edit>Transform** menülerinden yaralanabilirsiniz. **Free Transform** menüsü sayesinde tutamakları fare ile sürükleyerek elle döndürme yapabilirsiniz.

Edit>Transform menüsünün alt menüleri yardımıyla da farklı türde (perspektif değiştirme, eğrilik ve çarpıtma) elle döndürme yapabileceğiniz gibi, tuval döndürme işleminde olduğu gibi `180` ve `90 derece` döndürme işlemlerini de kolaylıkla gerçekleştirebilirsiniz.

Kırmızı Göz Giderme

Kırmızı göz, genellikle az ışıklı ortamlarda çekilen fotoğraflarda flaşın gözdeki retina tabakasından yansıması sebebiyle oluşur. Kırmızı göz kusuruna sahip bir fotoğrafta bu kusuru gidermek için **Red Eye Tool** arcını  kullanabilirsiniz. Araç seçili iken imleci kırmızı göze tıklayın. Araç seçili iken üstteki **Options** panelinden aracın etki edeceği alanı büyütüp küçültebilirsiniz.

Renk Değiştirme

Renk değiştirme işlemi için **Color Replacement Tool**  aracını kullanabilirsiniz. Araç seçili iken **Options** panelindeki **Sampling:Continuous**  tuşu boyama işlemi sırasında sürekli örnekleme yapar. **Sampling:Once**  tuşunu aktif ederek aracı kullanırsak, sadece imleci ilk tıkladığınız yerdeki rengi içeren alanları boyar.

Bu sayede boyanmasını istemediğiniz farklı renklerdeki komşu alanlara taşma yapmadan renk değiştirme işlemini hızlı bir biçimde gerçekleştirebilirsiniz. **Sampling:Background Swatch**  tuşu ise sadece geçerli arka plan renginde olan alanları boyamanıza olanak tanır.

Lekeleme İşlemi

Smudge Tool  aracı, henüz kurumamış boyalı bir zemine bir cismin dokundurulmasıyla oluşan etkiye benzer görüntüler elde etmek için kullanılır. **Options** panelindeki **strength** kısmından leke efektinin kuvvetini, **simple all layers** kutucu ile de lekeleme işleminin tüm katmanlar kullanılarak yapılmasını sağlayabilir, **finger painting** kutucuğu ile de gerektiğinde ön plan rengi ile lekeleme işlemi yapabilirsiniz. Bu işlemi, imleci sürüklerken **ALT** tuşuna basılı tutarak da yapabilirsiniz.

Bulanıklaştırma İşlemi

Resimlerinizde bulanık bir görüntü elde etmek ya da bazı keskin hatları yumuşatmak amacıyla **Blur Tool**  aracını kullanabilirsiniz. **Options**

panelinden fırça kalınlığını ayarlayabilir, bulanıklaştırma işlemini tüm katmanları kullanarak yapıp yapmayacağınızı (*simple all layers*), bulanıklık efektinin kuvvetini (*strength*) ayarlayabilirsiniz. Araç ile bulanıklaştırmak istediğiniz bölgeyi ne kadar çok boyarsanız o kadar fazla bulanıklık etkisi elde edersiniz.

Keskinleştirme İşlemi

Sharpen Tool aracı, kontrastı artırarak daha keskin görüntüler elde etmenize olanak tanıyan bir araçtır. Aracı keskinleştirilecek bölgeye daha fazla uygulayarak daha keskin bir görüntü elde edebilirsiniz. Bu araç için ayarlamaları da yine **Options** panelinden yapabilirsiniz.

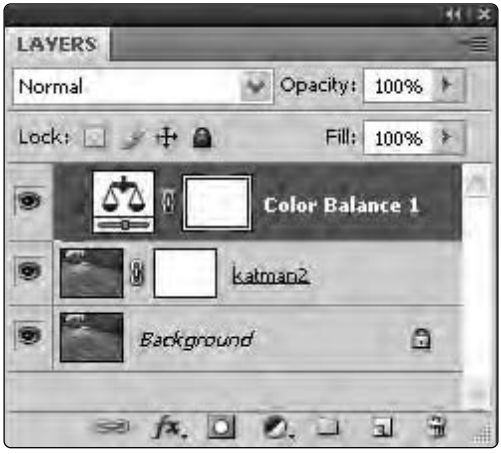
Soldurma, Yakma ve Doygunluk İşlemleri

Çalışmanızda daha açık ve soluk bir görüntü elde etmeniz gerektiğinde **Dodge Tool** aracını, daha fazla koyulaştırmak gerektiğinde ise **Burn Tool** aracını, kontrastı artırarak ya da azaltarak renk doygunluğunu değiştirmek içinse **Sponge Tool** aracını kullanabilirsiniz.

Katmanlar

Üst üste sıralanmış tabakalar olarak tarif edebileceğimiz **katmanlar**, çalışmanızı daha iyi organize edebilmenizi sağlarlar. Bir çalışmaya istediğiniz kadar katman ilave edebilirsiniz. Birçok görüntüyü birleştirmek, resme metin eklemek, resmin pikselleriyle oynamadan renk ve ton ayarlamaları yapmak, resme gölge ve ışık uygulamak gibi bir çok işlemde katmanlar size kolaylıklar sağlayacaktır.

Katmanları **Layers** panelinden yönetebilirsiniz. **Layers** paneliniz açık değilse **Window>Layers** menüsü yardımıyla açabilirsiniz. Katmanlar panelde, katmanı temsil eden küçük bir resimle ve katmana ait bir isimle gösterilirler. **Layers** paneli yardımıyla yeni katmanlar ekleyebilir, mevcut katmanlardan istediklerinizi silebilir, katmanları çoğaltabilir, katmanları gizleyebilir, isimlerini ve sıralarını değiştirebilirsiniz.



Katmanların sağ kısımlarında gördüğünüz kilit simgesi, o katmanın müdahale edilemez olduğu anlamına gelir. Dolayısıyla kilit işaretini kaldırmadan bu katman üzerinde işlem yapamazsınız. Kilit işaretini kaldırmak için panelin sol üst kısmındaki dört adet kilit seçeneğinden aktif olana bir kez tıklamanız yeterli olacaktır. Evet gördüğünüz gibi panelde dört adet kilit mevcuttur.

Bunlardan **Lock transparen pixels** katmanın sadece saydam kısımlarının düzenlenmesini engeller. Böylece resmin saydamlığı korunmuş olur. **Lock image pixels** seçeneği ise katmandaki piksellerin boyama araçları kullanılarak değiştirilmesini engellemek içindir. **Lock position** katmandaki piksellerin taşınmasını engeller. **Lock all** ise katmanın tüm özelliklerini kilitlemek için kullanılır. Bir katmanı gizlemek için katmanın sol başında bulunan **göz** simgesine tıklamanız yeterli olacaktır.

Arkaplan ve Katman Dönüştürme

Gerektiğinde bir katmanı arka plana, bir arka planı da katmana dönüştürebilirsiniz. Bir katmanın adını `Background` olarak değiştirmekle o katmanı arka plana dönüştürmüş olmazsınız. Bir katmanı arka plana dönüştürme işlemini **Layer>New>Background From Layer** menüsü yardımı ile yapabilirsiniz. Bir arka planı katmana dönüştürmek gerektiğinde ise **Layer>New>Layer From Background**... menüsünü kullanabilir ya da katmana çift tıklayabilirsiniz.

Yeni Grup-Katman Oluşturma ve Silme

Yeni bir katman oluşturmak için **Layer>New>Layer** menüsünü kullanabilir ya da Layer panelindeki yeni layer oluşturma tuşuna basabilirsiniz. Yeni bir grup oluşturmak için de **Layer>New>Group** menüsünü ya da **Layer** panelindeki yeni grup oluşturma tuşunu kullanabilirsiniz. Oluşturdunuz katmanların sırasını değiştirmek ya da bir gruba dahil etmek için katmanı fare ile sürükleyerek yeni yerine bırakmanız yeterli olacaktır.

Mevcut bir katmanı silmek için katman seçiliyken **Layer** panelindeki silme tuşuna basmanız yeterli olacaktır. Bir grup seçili iken aynı tuş yardımıyla bir grubu da silebilirsiniz.

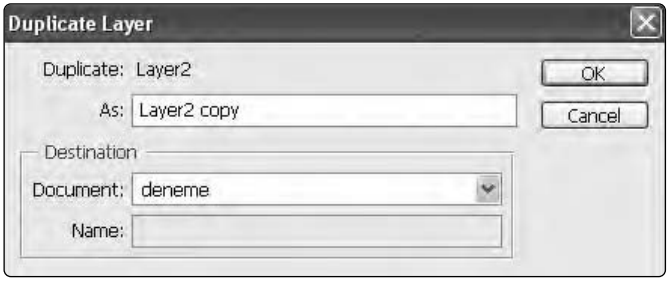


Bir grubu silerken açılan iletişim penceresi yardımıyla hem grubu hem de gruptaki katmanları silebileceğiniz (*Group and Contents*) gibi dilerseniz sadece grubu (*Group Only*) da silebilirsiniz.

Layer panelindeki **Opacity** sürgüsünü kullanarak katmanların opacity değerini artırabilir ya da azaltabilirsiniz. Bu değer `%1` ile `%100` arasında olur. Değerin `%1` olması demek katmanın saydam olması demektir. Değeri `%100` yaptığınızda katman tamamen opak olacak, yani hiç saydamlık özelliği kalmayacak demektir.

Katmanları Çoğaltmak

Bir katmanı çoğaltmak için **Duplicate Layer** komutundan yararlanacağız. Bu komuta, katman üzerinde fare ile sağ tıklayarak ya da **Layer** panelinin sağ üst köşesindeki panel menüsününden ulaşabilirsiniz. Açılan menüden **Duplicate Layer** komutuna tıkladığınızda **Duplicate Layer** penceresi açılacaktır.



Açılan pencereden katmanınıza bir isim verip **OK** tuşu ile işlemi sonlandırdığınızda katmanınızın bir kopyasını elde etmiş olacaksınız. Bu işlemi çoğaltacağınız katmanı yeni layer oluşturma tuşuna sürükleyerek de yapabilirsiniz.

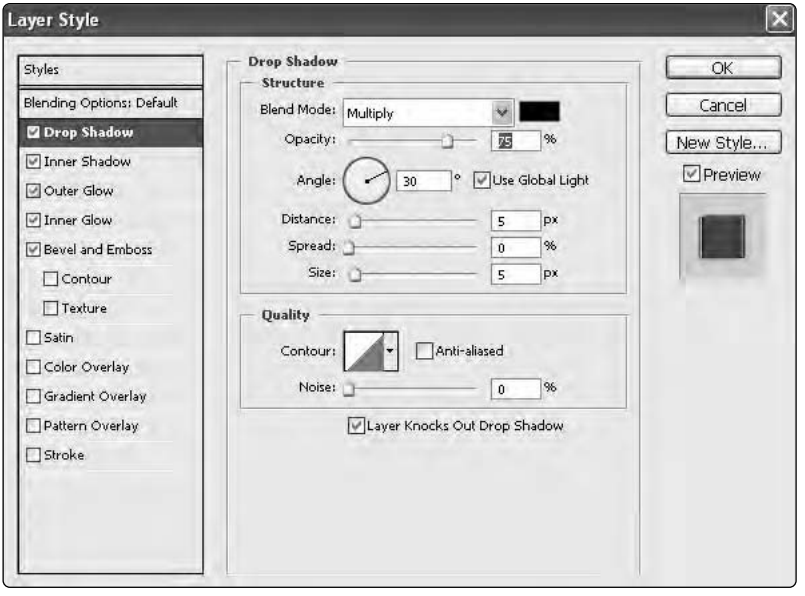
KATMANLARI BAĞLAMA

Bir çalışmanın parçaları olan katmanları bazı durumlarda birbirlerine bağlamak gerekebilir. Böylece bağlı katmanlar siz ayırmadıkça birlikte taşınabilir ve birbirleriyle olan ilişkili hizalama konumlarını da korumuş olurlar. Bu işlem için bağlanacak katmanları **SHIFT** tuşu ile seçtikten sonra **Layers** panelindeki bağ tuşuna tıklamanız yeterlidir. Böyle bağladığınız katmanların yanlarında da bir zincir işareti belirecektir. Bunun anlamı artık bu katmanlara birlikte işlemler uygulanabileceğidir.

KATMAN STİLLERİ

Photoshop ile katmanlarınıza farklı görünümler vermenizi sağlayan gölge, ışıma, kabartma... vb bir çok efekt ekleyebilirsiniz. Bu efektleri katman ya da katman gruplarına katman stilleri yardımı ile uygulayabilirsiniz. **Layers** panelindeki **layer style** tuşu ile açılan menüden istediğiniz stili seçebilirsiniz. Bu işlemi layer'a çift tıklayarak da yapabilirsiniz.

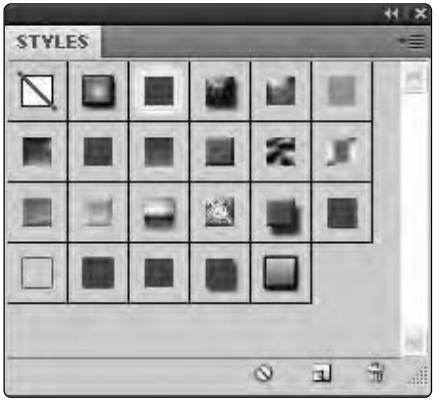
YENİ BİR KATMAN STİLİ OLUŞTURMAK



Yeni bir katman stili oluşturmak için örneğin bir metin oluşturun ve oluşturduğunuz metnin katmanını çift tıklayarak **Layer Style** penceresini açın.



Metninize istediğiniz stil özellikleri verdikten sonra penceredeki **New Style** tuşu ile **New Style** penceresini açın. Yeni stilinize bir isim verin ve pencereyi **OK** tuşu ile kapatın. Şimdi tekrar bir metin oluşturun ve metninize herhangi bir stil vermeyin.



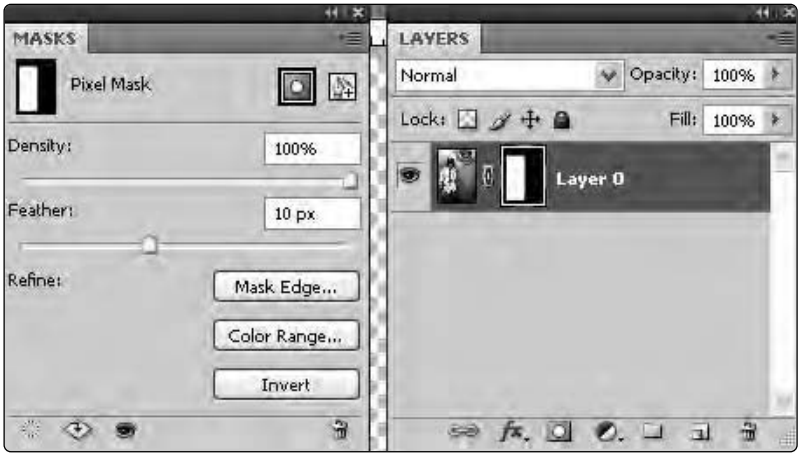
Şimdi katman seçiliyken **Styles** panelinden hazırlamış olduğunuz stili tıklayın. Metniniz, tıkladığınız stilin özelliklerini kazanacaktır.

Katman Maskeleri

Katmanlara da bazı katman bölümlerini gizlemek ve diğer katmanların görünür olmasına imkan tanımak amacıyla maske ekleyebilirsiniz. Bu şekil bir birleştirme çok sayıda resmi tek bir çalışmada birleştirmek ve birtakım düzenlemeleri yapabilmek için önemli ve kullanışlı bir tekniktir.

Maskelerde siyah ile boyama yapmak demek o alanı gizlemek, beyaz ile boyamak ise o alanın görünür olmasını sağlamak demektir. Gri tonları ise farklı saydamlık seviyelerini ifade etmektedir.

Maske Oluşturma



Bu işlemleri yönetmek amacıyla **Masks** panelinden yaralanabilirsiniz. Yeni bir maske oluşturmak için **Layer** panelindeki **katman maskesi ekleme** tuşuna ya da **Masks** panelindeki **piksel maskesi** tuşuna tıklayabilirsiniz. Oluşturduğunuz bu maske tüm katmanı gösterecektir. Maskenin tüm katmanı gizlemesi için, maskeyi oluştururken **ALT** tuşuna da basılı tutmalısınız. Bu takdirde katmana bağlanan maske siyah renkte olacaktır.

Eğer maskenizin sadece seçtiğiniz alanı göstermesini istiyorsanız, bir seçim aracı ile seçiminizi yaptıktan sonra **Layer** panelindeki **katman maskesi ekleme** tuşuna ya da **Masks** panelindeki **piksel maskesi** tuşuna tıklayın. Seçtiğiniz alanın değil de seçim dışı kalan alanların gösterilmesi içinse maskeyi oluştururken **ALT** tuşuna da basılı tutun.

Bir maskeyi başka bir katmana taşımak için maskeyi bir katmana sürüklemeniz yeterlidir. Maskeyi taşımak yerine kopyalamak istiyorsanız, maskeyi sürüklerken **ALT** tuşuna da basılı tutun.

Maske Düzenleme

Bir maskeyi düzenlemek için **Layer** panelinde maskenin üstüne ya da **Masks** panelinde **piksel maskesi** tuşuna tıklayın. Maskenin çevresinde bir kenarlık belirecektir. Artık maskeniz düzenlemeye hazır demektir. Şimdi bir fırça aracı seçin ve artık maskelenmesini istemediğiniz ve maskeden çıkartmak istediğiniz alanları beyaz renk ile boyayın. Eğer tam tersini yapmak, yani görünen bazı kısımları maskeye eklemek ve dolayısıyla gizlemek isterseniz, söz konusu alanları bir fırça aracı yardımıyla siyah renge boyayın.

Çalışmanız esnasında maskeyi değil de katmanda düzenlemeler yapmanız gerektiğinde katmanı seçili hale getirtin. Bunun için **Layer** panelinde katmanı temsil eden küçük resmin üstüne fare ile bir kez tıklamanız yeteli olacaktır. Resmin çevresinde oluşan ince kenarlık katmanın aktif olduğunu gösterir.

Fırçalar

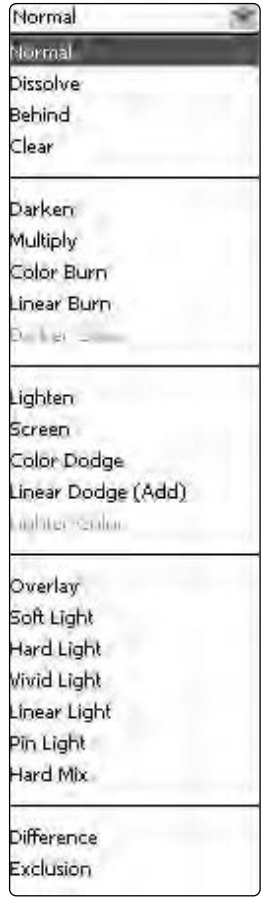
Photoshop fırçaları ile boyama, renk düzenleme, doldurma, bulanıklaştırma, lekeleme, silme... vb bir çok işlemi kolaylıkla yapabilirsiniz.

Brush Tool aracını ya da **Pencil Tool** aracını seçip, araç seçili iken üstteki **Options** panelinden gerekli ayarlamaları yaptıktan sonra imleci tuvale tıklatıp sürükleyerek boyama işlemini gerçekleştirebilirsiniz. Düz bir çizgi çekmek için tuvalde bir noktaya tıkladıktan sonra **SHIFT** tuşuna basılı tutarak imleci bir başka noktaya tıklatın. Böylece bu iki nokta arasında düz bir çizgi çizmiş olacaksınız.

Boyama Araç Seçenekleri



Bir boyama aracını seçtiğinizde üstteki **Options** panelinde beliren seçenekler yardımı ile seçtiğiniz araçla ilgili bir takım ayarlamalar yapabilirsiniz. Şimdi bu ayarlamaların neler olduklarına bir bakalım.



Mode: Renk karıştırma yöntemini seçmek içindir. Açılır kutucuğu tıklayarak resimde gördüğünüz menüden istediğiniz karıştırma modunu seçebilirsiniz.

Opacity: Boyama yaparken uygulanacak rengin saydamlık değerini ayarlamak içindir. Opacity değeri yükseldikçe saydamlık azalacaktır.

Flow: Rengin uygulama oranını belirlemek içindir. Uygulanan renk miktarı en fazla opacity değeri kadar olmak kaydıyla **flow oranı** baz alınarak oluşturulur.

Pistole tuşu tıklı iken fareyi basılı tuttuğunuz sürece boyama işlemi devam eder. Tuş aktif değilse boya miktarı fareye her tıklanışta artacaktır.

Auto Erase: Sadece **Pencil Tool** aracı için söz konusudur. Ön plan rengindeki alanları seçili arka plan rengi ile boyar.

Desenlerle Boyama

Desenlerle boyama yapmak için bir desen seçerek **Pattern Stamp Tool** aracını kullanabilirsiniz. **Pattern Stamp Tool** aracını seçip Üstteki **Options** panelinden **Mode**, **Opacity**, **Flow** ayarlarını dilediğiniz gibi yapabilirsiniz. Ardından desen açılır kutucuğundan bir desen seçin. Hemen yandaki **Aligned** kutucuğunu tıklarsanız fareyi bırakıp sonra tekrar basarak boyamaya devam ettiğinizde seçeceğiniz desenin devamlılığını muhafaza etmiş olursunuz. Fareyi her bırakışınızda boyamayı yeniden başlatmak için bu kutucuktaki onayı kaldırmalısınız.

Silgiler

Eraser Tool

Eğer bir arka planda çalışıyorsanız **Eraser Tool** aracı tuvali arka plan rengi ile boyayacaktır. Kilidi kaldırarak arka planı bir katmana dönüştürürseniz araç, rengi siler ve geriye saydam pikseler kalır.

Background Eraser Tool

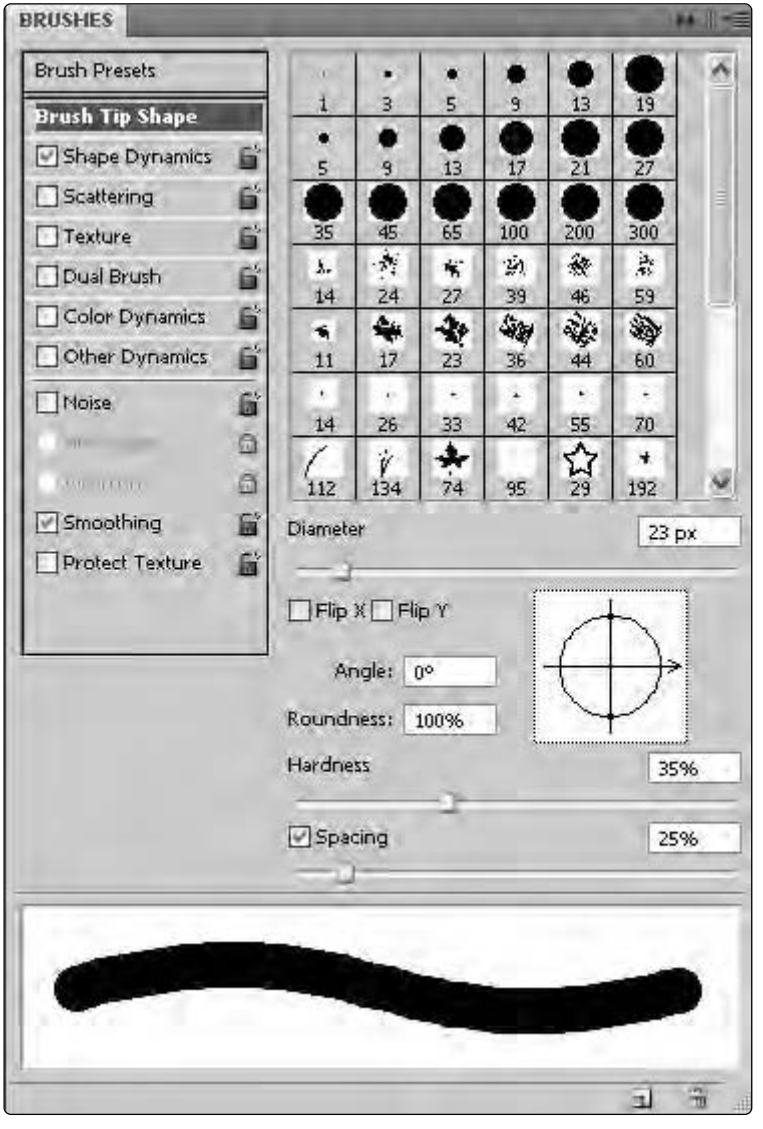
Bir katmandaki pikselleri ön plandaki nesneyi koruyarak silmek için **Background Eraser Tool** aracını kullanmalısınız. **Options** paneli yardımıyla farklı tolerans ve örnekleme değerleri ile farklı sonuçlar elde edebilirsiniz.

Magic Eraser Tool

Magic Eraser Tool aracı bütün benzer pikselleri siler. Burada ne kadar benzer pikselin silineceğini belirleyecek olan renk aralığını **Options** panelinden **Tolerance** değeri ile belirleyebilirsiniz. Düşük bir tolerance değeri belirlerseniz, araç seçili iken imleci tıkladığınız yerdeki pikselin rengine en yakın birkaç rengi silecektir.

Tolerance değeri yükseldikçe daha çok benzer renk silinecek, yani renk aralığı genişleyecektir. Pikselleri tamamen silmek için **Opacity** değerini `%100` olarak ayarlamalısınız. Düşük bir **Opacity** değeri seçerseniz pikseller kısmi olarak silinecektir. Sadece imleci tıklattığını piksele bitişik olan pikselleri silmek için **Contiguous** kutucuğunun seçili olması gerekir.

Brushes Paneli

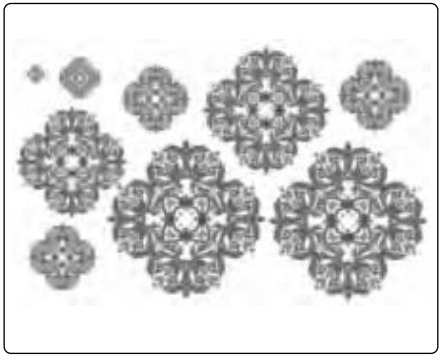


Brushes panelini açmak için **Window>Brushes** menüsünü kullanın. Bu panel yardımıyla fırça ayarlarında ve fırça uçlarında değişiklikler yapabilirsiniz. Ayrıca dilerseniz kendi fırçalarınızı da oluşturabilirsiniz.

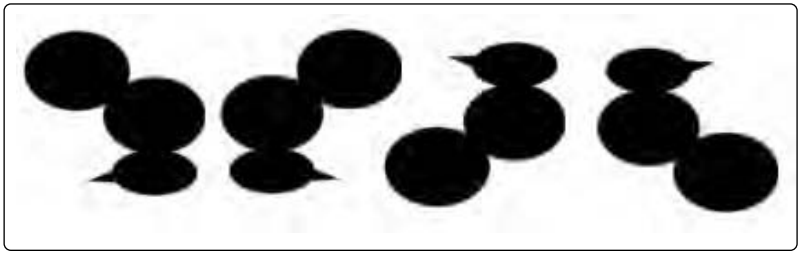
Örneğin bir görüntüden fırça oluşturmak için önce görüntünün fırça yapacağımız kısmını bir seçim aracı ile seçin. Ardından **Edit>Define Brush Preset**... menüsünü tıklayın. Açılan pencereden fırçanıza bir isim verin. Artık fırçanızı kullanabilirsiniz. Dilerseniz aynı yöntemle bir dosyayı da fırçaya dönüştürebilirsiniz.

Mesela çalışmalarınızda sürekli kullandığınız bir logoyu açıp **Edit>Define Brush Preset**... menüsü ile açılan pencereden bir isim girerek logonuzu bir fırçaya dönüştürebilirsiniz. Şimdi artık **Brushes** paneli yardımıyla yapabileceğimiz fırça ayarlarına göz atabiliriz.

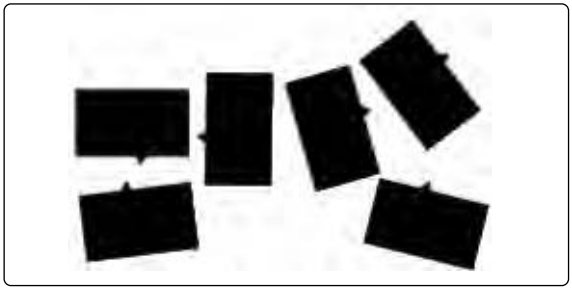
Brush Tip Shape



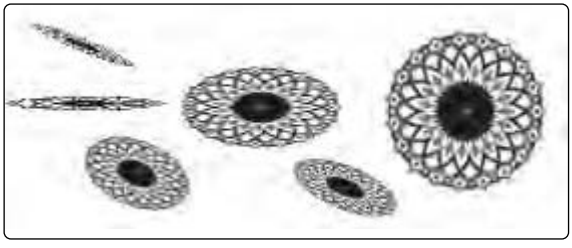
Fırçaların çaplarını ayarlamak için **Diameter** özelliğini kullanabilirsiniz. Kutucuğa piksel cinsinden bir değer girerek ya da alttaki sürgüyü hareket ettirerek seçtiğiniz fırçanın çapını ayarlayabilirsiniz. Fırçayı orijinal boyutlarına döndürmek içinse **Use Sample Size** tuşuna tıklayın.



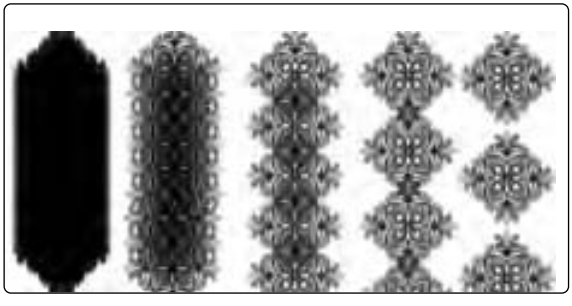
Fırçanın yönünü x ekseninde değiştirmek için **FlipX**, y ekseninde değiştirmek içinse **FlipY** kutucuklarını seçili hale getirmelisiniz. Yukarıdaki resimde gördüğünüz ilk şekil fırçanın orijinalini göstermektedir. İkinci şekil **FlipX** kutucuğu seçili iken, üçüncü şekil **FlipY** kutucuğu seçili iken son şekil ise hem **FlipX** hem de **FlipY** kutucukları seçili iken elde edilen sonucu göstermektedir.



Angle: Fırçanın yatay eksendeki açısını ayarlamak için kullanılır. Derece cinsinden bir değer girerek ya da yandaki ön izleme kutusundan ekseni fare ile tutup sürükleyerek açıyı istediğiniz gibi ayarlayabilirsiniz.



Roundness: Fırçanızın eksenleri arasındaki oranı ayarlamak içindir. Ön izleme kutusundan ya da kutucuğa yüzde cinsinden bir değer girerek ayarlayabilirsiniz. Kutucuğa gireceğiniz `%100` değeri daireyi, `%0` değeri ise doğruyu ifade edecektir. Bu iki değer arasındaki diğer değerler daireden doğruya doğru eliptik fırçalar elde etmenizi sağlayacaktır.



Spacing: Fırça darbeleri arasındaki mesafeyi ayarlamak için kullanılır. Kutucuğa yüzde cinsinden bir sayı girerek ya da sürgüyü kaydırarak is-

tediğiniz aralığı ayarlayabilirsiniz. Resimde soldan sağa doğru incelersek, ilk fırça `%1` spacing değerine sahiptir. İkinci fırça `%25`, üçüncüsü `%50`, dördüncüsü `%75` ve son fırça da `%100` spacing değerine sahiptir. Spacing değeri olarak azami `%1000` değerini girebilirsiniz.

Blend Mode

Boyama ya da düzenleme araçlarının çalışmanızdaki piksellere etkilerini karıştırma modları belirler. **Options** panelindeki açılır **Mode** kutusu yardımıyla fırçanızla beraber kullanacağınız karıştırma modunu seçebilirsiniz.

Normal: Piksel düzenlemede ve boyamada varsayılan ayarları barındıran moddur.

Dissolve: Piksellerin saydamlığına ve boyamada kullanılan renklere göre rasgele karışım yapar.

Behind: Üzerinde çalıştığınız katmanın sadece saydam olan kısımlarını boyar.

Clear: Pikselleri saydam hale getirecek şekilde temizler.

Darken: Bu modda çalışırken, seçilen karışım renginden daha koyu renkteki pikselleri boyamaya kalktığınızda her hangi bir değişme olmayacaktır. Bir başka ifade ile bu karışım modu karışım renginden daha açık renkteki pikselleri boyamak içindir.

Lighten: Karışım renginden daha açık renkler değiştirilmez. Daha koyu renkler ise değiştirilir.

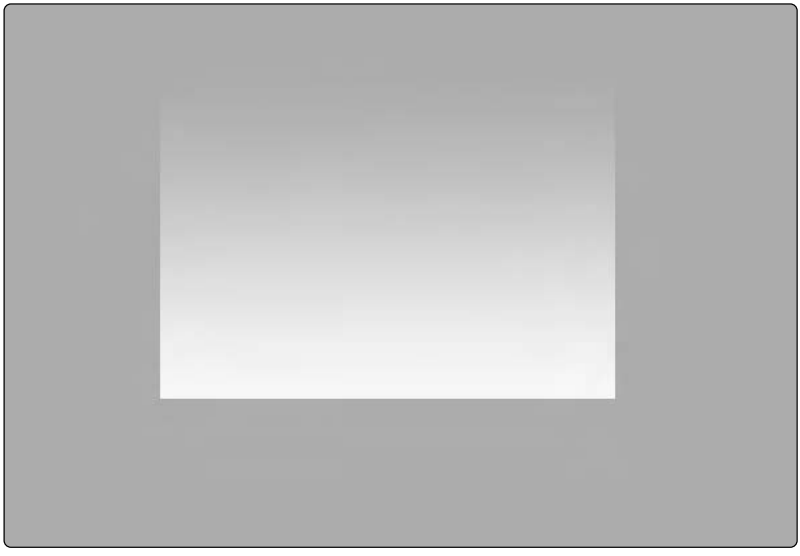
Soft Light: Karışım renginin koyu ya da açık olmasına bağlı olarak renkleri koyulaştırmak ya da açmak içindir.

Hard Mix: Pikselleri karışım renginin birincil rengine dönüştürecek şekilde boyar.

Hue: Karışım rengi ile piksellerin aydınlık ve doygunluk değerlerini baz alarak bir renk üretir.

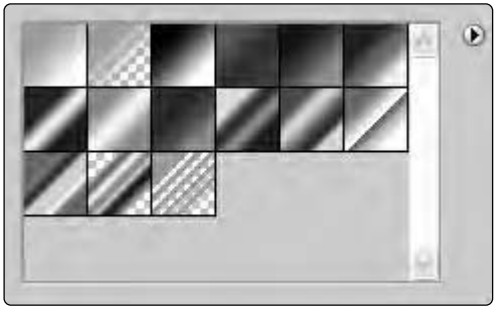
Saturation: Karışım renginin doygunluk değerleri ile piksellerin renk tonlarını baz alarak bir renk üretir.

GRADIENT İŞLEMLERİ



Gradient işlemi ile birçok rengi kademeli olarak karıştırma yapmış oluruz. Gradient işlemi uygulamanın yapılacağı alana imleci sürükleyerek yapılır. Gradient, fareyle ilk tıklanılan nokta ile farenin bırakıldığı nokta arasına uygulanır. Bu iki nokta arasındaki tüm bölgelere gradient uygulanmasını istemiyorsanız, öncelikle uygulamanın yapılacağı alanı seçin. **Gradient Tool** aracını seçip uygulamayı gerçekleştirin.

Uygulayacağınız gradient'in ayarlarını araç seçili iken üstteki **Options** panelinden yapabilirsiniz.





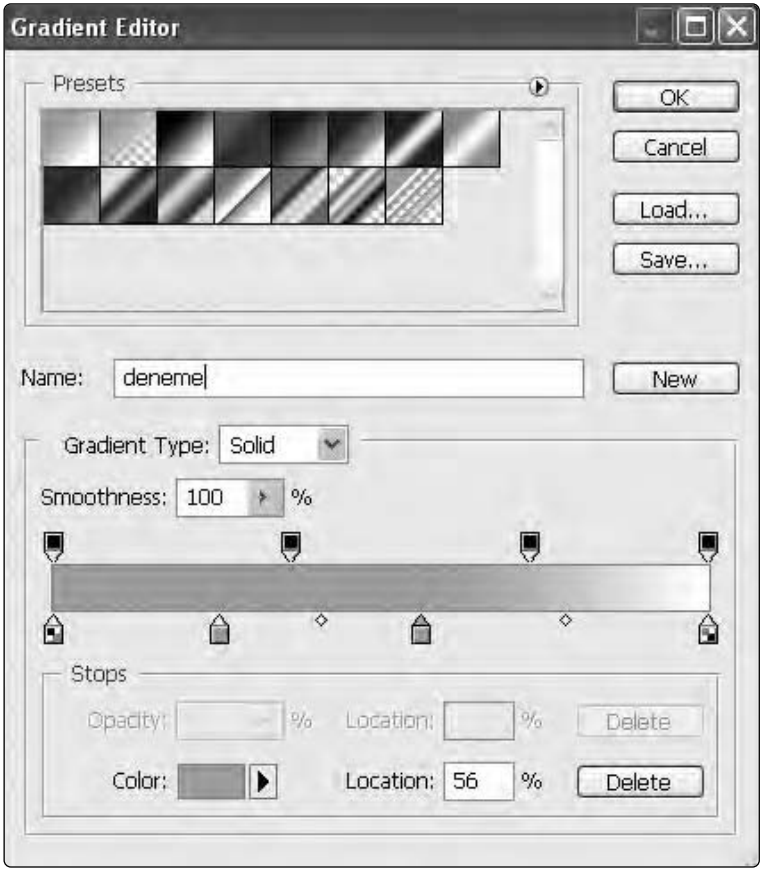
Açılır gradient penceresinin sağındaki ok tuşu yardımıyla bir gradient seçin. Açılan pencerenin sağ üst kısmında bulunan küçük ok işaretine tıkladığınızda açılan menüde Photoshop'ta bulunan hazır gradientlerin listesi ve bazı ayarlamalar yer almaktadır. Buradaki menülere kısaca bir göz atalım.

Text Only ile gradientleri sadece isimleriyle listeleyebilirsiniz. **Small Thumbnail** seçeneği gradientleri birer küçük simge ile listeler. **Large Thumbnail** seçeneği de gradientleri birer büyük simge ile listeler. **Small List** küçük, **Large List** ise büyük ve her ikisi de resimli olarak listeleme yapar.

Reset Gradients menü öğesini kullanarak kendi yaptığınız tüm gradientlerden kurtulabilir ve Photoshop'un varsayılan gradientleri ile baş başa kalabilirsiniz! **Load Gradients** ile de yeni bir gradient ekleyebilirsiniz. **Save Gradients** seçeneği ise hazırladığınız gradientleri *.grd* uzantısı ile kaydedebilmeniz içindir. **Replace Gradients** yardımıyla da *.grd* uzantılı bir gradient dosyasını Photoshop'a yükleyebilirsiniz.



Option panelinden **Gradient** açılır penceresinde seçili durumda bulunan gradientin üstüne bir kez tıklayarak **Gradient Editör** penceresini açabilirsiniz.



Açılan pencerede **Name** kutusuna oluşturacağınız gradientin adını girin. **Gradient Type** kutucuğundaki seçeneklerden **Solid** ya da **Noise** seçeneklerinden birini seçin. Normal düzgün bir gradient yapmak için **Solid** seçeneğini kullanabilirsiniz. Pürüzlü yüzeylere sahip gradientleri oluştururken ise **Noise** seçeneğini kullanmalısınız.

Alttaki renk paletinden gradientinizin rengini ve rengin başlangıç ve bitişini tutamaçlar yardımıyla ayarlayabilirsiniz. Paletin alt ve üst kenarlarında iken imleç el işaretine döndüğünde tıklayarak yeni bir tutamaç ekleyebilirsiniz. Bu tutamaçlar gerektiğinde daha detaylı gradientler oluşturmanızı kolaylaştıracaktır.

Tutamaçları silmeniz gerektiğinde bunu, penceredeki **Delete** tuşlarını kullanarak yapabilirsiniz. Paletin altı kısmındaki tutamaçlar ile renkleri, üstteki tutamaçlar ile de saydamlık ayarlarını yapabilirsiniz. Ayarlamalarınızı tamamladığınızda **New** tuşuna basarak yeni gradientinizi oluşturmuş olacaksınız.

Ardından gradient tipini belirleyin. Kullanabileceğiniz tiplerden **Linear Gradient** başlangıçtan bitişe kadar düz bir doğrultuda karıştırma uygular. **Radial Gradient** başlangıçtan bitişe kadar dairesel biçimde karıştırma uygular. **Angle Gradient** başlangıçtan bitişe kadar saat yönünün tersinde açılı olarak karıştırma uygular. **Reflected Gradient** başlangıçtan bitişe kadar yansımalı olarak karıştırma uygular. **Diamond Gradient** başlangıçtan bitişe kadar elmas biçimli karıştırma uygular.

Bir sonraki aşama olarak karışım modunu ve **Opacity** değerini belirleyin. Eğer renk geçişlerinin sıralamasını tersine çevirmek isterseniz **Reverse** kutucuğunu seçili hale getirin. Daha düzgün bir renk geçişi için **Dither** kutucuğunun seçili olmasına dikkat edin. Renk geçişi için imleci çalışmanız üzerinde istediğiniz bir açıyla sürükleyebilirsiniz. Sadece `45°` ve katlarıyla uygulama yapabilmek için **`SHIFT`** tuşuna basılı tutun.

Boya Kovası

Paint Bucket Tool (*Boya Kovası*) aracı benzer ve bitişik pikselleri renk ile ya da desenle doldurmak için kullanılır.



Araç seçili iken üstteki **Options** panelinden çalışmanızı ya da çalışmanızın seçtiğiniz kısmını belirlediğiniz ön plan rengi ile doldurmak için, seçim kutucuğundan **Foreground** seçeneğini seçin. Eğer seçiminizi desenle dolduracaksanız önce **Pattern** seçeneğini ardından da yandaki açılır kutucuktan uygulayacağınız deseni seçin. Karışım modunu ve saydamlık değerini de belirledikten sonra artık aracı kullanmaya başlayabilirsiniz.

DESENLER

Photoshop size pek çok hazır desen sunmaktadır. Ayrıca siz de kendi desenlerinizi oluşturabilirsiniz. Bunun için **Rectangular Marquee Tool** aracı ile desen olarak kullanacağız bölgeyi seçin. Ardından **Edit>Define Pattern** menüsünü tıklayın. Açılan **Define Name** penceresinden deseninize bir ad verin.



Gradient Tool aracında olduğu gibi **Paint Bucket Tool** aracında da **Options** panelinde desen seçme açılır kutusunda yer alan resimde gördüğünüz menü yardımıyla bir deseni kaydedebilir ya da bir pattern dosyasını Photoshop'a *.pat* uzantısı ile yükleyebilirsiniz.

FİLTRELER

Filtreler çalışmalarınıza, özel bazı efektler ekleme (karakalem, tablo... vd) ve görüntü düzeltme (ışık efektleri, deformasyon... vd) işlemlerini hızlıca ve kolaylıkla yapabilmenizi sağlarlar. Programdaki tüm filtrelere **Filter** menüsü altından ulaşabilirsiniz. Bir filtreyi bir katmana uygulayabilmeniz için o katmanın seçili olması gerekir. Eğer filtre katmanın tamamına değil de bir kısmına uygulanacaksa öncelikle o alanı seçim araçları ile seçmelisiniz.

FİLTRE GALERİSİ



Filtre menüsündeki bir çok filtreyi, filtre galerisi yardımıyla ön izlemeli olarak kullanabilirsiniz. Filtre galerisini **Filter>Filter Gallery** menüsü yardımıyla açabilirsiniz. Bu noktada yolumuza **Filter** menüsündeki filtre gruplarını tanıyarak devam edelim.

ARTISTIC FİLTRE GRUBU

Bu gruptaki filtreler genellikle sanatsal etkiler elde etme amacıyla kullanılabilir.

Colored Pencil: Bu filtre ile renkli boya kalemleri ile çizilmiş ve çapraz olarak taranmış resim etkisi elde edebilirsiniz.

Cutout: Kesilmiş renkli kağıtlardan oluşturulmuş etkisi verir.

Dry Brush: Resimdeki sık kullanılan renklerle kuru fırça tekniği ile boyanmış etkisi oluşturur.

Fresca: Çalışmanızı sık bir biçimde uygulanmış fırça darbeleri ile boyar.

Neon Glow: Nesnelere ışıma etkisi vermek içindir. Burada ışıma etkisi için istediğiniz bir rengi seçebilirsiniz.

Paint Daubs: Seçeceğiniz fırça tipine ve fırça ebatlarına göre boyama yapar.

Palette Knife: Çalışmanızdaki detayları gideren bir biçimde boyama yapar.

Plastic Wrap: Çalışmanıza parlak plastik bir kaplama deseni uygular.

Poster Edges: Çalışmalarınızdaki renk sayısını azaltarak poster görünümü kazandırmak için kullanabilirsiniz.

Rough Pastels: Çalışmanızda pastel boya ile yapılmış etkisi elde etmek için kullanabilirsiniz.

Sponge: Bir süngerle boyanmış hissi veren belirgin dokulara sahip bir görüntü elde etmek için kullanabilirsiniz.

Watercolor: Çalışmanızda sulu boya ile yapılmış etkisi elde etmek için kullanabilirsiniz.

Blur Filtre Grubu

Bu gruptaki filtreler, resimdeki komşu piksellerin renklerinin ortalaması olan yumuşak geçişler uygulayıp resmi bulanıklaştırarak çeşitli etkiler elde etmenize olanak tanırlar.

Average: Resimdeki renklerin ortalamasını bulur ve resmi bununla doldurur.

Blur & Blur More: Blur ve **Blur More** filtreleri renk geçişlerini yumuşatır. **Blur More** filtresi **Blur** filtresine nazaran daha güçlü bir etkiye sahiptir.

Box Blur: Filtreyi seçtiğinizde açılacak olan penceredeki ayar sürgüsü yardımıyla istediğiniz bulanıklıkta bir görünüm elde edebilirsiniz.

Gaussian Blur: Filtreyi seçtiğinizde açılacak olan penceredeki ayar sürgüsü yardımıyla istediğiniz bulanıklık derecesine sahip puslu görüntüler elde edebilirsiniz.

Lens Blur: Çalışmanızdaki bazı nesnelerin odak olması diğer kısımların ise geri planda kalarak bulanıklaştırılması için kullanabilirsiniz.

Motion Blur: Resminizi, filtreyi seçtiğinizde açılacak olan pencere yar-

dımıyla belirleyeceğiniz yönde ve belirleyeceğiniz yoğunlukta bulanıklaştırmak içindir.

Radial Blur: Radial bir bulanıklık görüntüsü elde etmek içindir. Buradaki etki sizin filtreyi seçtiğinizde açılacak olan pencere vasıtası ile belirleyeceğiniz döndürme derecesine ve döndürme merkezine bağlı olarak oluşacaktır.

Shape Blur: Öncelikle listeden hazır bir şekil seçin ve boyutunu ayarlayın. Şeklin boyutunu **Radius** sürgüsünü kullanarak ayarlayın. Boyut fazla olursa bulanıklık da o denli fazla olacaktır. Ayrıca hazır şekiller haricinde dilerseniz farklı şekiller de yükleyebilirsiniz.

Smart Blur: Daha ayrıntılı bir şekilde bulanıklaştırma yapmanıza olanak tanır.

Surface Blur: Kenar piksellerini koruyup bulanıklaştırma işlemini gerçekleştirir.

Brush Strokes Filtre Grubu

Bu filtre grubu da, ressam elinden çıkmışcasına birtakım sanatsal fırça darbeleri elde etmenize olanak tanır.

Accented Edges: Görüntünün kenarlarını daha vurgulu hale getirir. Parlaklığı arttırıp azaltarak istediğiniz etkiyi elde edebilirsiniz.

Angled Strokes: Görüntünün açık ve koyu alanlarını karşıt yönlerdeki fırça darbeleri ile boyar.

Cross Hatch: Görüntü üzerinde ayrıntıları yok etmeden, renkli kenarları kalemle tarama yapılmış etkisi verecek şekilde düzenler.

Dark Strokes: Koyu ve açık alanları kısa ve uzun fırça darbeleri ile boyar.

Ink Outlines: Görüntünün ana hatlarını mürekkeple çizilmiş etkisi verecek şekilde boyar.

Spatter: Görüntüye püskürtme tarzı boyama etkisi verir.

Sumi-e: Doygun fırça darbeleri ile siyah mürekkep ağırlıklı bulanık bir boyama üslubu kullanır.

Distort Filtre Grubu

Bu filtre grubu, görüntüleri deformasyona uğratarak farklı şekiller elde etmenize olanak tanır.

Displace: Görüntünün nasıl deforme edileceğini açılır pencere yardımıyla seçeceğiniz görüntü belirleyecektir.

Glass: Görüntüye bir camdan bakılıyormuş izlenimi oluşturur. Dilerseniz bir PSD dosyası formatında kendi cam efektinizi de oluşturup kullanabilirsiniz.

Lens Correction: Bu filtre yardımıyla görüntülerde rastlanan mercek hatalarını düzeltebilirsiniz.

Ocean Ripple: Resme dalga efekti vererek su altından görünüyormuş izlenimi doğurur.

Pinch: Görüntüyü içe ya da dışa doğru kaydırır.

Ripple: Dalga deseni oluşturmak içindir.

Sheat: Açılan penceredeki kutuda göreceğiniz eğri yardımıyla görüntüyü deforme edebilirsiniz. Eğri üzerinde istediğiniz noktalara tıklayıp sürükleyerek istediğiniz etkiyi elde edebilirsiniz.

Twirl: Görüntüde **Angle** sürgüsünü kullanarak istediğini açıda bir burgu elde edebilirsiniz.

Wave: Ripple filtresinden daha fazla denetim imkanı (dalga boyu, dalga yüksekliği, dalga türü) veren bu filtre ile daha hassas dalga efektleri elde edebilirsiniz. Dilerseniz **Randomize** tuşu ile rastgele değerlere sahip bir efekt de üretebilirsiniz.

Zigzag: Zik zak şeklinde dalgalı bir görüntü elde etmek içindir.

Noise Filtre Grubu

Bu gruptaki filtreler, rastgele dağıtılmış renklerle birtakım etkiler oluşturmayı amaçlamaktadır. Ayrıca bu filtre aracılığı ile resimlerdeki bazı problemleri (örn: çizikler) de giderebilirsiniz.

Despeckle: Görüntüde renk geçişlerinin olduğu bölgeler dışındaki kısımları bulanıklaştırır. Bu sayede parazitlerin giderilmesini sağlamış olur.

Median: Görüntüyü oluşturan piksellerin parlaklık değerlerini karıştırır. Fotoğraflardaki hareket etkisini azaltma noktasında bu filtreden faydalanabilirsiniz.

Reduce Noise: Açılan pencereden yapacağınız ayarlamalar doğrultusunda görüntüdeki parazitleri azaltır.

Pixelate Filtre Grubu

Bu filtre grubundaki filtreler pikselleri kare, nokta... vb şekillerde gruplayarak gösterirler.

Crystallize: Pikseller bu filtre uygulandığında renkli çokgenler şeklinde gruplandırılırlar.

Facet: Görüntülerin üzerilerinde elle boyama yapılmış hissi veren bir filtredir.

Fragment: Pikselleri birbirlerine kaydırarak bulanık bir görüntü oluşturur.

Mezzotint: Açılan pencereden bir nokta tipi seçerek rastgele bir desen şeklinde dönüştürebilirsiniz.

Mosaic: Görüntüyü oluşturan pikselleri kare mozaik şeklinde görüntüler.

Pointillize: Görüntüyü oluşturan pikselleri noktalar şeklinde görüntüler.

Render Filtre Grubu

Clouds & Differance Clouds: Rastgele değerler kullanarak bulutvari görünümde desenler oluştururlar.

Fibers: Görüntüden lif görünümlü desenler oluşturur.

Lens Flore: Işığın merceğe yansıması efekti oluşturur. Açılan pencerede görüntünün küçük resmindeki **+** (artı) işaretinden faydalanarak parlak ışığın oluşacağı merkez noktasını belirleyebilirsiniz.

Lighting Effects: Değişik ışık sitillerini ve türlerini kullanarak çok farklı aydınlatma etkileri elde edebilirsiniz.

Sharpen Filtre Grubu

Bu filtre grubu, bulanık görüntüleri netleştirmek için kullanılır.

Sharpen & Sharpen More: Görüntülere keskinleştirme ve daha çok keskinleştirme etkisi verir.

Sharpen Edges & Unsharp Mask : Sharpen Edges filtresi görüntünün tamamını değil renk geçişlerinin olduğu noktaları keskinleştirir. Ayrıntılı bir keskinleştirme işlemi içinse **Unsharp Mask** filtresindeki **Amount Radius** ve **Threshold** sürgülerini kullanın.

Smart Sharpen: Görüntüdeki aydınlık ve gölgeli kısımların ne miktarda keskinleştirileceğini ayrıntılı olarak belirlemenize olanak tanır.

Sketch Filtre Grubu

Bas Relief: Koyu renkli bölgeler için ön plan rengini, açık renkli bölgeler içinse arka plan rengini kullanarak kabartma efekti oluşturur.

Charcoal: Bir kağıt üzerine uygulanan boyalar ile oluşturulmuş poster etkisi benzeri bir etki oluşturur.

Chrome: Görüntüden krom kaplı bir yüzey etkisi oluşturur.

Conte Crayon: Bu filtrede de koyu renkli bölgeler için ön plan rengi, açık renkli bölgeler içinse arka plan rengi kullanılır. Bu şekilde tebeşirle oluşturulmuş gibi bir doku elde edilir.

Photocopy: Fotokopi ile çoğaltılmış etkisi oluşturur.

Plaster: Koyu renkli bölgeleri çıkıntı, açık renkli bölgeleri ise girinti şeklinde gösteren sıva etkisi oluşturur.

Stamp: Istampa kullanılarak yapılmış damga etkisi oluşturur.

Water Paper: Renkleri dağıtır ve birbirlerine karıştırır. Böylece ıslak bir kağıda yapılmış resim etkisi verir.

Stylize Filtre Grubu

Diffuse: Renkleri dağıtarak karıştırır ve daha yumuşak bir görüntü oluşturur.

Emboss: Görüntüye kabarta etkisi verir.

Extrude: Üç boyutlu doku etkisi oluşturur.

Find Edges: Görüntüde renk geçişlerinin olduğu alanları koyu çizgiler ile çizerek çizim efekti oluşturur.

Glowing Edges: Görüntüye neon ile aydınlatılmış izlenimi kazandırır. Filtre birkaç kez tekrar uygulanarak etki değiştirilebilir.

Solarize: Fotoğraflardaki negatif ile pozitif görünümleri katıştırarak solarizasyon etkisi oluşturur.

Wind: Rüzgar efekti oluşturmak için kullanılır.

TEXTURE FİLTRE GRUBU

Craquelure: Kabartmalı ve ince çatlaklardan oluşan bir etki sağlar.

Grain: Görüntüye değişik tiplerde doku eklemek içindir.

Mosaic Tiles: Küçük karolardan oluşan bir görüntü oluşturur.

Patchwork: Bu filtre ile küçük karelerden oluşan bir görüntü elde edilir.

Stained Glass: Vitray etkisi elde etmek içindir.

Texturizer: Görüntüye seçilen dokuyu uygular.

ÇİZİM

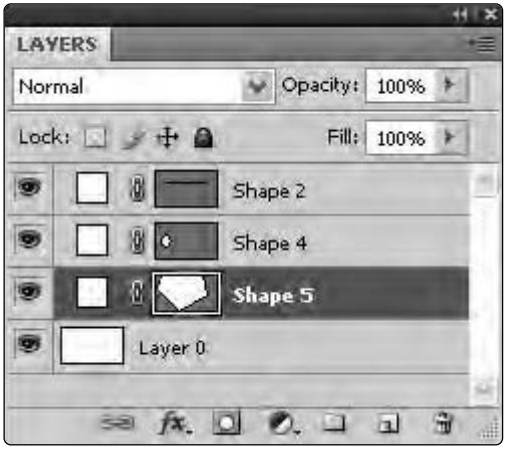
Photoshop ile çizim işlemleri için kalem ve şekil araçlarını kullanacağız. Bu araçlar ile vektörel şekiller, yollar, ayrıca web tasarımlarınız için menüler, düğmeler... vb bir çok uygulamayı üretebilirsiniz.

ÇİZİM MODLARI

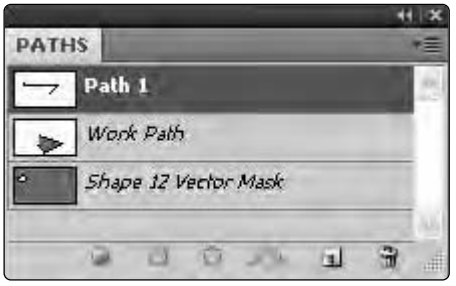


Kalem ve şekil araçlarını kullanırken karşımıza üç farklı mod çıkmaktadır. İhtiyacınıza göre **Options** panelinden gerekli olan modu seçip kullanabilirsiniz.

Bunlardan **Shape Layers**  şekilleri ayrı katmanlarda oluşturur. Özellikle web sayfaları için yapacağınız çalışmalarda daha kolay ve daha detaylı bir biçimde düzenleme olanağı sunduğundan doğru tercih olacaktır. Bir şekil katmanı, renk dolgusu ve bir vektör maskesinden oluşur. Şekil çizerken merkezden dışarı doğru çizmek için **ALT** tuşuna da basılı tuttun.



Bu modda çalışırken katmana bir başka şekil eklemek için **Options** panelinden **Add to shape area** (**+**) tuşunu seçili hale getirin. Artık seçili olan katmana bir başka şekil çizebilirsiniz. Bir şekli çıkarmak içinse **Subtract from shape area** (**–**) tuşunu seçili hale getirin.



Paths modu ise **Paths** panelinden görüp kontrol edebileceğiniz geçici vektör yolları çizmek içindir.

Fill Pixels modu da aynı katman üzerinde işlem yapmak içindir. Bir fırça aracının yaptığına benzer bir iş yapmış olur.

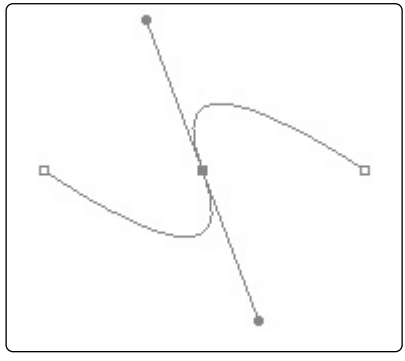
Kullanabileceğiniz en önemli araçlardan biri olan **Line Tool** aracı ile ilgili ayarları yapabilmek için de araç seçili iken **Options** panelindeki **Geometry Options** ters okunu tıklayın.

Açılan pencereden çizginize ok başı ve sonu ekleyebilirsiniz. Ayrıca ok başlarının iç bükey oranı, genişlik ve uzunluk değerlerini % (yüzde) cinsinden belirleyebilirsiniz.



Diğer şekil araçlarıyla kullanılan şekil özelliklerinden **Unconstrained**, dikdörtgen, elips ya da bir özel şeklin boyutlarını fare ile sürükleyerek belirleyebilmek içindir. **Fixed Size** ile ebatları **W:** ve **H:** kutularına gireceğiniz değerlere göre belirlenecek şekiller çizebilirsiniz. **From Center** kutucuğunu seçili hale getirerek de merkezden başlayarak genişleyen şekiller çizebilirsiniz.

KALEM ARACI



Şimdi de **Pen Tool** aracının nasıl kullanıldığına bir bakalım. Aracı tuval üzerinde bir noktaya, ardından da bir başka noktaya tıklattığınızda iki nokta arasında düz bir çizgi oluşturulacaktır. Bu noktaları devam ettirerek tıkladığınız bağlantı noktalarından oluşan istediğiniz gibi düz ya da eğri bir yol oluşturabilirsiniz.

Yollar açık ya da kapalı olabilirler. Açık yolların iki ayrı bitiş noktası olmaktadır. Kapalı yollar (örn: çember) ise süreklidir. Bir yolu kapatmak için **Pen Tool** aracını ilk bağlantı noktasının üstüne getirin. Bu durumda kalem aracının yanında içi boş küçük bir daire görünecektir. Artık bağlantı noktasına tıklayarak yolu kapatabilirsiniz.

Bir yola yeni parçalar eklemek için **Add Anchor Point Tool** aracını Resim545 kullanabilirsiniz. Araç ile yolun istediğiniz bir noktasına tıklayın. Bu takdirde yeni bir nokta eklenecektir. Artık bu noktayı sürükleyerek yolunuzda gerekli değişiklikleri yapabilirsiniz. Bir bağlantı noktasını silmek içinse, **Delete Anchor Ponit Tool** aracı ile silmek istediğiniz noktanın üzerine gelip bir kez tıklamanız yeterlidir.

Yaptığınız bir seçimi yola dönüştürmek isterseniz, **Paths** panelinde bulunan **Makes Form Selection** tuşunu tıklayın. Seçiminiz **Work Path** adıyla yola dönüştürülecektir. Yolun üzerine çift tıklayarak yolun ismini değiştirebilirsiniz.



Bir nesnenin etrafında, nesneye yapışan bir yol çizmek istediğinizde **Freeform Pen Tool** aracı seçili iken **Options** panelindeki **Magnetic** kutucuğunu seçili hale getirin. Bu takdirde imleç, mıknatıslı kalem aracına dönüşecektir.

Ardından **Freeform Pen Options** penceresindeki ayarları dilediğiniz gibi değiştirin. Artık nesnenizin etrafında bir yol çizebilirsiniz.



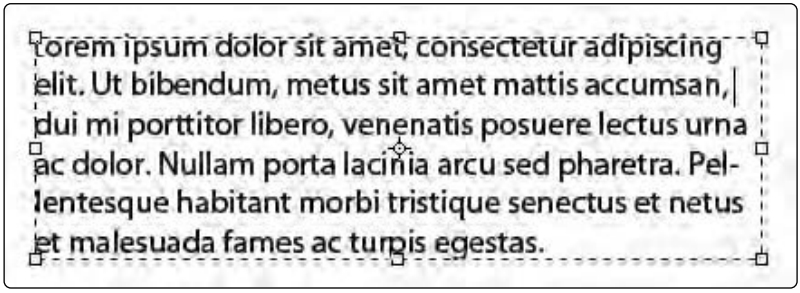


Bu yolu seçime dönüştürmek içinse **Paths** panelinden **Load Path As Selection** tuşunu tıklamanız yeterli olacaktır.

METİNLER

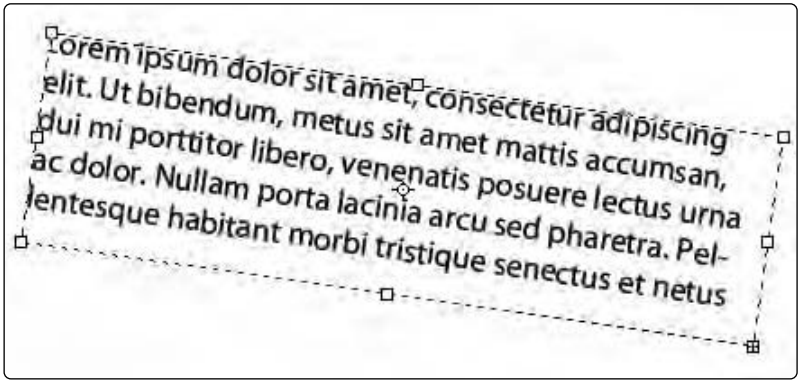
Photoshop ile yazı oluşturduğunuzda bu yazı **Layers** paneline yeni bir katman olarak eklenecektir. Bir yazı oluşturmak için **Horizontal Type Tool** ya da **Vertical Type Tool** araçlarını kullanacağız.

Aracı seçtikten sonra imleci yazıyı oluşturacağınız yere tıklayın ve yazınızı girin. Yeni bir satıra geçmek istediğinizde **Enter** tuşuna basın.

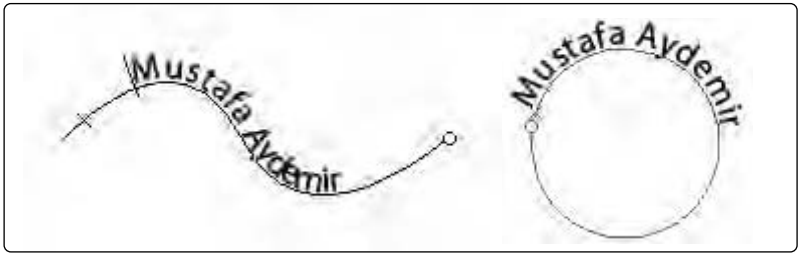


Bir metin oluştururken imleci metni oluşturacağınız yere tıklamak yerine sürükleyerek bir alan seçebilirsiniz. Oluşturduğunuz bu alan, gireceğiniz metnin de sınırlarını belirleyecektir. Yazdığınız metin bu sınıra dayandığında bir alt satırdan devam edecektir.

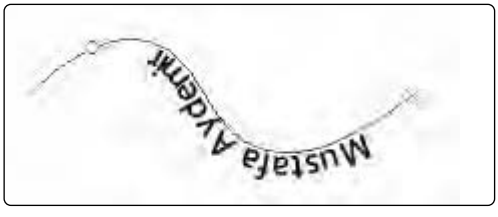
Sınır belirleme alanının köşelerinde yer alan tutamaçlar yardımıyla alanın ebatlarını değiştirebilirsiniz. Köşelerdeki bu tutamaçların üzerine geldiğinizde imleç çift taraflı ok işaretine dönüşecektir. Artık fareye basılı tutarak alanı istediğiniz gibi büyütebilir ya da küçültebilirsiniz. Söz konusu alanın ebatlarını değiştirirken alanın mevcut oranlarını korumak için sürükleme esnasında **SHIFT** tuşuna da basılı tutmalısınız.



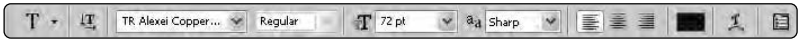
Dilerseniz metin alanını döndürebilirsiniz. Bunun için metin alanının köşelerindeki tutamaçları imleç çift taraflı eğri ok biçimde iken tutun ve istediğiniz yönde sürükleyerek döndürün.



Bir metni bir yola ya da şekle ekleyerek de kullanabilirsiniz. Bu durumda metin eklediğiniz yola göre devam edecektir. Metni yol boyunca ilerletmek için imleci metnin başına getirdikten sonra klavyeden boşluk tuşuna basmalısınız.



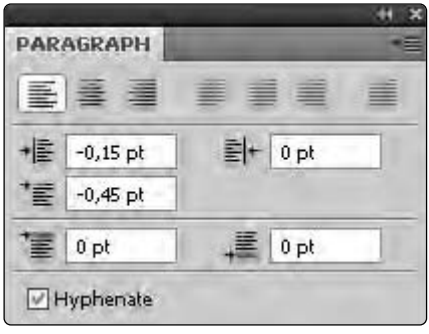
Yol üzerindeki yazıyı taşımak veya döndürmek için öncelikle **Path Selection Tool** ya da **Direct Selection Tool** aracını seçin. **İmleç** şeklini alınca fare ile metne tıklatıp sürükleyin. Döndürmek içinse metni dikine sürükleyin. Bir yolu metin ile beraber bir başka konuma taşımak içinse **Path Selection Tool** aracıyla yola tıklayıp istediğiniz konuma sürükleyin.



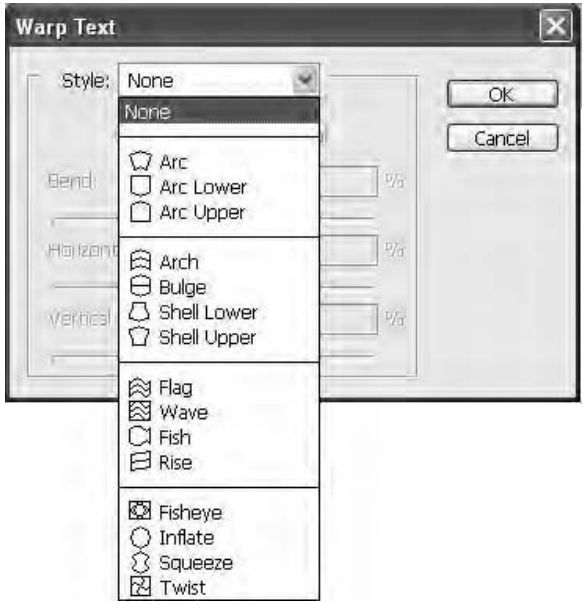
Metinlerinizi oluşturan karakterlerle ilgili düzenlemeleri metin aracı seçili iken **Options** panelinden yapabilirsiniz. Panelden metnin yazı tipi, punto, hizalama, renk... vd ayarlarını yapabilirsiniz.



Daha detaylı ayarlamalar yapmak için **Character** panelini kullanabilirsiniz. Panelde, yanında değer olan seçenekleri kullanırken değer kutucuklarına doğrudan bir rakam girebileceğiniz gibi simgeleri tıklayıp sağa-sola kaydırarak da mevcut değerleri değiştirebilirsiniz. Örneğin bahsettiğim şekilde panel vasıtası ile **dikey ölçekleme** ve **yatay ölçekleme** yapabilirsiniz.



Photoshop ile metinleri oluşturan karakterleri detaylı olarak biçimlendirebildiğiniz gibi paragrafları da biçimlendirebilirsiniz. Bunun için **Paragraph** panelinden yaralanabilirsiniz. Bu panel yardımıyla paragrafları hizalama, paragraflara girinti verme, tire koyma, paragrafların üstlerinde ve altlarında yer alacak boşluklar gibi ayarlamaları hızlı bir biçimde yapabilirsiniz.



Hazırladığınız metinlere uygulayabileceğiniz bir diğer işlem de çarpıtma işlemidir. Bu işlem için bir metin aracı seçili iken **Options** panelindeki çarpıtma tuşunu kullanarak çarpıtma sitil seçeneklerinin bulunduğu pencereye ulaşabilirsiniz.



Bu pencereden bir stil seçerek metninize uygulayabilirsiniz.

Bir metni gerektiğinde şekle dönüştürebilirsiniz. Bu işlem için **Layer>Type>Convert to Shape** menüsünü kullanabilirsiniz. İşlemi gerçekleştirdiğinizde yazı katmanı vektör maskeli bir katmana dönüşür. Oluşan vektör maskesini gerektiğinde düzenleyebilirsiniz. Fakat şekle dönüştürdüğünüz için metindeki karakterleri düzenlemeniz artık mümkün olmayacaktır.

WEB SAYFALARI

SAYFALARI DİLİMLEME

Photoshop ile hazırladığınız web sayfalarını dilimleyerek daha küçük parçalara bölebilirsiniz. Bu sayede sayfalarınız tarayıcılarda daha hızlı bir şekilde yüklenecektir. Bu arada gerekli HTML ve CSS kodları da üretilecektir. Bunun için Photoshop'un **File>Save for Web** komutunu kullanabilirsiniz.



Şimdi öncelikle **Slice Tool** aracını seçin ve tasarımınızı dilimleyin. Bu işlem ile kullanıcı tabanlı bir dilimleme yapmış oldunuz. Dilimleme işleminden sonra **Slice Select Tool** aracı ile dilimleri yeniden hizalayabilir, taşıyabilir ya da yeniden boyutlandırabilirsiniz. Photoshop sizin dilimlediğiniz alanlar haricindeki alanları otomatik olarak dilimleyecektir.

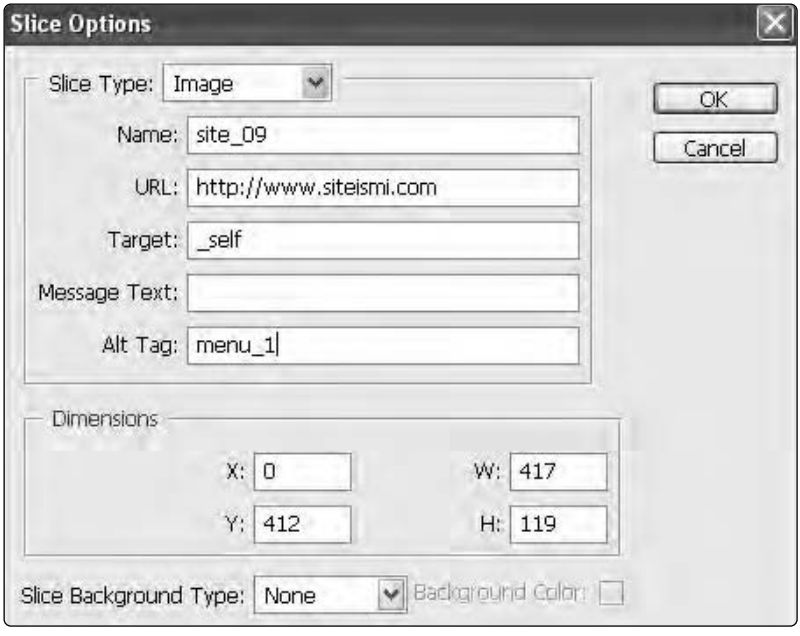
Bir başka dilimleme yöntemi olarak kılavuz çizgilerini kullanabilirsiniz. Kılavuz çizgilerini sürükleyip bırakarak dilimlemek isteğiniz yerlere konumlandırın. Daha sonra **Slice Tool** aracını seçip, **Options** panelindeki **Slices From Guides** tuşuna tıklayın.

Photoshop ile dilim aracındaki gibi esnek olmasa da katmanları kullanarak da dilimler oluşturabilirsiniz. Bunun için **Layers** panelinden katmanı seçin ve **Layer>New Layer Based Slice** komutunu tıklayın.

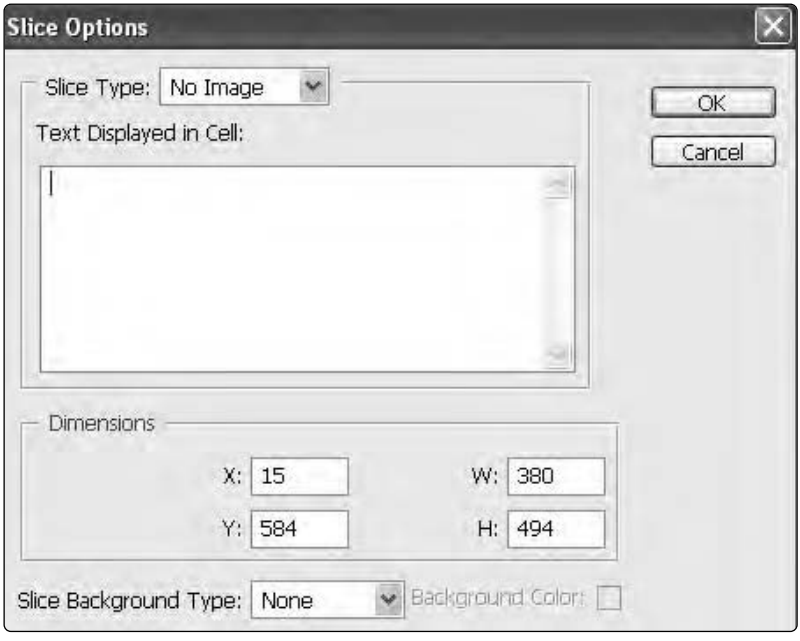
Oluşturduğunuz bir dilimi **ALT** tuşuna basılı tutup sürükleyerek çoğaltabilirsiniz. **Slice Select Tool** aracı ile **SHIFT** tuşuna da basılı tutarak birden çok dilim seçebilirsiniz. Bu şekilde birden fazla dilim seçili iken fare ile sağ tıklayarak açılan pencereden **Combine Slices** komutu ile dilimleri birleştirebilirsiniz.



Web sayfanız üzerinde birçok dilimleme işlemi yaptığınızda bazı dilimler üst üste gelebilir ve bunları düzenlemekte zorluk çekebilirsiniz. Bu noktada **Options** panelindeki resimde gördüğünüz tuşlar yardımı ile dilimlerin dilim yığını içindeki konumunu değiştirebilirsiniz. Dilimi en üste yada bir üste taşıyabilir ya da en alta gönderebilirsiniz. Bir dilimi ya da dilimleri silmek için dilimler seçili iken **Delete** tuşuna basmanız yeterli olacaktır. Dilimleri silmek içinse **View>Clear Slices** menüsünü kullanabilirsiniz. Dilimleri kilitlemek gerektiğinde ise **View>Lock Slices** menüsünü kullanın.



Slice Select Tool aracı ile bir dilime çift tıklayarak **Slice Options** penceresini açabilirsiniz. Bu pencereden dilim türünü seçebilirsiniz. Varsayılan dilim türü **Image**'dır. Tür olarak Image seçerek resme bir URL bağlayabilir ve dilerseniz resmin arka plan rengini belirleyebilirsiniz. **Target** kutucuğuna `_blank` ifadesini girdiğinizde, bağlantı yeni bir tarayıcı penceresinde görüntülenecektir. Kutucuğa `_self` ifadesini girdiğinizde ise aynı tarayıcı penceresinde açılacaktır.



Bir web sayfası sadece resimlerden oluşmaz. **Slice Options** penceresinde dilim tipi olarak **No Image** seçerek bu dilimin metin dilimi ya da boş hücre dilimi olmasını sağlayabilirsiniz.

GÖRÜNTÜLERİ EN İYİLEŞTİRME

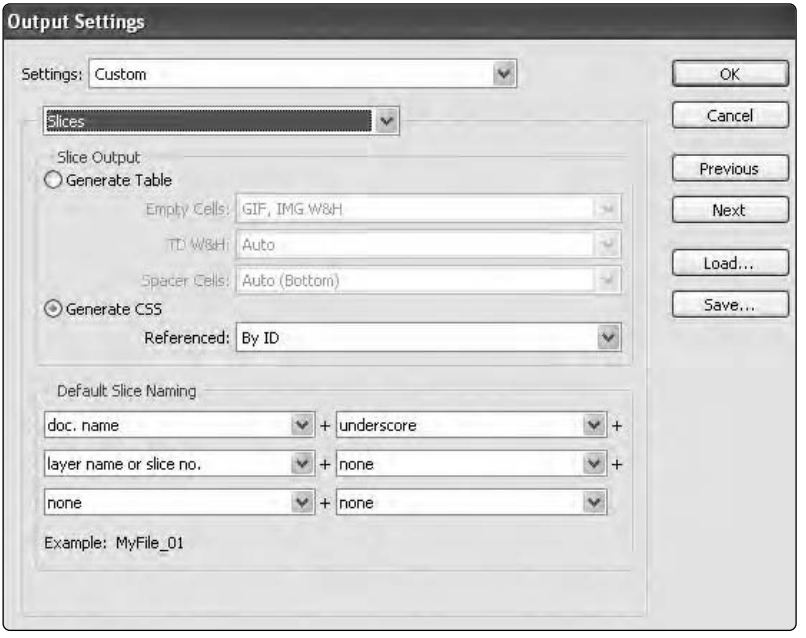
Web için hazırlayacağımız görüntülerin hem mümkün olan en iyi görüntü kalitesine hem de mümkün olan en küçük dosya boyutuna sahip olmaları gerekir. Bu işlem için **File>Save for Web & Devices** menüsünden yaralanabilirsiniz.



Açılan pencere yardımıyla görüntüleri en iyileştirebilir, farklı dosya formatları için deneyip farklı dosya formatlarındaki yükleme sürelerini de gözlemleyebilir ve ardından kaydedebilirsiniz. Kaydetmeden önce tarayıcı ön izlemesi de yapabilirsiniz.

Görüntüyü Kaydetme

Gerekli ayarlamaları bitirdiğinizde **Save** tuşuna tıklayın. Açılan pencereden kayıt türü olarak H**TML and Images** seçeneğini ardından da **Settings** bölümünden **Other** seçeneğini seçin.



Other'ı seçtiğinizde yeni bir pencere açılacaktır. Bu pencerede **Next** tuşu ile Slice ayar penceresine gelebilirsiniz. Burada **Slice Output** kısmından **Generate CSS** radyo düğmesini seçerek HTML çıktınızın CSS kodlarını da içermesini sağlayabilirsiniz. Böylece işlemi tamamladığınızda oluşturduğunuz dilimleri içeren bir images klasörü ile HTML dosyanızı oluşturmuş olacaksınız.

PHOTOSHOP CS4 AUTOMATE

Automate özellikleri sayesinde Photoshop CS4 ile bir çok işlemi otomatik olarak yaptırabilirsiniz.

AUTOMATE>BATCH

Batch özelliği ile birçok fotoğrafa tek bir komut yardımıyla bir çok işlemi tek seferde uygulayabilirsiniz. Örneğin birçok fotoğrafınızın ebatlarını değiştirmeniz ya da tüm fotoğraflar üzerine aynı metni eklemeniz gerekiyorsa bunları tek tek açıp işlem yapmak bir hayli zamanınızı alacaktır. Fakat Photoshop ile bu işlemi bir çırpıda yapabileceksiniz.

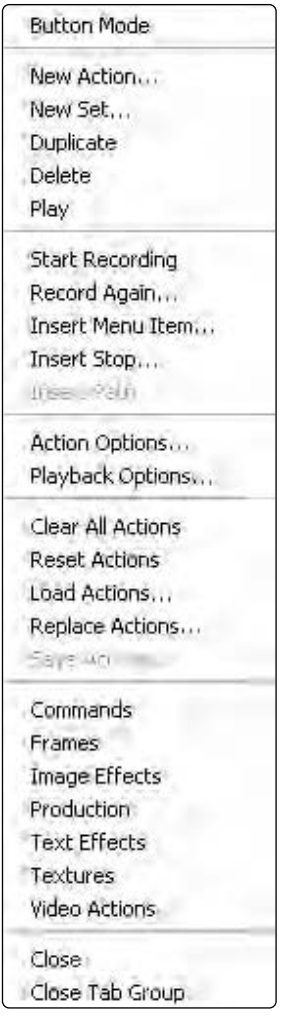


Konuyu detaylandırırken öncelikle **Actions Panel**'ini tanıyalım. Çünkü işlemlerimizde bu panelden yaralanacağız. Burada yapmamız gereken değişiklikleri tek bir fotoğrafa uygulamak ve ardından **Actions** paneline kaydetmek olacaktır. Ardından **Batch** komutu yardımıyla bu işlemi istediğiniz fotoğraflara uygulayabilirsiniz. Burada dikkat edeceğimiz husus, fotoğraflara uygulanacak farklı işlemleri değil aynı işlemler için Actions panelini kullanmaktır.

Şimdi de **Actions** panelinin alt kısmında bulunan ve işlemlerde kullanacağımız tuşlara göz atalım. **Stop** tuşu çalışmanıza uygulayacağınız işlemleri **Actions** paneline kaydederken kaydedilmemesi gereken bir işlem olduğunda ya da çalışmanızı tamamladığınızda bu tuş yardımıyla işlemi durdurabilirsiniz.

Begin Recording tuşu kayıt işlemini başlatmak için kullanılan tuştur. **Play Selection** tuşu ise aktif olan çalışmanıza bir işlemi uygulamak içindir. Bir sonraki tuşumuz olan **Creat New Set** ise yeni bir action grubu oluşturmak içindir. **Creat New Action** tuşunu da yeni bir action tanımlamak istediğimizde kullanacağız.

Şimdi de gelelim actions panelindeki menüye. Menüdeki komutları sırasıyla inceleyecek olursak:



Button Mode: Tanımlanan action'lar renk sıralamasıyla görüntülenir. Buradaki renkleri **Creat New Action** tuşunu tıkladığımızda açılan pencereden belirliyoruz.

New Action: Yeni bir action tanımlamak içindir.

New Set: Yeni bir action grubu oluşturur.

Duplicate: Seçilen action'ı kopyalamak içindir.

Delete: Seçilen Action'ı siler.

Play: Seçilen Action'ı çalışmanıza uygulanmak için başlatır.

Start Recording: Kayıt işlemini başlatmak içindir.

Insert Menu Item: Action işlemi haricinde bir menü öğesi ekler.

Insert Stop: Menü eklemeyi durdurur.

Insert Path: Path eklemek içindir.

Action Options: Action Options penceresini açıp gerekli düzenlemeleri yapmanıza olanak tanır.

Playback Options: Action'ı yeniden oynatabilmek içindir.

Clear All Action: Mevcut tüm action'ları silmek içindir.

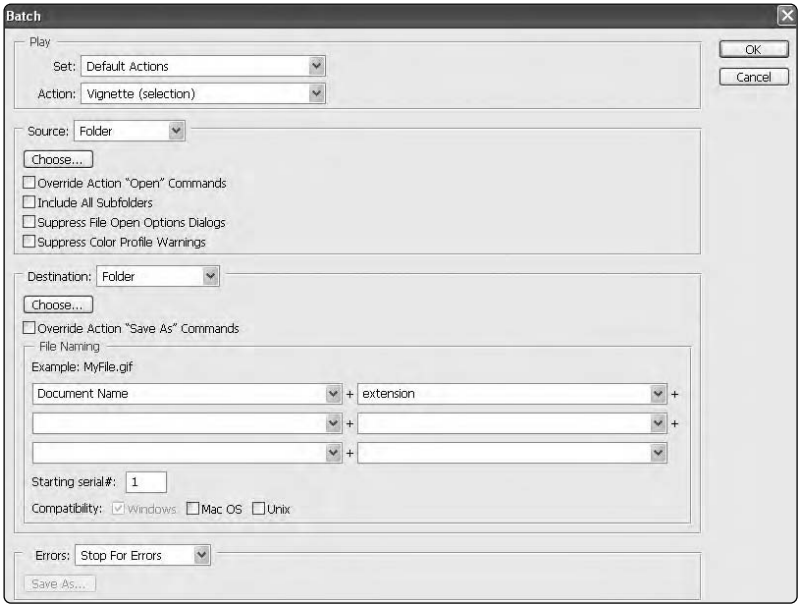
Reset Action: Action'ı başlangıç özelliklerine döndürmek amacıyla kullanılır.

Load Action: Bir action'ı yüklemekte kullanılır. Action'lar **Photoshop CS4>Presets>Action** dizininde bulunurlar.

Replace Action: Action'ı değiştirmek içindir.

Save Action: Action'ı kaydetmek içindir.

Hazır action'lar: Menüdeki **Commands**, **Frames**, **Images Effects**, **Production**, **Text Effects**, **Texture**, **Video Action** öğeleri Photoshop CS4 ile birlikte gelen hazır action'lar olup, dilediğinizi seçerek kullanabilirsiniz.



Şimdi de **File>Automate>Batch** komutu ile resimde gördüğünüz Batch penceresini açalım ve pencere yardımı ile yapabileceklerimize bir bakalım. Böylece action uygulanacak onlarca çalışmayı açmanıza da gerek kalmayacaktır.

Set: Uygulanacak action grubunu belirler.

Action: Uygulanacak action'ı belirlemek içindir.

Source: Çalışmaların konumu belirtmek içindir.

Import: Kamera, scanner... vb harici kaynaklardan alınan çalışmalara action uygulanmasını sağlamak içindir.

Bridge: Bridge'den alınan çalışmalara Action uygulamayı sağlar.

Choose: Çalışmaların konumu göstermek içindir.

Override Action "Open" Commands: Bir action kaydedilirken **Open** komutu kullanıldıysa bu seçim kutucuğu işaretlenmelidir. Aksi halde **Batch** uygulaması yapılamayacaktır.

Include All Subfolders: Seçilen dizinin alt dizinlerine de action uygulanacaksa bu kutucuk seçilmelidir.

Suppress File Open Options Dialogs: Action uygulaması esnasında diyalog pencerelerinin açılmasını engeller.

Supress Color Profile Warnings: Renk profil hata diyalog pencerelerinin açılmasını engeller.

Destination: Bu açılır kutucuk, action uygulanmış çalışmaların kaydedilme durumunu seçmek içindir. **None** seçeneği, action uyguladıktan sonra kaydetmez. **Save and Close** seçeneği ise action uyguladıktan sonra çalışmaları kaydedip kapatmak içindir. **Folder** ise belirlediğiniz klasörü göstermek için **Choose** tuşunu aktif etmek içindir.

Choose: Action uygulanmış çalışmaların kaydedileceği yeri belirlemek içindir.

Override Action "Save As" Commands: Action oluşturulurken **Save** ya da **Save As** komutu kullanılmadıysa bu seçim kutucuğu işaretli olmalıdır. Aksi takdirde çalışmalarınızı kaydetme işlemini elle (*manuel* olarak) yapmanız gerekir.

File Naming: Çalışmalarınıza belge ismi, tarih, seri no eklemek için bu bölümü kullanabilirsiniz.

Starting Serial: Çalışmalara bir başlangıç numarası vermek içindir.

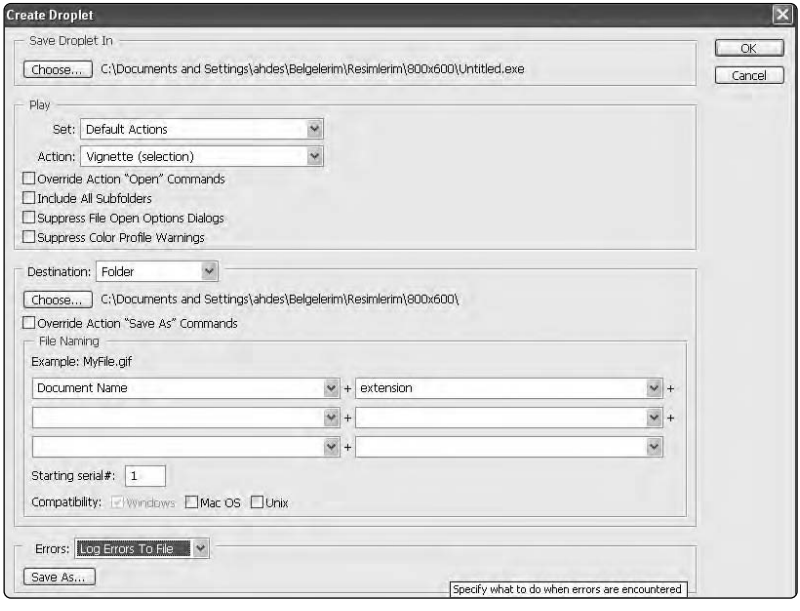
Compatibility: Çalışmalarınız için platform (*Windows*, *Mac*, *Linux*, *Unix*) seçimi içindir.

Errors açılır kutucuğundaki **Stop For Errors** seçeneği ile bir hata meydana gelirse action'ı durdurabilirsiniz. **Log Errors to File** ile de hataların kaydını tutabilirsiniz.

Penceredeki tüm ayarlamaları bu açıklamalar ışığında tamamladıktan sonra **OK** tuşunu tıkladığınızda işlemler otomatik olarak uygulanacaktır.

AUTOMATE>CREAT DROPLET

Oluşturduğunuz action'ları çalışmalarınıza daha hızlı bir biçimde uygulayabilmek için **Creat Droplet** komutundan yararlanabilirsiniz. Burada sistemin işleyişi, hazırlanan action'ların ***.exe*** uzantılı küçük bir program olarak kaydetmek ve yukarıda anlattığım batch işlemlerini her seferinde tekrar yapmak yerine içinde işlem yapılacak uygulamaları barındıran klasörü hazırladığınız bu **Droplet** üzerine sürükleyip bırakmak şeklinde olmaktadır.



Hemen **File>Automate>Creat Droplet** ile penceremizi açalım. Gördüğünüz gibi penceredeki ayarlamalar Batch penceresini ile aynıdır. Farklı olarak **Save Droplet In** ile oluşturacağınız *.exe* uzantılı programın adını ve kaydedileceği yeri belirleyebilirsiniz.

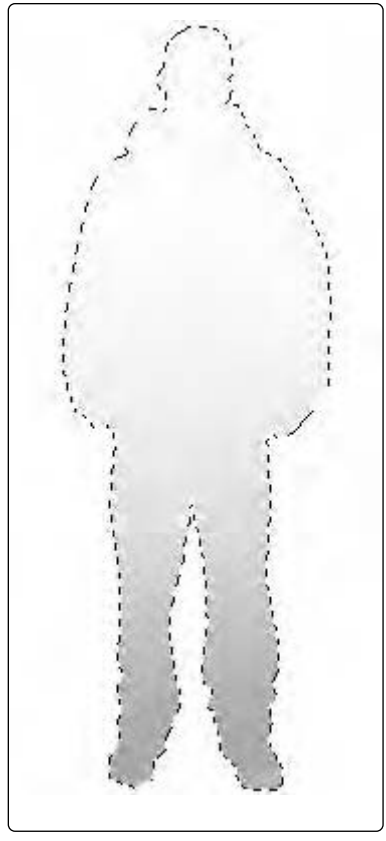
UYGULAMALAR

GÖLGE EFEKTİ

İlk olarak resminizi açın ve resminizi ihtiva eden katmanı **Duplicate Layer** komutu ile çoğaltın.

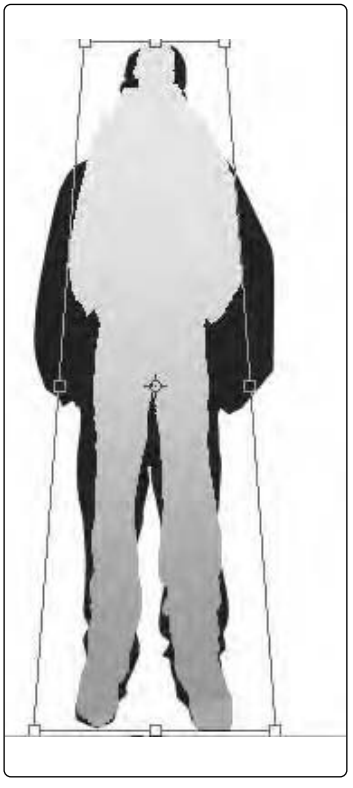
Çoğalttığınız katmanda **Magic Wand Tool** aracı ile resmi seçin ve **Gradient Tool** ile gradient uygulayıp seçimi **Select>Deselect** menüsü yardımı ile kaldırın.

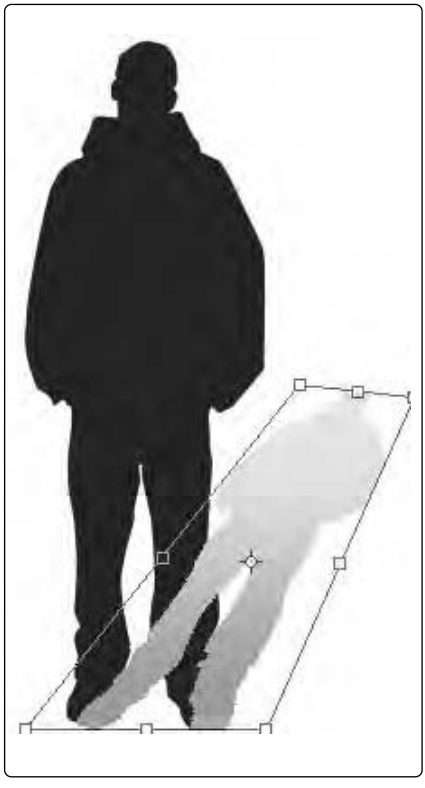




Ardından **Edit>Transform>Perspective** ile resmin üst kısmını daraltın.

Bir sonraki aşama olarak da **Edit>Transform>Distort** ile kontrol noktalarından çekiştirerek görüntüyü yatırın.





Şimdi de **Filter>Blur>Guassian Blur** filtresi uygulayın. Açılan pencerede Radius değerini `2,5` olarak değiştirin. Son olarak filtre uyguladığınız bu katmanı **Layers** panelinde asıl resminizin altına alın.



Resmi Karakaleme Çevirmek

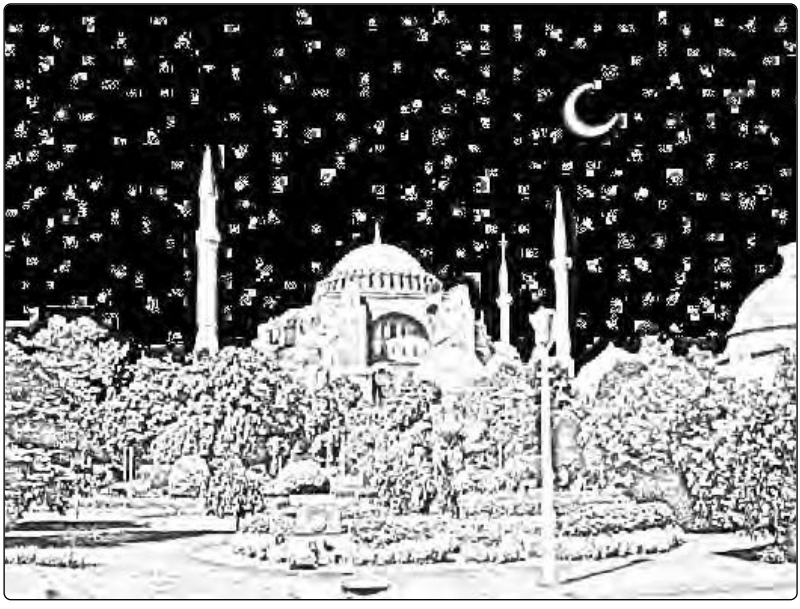


Öncelikle resminizi açtıktan sonra **Layers** panelinden katman üzerine çift tıklayarak resmin arka plandan katmana dönüşmesini sağlayın. Ardından katmanı sağ tıklayıp açılan menüden **Duplicate Layer** komutu ile çoğaltın. Aynı işlemi tekrar ederek katmanı bir kez daha çoğaltın.

Şimdi ilk kopya katmanı seçip **Image>Adjustments>Desaturate** komutu ile katmanı siyah-beyaz yapın. Aynı işlemi diğer kopya katmana da uygulayın.

Daha sonra üstteki kopya katmanı seçip **Image>Adjustments>Invert** komotunu uygulayın.

Ardından katmanın **Blending** değerini **Color Dodge** olarak değiştirin.



Son olarak yapmanız gereken **Filter>Blur>Gaussian Blur** komutu ile **Gaussian Blur** penceresini açıp **Radius** değerini deneyerek resme en uygun olacak şekilde ayarlamak olacaktır.

METNİ RESİMLE DOLDURMAK

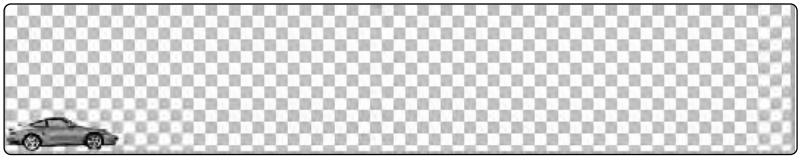


İlk olarak resminizi açın ve üzerine bir metin girin. Böylece elinizde bir resim bir de yazı katmanı olacaktır.

Ardından resim katmanını yazı katmanının üzerine çıkarın. Son olarak da imleç iki katman arasında iken **ALT** tuşuna basılı tutup imleç değiştikten sonra iki katman arasına tıklayın. Sonuç resimdeki gibi olacaktır.



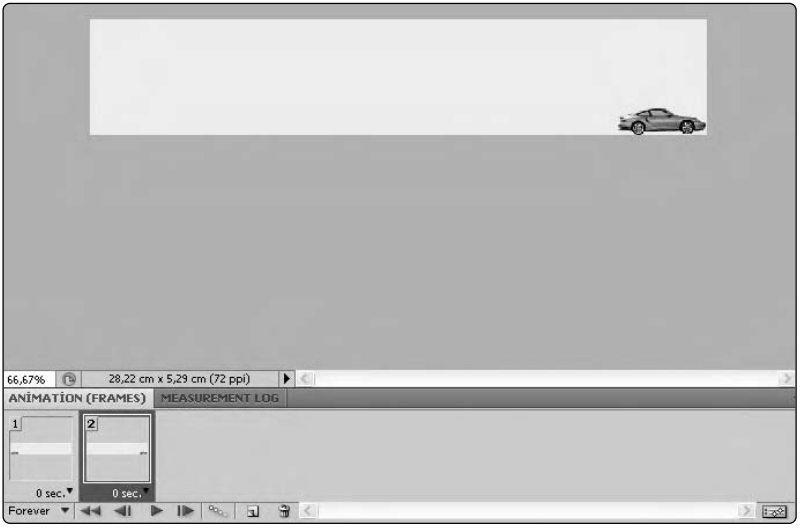
Animasyon Uygulaması



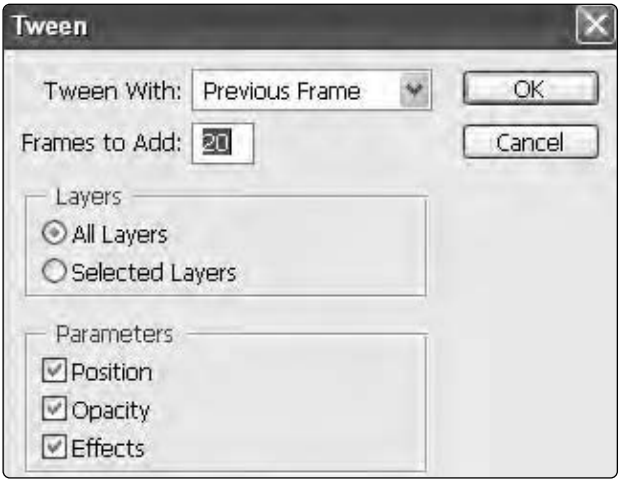
Öncelikle animasyonda kullanacağınız nesneyi seçin ve arka planından temizleyip transparent yapın. Ben örnek olarak bir araba resmi kullandım. Ardından bir yeni katman oluşturup resminizi bu katmana yapıştırın. Sonraki aşama olarak animasyonunuza bir arka plan rengi verebilmek için yeni bir katman daha ekleyin. Eklediğiniz yeni katmanı resminizin olduğu katmanın altına alın.



Ardından **Animation** panelinden kare çoğaltma tuşu ile animasyon karenizin bir kopyasını çıkartın. Eğer **Animation** paneliniz açık değilse **Window>Animation** menüsü yardımıyla açabilirsiniz.



Şimdi de kopya kare seçili iken nesnenizi çalışma alanının sonuna götürün.



Ardından **Animation** panelindeki **Tween** tuşuna  tıklayarak Tween penceresini açın. Pencerede **Frames to Add** kutucuğuna 20 değerini girin. Bu değer animasyon karelerinin sayısını ifade etmektedir. Karelerin sayısı uygun bir değerde ise animasyon daha gerçekçi görünecektir.



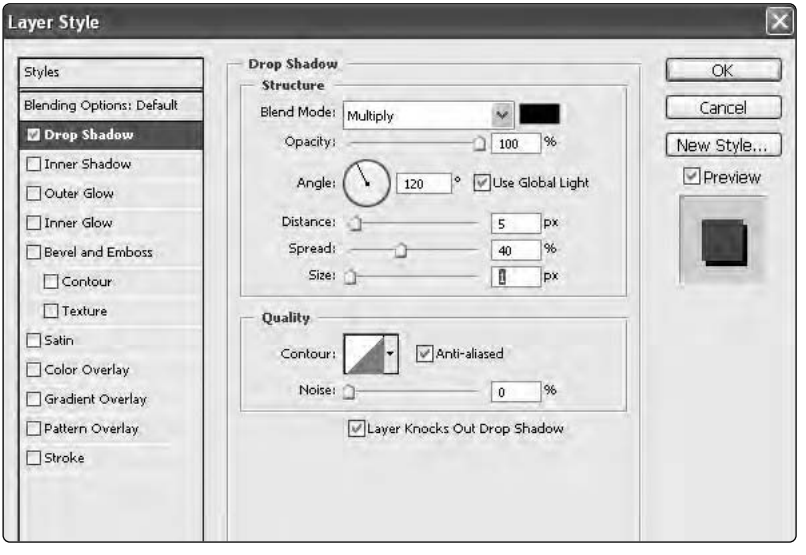
Resimde gördüğünüz gibi kareler verdiğiniz değere göre çoğaltıldı. Artık animasyonu oynatma tuşları  yardımıyla test edebilirsiniz. Artık son işlem olarak **File>Save for Web & Devices** menüsü yardımıyla animasyonu GIF formatında kaydedebilirsiniz.

Diğer uygulamalar gibi bu uygulamanın dosyasını da kitabınızın DVD ekinin **Uygulamalar** klasöründe bulabilirsiniz.

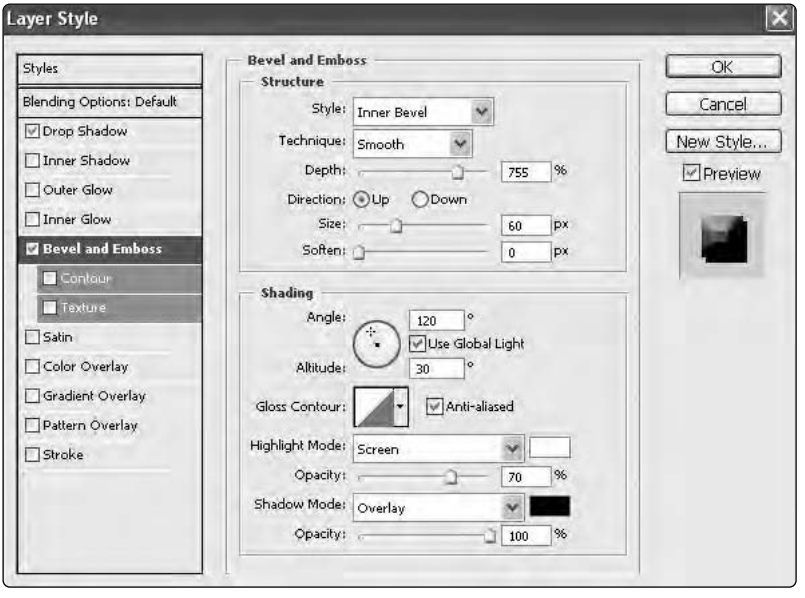
METİN DÜZENLEME



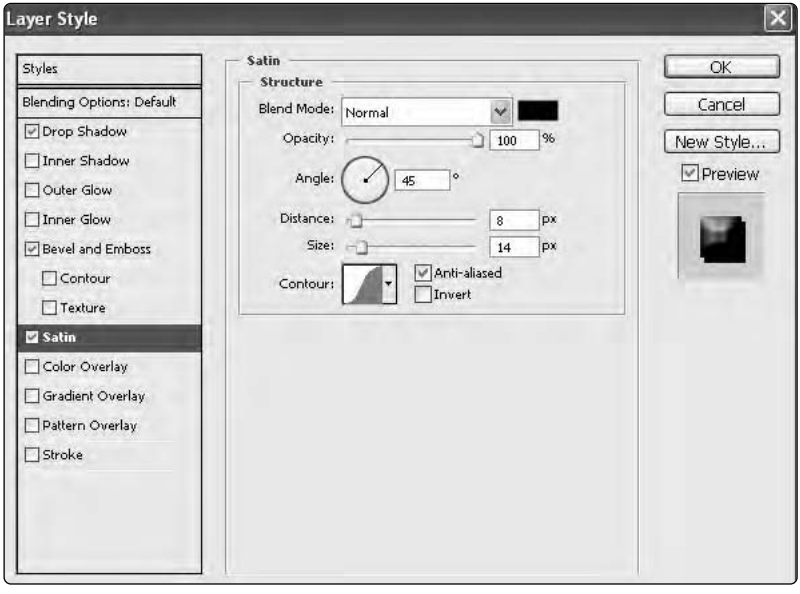
Uygulamanız için istediğiniz ebatlarda yeni bir katman açıp yine istediğiniz edatlarda ve yazı tipinde olacak şekilde metninizi girin. Ardından yazı katmanı seçili iken katmana çift tıklayarak ya da **Layer>Layer Style>Blending Options** menüsü yardımıyla **Layer Style** penceresini açın.



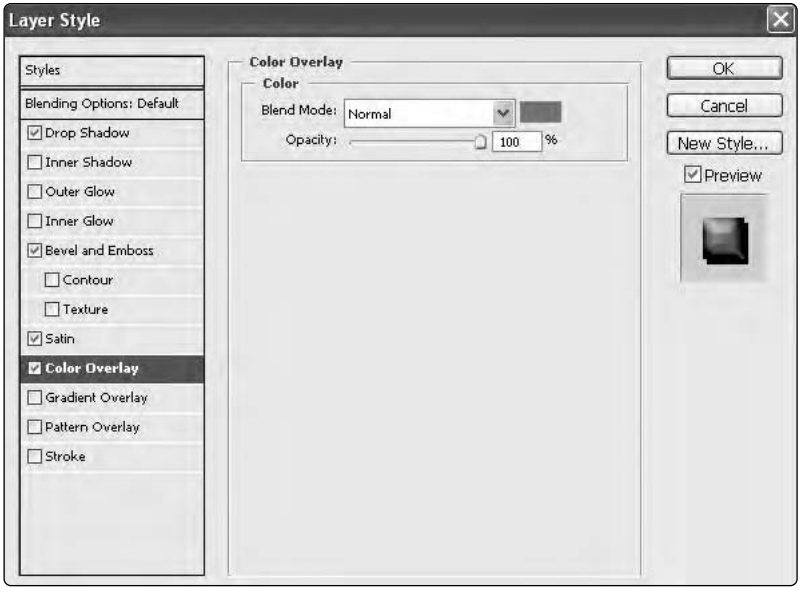
Şimdi metninize şekil vermeye **Drop Shadow** ile başlayabilirsiniz. **Drop Shadow** ayarlarını resimdeki gibi yapın. Bu ayarlar ile metne uygulayacağınız gölge efektinin yönünü, rengini, şiddetini, saydamlığını... vb özelliklerini ayarlamış oluyorsunuz.



Şimdi de **Bevel and Emboss** ile devam edin.

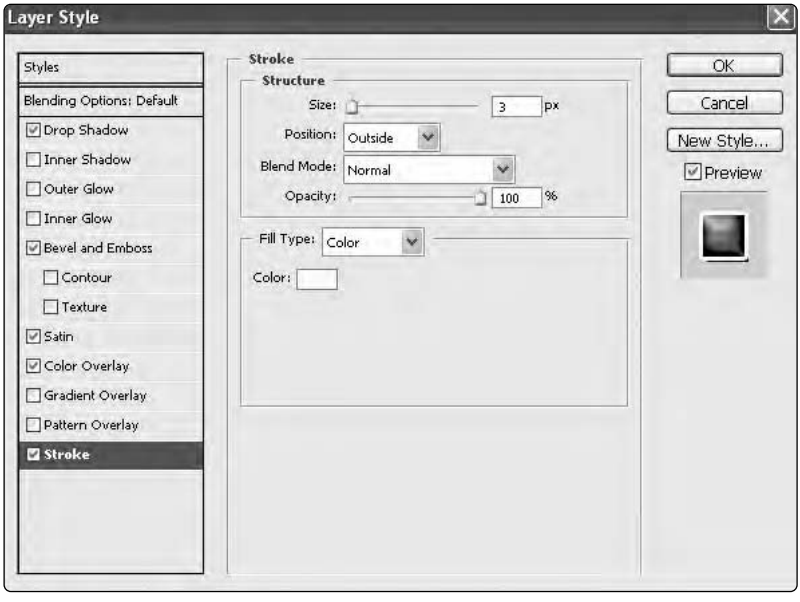


Bir sonraki aşama olarak **Stain** için gerekli ayarlamaları yapın.



Bu sefer sıra **Color Overlay** ayarlarında.

Burada ben kırmızı rengi seçtim. Siz dilediğiniz başka bir rengi de seçebilirsiniz.



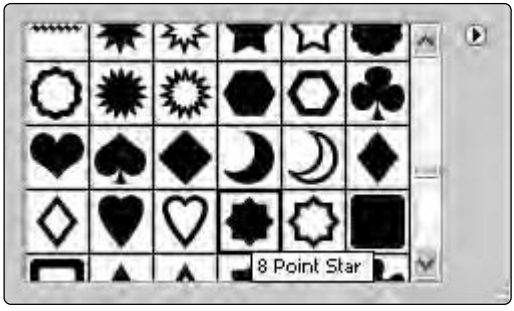
Ve son olarak **Stroke** bölümündeyiz. Renk olarak `beyaz`'ı seçtik.



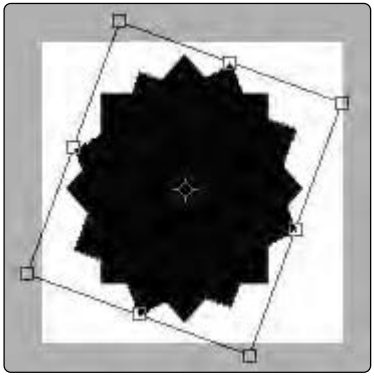
Ve evet işte sonuç. Tabi bu sonuç kişisel beğeniyi yansıtmakta. Siz DVD ekindeki PSD dosyası üzerinde oynayarak daha farklı etkiler ve sonuçlar elde edebilirsiniz.

Web 2.0 Badge Yapımı

İlk iş olarak **File>New** ile `150*150` ebatlarında yeni bir belge açın. Ardından **Custom Shape Tool** aracına tıklayın.

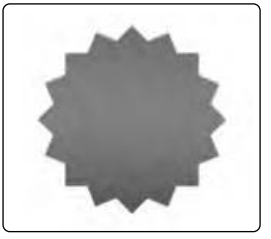


Araç seçili iken **Options** panelinden önce **Fill Pixels** modu tuşunu aktif hale getirin. Ardından da **Shape** açılır kutucuğuna tıklayın. Pencereden 8 köşeli yıldız şeklini seçin. Şekil görünmüyorsa pencerenin sağ üst kısmındaki küçük ok işaretine tıklayın. Açılan menüden **All** seçeneğini ile mevcut tüm şekiller **Shape** penceresinde görünür olacaktır. **Geometry Options** ters okunu tıklayarak **Unconstrained** radyo düğmesini seçin ve fare ile sürükleyerek istediğiniz boyutlarda yıldızınızı çizin.

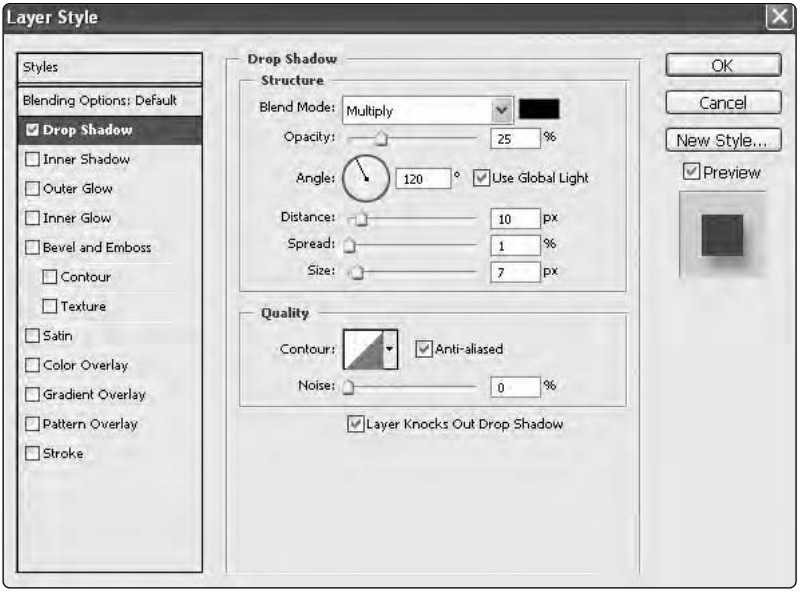


Katmana sağ tıklayarak **Layer** menüsünü açın ve **Duplicate Layer...** ile katmanın bir kopyasını çıkartın. Ardından **Edit>Transform>Rotate** menüsünü tıklayın. Beliren tutamaçlar yardımıyla ikinci katmanı döndürüp resimdeki hale getirin.

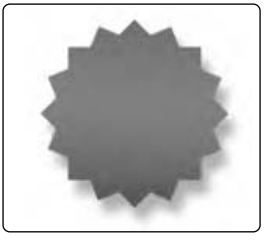
Şimdiki amacımız bu iki şekli birleştirmek. Bunun için **Layers** panelinde **CTRL** tuşunu basılı tutarak iki katmanı da seçili hale getirin ve **CTRL+E** tuşları yardımıyla şeklinizi oluşturun.



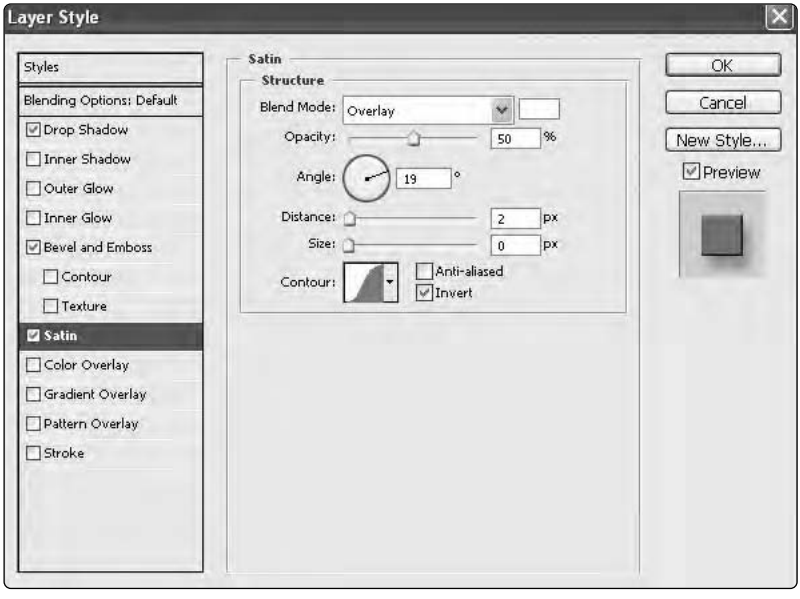
Sırada renk ayarları var. Ama öncelikle **Layers** panelindeki **Lock Transparent Pixel** tuşunu aktif edin ki gradient efekti sadece yıldızınıza uygulansın. Yıldızınız için istediğiniz ön plan rengi belirleyin ve ardından **Gradient Tool** aracını kullanarak gradient efekti oluşturun.



Renk işleminden sonra çalışmanıza biraz da efekt kazandıralım. Önce katmana çift tıklayarak pencerenizi açın ve **Drop Shadow** uygulayın.



Çalışmanın **Drop Shadow** uygulanmış hali resimdeki gibi oldu. Ardından **Size** değerini `2 piksel` olarak **Bevel and Emboss** uygulayın.



Şimdi de resimdeki gibi **Stain** uygulayın.



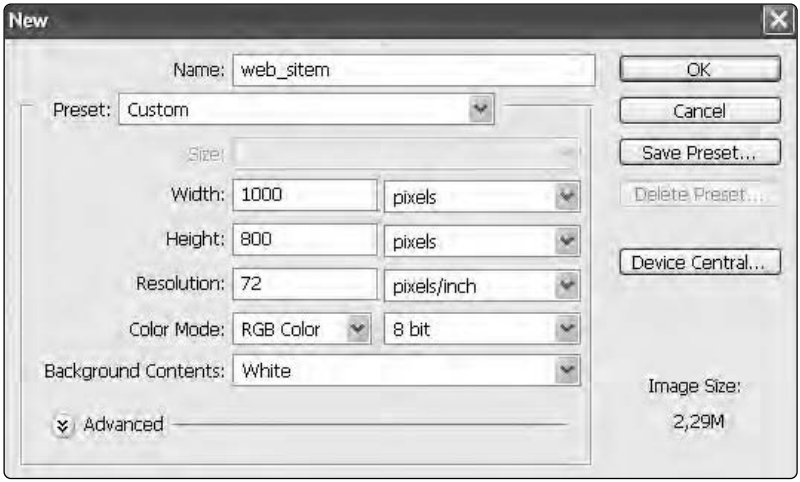
Artık pencereyi **OK** ile kapatabilirsiniz. Bir sonraki aşama olarak yapmanız gereken yeni bir layer daha oluşturmak.

Ardından **Elliptical Marque Tool** aracı ile şekin üzerinde bir alan seçip beyaz renk ile doldurun. Daha sonra imleç iki katman arasında iken **ALT** tuşuna basılı tutup imleç değiştikten sonra iki katman arasına tıklayın. Böylece iki katmanı ilişkilendirmiş oldunuz. Son olarak bu katmanın **opacity** değerini `25` olarak ayarlayın.

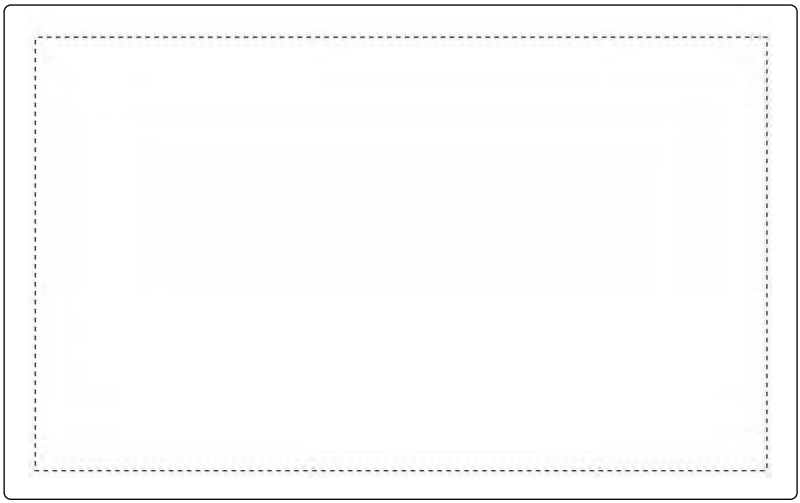


İşte eseriniz hazır! Dilerseniz yıldızın üzerine bir metin de ekleyebilirsiniz.

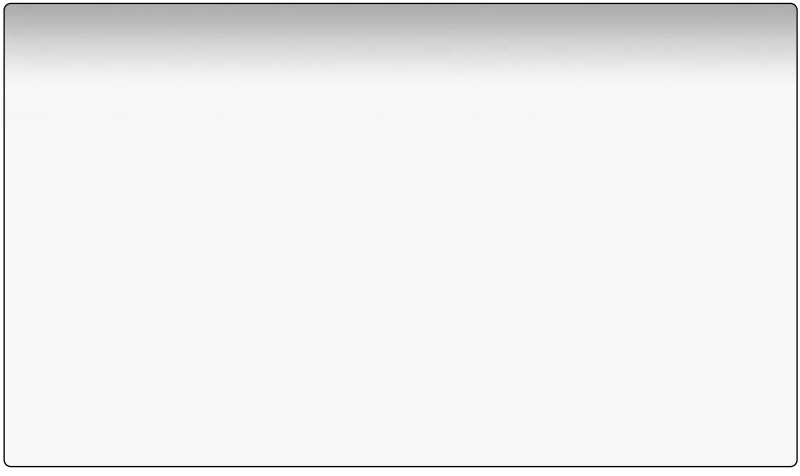
WEB SİTE TASARIMI



Son olarak dilerseniz birlikte bir web arayüz tasarımı yapalım. Öncelikle `1000*800` ebatlarında beyaz arka plana sahip yeni bir belge açın.



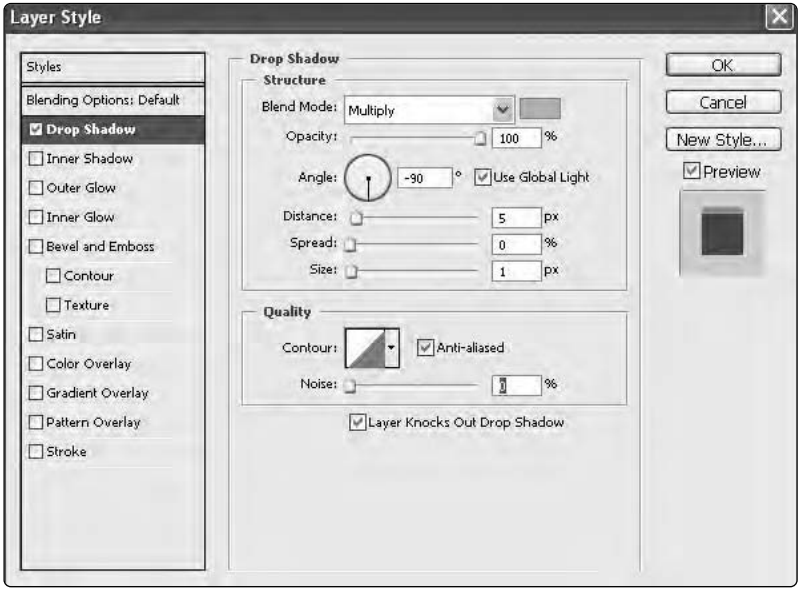
Ardından yeni bir katman oluşturup **Rectangular Marquee Tool** aracı ile resimdeki gibi sitenin ana gövdesini seçin.



Daha sonra **Gradient Tool** aracını ve gradient efektinde kullanacağınız renkleri seçip, seçim alanında **SHIFT** tuşuna basılı tutarak imleci yukarıdan aşağıya doğru sürükleyerek efekti resimdeki gibi oluşturun.



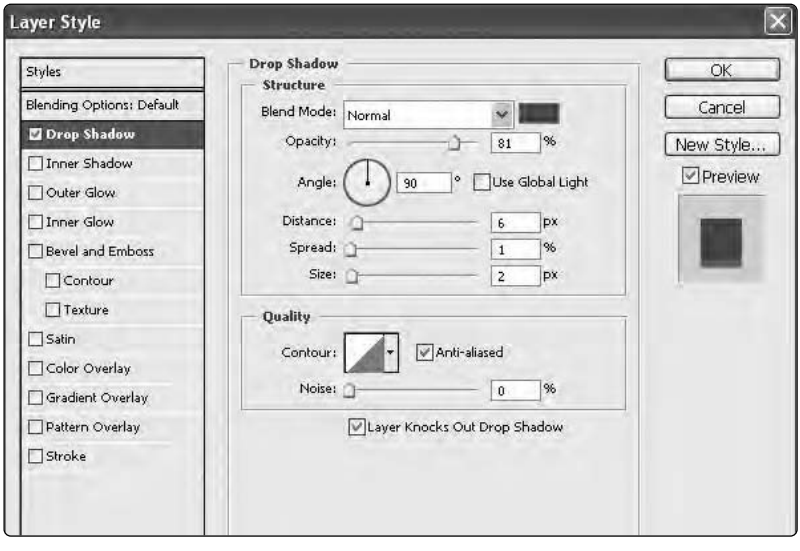
Bir sonraki aşama olarak yeni bir katman oluşturup sitenin en altında yer alacak footer kısmını da yine **Rectangular Marquee Tool** aracı ile seçip, seçimi istediğiniz bir renk ile doldurun.



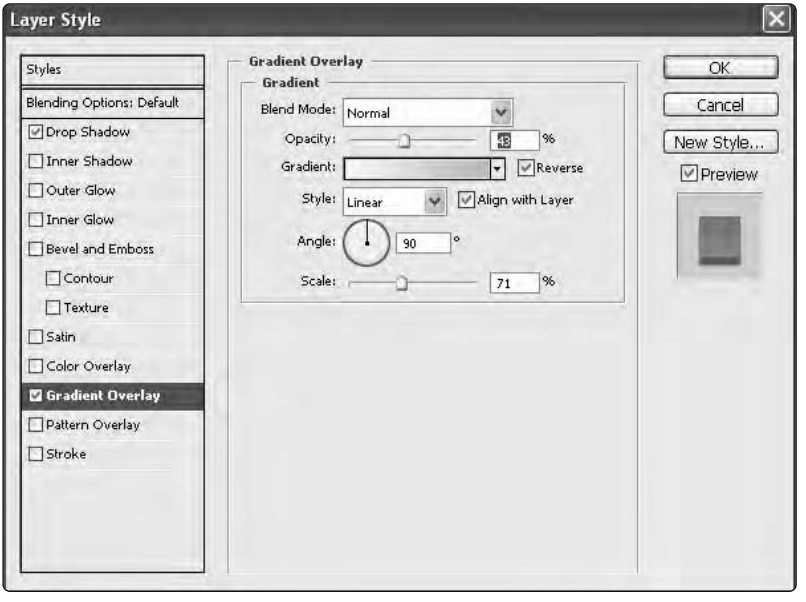
Dilerseniz footer kısmının daha belirgin ve estetik görünmesi için çeşitli efektler uygulayabilirsiniz. Biz burada örnek olarak resimdeki gibi **Drop Shadow** uyguladık.



Şimdi bir katman daha oluşturun ve üstte menülerin yer alacağı bölümü de seçip istediğiniz bir renkle doldurun.



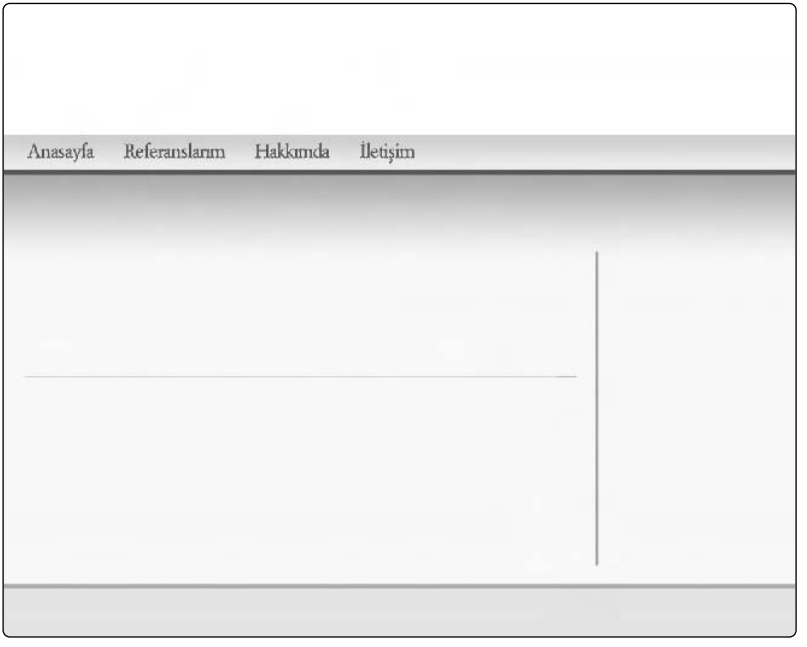
Menü çubuğunu da ana gözdeden ayırmak için **Drop Shadow** efekti uygulayın.



Menü çubuğuna bir de gradient efekti vermek için resimdeki gibi **Gradient Overlay** uygulayın.



Menünün son hali resimdeki gibi olmalıdır. Şimdi yeni bir katman daha oluşturup seçeceğiniz **Brush Tool** aracı ve ince uçlu fırçalarla istediğiniz renkte, resimdeki gibi sayfayı bölümleyen çizgiler çizin. Düz çizgiler çizebilmek için aynı anda **SHIFT** tuşunu da basılı tutmalısınız.



Artık resimdeki gibi menüye, gerekli menü öğelerini **Horizontal Type Tool** aracı ile istediğiniz fontu kullanarak yazabilirsiniz.



Sıra **header** kısmını oluşturma geldi. Tasarımın üst kısmında boş kalan alanı seçip istediğiniz bir renkle doldurun. **Horizontal Type Tool** aracı ile istediğiniz font ile site adını girin.

Ayrıca header'a daha önce öğrendiğimiz bilgilerin ışığında resim, fırça ya da filtre ekleyebilirsiniz. Ben örnek olması için sadece bir metin ve bir fırça kullandım.



Son aşama olarak da sitenin gövdesine birkaç resim ve metin ekleyelim. Alacağınız sonuç resimdekine benzer olacaktır. Siz değişik ebatlarda resimler ekleyerek ya da sütun sayısını değiştirerek daha özgün tasarımlar oluşturabilirsiniz.

Artık önceki sayfalarda öğrendiğiniz gibi **Slice Tool** aracı ile resmi dilimleyebilir ve **File>Save for Web** komutunu kullanarak çalışmanızın HTML çıktısını elde edebilirsiniz.

10

FLASH CS4

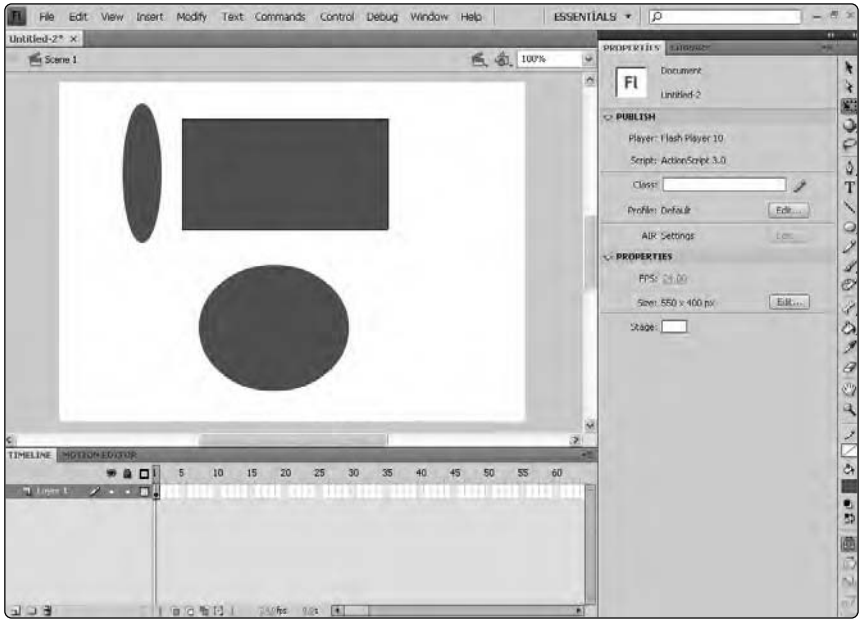
Çalışma Alanı 557
Çalışma Sahnesi 560
Dosya İthal Etme 563
Yayınlama Ayarları 564
Paneller 565
Tıme Line 571
Grafikler 576
Pencil Tool 576
Rectangle Tool & Oval Tool 577
Free Transform 579
Pen Tool 581
Seçim İşlemleri 582
Semboller 584
Yeni Sembol Oluşturma 584
Şekilleri Sembole Dönüştürme 585
Sembol Özellikleri 586
Semboller ve Time Line 588
Animasyonlar 588
Classic Tween 589
Şekil Animasyonları 593
Metinler 595
Metinlere Maske Uygulama 596
Ses & Video 597
Ses Dosyaları ile Çalışmak 597
Videolar ile Çalışmak 599
CS4 Yenilikler 600
Deco Tool 600
3D Rotation Tool 603
3D Translation Tool 603
Motion Editor Paneli 603
Motion Presets Paneli 605
Spray Brush Tool 607
Bone Tool 607

Adobe **Flash CS4** programı ile farklı türde animasyonlar üretebilirsiniz. Ayrıca web siteleriniz için gerekecek banner, buton... vb çalışmalardan tutun da sıfırdan bir web sitesi tasarlamaya kadar bir çok tasarım çalışmanızda da Flash programından yaralanabileceksiniz.

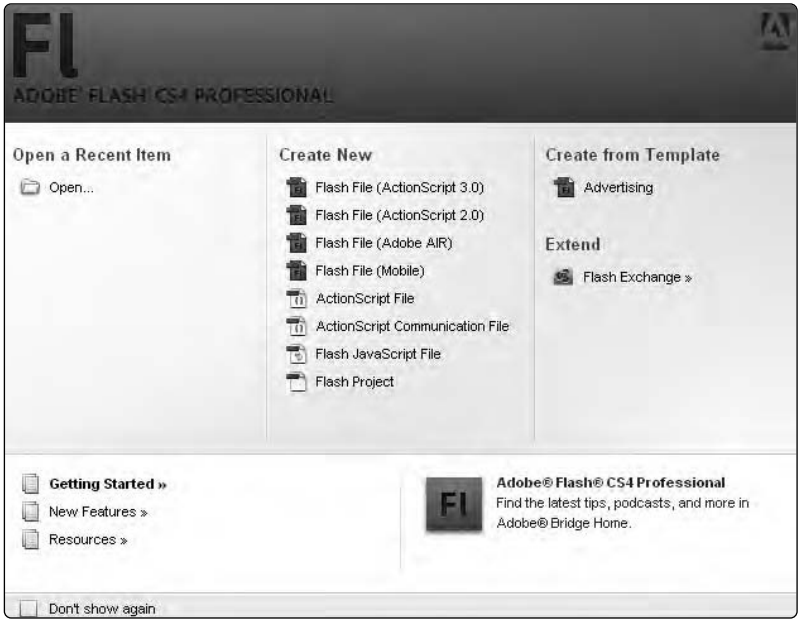
Detaylara geçmeden önce, şu ana kadar üç adet Adobe programı görmüş olduğumuzdan hareketle, söz konusu programlardan aşina olduğunuz benzer hususlara çoğu konuda hiç değinmeyeceğimi, bir çok konuda da kısaca değinerek, özellikle **Flash** programına has farklılıklar ve özellikler üzerinde duracağımı belirteyim. Böylece sıkıcı tekrarlardan ve kitabın gereksiz yere uzamasına sebep olmaktan da kurtulmuş olacağız.

Çalışma Alanı

Adobe **Flash CS4** programının çalışma alanı kitapta gördüğümüz diğer Adobe programlarıyla bir çok benzerlikler göstermektedir. **Flash CS4** ile de işlerimizi paneller, araç çubuğu, menüler... vd özellikleri kullanarak yapacağız.

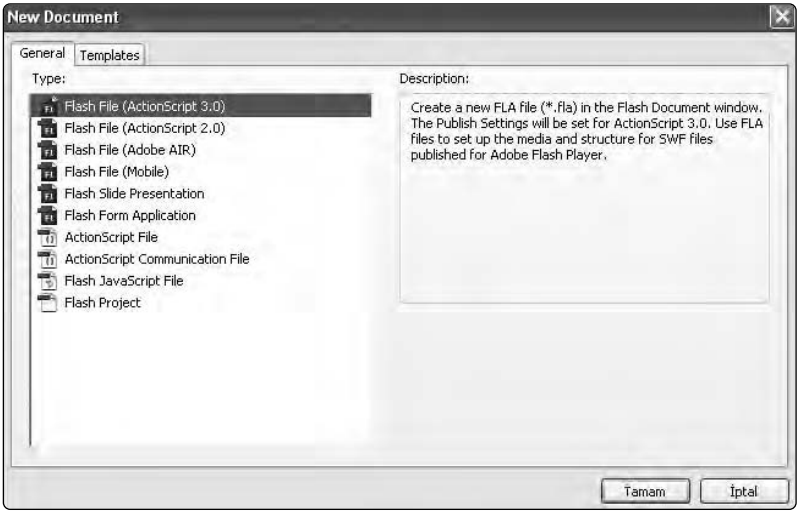


Çalışma alanını, **Animator**, **Classic**, **Debug Designer**, **Developer**, **Essential** seçeneklerinden her hangi biri ile kullanabilirsiniz. Burada hangi seçeneği tercih edeceğiniz daha çok yapacağınız çalışmanın içeriği ile ilgili olacaktır. Resimde **Essential** seçeneğini görüyorsunuz. Ayrıca gerektiği durumlarda **New Workspaces**... ile kendinize özel bir çalışma alanı da hazırlayabilirsiniz.



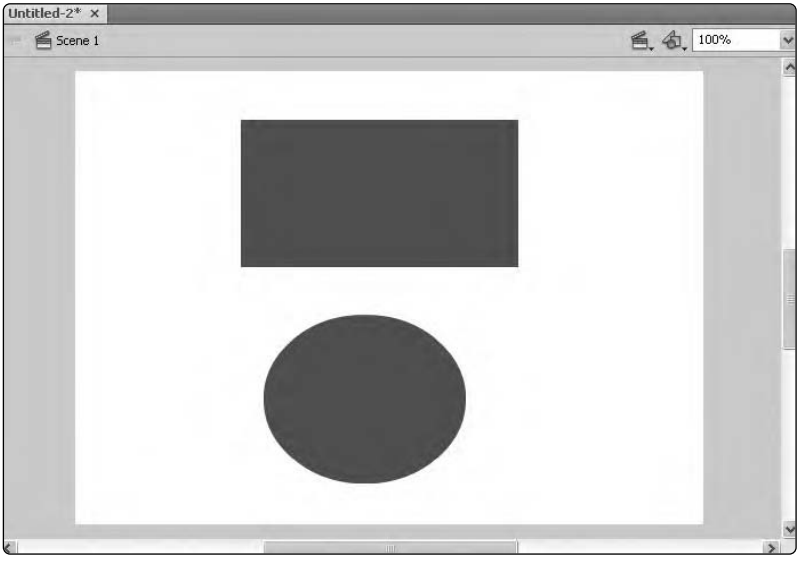
Flash CS4'ü de ilk açtığımızda bizi hoş geldin ekranı ile karşılar. Ekranın en solunda son kullanılan belgeleri açmanızı sağlayan **Open a Recent Item** bölümü, orta bölümde ise yeni bir belge (*Create New*) oluşturabilmenizi sağlayan seçenekler yer alır.

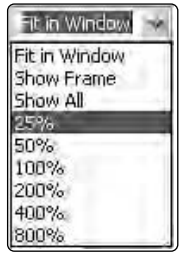
Sağ bölümden belge oluşturmanızı hızlandıran Flash programındaki hazır şablonlara ulaşabilirsiniz. **Extend** bölümü ise Flash programına ekleyebileceğiniz bazı uygulamaları içeren Adobe site linkine ulaşmanızı sağlamaktadır.



Artık ekranın orta bölümünden **Flash File** (*ActionScript 3.0*) seçeneği ile yeni bir belge açabiliriz. Dilerseniz yeni belge açma işlemini **File>New** menüsü yardımıyla açılan pencereden istediğiniz belge tipini ya da şablonu seçerek de yapabilirsiniz.

ÇALIŞMA SAHNESİ

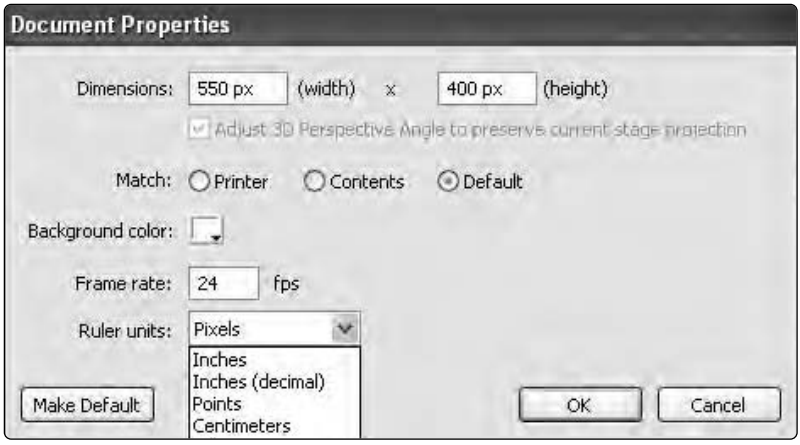




Flash programında çalışmalarımızı yapacağımız alanı sahne olarak adlandırıyoruz. Sahneyi kullanırken, gördüğümüz diğer Adobe programlarında olduğu gibi alanı yaklaştırıp uzaklaştırabilir, daha hassas çalışmalar yaparken ızgara, kılavuz ve cetvelleri kullanabilirsiniz.

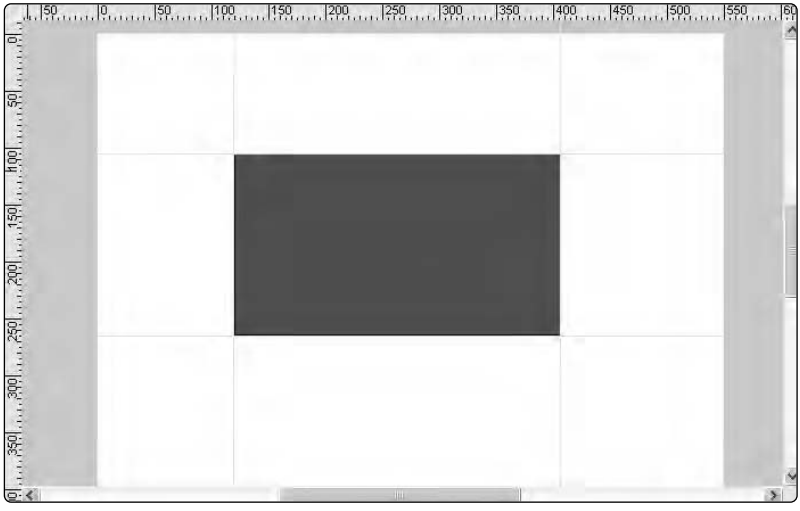
Yaklaştırma ve uzaklaştırma konusunda **Zoom Tool** aracını kullanabileceğiniz gibi sahnenin sağ üst kısmında yer alan ve resimde gördüğünüz açılır pencereden de yararlanabilirsiniz. Açılır menüdeki **Fit in Window** seçeneği ile sahnenin tümünü kaydırma çubukları olmadan görüntüleyebilirsiniz. **Show Frame** ile de karenin tamamını kaydırma çubukları ile birlikte görebilirsiniz. Sahnenin dışında kalan bölümleriyle beraber görünmesi için **Show All** seçeneğini tercih etmelisiniz.

CETVELLERİ KULLANMA



Cetvelleri görünür hale getirmek ya da gizlemek için **View>Rulers** menüsünü, cetvellerde kullanılacak ölçü birimi olarak da `milimetre`, `santimetre`, `inç` ya da `piksel` kullanabilirsiniz. Bu seçimi **Modify>Document**... menüsü ile açılan **Document Properties** penceresi yardımıyla yapabilirsiniz.

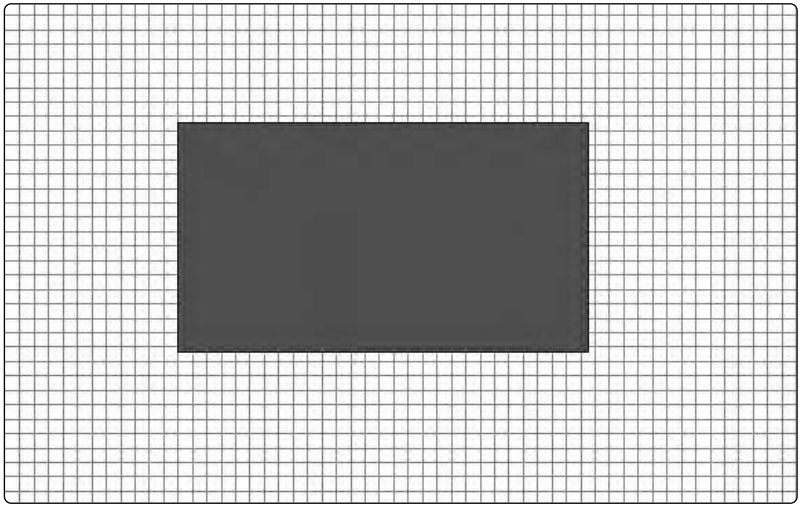
Kılavuzları Kullanma



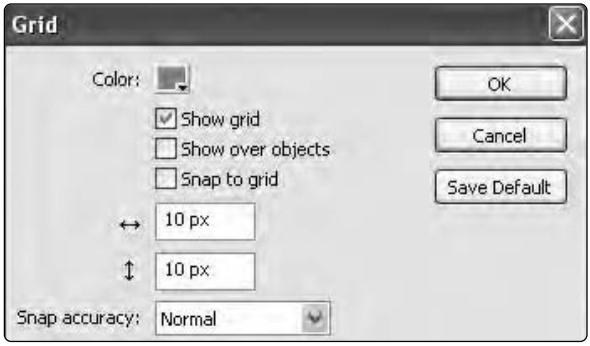
Cetvellerden fare ile sürükleyerek kılavuzları sahnenize taşıyabilirsiniz. Kılavuzları kaldırmak için de ilk işlemin tersini yapın. Yani kılavuzları cetvellere doğru sürükleyin.

Kılavuzları kilitlemeniz gerektiğinde ise **View>Guides>Lock Guides** menüsünden yaralanabilirsiniz. Kılavuzları temizlemek için de **View>Guides> Clear Guides** menüsünü kullanabilirsiniz.

Izgara Kullanma



Izgaralar sahnede nesnelerin arkasında düz çizgiler şeklinde görünürler. Izgaraları, **View>Grid>Show Grid** menüsü yardımıyla açabilirsiniz.



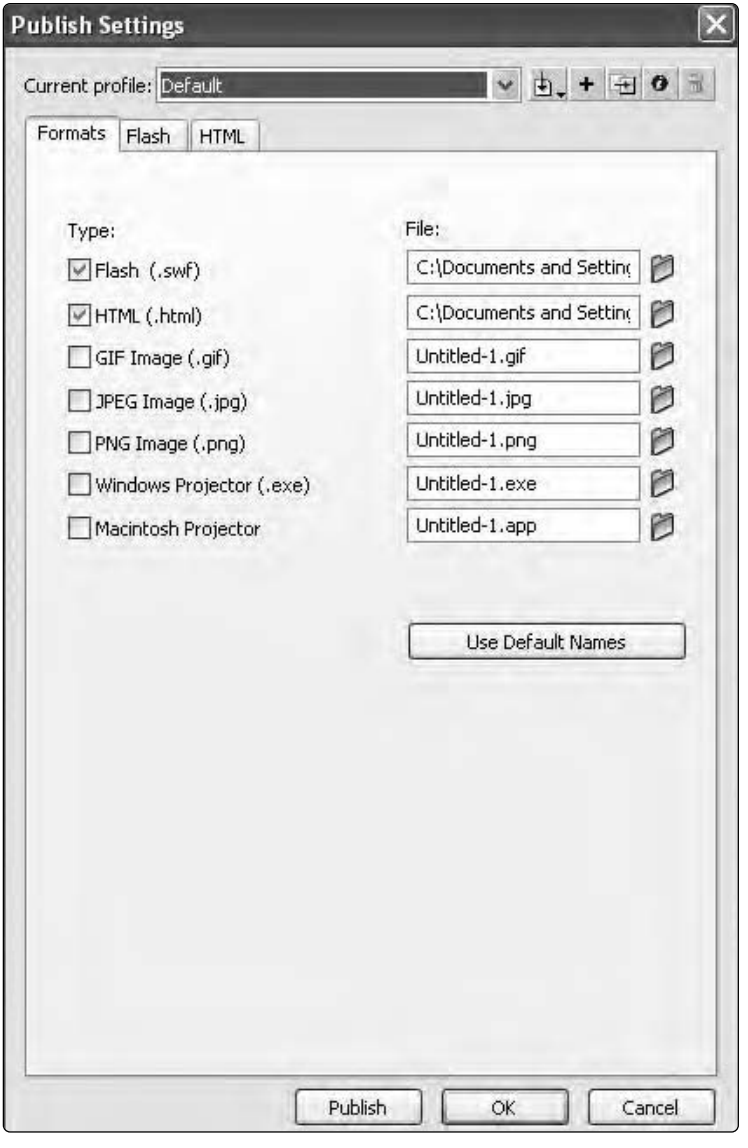
View>Grid>Edit Grid menüsü yardımıyla da ızgaraları düzenleyebileceğiniz **Grid** penceresini açabilirsiniz.

Dosya İthal Etme

Çalışmanıza dışardan dosya dahil etmeniz gerektiğinde bunu **File>Import>Import to Stage** menüsü ile yapabilirsiniz. Belgeyi kütüphaneye dahil etmek içinse **File>Import>Import to Library** menüsü kullanabilirsiniz.

Yayınlama Ayarları

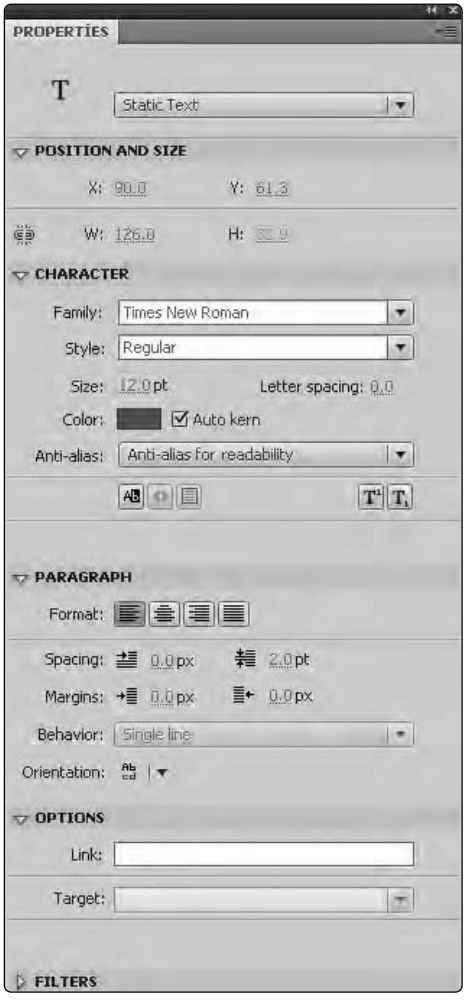
Yapmış olduğunuz çalışmayı tamamladığınızda, öncelikle tarayıcılarda nasıl göründüğünü test edebilirsiniz. Bunun için **File>Publish Preview>Default** menüsünden yararlanabilirsiniz.



Kontrollerinizi tamamladığınızda, çalışmanızı yayınlamak için gereken birkaç ayarlamayı yapmak üzere **File>Publish Settings** menüsü yardımıyla **Publish Settings** penceresini açabilirsiniz. Pencerenin resimde görünen ilk sekmesinde çalışmanızda yer alan ve yayınlanacak dosya türleri ile dosyaların aktarılacağı konumlar yer almaktadır. Diğer tablarda ise ilgili dosya formatları için gereken ayarlamalar bulunmaktadır.

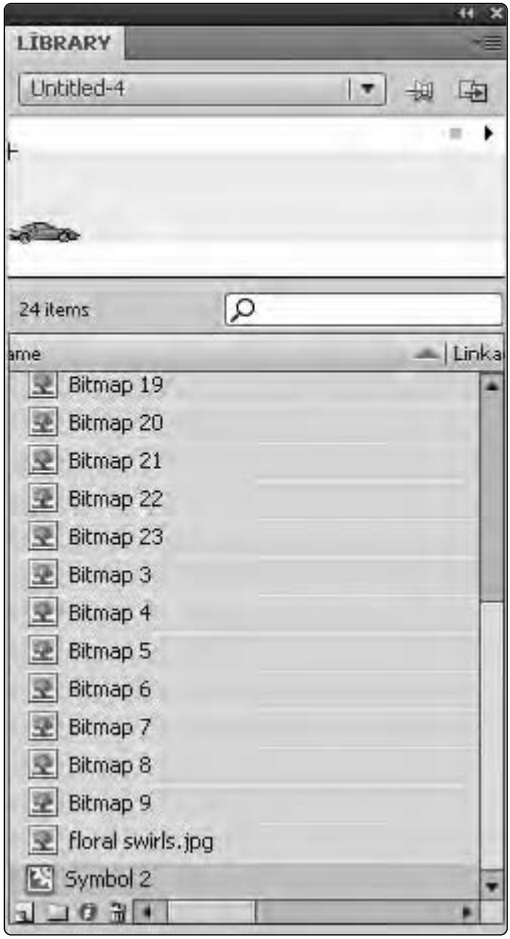
PANELLER

PROPERTIES PANELİ



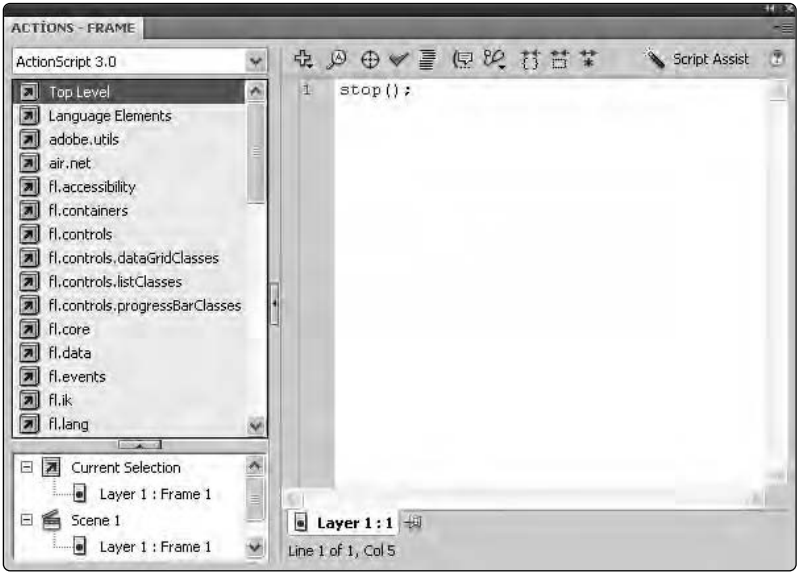
Daha önce gördüğümüz Adobe programlarından tanıdığınız **Properties** panelinin mantığı, çalışmanızla ilgili işlemleri yaparken, gerekli menü ya da panellere gitmeden daha hızlı bir biçimde çalışmanızı sürdürebilmenizi sağlamaktır. Bu amaçla, aktif olan işlem ile ilgili en sık kullanılan özelliklerin bir araya toplar. Paneldeki özellikler sabit olmayıp yaptığınız işleme göre değişecektir. Örneğin resimde, bir metin çalışması esnasında panelin sahip olduğu özellikleri görmektesiniz.

LIBRARY PANELİ



Library paneli için, **File>Import** menüsü ile içe aktardığınız bitmap, ses ya da video dosyaları ile Flash'ta oluşturduğunuz sembollerin saklandığı ve yönetildiği kütüphanedir diyebiliriz. Bu panel aracılığı ile panele dahil ettiğiniz öğeleri sıralayabilir, öğelerin belgenizdeki kullanım sıklığını takip edebilir, panelde arama yapabilir ya da öğelerin ön izlemesini görebilirsiniz. **Library** panelindeki bir öğeyi sahnenize dahil etmek için söz konusu öğeyi sürükleyip sahnenize bırakmanız yeterlidir.

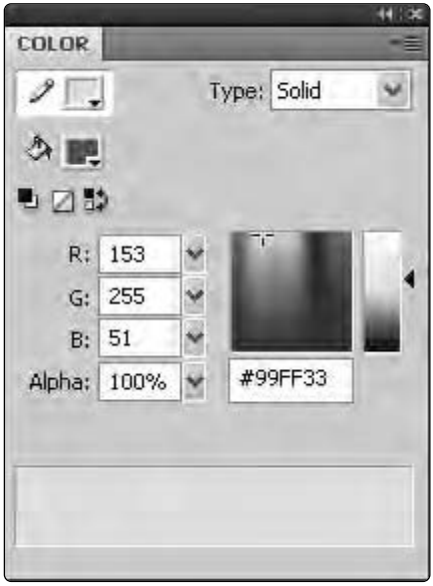
Actions Paneli



Çalışmalarınıza ActionScript kodları eklemek için kullanacağınız paneldir. Kod ekleyeceğiniz kareyi ya da nesneyi seçtikten sonra bu panel yardımıyla kodlarınızı oluşturabilirsiniz. Panelde editör yardımı ile kodlar bizzat elle yazılabileceği gibi, Flash'ın uzman olmayan kullanıcılar için hazır olarak sunduğu kodları panelden seçerek de editöre otomatik olarak yazdırabilirsiniz.

Yeri gelmişken kısaca ActionScript'ten bahsedelim. ActionScript, Flash programında kullanılan ve kendine has yazım kuralları olan bir kodlama dilidir. ActionScript yardımıyla Flash programında dinamik uygulamalar geliştirebilirsiniz.

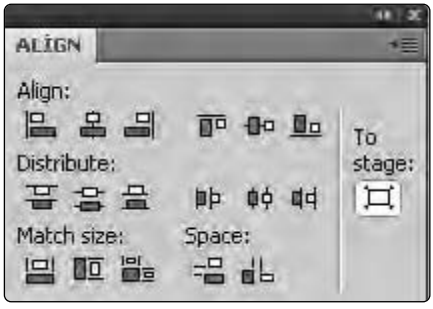
Color Paneli



Bu panel aracılığı ile sahnenizde oluşturacağınız nesnenin **çerçeve rengini** (*stroke color*) ve **dolgu rengini** belirleyebilirsiniz. **Black and white** tuşuna tıklayarak bir çırpıda çerçeve rengini siyaha dolgu rengini ise beyaza çevirebilirsiniz. **No color** tuşu ile de nesnenin bir çerçeve rengine sahip olmamasını sağlayabilirsiniz. **Swap color** tuşu ile çerçeve rengi ile dolgu rengini birbirleri arasında değiştirebilirsiniz. **Type** açılır kutucuğundan dolgu renginin tipini belirleyebilirsiniz.

Paneldeki renk paleti yardımıyla farklı renkler oluşturup kullanabileceğinizi ise siz zaten biliyorsunuz.

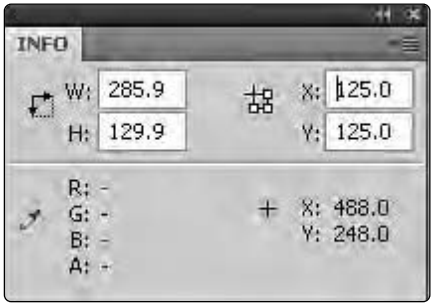
Align Paneli



Bu panel size sahnedeki nesnelerin konumlarını ayarlama imkanı verir. Nesnenin konumunu ayarlayabilmek için öncelikle **Selection Tool** aracı ile nesneyi içine alan bir seçim alanı oluşturmanız gerekir. Bir başka ifade ile nesnenin seçili olması gerekir.

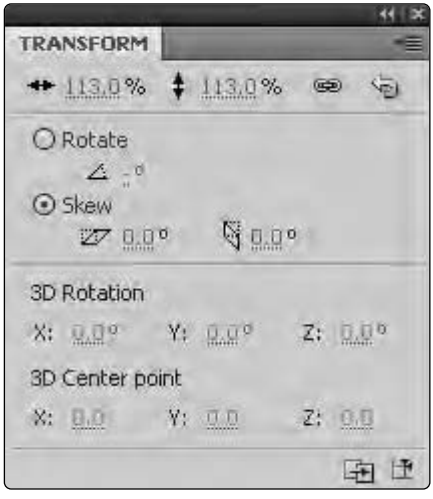
Öncelikle paneldeki **To stage** tuşuna basarak hizalamaların sahne içinde olmasını sağlayın. Ardından zaten aşina olduğunuz hizalama tuşları yardımıyla nesnenizi istediğiniz gibi konumlandırabilirsiniz.

Info Paneli



Info paneli ise bildiğiniz gibi nesnenizle ilgili bazı temel bilgileri içermektedir. Bu panel yardımıyla nesnenizin yükseklik ve genişlik değerlerini, ayrıca da nesnenin sahne üzerindeki koordinatlarını hızlı bir biçimde değiştirebilirsiniz.

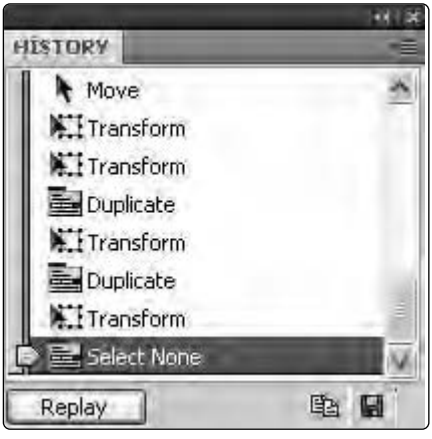
Transform Paneli



Transform panelinden ise nesnenin ebatlarını yüzdesel olarak değiştirebilirsiniz. **Constrain** tuşu yardımıyla da ebat ayarlama işlemi sırasında nesnenin en boy oranının korunup korunmayacağını belirleyebilirsiniz.

Panel yardımıyla ayrıca nesne üzerinde döndürme ve eğme işlemleri de yapabilirsiniz.

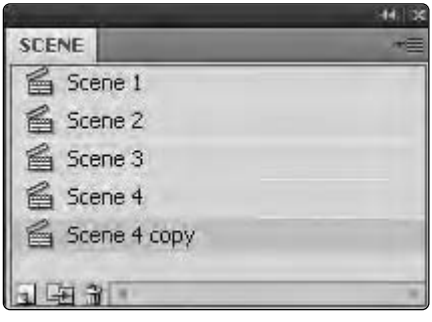
History Paneli



History paneli de diğer Adobe uygulamalarından bildiğiniz bir başka paneldir ve Flash programında yapacağınız uygulamalarda da yine size çok faydası dokunacaktır.

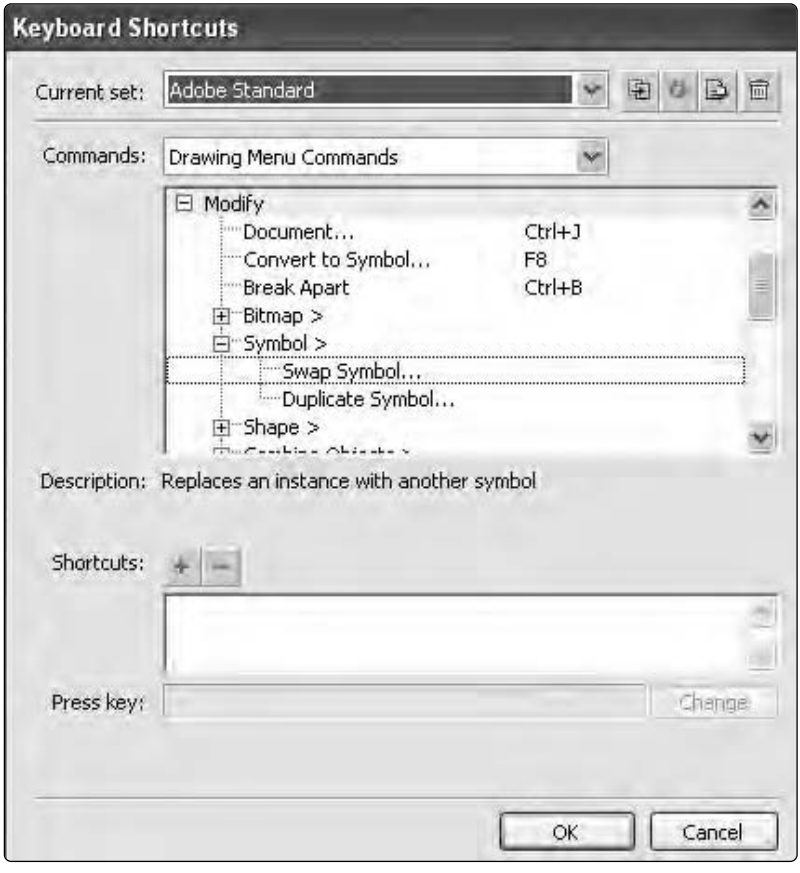
Çalışmalar esnasında oluşabilecek olası hatalardan kurtulmak ya da bir başka sebeple çalışmanın önceki aşamalarına dönmek bazen hayat kurtarıcı olabilir!

Scene Paneli



Scene (*sahne*) paneli ile Flash belgenizin kaç sahneden oluştuğunu görebilir ve sahneleri kontrol edebilirsiniz.

KLAVYE KISAYOLLARI

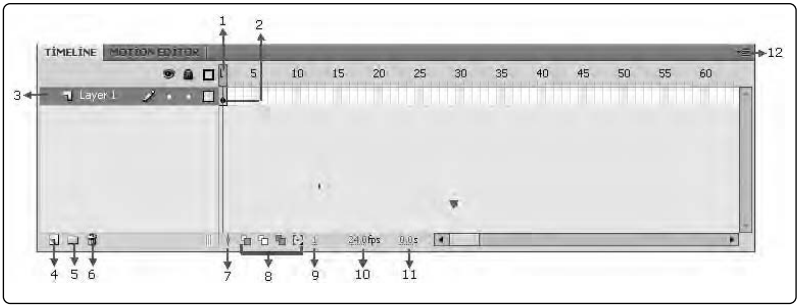


Klavye kısayolları oluşturmak için **Edit>Keyboard Shortcuts menüsü** ile açılan **Keyboard Shortcuts** penceresinden yaralanabilirsiniz. Tahmin edeceğiniz gibi bu pencere yardımıyla mevcut kısa yolları değiştirebilir, yeni kısa yollar da oluşturabilirsiniz.

TIME LINE

Time Line kavramını Türkçe'ye zaman çizelgesi olarak çevirebiliriz. Peki Time Line panelini ne amaçla kullanacağız? Bu soruyu kısaca çalışmalarımızı: **frame** (*kare*), **keyframe** (*anahtar kare*) ve **blank keyframe** (*boş anahtar kare*) ve **katmanlar** yardımıyla yönetmek için kullanacağız, şek-

linde cevaplayabiliriz. Çalışmanızda yer alacak her bir **Layer** (*katman*) kendine ait bir zaman çizelgesine sahip olacaktır.



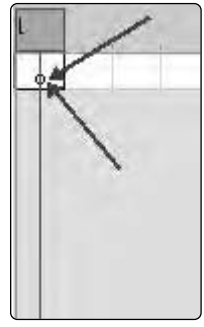
1. Oynatma kafası **2.** Seçili kare **3.** Seçili layer **4.** Yeni katman oluşturma tuşu **5.** Yeni layer grubu oluşturma tuşu **6.** Grup ya da layer silme tuşu **7.** Oynatma kafasına kadar kaydırma tuşu **8.** Çoklu kare görünümü tuşları **9.** Geçerli kare göstergesi **10.** Kare hız göstergesi **11.** Geçen süre göstergesi **12.** Kare görünüm tipi açılır menüsü tuşu

Resimde üst kısmında `1, 5, 10, 15` diye devam eden numaralar gördüğünüz her bir alan o katmandaki bir kareyi ifade eder. Yani Flash programı ile hazırlayacağınız çalışmanızın her bir katmanının toplam kaç kareden oluştuğunu gösterir. Bu panel yardımıyla sadece çalışmadaki katmanların kare sayısını değil aynı zamanda da karelerin ardı ardına geliş hızlarını ve sürelerini de kontrol ederiz. Yani belli bir zaman aralığında adeta bir film şeridi gibi ne kadar karenin gösterileceğini belirleriz. Resimde 1 numara ile gösterdiğim oynatma kafası, o anda görüntülenen kareyi gösterir.

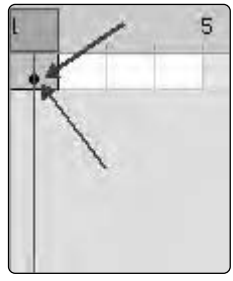
Kare gösterim hızının (*frame rate*) mesela `12 fps` olması, saniyede `12` adet kare görüntüleneceği anlamına gelmektedir. Bir başka ifade ile her kare ekranda 1/12 saniye kalır ve bu süre sonunda oynatma kafası bir sonraki kareye geçer. Hazırladığınız çalışmayı kontrol ederken oynatma kafası Time Line boyunca soldan sağa doğru hareket edecektir.

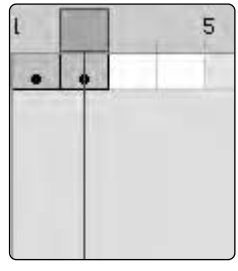
Oynatma kafası hareket ederken panelin alt kısmında bulunan durum göstergesi size, kare numarası ve kare hızı ile ilgili olarak geçerli kare hakkındaki bilgileri görüntüleyecektir.

Keyframe (*anahtar kare*) farklı içeriklere sahip kareleri ifade eder. **Frame** (*kare*) ise **Keyframe**'lerin arasını dolduran ve değişiklik içermeyen karelerdir.



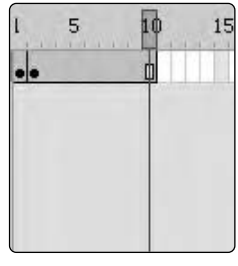
Flash ile bir belge açtığımızda resimde gördüğünüz gibi ilk anahtar kare program tarafından oluşturulur. Dikkat ederseniz anahtar karenin ortasındaki küçük dairenin içi boştur. Bu, anahtar kerenin (henüz herhangi bir işlem gerçekleştirmediğimiz için) bir **blank keyframe** (*boş anahtar kare*) olduğunu gösterir.





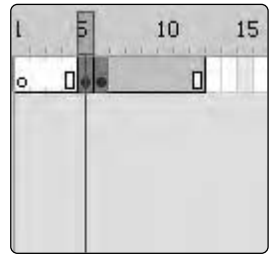
Şimdi sahnemize **Rectangle Tool** aracı ile bir dikdörtgen çizelim ve karemize bakalım. Resimde gördüğünüz gibi bu sefer dairenin içi doldu. Yani biz bir değişiklik yaptık ve karemiz keyframe bir kare olmuş oldu.

Şimdi karenin üzerinde iken farenin **SAĞ** tuşunu tıklayın. Bu takdirde bir menü açılacaktır. Açılan menüden **Insert Keyframe** komutu ile yeni bir keyframe oluşturun. Oynatma kafası ikinci **keyframe** üzerinde iken **Selection Tool**'a aracı ile, biraz evvel sahnemizde çizdiğimiz dikdörtgeni sahne üzerinde bir başka konuma taşıyın.



Şimdi fare ile ilk kareye tıkladığınızda dikdörtgenin ilk konumunda, tekrar ikinci kareye tıkladığınızda ise karenin taşıdığımız konumda olduğunu göreceksiniz. Bunun anlamı, bir **keyframe**'de yapacağınız değişikliğin diğer **keyframe**'i etkilemeyeceğidir.

Şimdi de **Time Line** üzerindeki `10` numaralı kareye sağ tıklayıp açılan menüden bu sefer **Insert Frame** komutunu seçelim. Bu takdirle `2` numaralı keyframe ile `10` numaralı **frame** arası **frame**'ler ile doldurulacaktır.



Fakat bu kareler keyframe olmadığından oynatma kafasını hareket ettirdiğinizde `2.` kare ile `10.` kare arasında, çalışmanızda her hangi bir değişiklik olmadığını göreceksiniz. Seçili katmanın zaman çizelgesine yeni kareler eklemenin bir diğer yolu da eklenecek son kareyi tıklayıp **`F5`** tuşuna basmaktır. Örneğin `12.` kareyi tıklayıp **`F5`** tuşuna basarsanız `10.` karedeki oynatma kafası `2` kareyi daha çizelgeye dahil ederek `12.` kareye gelecektir.

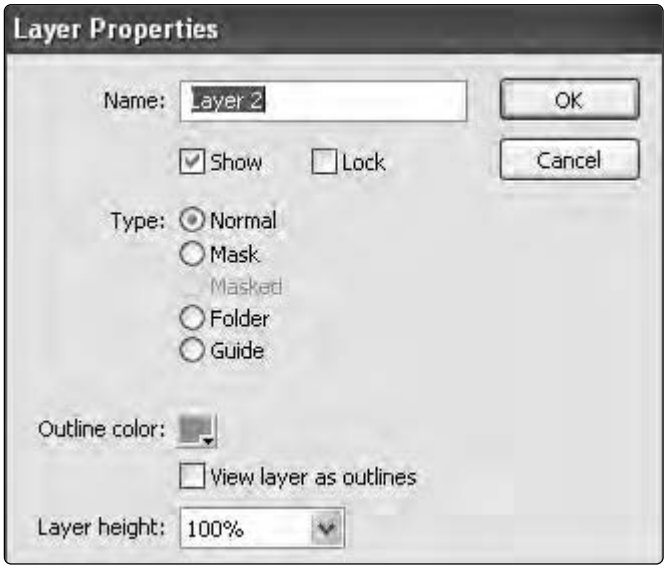
Bu aşamada bir de karelerin nasıl taşınabileceğini görelim. İlk olarak `1.` anahtar kareyi sonra da **`SHIFT`** tuşuna basılı tutarak `2.` anahtar kareyi seçin. Ardından iki anahtar kareyi sürükle bırak yöntemiyle istediğiniz karelere taşıyın. Bu takdirde resimde de gördüğünüz gibi zaman çizelgesinin baş tarafındaki kareler **blank keyframe** (*boş anahtar kare*) olacaktır.

Yani oynatma kafası ile bu karelere geldiğinizde artık sahnede çizdiğiniz dikdörtgeni göremeyeceksiniz. Resimdeki örneğimizden hareketle konuşacak olursak, oynatma kafası `1.` ile `4.` boş ana kareler arasında iken sahnede çizdiğimiz dikdörtgen görünmeyecektir. Fakat `5.` ana kareye geldiğinizde dikdörtgenimizi çizdiğimiz ilk konumunda görebileceğiz.



Oluşturduğunuz bu boş ana kareleri silmeniz gerektiğinde ise ilk olarak `1.` boş ana kareyi ardından **`SHIFT`** tuşuna da basılı tutarak `4.` boş ana kareyi seçin. Sonra, `4.` boş ana kare üzerinde iken fare ile sağ tıklayıp menüyü açmanız ve **Remove Frames** komutuna tıklamanız yeterli olacaktır.

Time Line panelindeki karelerin görünümünü değiştirmek için bir önceki resimde `12` numara ile gösterilen tuşa tıklayın. Açılan pencereden **Tiny**, **Small**, **Normal**, **Medium** ya da **Large** seçeneklerinden birini seçerek kare görünümünü büyültüp küçültebilirsiniz. Kare hücrelerinin satır yüksekliğini azalmak istediğinizde ise **Short** seçeneğini seçmelisiniz.

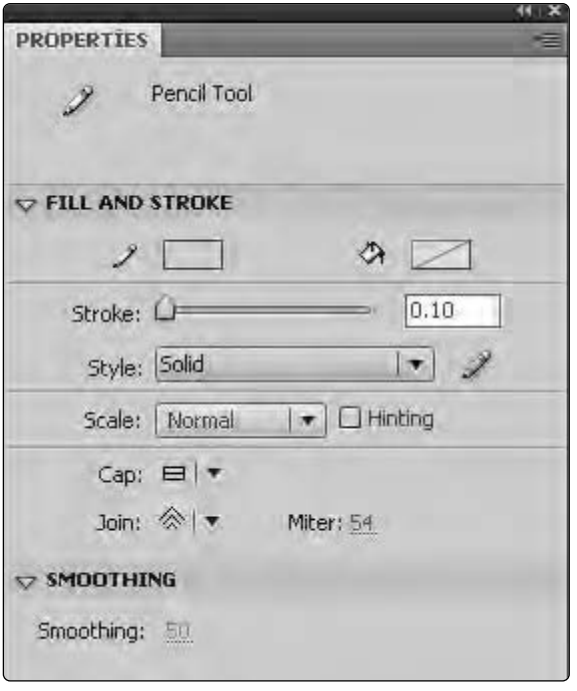


Bir de, **Time Line** panelindeki katmanların yüksekliklerini ve katmanların isimlerini nasıl değiştirebileceğimize bakalım. Bunun için yüksekliğini değiştireceğiniz katmanın sol tarafındaki küçük katman simgesine çift tıklayın.

Açılan **Layer Properties** penceresinden **Layer height** açılır kutucuğu yardımıyla katman yüksekliğini değiştirebilirsiniz. Bir katmanın ismini değiştirmek için, katman isminin üstüne çift tıkladığınızda katman isminin değiştirilebilir hale geldiğini göreceksiniz. Bu durumda imleç yanıp sönerken katmanın mevcut ismini silip yeni ismi girebilirsiniz.

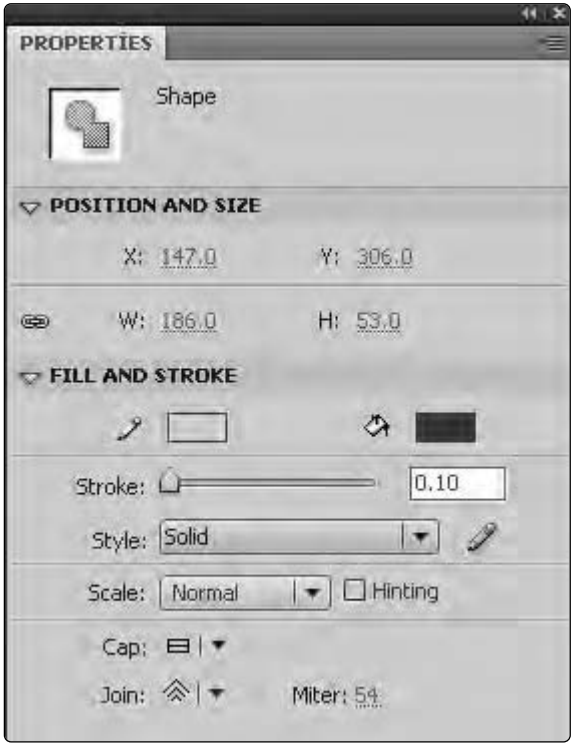
GRAFİKLER

PENCIL TOOL



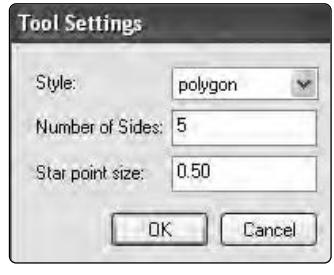
Pencil Tool aracı aynen ismi gibi kurşun kalem tarzı çizimler yapmanıza olanak tanır. Resimde bu araca ait **Properties** panelini görmektesiniz. Panel yardımı ile **Style** açılır kutucuğundan farklı çizgi sitilleri belirleyebilirsiniz. **Stroke** sürgüsü yardımıyla çizgi kalınlığını ve çizgi rengini de hızlıca ayarlayabilirsiniz.

Rectangle Tool & Oval Tool



Rectangle Tool & Oval Tool araçları da sizin bildiğiniz araçlar. Hatırlayacağınız gibi bu araçları dikdörtgen, kare, oval ve daire çizmek için kullanmıştınız. Resimde gördüğünüz gibi grafiğinizi seçili hale getirdikten sonra **Properties** panelinde yer alan **Position and Size** bölümünden grafiğinizin ebatlarını ve sahne üzerindeki konumunu ayarlayabilirsiniz.

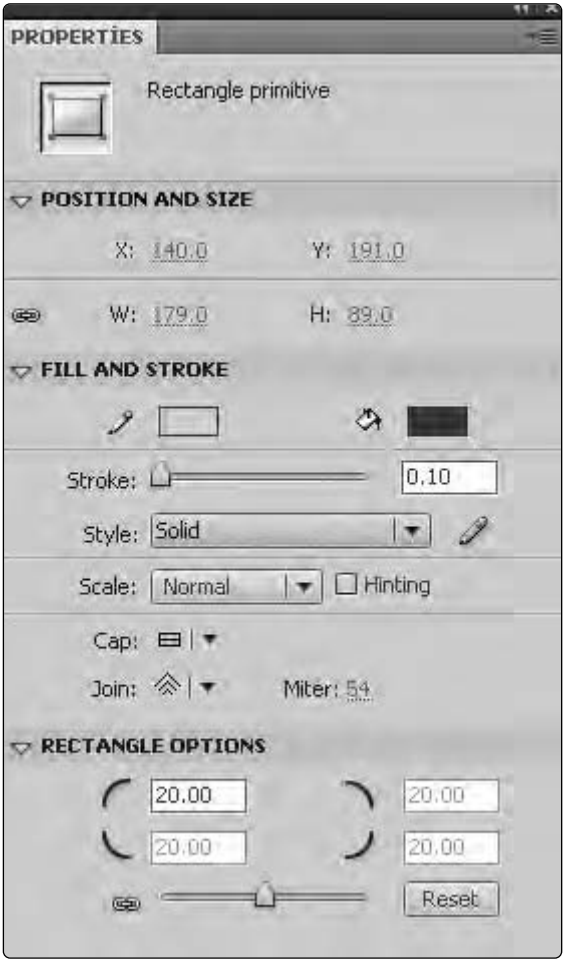
Aynı araç grubunda yer alan **PolyStar** aracı ile de çokgenler ya da yıldızlar çizebilirsiniz. Çokgen mi yıldız mı çizeceğinizi, araca ait **Properties** panelinin **Tool Settings** bölümündeki **Options**... tuşu ile açılan **Tool Settings** penceresinden seçebilirsiniz.



Yine **Properties** panelinden çizdiğiniz grafiklerin dolgu (*fill color*) ve çerçeve-kontur (*stroke color*) renklerini belirleyebilirsiniz.

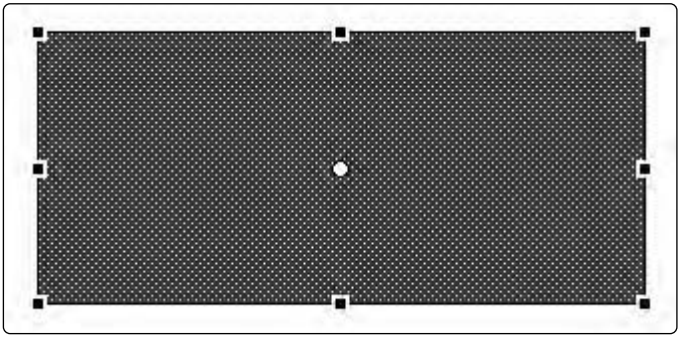


Çizdiğiniz bir şekil seçili değilken fare ile şeklin kenarına yaklaştığınızda imleç bir parantez işaretine dönüşecektir. Bu durumda oluşturduğunuz şeklin sınırlarından tutup sürükleyerek şeklini istediğiniz gibi değiştirebilirsiniz.



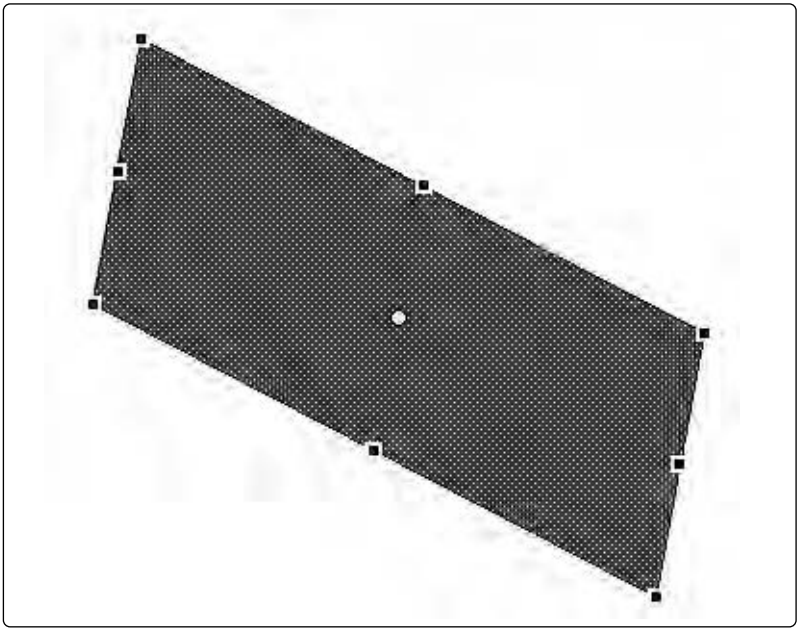
Flash programında farklı olarak ayrıca **Rectangle Primitive Tool** ve **Oval Primitive Tool** araçları yer almaktadır. Bu araçların köşelerindeki tutamaçlar yardımıyla ya da **Properties** panelini kullanarak köşelerine dışa ya da içe doğru açılandırma yapabilirsiniz.

FREE TRANSFORM

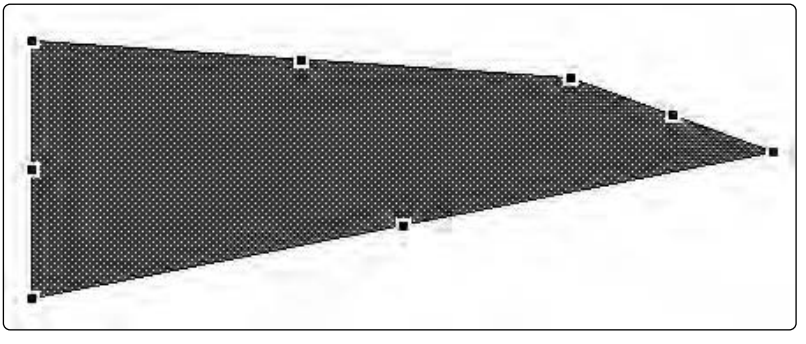


Free Transform aracı bir nesnenin boyutlarını küçültmek, büyütmek, eğmek ya da bükmek için kullanılır. Öncelikle sahnenize bir dikdörtgen çizin ve dikdörtgeni seçili hale getirin. Ardından **Free Transform** aracını tıklayın. Bu durumda resimde gördüğünüz gibi nesneni etrafında tutamaçlar belirecektir. Ve araçlar panelinin en alt kısmında bu araca ait özellikler de aktif olacaktır.

Şimdi **Scale** tuşuna tıklayalım. Bu özellik nesneyi x ve y ekseninde, yani yatay ve düşey olarak büyültüp küçültmek içindir. Bu özellik seçili iken tutamaçları sürükleyerek boyutları dilediğiniz gibi ayarlayabilirsiniz.

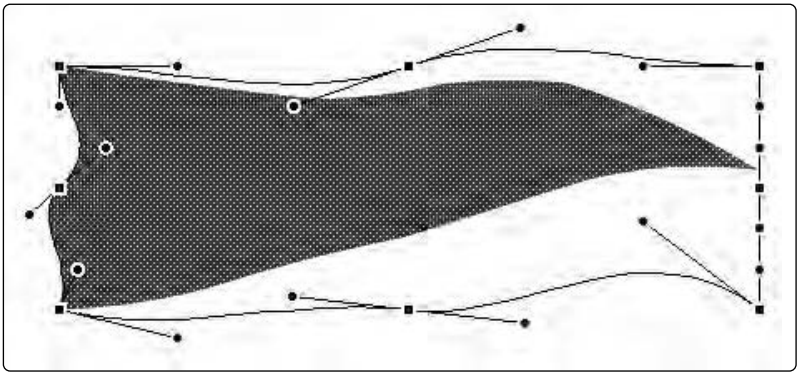


Nesneye yatay ve dikey eğim vermek içinse **Rotate and Skew** özelliğini seçmelisiniz. Boyutları değiştirmeden döndürme işlemi yapmak için imleci nesnenin bir köşesine getirin. Bu durumda imleç eğik ok işareti şekline dönüşecektir. Bu durumda köşe tutamacından tutarak nesneyi döndürebilirsiniz.

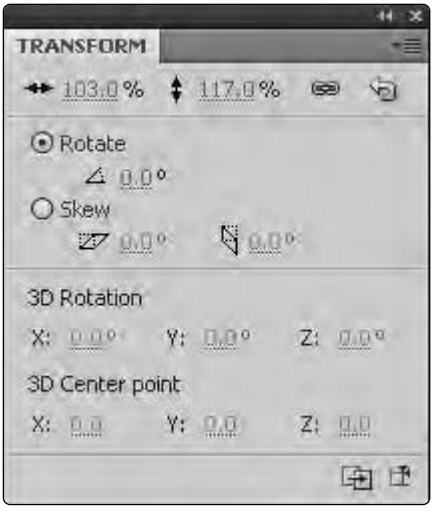


Nesnenin köşelerini diğer köşelerden bağımsız olarak eğmek için de **Distort** özelliğinden yararlanabilirsiniz. Bu durumda imleç içi boş

bir üçgene dönüşecektir. Artık eğmek istediğiniz köşeyi tutarak sürükleyebilirsiniz.



Daha detaylı bir dönüştürme işlemi yapmanız gereken bir durum söz konusu olduğunda bu sefer **Envelope** özelliğini seçmelisiniz. Bu sayede nesnenizi çok daha detaylı bir biçimde şekillendirebilirsiniz.



Bu işlemleri yürütürken, paneller bölümünde anlattığımız **Transform** panelinden yaralanabilirsiniz.

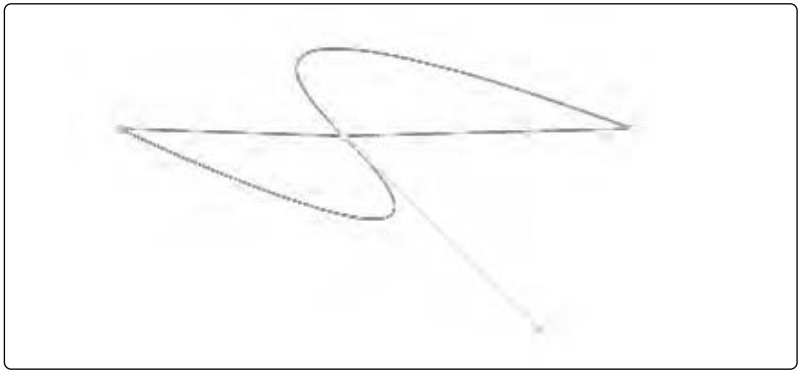
Paneli **Window> Transform** menüsü yardımı ile açabilirsiniz.

Pen Tool

Pen Tool aracı da sizin diğer programlardan tanıdığınız bir araç. Bu aracı Flash ile de vektörel çizimler yapmak için kullanırız. Aracı seçip sahne üzerinde bir noktaya tıkladıktan sonra bir başka noktaya daha tıklar-

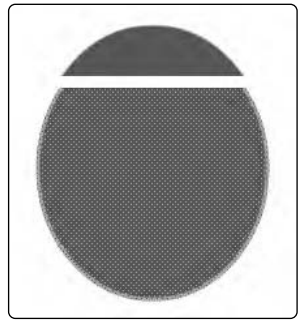
sanız, Flash bu iki nokta arasında düz bir çizgi oluşturacaktır. Tıklamaya devam edip çizgilerle bir üçgen, dikdörtgen ya da herhangi bir başka sekil oluşturduğunuzda şeklin tamamlanması için son olarak araç ile ilk tıkladığınız noktaya bir kez daha tıklayarak şekli kapatmanız gerekir.

İlk tıkladığınız noktayı net olarak anlayabilmeniz için bu noktaya geldiğinizde imlecin kenarında içi boş küçük bir daire belirecektir. Şekli kapattığınızda Flash şekli, bir kenar çizgisi ve bir dolgu rengi ile dolduracaktır.



Hatırlayacağınız gibi noktaların üzerine tıkladığınızda fareyi bırakmadan sürükleyerek çizgileri dilediğiniz gibi eğebilirsiniz.

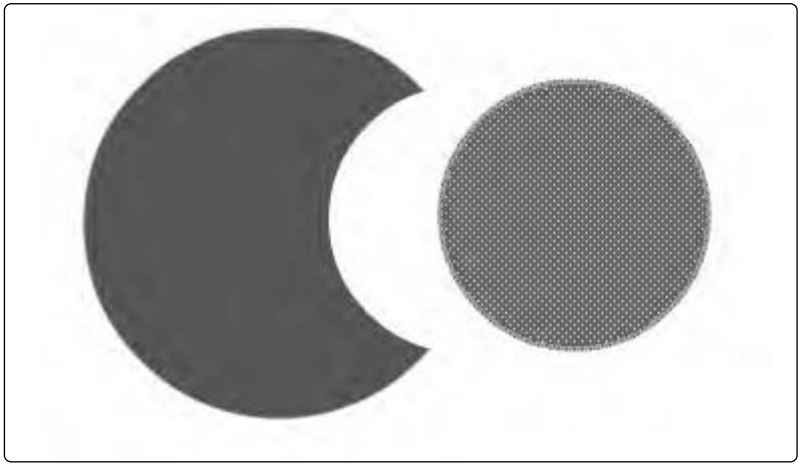
SEÇİM İŞLEMLERİ



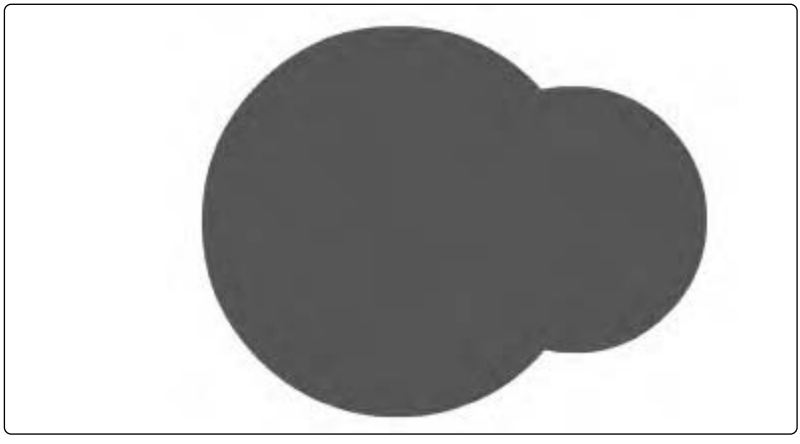
Bir şekli seçmek için Flash programında şekli **Selection Tool** aracı ile seçmelisiniz. Örneğin bir dikdörtgeni bu şekilde seçerseniz hem dolguyu hem de çerçeve-kontur kısmını seçmiş olursunuz. Bu seçim, şekli konturuyla birlikte taşımak için gereklidir.

Şeklin sadece bir kısmını seçtikten sonra taşırsanız, şekli parçalamış olursunuz. Şeklin dolgusuna bir kez tıklarsanız sadece dolguyu seçmiş olursunuz. Şeklin bir kenarına tıkladığınızda ise sadece o kenar seçilmiş olur. Tüm kenarların seçilmesi için bir **kenar çizgisi** (*kontur*) üzerine çift tıklamalısınız.

Seçim işlemini rasgele serbest bir seçim alanı oluşturacak şekilde **Lasso Tool** aracı ile de yapabilirsiniz. Seçimden sonra bu alanı taşıyabilir ya da **DELETE** tuşuna basarak silebilirsiniz. **Lasso Tool** aracı seçiliyken araçlar panelinin en alt kısmında beliren **Polygon Mode** ile serbest çokgen seçimleri de yapabilirsiniz.



Bir şekli (örn: daire) farklı renkteki bir diğer şeklin üzerine getirip bırakır ve sahnenin her hangi bir yerine tıklayarak şeklin seçili olma durumuna son verirseniz, ardın da tekrar çift tıklama ile şekli seçer ve sürüklerseniz diğer şeklin üzerine gelen kısım resimde gördüğünüz gibi kesilecektir.



Renklerin aynı olması ve kenar çizgilerinin bulunmaması durumunda ise kesme işlemi değil resimdeki gibi birleşme işlemi gerçekleşecektir.

SEMBOLLER

Semboller, çalışma esnekliği sağlamalarının yanı sıra daha düşük boyutlara sahip dosyalarla çalışma imkanı verirler. Bir sembol oluşturulduktan sonra, söz konusu sembolü projenizde dilediğiniz kadar kullanabilirsiniz. Her sembol kendine ait bir katmana ve zaman çizelgesine sahiptir.

YENİ SEMBOL OLUŞTURMA



Yeni bir sembol oluşturmak için **Insert>New Symbol** menüsünü kullanabilirsiniz. Komutu tıkladığınızda karşınıza resimdeki **Create New Symbol** penceresi gelecektir. Pencerede ilk yapabileceğiniz işlem, sembolünüze bir isim vermektir. **Type** bölümündeki açılır kutucuktan ise sembolün türünü seçebilirsiniz. Yeni sembol ya da sembol klasörü oluşturmak, bir sembolü silmek gibi işlemleri **Library** panelinden de yapabilirsiniz.

Pencereden seçebileceğiniz ilk tür olan **Movie Clip** en esnek sembol türüdür. Bu tür ile farklı animasyonlar oluşturabilirsiniz. **Button** türü, buton oluşturmak için kullanılır. Kendine has yapıda bir zaman çizelgesine sahiptir. **Graphic** sembol türünü ise animasyon gerektirmeyen yerlerde kullanabilirsiniz.



Movie Clip türünü seçtiğimizde, resimde gördüğünüz gibi belge çubuğunda oluşturduğumuz **sembol** (*Symbol 1*) **sahne adının** (*Scene 1*) he-

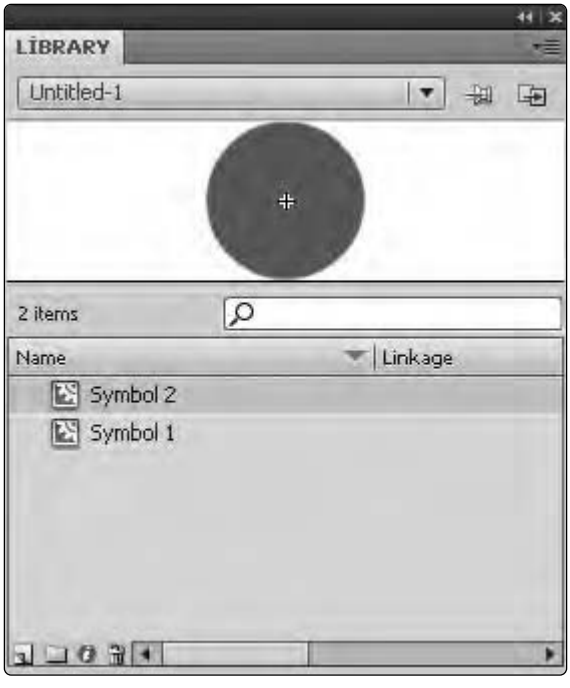
men yanında görüntülenmektedir. Sahne adının solunda yer alan ok işaretine tıkladığınızda sembolden çıkıp sahneye geri dönmüş olursunuz.

Bir sembol oluşturduğunuzda, oluşturduğunuz sembol kitaplığa da eklenecektir. Dolayısıyla sahneye döndükten sonra tekrar sembole geçiş yapmak için **Library** panelinden yaralanabilirsin. Yapmanız gereken panelde sembolün resmine çift tıklamaktan ibarettir.

Sahne ile sembol arasındaki geçişi belge çubuğunun sağ ucunda bulunan **Edit Scene** ve **Edit Symbol** tuşlarını kullanarak da yapabilirsiniz.

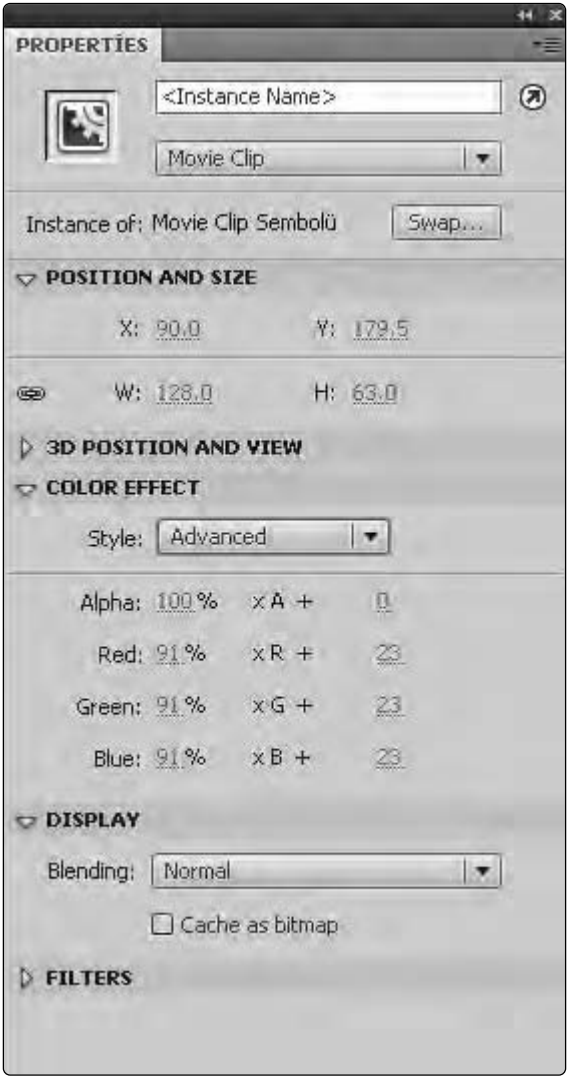
Şekilleri Sembole Dönüştürme

Yukarıda anlatılan şekilde sıfırdan yeni bir sembol oluşturabileceğiniz gibi, çizdiğiniz şekilleri de sonradan sembollere dönüştürebilirsiniz. Meseleyi bir örnekle ele almak üzere sahnenizde bir daire çizin ve daireyi seçili hale getirin. Ardından **Modify>Convert to Symbol** menüsü ya da klavyeden **F8** tuşunu tıklayın.



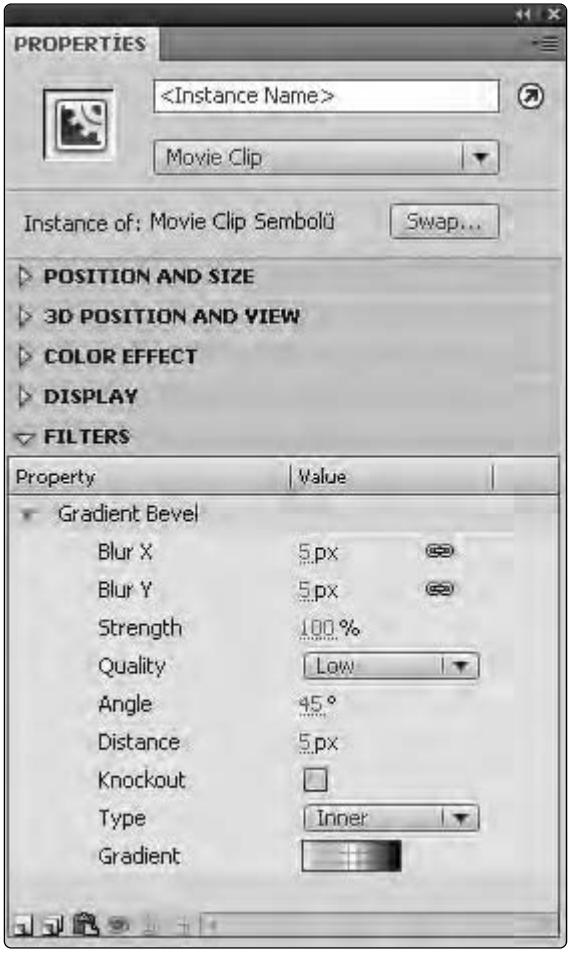
Açılan **Convert to Symbol** penceresinden sembolün adını girin ve tipini seçin. **Registration** bölümünden ise oluşturacağınız sembolün merkez noktasını tıklayarak seçin. Şimdi **Library** paneline bakarsanız yeni sembolü göreceksiniz.

Sembol Özellikleri



Sembollerin özelliklerini düzenlemek için tahmin edeceğinizi gibi Properties panelinden yararlanacağız. Paneldeki **Color Effect** kısmındaki **Style** açılır kutucuğundan sembolün parlaklığını, renk tonunu, saydamlığını ayarlayabilirsiniz. **Blending** açılır kutucuğundan ise istediğiniz karışım modunu seçebilirsiniz.

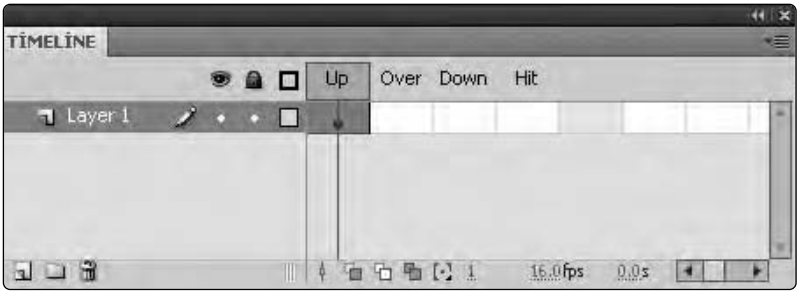
Farklı olrak **Graphic** türündeki sembollere karışım modu uygulayamazsınız. Ayrıca panel yardımı ile sembolün ebatlarında ve konumunda da istediğiniz değişiklikleri yapabileceğinizi hatırlatayım.



Bir başka önemli husus da **Grafhic** ve **Button** türlerinde sembollere **Properties** paneli yardımıyla **filtre** eklenebileceğidir. Panelin **Filters** kısmının en altında yer alan **Add filter** tuşuna tıkladığınızda açılan menüden uygulayacağınız filtreyi seçebilirsiniz. Gerektiğinde bir sembole birden fazla filtre de ekleyebilirsiniz.

Semboller ve Time Line

Sembollerin **zaman çizelgeleri** (*time line*) butonlar haricinde her hangi bir özel farklılık göstermezler. Buton sembollerinde alışılan zaman çizelgesinden farklı olarak dört çeşit kare söz konusudur.

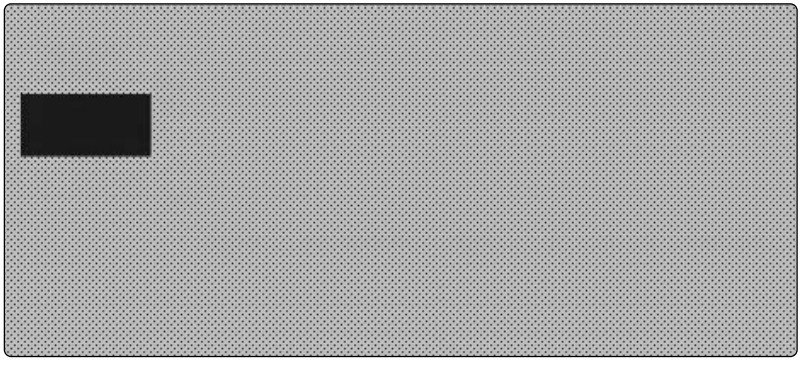


Buton sembolünü çift tıkladığınızda resimdeki gibi bir **time line** göreceksiniz. Burada **Up**, butonun normal halini, **Over fare** ile butonun üzerine gelindiği anı, **Down**, fare ile butona tıklandığında nasıl görüneceğini ifade eder. **Hit** ise butonun etkin olacağı yerlerin belirlenmesi içindir.

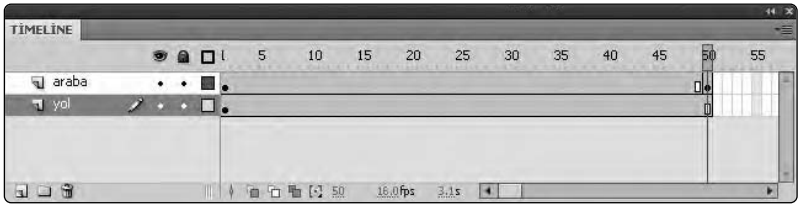
Animasyonlar

Animasyon terimi için kısaca; nesnelerin belli bir zaman aralığında gerçekleştirdikleri hareketler ve değişikliklerdir diyebiliriz. İlk olarak, Flash programının eski sürümlerindeki **Motion Tween** animasyonu **CS4** sürümünde daha da geliştirilmiştir. Eski sürümlerdeki halinin ise **Classic Tween** adıyla **CS4** sürümünde de yer aldığını belirtelim ve ilk animasyon çalışmamızı buna göre yapalım.

Classic Tween



Öncelikle animasyon uygulayacağınız nesnelerin mutlaka sembol olmaları gerektiğini ifade ederek başlayalım. Şimdi örnek olarak basit bir animasyon gerçekleştirelim. Amaç bir nesnenin bir noktadan diğer bir nokta hareket etmesidir. Mesela bir arabanın yol boyunca hareket etmesini örnek verebiliriz.



Öncelikle nesnenizi sembole dönüştürün. Ardından nesnenin bulunacağı son nokta için zaman çizelgesinde bir kare belirleyin. Örneğin bu işlem için `50` numaralı kareye, ardından da bir anahtar kare oluşturmak üzere **`F6`** tuşuna tıklayın. Alttaki zemin vazifesi gören (mesela yol) katmanın da `50.` kareye kadar görünür olması için söz konusu katman seçili iken ve `50.` kare üzerinde iken **`F5`** tuşuna basmalısınız.



Arından `50.` karede iken nesneyi olması gereken son noktaya sürükleyin. Şimdi artık nesne `1.` karede iken olması gereken ilk noktada, `50.` karede ise olması gereken son noktada olacaktır. Fakat oynatma kafasını bu iki kare arasında sürüklerseniz diğer karelerde bir değişiklik olmadığını, yani henüz ortada bir animasyonun olmadığını göreceksiniz.

Şimdi `1.` ile `50.` kareler arasındaki herhangi bir kareye tıklayarak oynatma kafasını getirdikten sonra, hareketli nesnenin katmanına ait zaman çizelgesinin üzerindeyken sağ tıklayın. Acılan menüden **Create Classic Tween** komutunu uygulayarak animasyonu gerçekleştirebilirsiniz.

Test ve Ön İzleme

Bir animasyon çalışmanızı tamamladığınızda ya da çalışmanın her hangi bir evresinde animasyonunuzun nasıl göründüğünü test etmek için **Control>Test Movie** menüsünden yaralanabilirsiniz. Bu işlem için **CTRL+ENTER** tuş kombinasyonunu da kullanabilirsiniz. Yine benzer şekilde bir sahneyi ön izlemek için **Control>Test Scene** menüsünü kullanabilirsiniz.

Maske Uygulama

Şimdi bir başka çalışmaya geçelim ve bu çalışma ile katman kullanımını görelim. Önce Bir manzara resmini sahnemize **Import** edelim ve ardından yeni bir katman daha ekleyerek adına dürbün diyelim. Zaman çizelgesinde dürbün adını verdiğimiz katmanın ilk karesinde iken herhangi bir dolgu rengine sahip bir oval çizin. Şekli maske olarak kullanacağımızdan dolgu renginin hangi renk olduğu fark etmeyecektir.

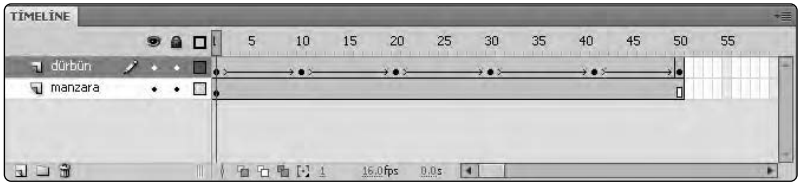


Şimdi **Oval**'i **CTRL+C** tuşları ile kopyalayıp **CTRL+SHIFT+V** tuşları ile üstüne yapıştırın. Seçim aracı ile ikinci ovali yana kaydırın. Artık dürbünümüz hazır.

Sıra animasyonu oluşturmaya geldi. Bunun için öncelikle dürbünü bir sembole dönüştürelim. İlk iş olarak **F8** tuşuna basın ve şekli sembole dönüştürmeyi sağlayacak olan **Convert to Symbol** penceresini açın. Sembole isim verip, tür olarak da Movie Clip seçin.

Bu aşamada şekil katmanının `50.` karesine gelip **F6** tuşuna basarak bir keyframe oluşturun. Oynatma kafasını `50.` kareye getirirseniz, arkadaki manzara resminin görüntülenmediğini göreceksiniz. Bunun olması için hatırlayacağınız gibi manzara katmanını seçili iken `50.` kareye gelip **F5** tuşu ile `1.` ile `50.` kareler arasını **frame**'lerle doldurun.

Şimdi `1.` ile `50.` kareler arasında herhangi bir kareyi seçip zaman çizelgesi üzerinde sağ tık yapın ve açılan menüden **Classic Tween** komutunu seçin. Böylece animasyon oluşmuş oldu. Fakat animasyon düz bir eksen üzerinde hareket etmeyeceği için çizelgenin farklı karelerinde birkaç tane keyframe daha oluşturmanız gerekmektedir.



Hemen işe koyulalım ve mesela `10.` kareye gelip yine **F6** tuşu ile bir keyframe oluşturalım. Tabi ki peşinden, seçim aracı ile şeklin `10.` kareye gelindiğinde hangi konumda olmasını istiyorsanız o konuma taşımayı da unutmayın. Bu işlemi `20.` `30.` ve `40.` kareler için de tekrarlayın ve her seferinde şekli resmin farklı bir yerine konumlandırarak dürbünün tüm manzarayı dolaşmasını sağlamış olun.



Animasyonu bitirdiğinize göre artık şekil katmanına maske uygulama işlemine geçebiliriz. Şimdi şekil katmanının üzerine gelin ve sağ tıklayarak menüyü açın. Ardından Mask komutunu tıklayın. Bunu yaparak şekli bir maskeye dönüştürmüş oldunuz. Artık manzaranın sadece, şeklinizle maskelediğiniz alanı görünmekte ve manzaranın diğer kısımları görünmemektedir. **Zaman çizelgesini** (*time line*) hareket ettirdiğinizde dürbün, belirlediğiniz rotada tüm manzarayı dolaşacaktır.

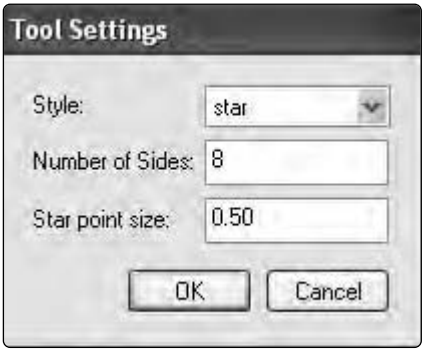
Parlaklık Uygulaması

Animasyonlarınızda renk, parlaklık, saydamlık gibi özellikleri de değiştirebilirsiniz. Buna örnek olarak güneşin doğuşunu temsil eden bir animasyon hazırlayalım. Bunun için sarı dolgu renginde bir daire çizin. Ardından yukarıda öğrendiğimiz şekilde ve **Graphic** türünde bir animasyon oluşturun.

Sonra `1.` karede iken daireyi sahnenin sağ alt kısmına, `50.` karede ise sol üst kısma sürükleyin. Böylece animasyonun eksenini belirlemiş oldunuz. **Properties** panelindeki **Style** açılır kutucuğundan **Brightness** seçeneğini tıklayın ve **Bright** sürgüsünü `-100` olarak ayarlayın. Artık animasyonu deneyebilirsiniz.

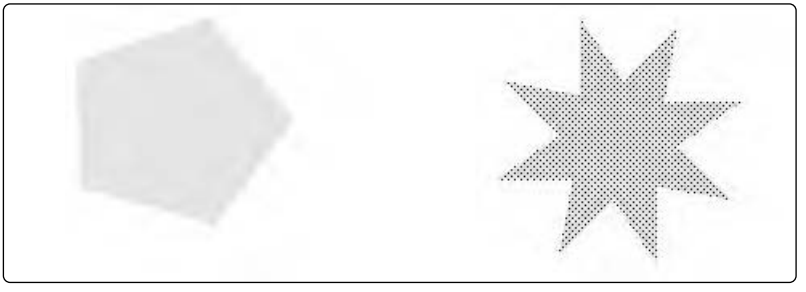
Diğer animasyonlar ile birlikte bu animasyonun uygulama dosyasını da DVD içinde **Uygulamalar> Flash** klasöründe bulabilirsiniz.

Şekil Animasyonları



Bu animasyon türü bir nesnenin şeklinin başka bir şekle dönüştürülmesi için kullanılır. Biz örnek olarak bir beşgeni bir yıldıza dönüştürelim. İlk olarak **PolyStar Tool** ile bir beşgen çizelim.

Ardından `50.` kareye gelip **F6** ile bir keyframe ekleyelim.

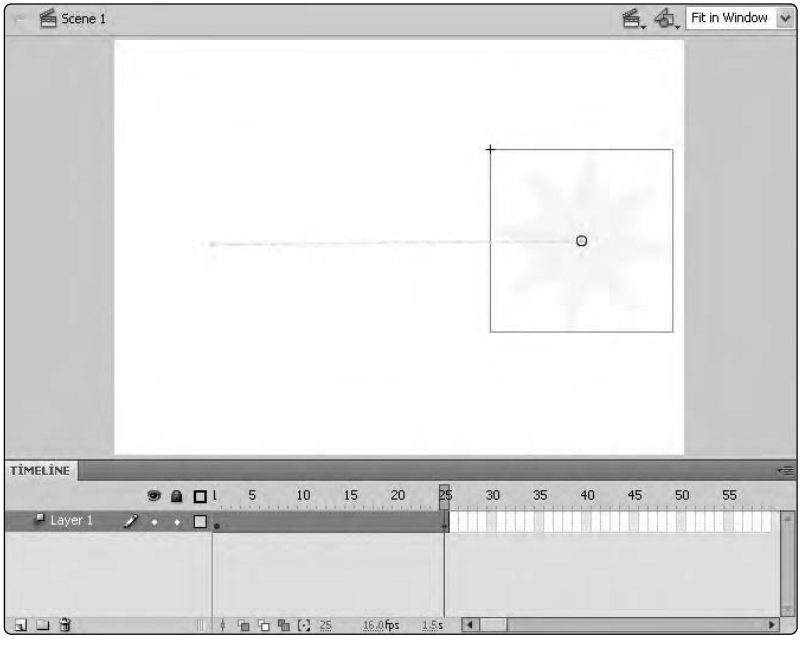


Sonrasında `50.` karede iken beşgeni silip, yine **PolyStar Tool** ile **Properties** panelindeki **Options**... tuşundan yaralanarak bir yıldız çizelim. Daha sonra `1.` ile `50.` kareler arasındaki herhangi bir karede sağ tık ile açılan menüden **Create Shape Tween** komutunu seçelim. Artık animasyonunuz hazır. Flash programı, `1.` ile `50.` kareler arasını doldurarak beşgenden yıldıza geçiş işlemini gerçekleştirecektir.

Animasyondaki ara şekillerin oluşmasında kontrol sağlamak isteseniz **Modify>Shape>Add Shape Hint** menüsünden yararlanabilirsiniz. Bu menü yardımıyla şekle ekleyeceğiniz noktalar (ip uçları) sayesinde, ilk şeklin hangi noktasının son şeklin hangi noktasına dönüşeceğini belirleyebilirsiniz. Bu size daha gerçekçi ve yumuşak dönüşümler oluşturma imkanı verecektir.

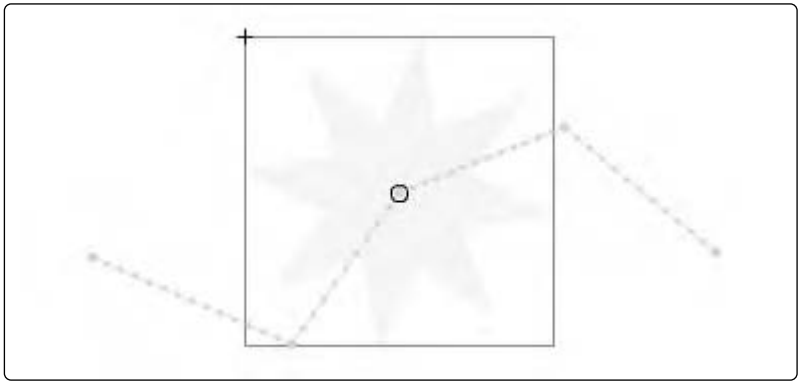
Son olarak, eklediğiniz noktaları kaldırmanız gerektiğinde **Modify>Shape>Remove All Hints** menüsünü kullanmanız gerektiğini belirteyim.

Motion Tween



Şimdi de **Flash CS4** ile gelen animasyon yapısını görelim. Sahnede bir yıldız oluşturarak işe başlayalım. Ardından yıldızın üzerinde sağ tıklayın ve açılan menüden **Create Motion Tween** komutunu uygulayın. Bu takdirde Flash, `2.` keyframe'i otomatik olarak oluşturacaktır. Zaman çizelgesinde, seçili kare alanını fare ile sürükleyerek kare sayısını arttırabilirsiniz. Örneğin biz seçili alanı `25.` kareye kadar sürükleyelim. Oynatma kafası `25.` karede iken bu sefer yıldızı animasyonda gelmesini istediğiniz son noktaya sürükleyin.

Gördüğünüz gibi çok daha kolay ve pratik bir şekilde animasyonu oluşturmuş olduk. Resimde gördüğünüz yıldızı takip eden çizgi, animasyondaki kareleri ve animasyonun hareket edeceği ekseni temsil etmektedir.



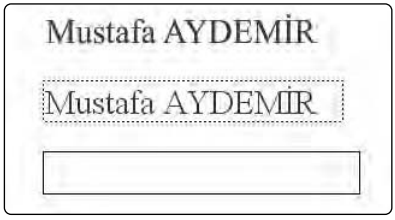
Şimdi zaman çizelgesindeki seçili alanı sürükleyerek animasyonu `50.` kareye kadar uzatın. Ardından oynatma kafasını `25.` kareye getirin. Şimdi yıldızı yukarı ya da aşağı doğru sürükleyin. Bu takdirde animasyona otomatik olarak yeni bir keyframe eklenecektir. Bir başka karede yine aynı işlemi tekrarlayın.

Bu sayede, animasyona yeni keyframe'ler eklemenin yanı sıra, aynı zamanda da animasyonun eksenini yukarı aşağı, zik zak şeklinde değiştirmiş oldunuz. Tabi dilerseniz keyframe'leri, daha önce öğrendiğiniz şekilde **`F6`** tuşuna basarak da oluşturabilirsiniz.

Dilerseniz animasyonu parçalara da ayırabilirsiniz. Bunun için ayrılacak kareye tıkladıktan sonra **`CTRL`** tuşuna basılı tutarak kareye tekrar tıklayın. Böylece ayrılacak kare seçilmiş olacaktır. Şimdi sağ tıklayıp menüyü açın. Menüden **Split Motion** komutunu uygulayın. Böylece iki ayrı animasyon elde etmiş oldunuz. Artık animasyonların birinde yapacağınız değişiklik diğerini etkilemeyecektir.

METİNLER

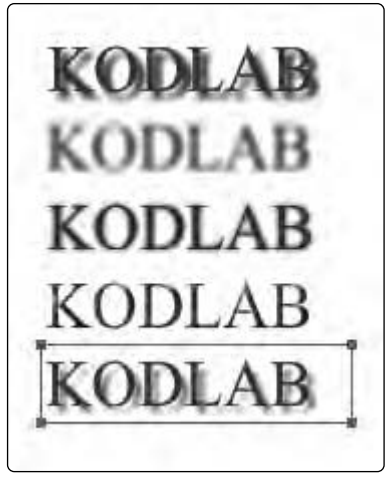
Flash ile metin oluşturmak, öğrendiğiniz diğer Adobe programlarından farklı değildir. Yine metin aracını (*Text Tool*) seçip tek tıklama yaparak bir ekleme noktası oluşturabilir ya da bir seçim yaparak bir metin alanı oluşturabilirsiniz. Metinlerinizi, hatırlayacağınız gibi **Properties** panelini kullanarak detaylı bir biçimde düzenleyebilirsiniz.



Gelelim Flash programında oluşturabileceğiniz metin türlerine. Bunları, **Properties** panelinin en üst kısmında göreceğiniz gibi; **static text**, **dynamic text** ve **input text** olarak sıralayabiliriz.

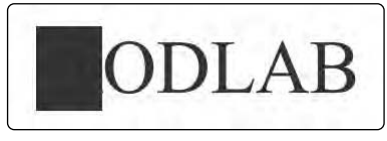
Static text, sahneye yazdığımız herhangi bir metin için kullandığımız standart metin türüdür. **Dynamic text**, actionscript kodları ile kontrol edilebilecek metin türünü ifade eder. **Input text** ise, formlardan bildiğiniz metin girişi yapılabilmesini sağlayan bir kutu oluşturur.

Metinlerinize, **Properties** panelinin **Options** kısmını kullanarak linkler, **Filters** kısmından da resimde gördüğünüz gibi filtreler ekleyebilirsiniz.



METİNLERE MASKE UYGULAMA

Şimdi de metinlere nasıl maske ekleyebileceğimizi görelim. Konuyu bir şekil (*Shape Tween*) animasyonu üzerinden açıklayalım.



İlk adım olarak sahnede metni oluşturun. Ardından maske için kullanılmak üzere ikinci bir katman oluşturun. Çalışmaya şekil aracı ile ilk harfin üzerine bir dikdörtgen çizerek devam edin.

Şimdi maske katmanı seçili iken, zaman çizelgesinde örneğin `50.` kareye gelip **`F6`** tuşu ile bir keyframe, ardından alttaki metnin yer aldığı katmanı seçili duruma getirip, metnin son kareye kadar görünebilmesi için `50.` karede F5 tuşu ile bir frame oluşturun.

Yine maske katmanı seçili iken, `50.` kareyi tıkladıktan sonra sadece ilk harfi kaplayan dikdörtgeni **Free Transform Tool** yardımıyla tüm harfleri kapsayacak şekilde uzattın. Şimdi ise daha önce öğrendiğiniz gibi `1.` ile `50.` kareler arasındaki herhangi bir kareye tıkladıktan sonra açılan menüden **Create Shape Tween** komutunu seçin. Böylece animasyonu tamamlanmış oldunuz.

Şimdi maske katmanının üzerine gelin ve sağ tıklayarak menüyü açın. Ardından **Mask** komutunu uygulayın. Böylece şekil bir maskeye dönüşmüş oldu. Artık animasyonu test edebilirsiniz.

Ses & Video

Ses Dosyaları ile Çalışmak

Flash dosyanıza ses eklemek istediğinizde yapmanız gereken ilk iş, ses dosyasını sahneye ya da kitaplığa ithal etmek olacaktır. Hemen başlayalım ve **File>Import>Import to Library**... komutu ile açılan Import to **Library** penceresi yardımıyla programa, çalışmanıza dahil edeceğiniz ses dosyasının bilgisayarınızdaki konumunu gösterin.

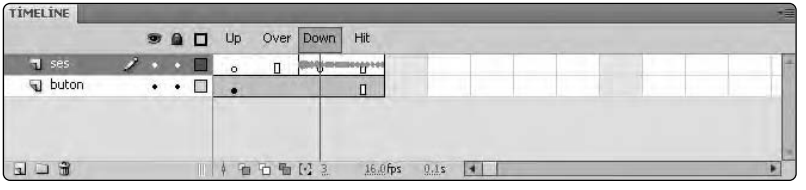
Library panelinden, yüklediğiniz ses dosyasını görebilirsiniz. Panelde dosya seçiliyken sağ üst köşedeki **play** tuşu ile sesi dinleyebilirsiniz. Sesi bu şekilde dinleyip test ettikten sonra çalışmanıza eklemeye karar verirseniz ilk yapmanız gereken, ses dosyası için yeni bir katman oluşturmaktır. Yeni oluşturduğunuz katman seçili iken **Library** panelinden ses dosyasını sahneye sürükleyip bırakın. Artık çalışmanızı test edebilirsiniz.

Yüklediğiniz ses üzerinde, zaman çizelgesinde sesin yüklendiği kareyi seçtikten sonra **Properties** paneli vasıtası ile bazı düzenlemeler yapabilirsiniz. Örneğin sesin kaç kez tekrarlanacağını **Repeat** kısmından belirleyebilirsiniz. Paneldeki **Effect** açılır kutucuğundan yaralanarak sese efekt uygulayabilirsiniz. Açılır kutudan istediğiniz efekti seçtikten sonra, düzenleme yapmak için kutucuğun yanındaki kalem tuşuna (*Edit Sound Envelope*) basın. Açılacak pencereden ses üzerinde düzenlemeler yapabilirsiniz.

Nesnelere Ses Ekleme

Şimdi de dilerseniz sembollerle çalışmayı görelim. Bu gereklidir çünkü, örneğin bir butona ses eklemek istiyorsanız butonu önce bir sembole dönüştürmek zorundasınız.

Önce buton niyetine bir daire çizelim. Ardından butona sağ tıklayarak **Convert to Symbol** komutunu uygulayın. Şimdi daireye çift tıklayarak sembole girin. Çizdiğimiz bu daire katmanının adını buton olarak değiştirin. Bir sonraki aşama olarak yeni bir katman oluşturun ve adını ses olarak değiştirin. Sonra butona ekleyeceğiniz ses dosyasını da kitaplığa ithal edin.



Buton sembollerinde dört farklı durum olduğunu daha önce görmüştük. Biz butona tıklandığında ses vermesini istediğimiz için ses katmanı seçili iken **Down** karesini seçip **F6** tuşu ile bir keyframe oluşturalım.



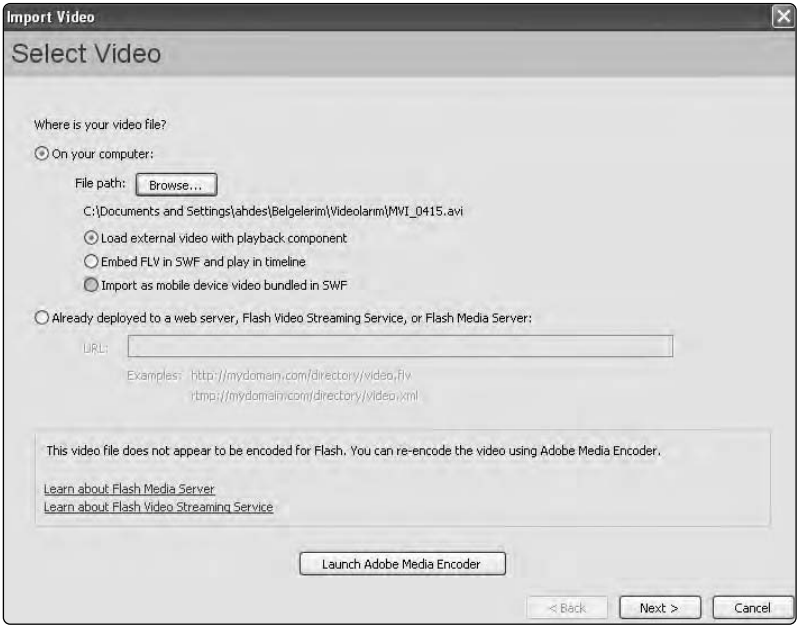
Yapmanız gereken **Down** karesini seçip ses dosyasını sahneye sürükleyip bırakmaktır. Artık **CTRL+ENTER** tuşları ile çalışmanızı test edebilirsiniz. Test ekranında butona her tıklayışınızda ses dosyası çalışacaktır.

Dilerseniz **Window>Common Libraries>Sounds** menüsü ile

resimde gördüğünüz ses kütüphane panelini açarak, burada mevcut olan toplam 186 adet farklı ses efektini de çalışmalarınızda kullanabilirsiniz.

VİDEOLAR İLE ÇALIŞMAK

Çalışmanıza bir video ekleyebilmeniz için ilk yapmanız gereken **File>Import>Import Video**... menüsünü tıklamak olmalıdır.



Bu takdirde karşınıza resimdeki pencere çıkacaktır. Bu pencerenin amacı, eklenecek videonun konumunu saptamaktır. Video bilgisayarınızdan seçilebileceği gibi, bir web sunucusundaki video da url adresi girilerek seçilebilir. **Next** tuşu ile yolunuza devam edebilirsiniz.



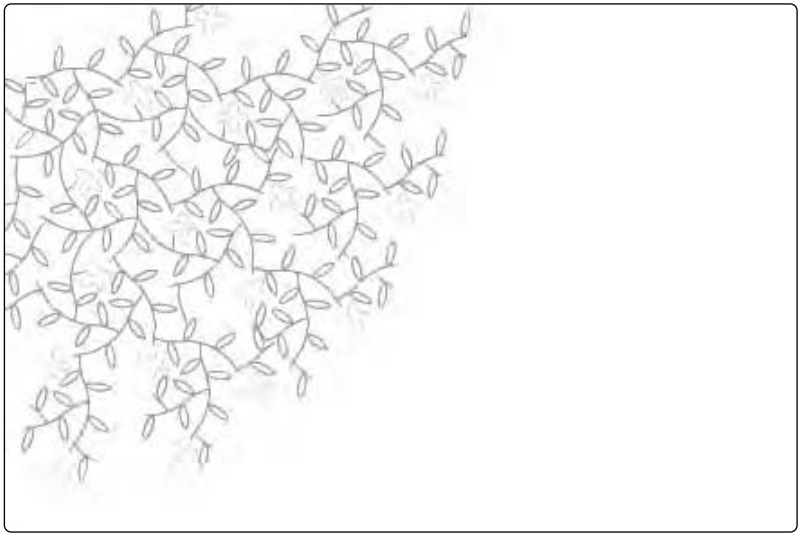
Daha sonra resimdeki ekrandan videonuza bir skin seçin. **Skin rengini** de belirledikten sonra yine **Next** ile yolunuza devam edin ve **Finish** ile işlemi tamamlayın. Video hem kitaplığa hem de sahneye eklenecektir. Çalışmanızı dilerseniz **CTRL+ENTER** tuşları ile test edebilirsiniz.

CS4 YENİLİKLER

Bu bölümde Flash programlarının önceki sürümlerinde olmayan bazı özelliklerine değineceğiz.

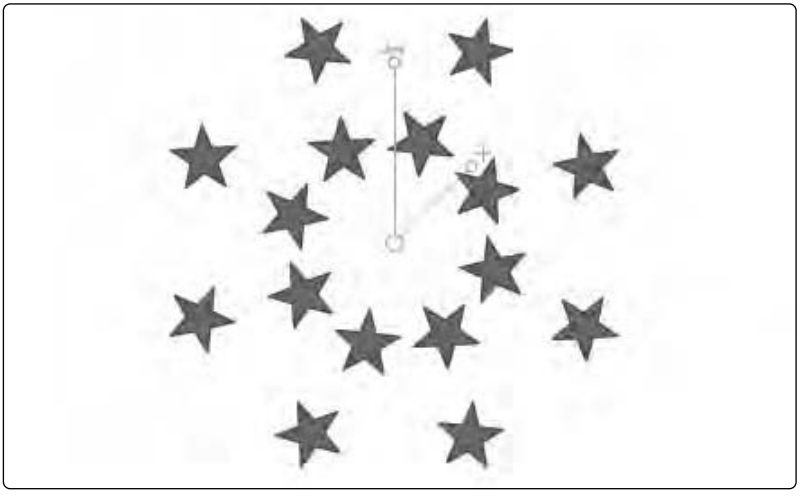
DECO TOOL

Deco Tool kullanarak farklı çizim efektleri oluşturabilirsiniz. Bu efektlerden ilk olarak **Vine Fiil** efektini görelim. Efekti, **Deco Tool** seçiliyken **Properties** panelindeki **Drawing Effect** kısmında bulunan açılır kutucuktan seçebilirsiniz.

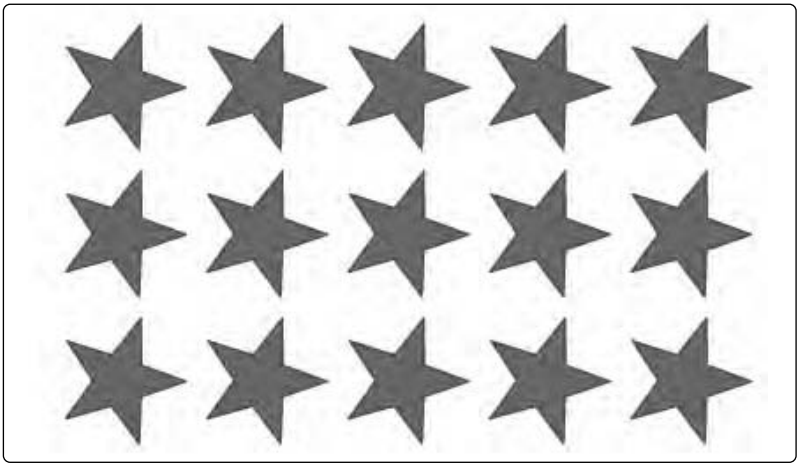


Properties panelinde gördüğünüz gibi bu efektin **Leaf** (*yaprak*) ve **Flower** (*çiçek*) olmak üzere iki seçeneği mevcuttur. Öncelikle biz efekti varsayılan değerleri ile uygulayalım. Bunun için imleci sahnede efektin başlamasını istediğimiz noktaya tıklıyoruz. Efekt sahne boyunca devam edecektir. Efektin sona ermesini isterseniz sahneye bir kez daha tıklamalısınız.

Bu efekti kendi seçeceğiniz yaprak ve çiçek desenleriyle de kullanabilirsiniz. Bunun için **Edit** tuşları ile daha önceden kütüphaneye eklediğiniz desenlere ulaşabilirsiniz. Ayrıca paneldeki **Animate Pattern** kutucuğunu işaretleyerek sahnenin desen ile doldurulması işleminin bir animasyon olarak saklanmasını sağlayabilirsiniz.

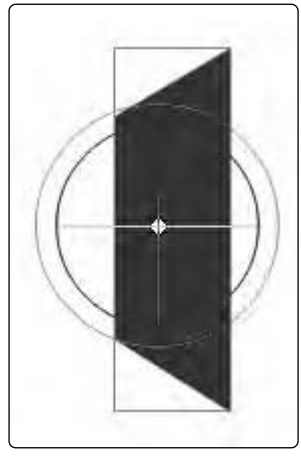


Gelelim ikinci efekt olan **Symmetry Brush** seçeneğine. Bu efekt yardımıyla bir simetri ekseni üzerinde birbirlerine simetrik olan desenleri sahne üzerinde oluşturabilirsiniz. Bu efektte de yine varsayılan desen kullanmayıp, bizim de resimde gördüğünüz örnekte yaptığımız gibi kütüphaneden bir sembol kullanabilirsiniz. **Eksenler** yardımıyla nesneleri döndürebilir ya da birbirlerine olan uzaklıklarının arttırıp azaltabilirsiniz.



Üçüncü ve son seçenek ise **Grid Fill** seçeneğidir. Bu efekt ile sahnenin şekiller tarafından simetrik olarak doldurulmasını sağlayabilirsiniz.

3D Rotation Tool

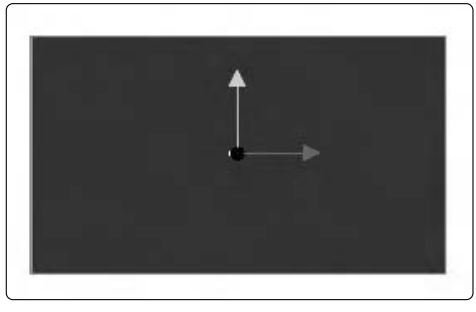


Flash CS4 ile gelen **3D Rotation Tool** aracı ile nesneleri `x`, `y` ve `z` koordinatlarında döndürebilirsiniz. Şimdi aracı denemek için ekrana bir dikdörtgen çizelim. Ardından dikdörtgeni **F8** tuşu yardımıyla hızlıca bir sembole (*movie clip*) dönüştürelim.

Şimdi **3D Rotation Tool** aracını seçelim. Objenin ortasında daireler ve çizgiler belirecektir. Bunlar **x** (*kırmızı*), **y** (*yeşil*) ve **z** (*mavi*) ekseninde döndürme yapabilmeniz içindir.

3D Translation Tool

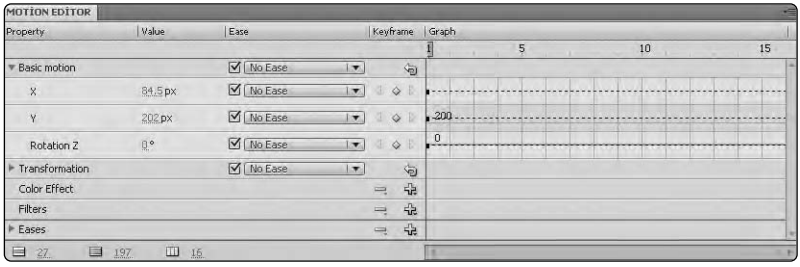
Diğer 3D aracı olan **3D Translation Tool** ile de üç boyutlu taşıma işlemleri yapabilirsiniz. Yine bir dikdörtgen çizelim ve yine dikdörtgeni **F8** tuşu yardımıyla bir sembole (*movie clip*) dönüştürelim.



Aracı seçtiğinizde yukarıdaki gibi `x`, `y` ve `z` koordinatlarını temsil eden oklar ve nokta ile nesneyi bu üç eksende de taşıyabilirsiniz. Dilerseniz aynı işlemi Properties panelindeki **3D Poaition and View** kısmından da yapabilirsiniz.

Motion Editor Paneli

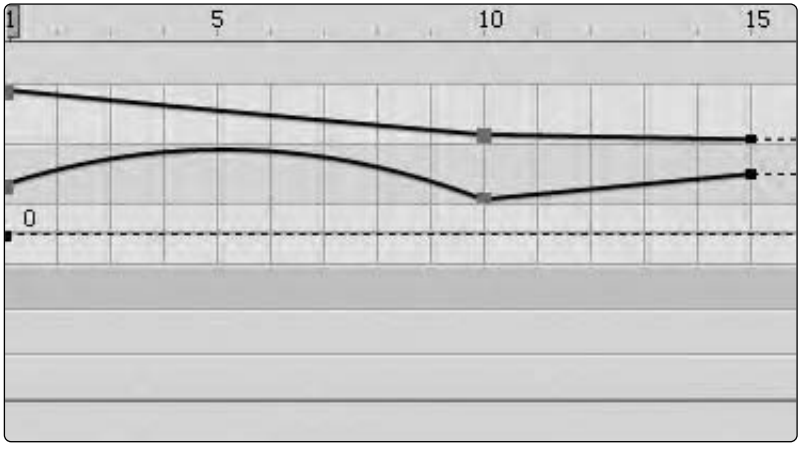
Motion tween olayını bu editör yardımıyla çok daha pratik ve detaylı bir biçimde yapabilirsiniz. Öncelikle bir dikdörtgen çizip **F8** yardımı ile movie clip sembolüne dönüştürün. Ardından zaman çizelgesinde ilk kareye tıklayıp **Create Motion Tween** komutunu uygulayın.

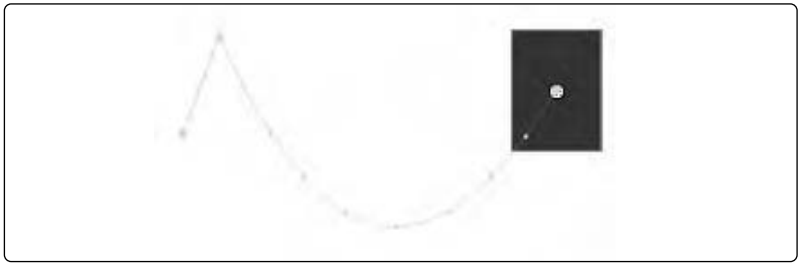


Şimdi **Motion** editor paneli aktif ve resimdeki gibi olacaktır. Bu sefer tüm animasyonu **Motion** editor panelinde yapacağız. Resimde gördüğünüz gibi editörün sağ tarafında zaman çizelgesini temsilen ve sol baştaki (*Basic motion* bölümündeki) koordinatları temsilen çizgiler yer almaktadır.

Şimdi sağ bölümde çizginin en başına, yani ilk kareye tıklayın. Burası tahmin edeceğiniz gibi animasyonun başlangıç noktasıdır. Ardından `15.` çizgiye gelip sağ tıklayın ve **Add Keyframe** uygulayın. Yine tahmin edeceğiniz gibi burası da animasyonun son noktası olacaktır.

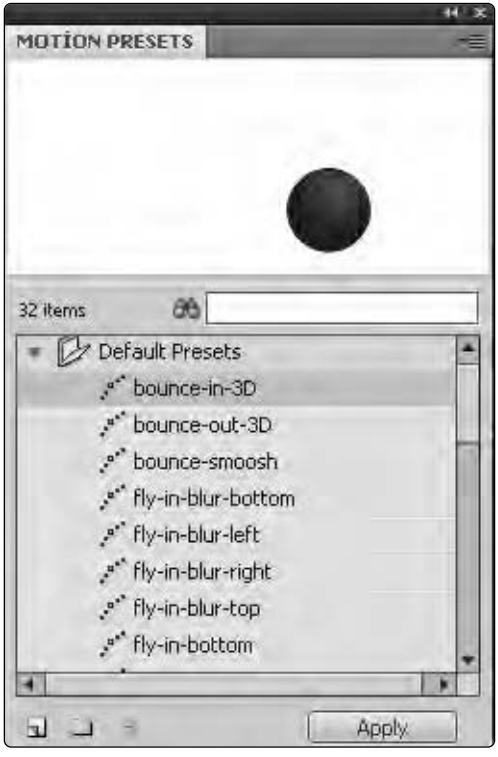
Şimdi sağda oynatma kafası `15.` çizgideyken, sol baştan x koordinatının sürgüsünü fare ile sürükleyerek nesneyi olması gereken son noktaya getirin. Evet kabaca işin özü bu. Gerçekten çok kolaylıkla ve hızlıca animasyonlar hazırlamak işten bile değil.





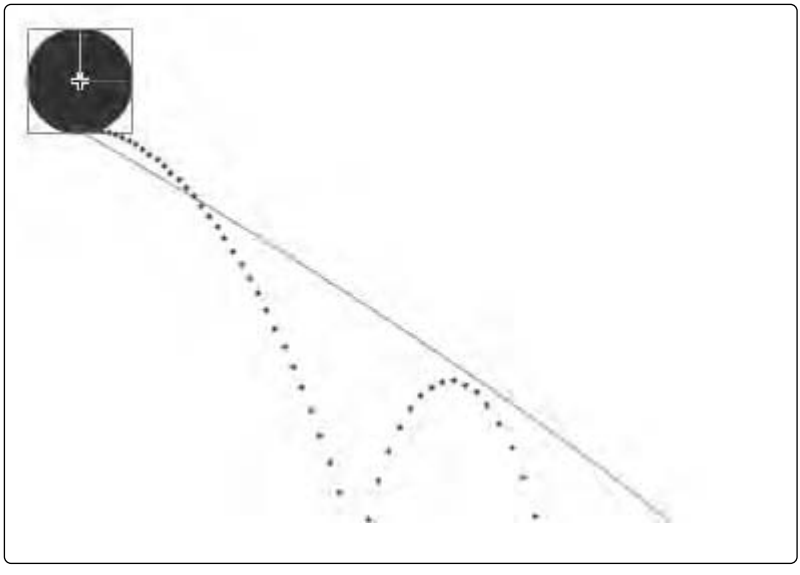
Editörde de yine sahnede yapabildiğiniz gibi animasyon yolunun eksenini istediğiniz gibi değiştirebilirsiniz. Bu işlemi sağ kısımdaki, üzerine keyframe eklediğimiz çizgileri fare ile hareket ettirerek yapabilirsiniz. Editörün **Transformation** kısmındaki **skew** sürgüleri ile de objeyi `x` ve `y` ekseninde döndürebilir, skale sürgüleri ile de yine `x` ve `y` ekseninde küçültüp büyültebilirsiniz.

Motion Presets Paneli



Flash CS4 ile gelen yeniliklerden bir de **Motion Presets** panelidir. Bu panelde programla birlikte gelen ve çalışmalarınızda kullanabileceğiniz hazır animasyonlar bulunmaktadır. Ayrıca kendi hazırladığınız animasyonları da bu panele kayıt edebilirsiniz. Bu takdirde animasyonlarınız istediğiniz zaman kullanabileceğiniz şekilde elinizin altında olacaktır. Panelde seçtiğiniz bir animasyonun ön izlemesini panelin ön izleme kısmında görebilirsiniz.

Öncelikle paneldeki **Default Presets** klasöründe bulunan hazır animasyonların nasıl kullanıldığına bakalım. İlk iş olarak bir daire çizelim ve bunu Movie Clip türünde bir sembole dönüştürelim.

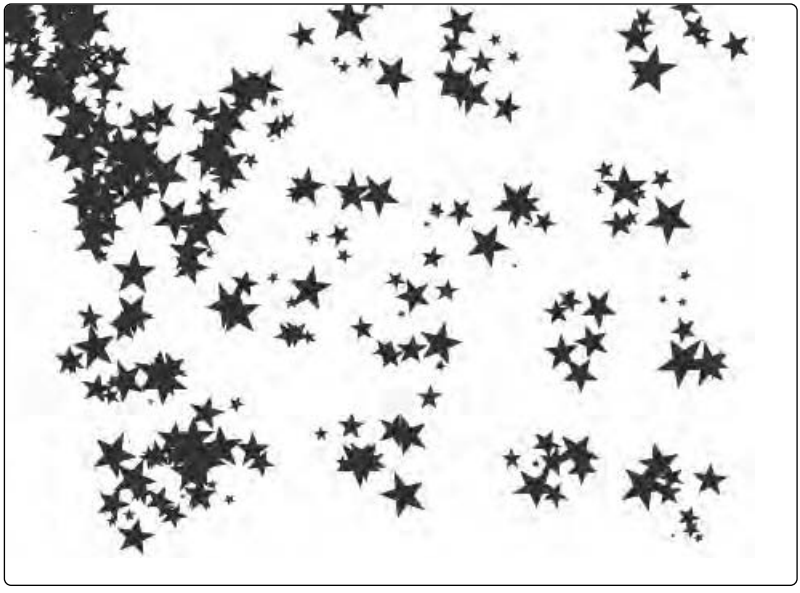


Şimdi **Motion Presets** panelinden istediğiniz bir animasyonu seçin ve panelin alt kısmındaki **Apply** tuşu ile sembole uygulayın. Ardından **CTRL+ENTER** yapıp animasyonun keyfini çıkarın!

Kendi hazırladığınız bir animasyonu panele eklemek içinse yapmanız gereken ilk iş animasyonda kullandığınız nesneyi seçmek olmalıdır. Nesne seçiliyken **Motion Presets** panelinin alt kısmındaki **Save selection as preset** tuşuna tıklayın.

Açılan pencere yardımıyla animasyonunuza bir isim verin. Artık bu animasyon da emrinize amade.

Spray Brush Tool



Şimdi de Flash CS4 ile gelen yenikliklerden biri olan **Spray Brush** aracını inceleyelim. Bu araç, varsayılan sembolü ya da sizin hazırladığınız bir sembolü püskürtme şeklinde sahneye serpiştirir.

Kendi hazırladığınız sembolü kullanmak için öncelikle **Properties** panelinde bulunan **Symbol** kısmındaki **Edit** tuşu ile kütüphaneden kendi sembolünüzü seçin. Ve aracı sahne üzerinde basılı tutup gezdirerek püskürtme işlemini uygulayın.

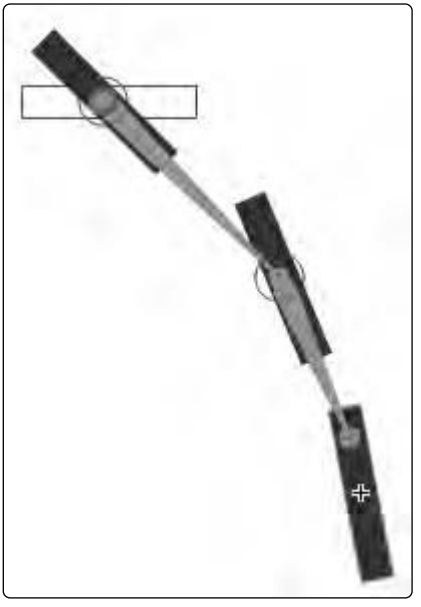
Bone Tool

Gördüğünüz gibi Flash CS4 ile gelen yenilikler anlatmakla bitmiyor. Şimdi de **Bone** aracından bahsedeceğiz. Bu aracın çalışma mantığı, objelerin birbirlerine bağımlı olarak hareket etmesini sağlamaktır diyebiliriz.

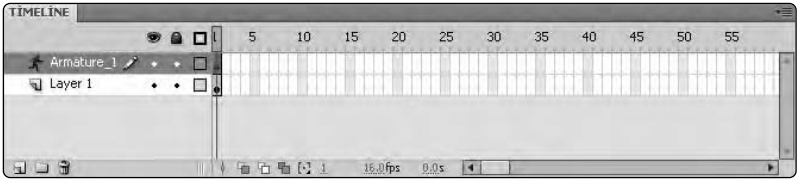
Konuyu hemen bir örnekle detaylandıralım. Öncelikle bir dikdörtgen çizip bunu **Movie Clip** türünde bir sembole dönüştürün. Ardından **CTRL+C** ve **CTRL+SHIFT+V** ile iki adet kopyasını oluşturun. Burada amaç, nesnelerden birisi hareket ettiğinde diğerlerinin de hareketini sağlamaktır.



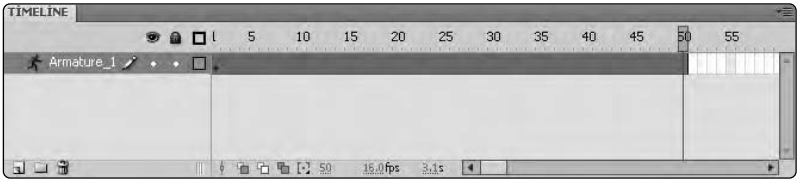
Şimdi **Bone** aracını seçip ilk nesnenin istediğiniz bir noktasına tıklayıp ikinci nesnenin bağlantı oluşturmak istediğiniz noktasına kadar sürükleyip bırakın. Bu sefer ikinci nesne ile üçüncü nesne arasındaki bağlantıyı sağlamak üzere işlemi tekrarlayın. İşlem sonrası sonuç resimdeki gibi olacaktır.



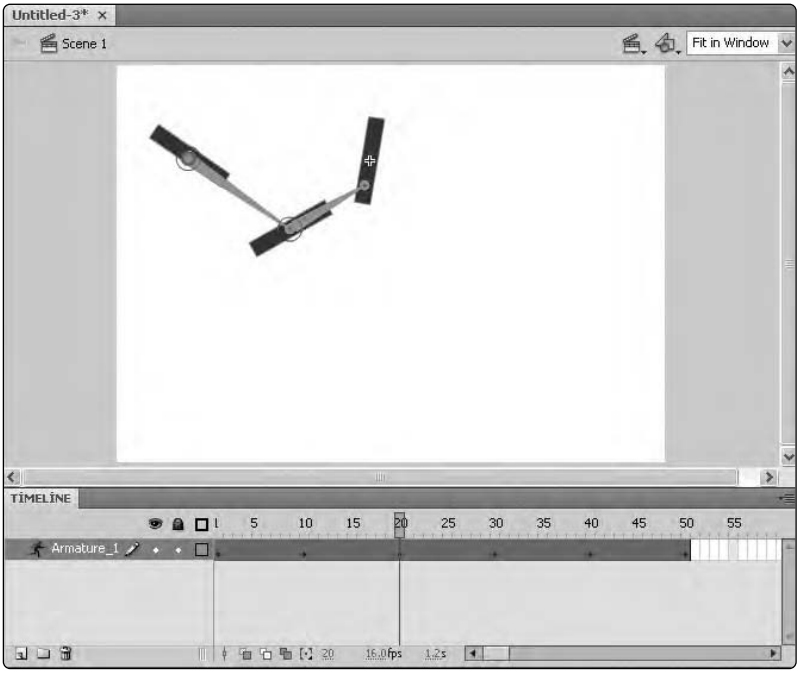
Şimdi **Selection Tool** ile üçüncü nesneyi tutup ekranın aşağısına ya da yukarısına doğru sürükleyin. Resimde de gördüğünüz gibi siz üçüncü nesneyi sürüklediğinizde ilk iki nesne de beraber hareket etmektedir.



Zaman çizelgesine bakarsak, siz bu işlemleri yaptığınızda, programın söz konusu kinematik işlemleri için bir katman oluşturduğunu göreceksiniz. Dolayısıyla artık sahneyi açtığınızda oluşan ilk katmana ihtiyaç kalmadığından onu silebilirsiniz.



Sıra geldi animasyon oluşturmaya. Daha önceki örneklerde yaptığımız gibi zaman çizelgesinde `1.` karedeki seçimi sürükleyerek `50.` kareye kadar uzatın. Böylece `50` karelik bir animasyon elde etmiş olacaksınız.



Şimdi **Bone** aracı ile bütünleşik hale getirdiğiniz nesnenin, hangi karede hangi konumda olmasını istiyorsanız, istediğiniz kareyi seçtikten sonra nesneyi o konuma sürükleyin. Bu işlemi yaptığınızda, otomatik olarak o karede yeni bir keyframe oluşacaktır. Örnek olarak `10`, `20`, `30`, `40` ve son olarak `50.` karede nesneyi her bir keyframe için farklı konumlara taşıyarak işlemi tekrarlayın.

Artık arkanıza yaslanıp animasyonu **CTRL+ENTER** ile seyredebilirsiniz. **Bone Tool** örneği olarak hazırladığımız animasyonu da yine DVD içinde bulabilirsiniz.

11

İÇERİK YÖNETİM SİSTEMLERİ

CMS Nedir? 611
CMS Türleri ve Örnek Bazı CMS Uygulamaları 612
E-Ticaret Sistemleri 612
Forum Sistemleri 614
Blog Sistemleri 618
Portal Sistemleri 620
Örnek CMS İncelemesi: Joomla 623
Tarihçe 623
Genel Terimler 624
Cpanelde Veritabanı Oluşturma 625
Uzak Sunucuda (Cpanel) Joomla Kurulumu 630
Joomla Genel Ayarlar 639
Joomla 1.5.x Genel Yapılandırma Ayarları (Site Sekmesi) 640
Joomla 1.5.x Genel Yapılandırma Ayarları (Sistem Sekmesi) 642
Joomla 1.5.x Genel Yapılandırma Ayarları (Sunucu sekmesi) 646
Joomla'da İçerik Planlaması (Bölüm Kategori İçerik Mantığı) 648
Resim, Ek Sayfa, Devamını Oku... Tuşları ve İşlevleri 655
Menüler ve Menü Yönetimi 658
Menü Öğe Çeşitleri 664
Üye Yönetimi 666
Eklenti ve Tema Yöneticisi 669
Joomla 1.5.x Temalarındaki Ana Dosyalar 673

CMS Nedir?

CMS (**C***ontent* **M***anagament* **S***ystems: İçerik Yönetim Sistemleri*) dinamik içerik yönetim yapılanması sunan web uygulamalarıdır. PHP, ASP.NET... vb web programlama dilleri ile yazılırlar. Sistemlerin yönetimi **admin** (*yönetim*) panelleri vasıtası ile yapılır. Bu sistemlerde içerik oluşturmak için sistemin yapısına müdahale etmek gerekmez. Adeta WORD ya da benzeri bir kelime işlem programında yazı yazar gibi kolay ve hızlı içerik üretme imkanı verirler.

CMS sistemleri; kolay kurulum, Admin Panel ile yönetim kolaylığı, içerik üretme kolaylığı, esnek tasarım (tema) özellikleri, bir çok dinamik ve etkileşimli uygulamayı (forum, anket, oylama, yorum, özel mesaj...) bünyelerinde barındırmaları...vb birçok avantajları sebebiyle yaygın olarak kullanılmaktadırlar.

İçerik yönetim sistemleri, işlevlerine göre değişik guruplara ayrılmaktadır. Bunların başlıcalarını; portal sistemleri, blog sistemleri, e-ticaret sistemleri ve forum sistemleri olarak sayabiliriz. Bunlar arasında ücret-

li olan sistemler de olmasına karşın çoğunluğu **açık kaynak** (*open source*) yazılımlardır ve biz kitabımızda ücretlileri değil de sadece açık kaynak CMS sistemlerini konu edineceğiz.

CMS Türleri ve Örnek Bazı CMS Uygulamaları

Bu bölümde, yukarıda kısaca saydığım CMS sistemlerine birkaç örnek vereceğim. Amacım örnek vereceğim CMS sistemleri hakkında kısa kısa bilgiler de vererek sizi söz konusu sistemlerle tanıştırmak.

E-Ticaret Sistemleri

Mal ya da hizmet tanıtım ve satışlarının internet ortamında yapılmasını sağlayan e-ticaret siteleri gün geçtikçe artmakta ve bu sektör günümüzde ciddi cirolara ulaşmış bulunmaktadır. E-ticareti denemeyi düşünüyorsanız aşağıdaki sistemleri de göz ardı etmemelisiniz. İşinizin büyüklüğüne ve özelliklerine göre bir seçim yapmak yararınıza olacaktır.

OsCommerce



osCommerce GNU GPL Lisansına sahip, açık kaynak kodlu bir e-ticaret betiğidir. **osCommerce**, ürünlerinizi internette satışa sunmanız için oluşturulmuş en iyi **betiklerden** (*script)* biridir. Gelişmiş admin paneli

sayesinde çok fonksiyonlu ama kullanımı kolay bir mağaza yönetimi imkanı sunar.

osCommerce istenildiği kadar çok sayıda ürünü satışa sunabilme, ürünleri kendi aralarında kategorilere ayırabilme, müşterilere sitede görüşlerini sunabilme imkanı tanıma, ürünleriniz ile ilgili olarak müşterilerinize düzenli e-postalar ile bilgi verebilme, müşterilere en çok satan ürünleri gösterebilme, istediğiniz kadar farklı dillerde ve farklı para birimleri ile yayın yapabilme, kargo ve vergi bilgileri desteği... vb birçok özelliğe sahiptir.

osCommerce'in özellikleri bunlarla sınırlı değil elbette... Aşağıdaki linkten betiğin resmi sitesine ulaşabilir, detaylı bilgileri bu site üzerinden alabilir ve betiği indirip deneyebilirsiniz.

Resmi sitesi: *http://www.oscommerce.com/*

ZEN-CART

Zen-cart kullanıcı dostu bir betiktir. Çoklu ödeme ve alışveriş seçeneği, indirim sistemi gibi özelliklere sahiptir. Kurulumu ve kullanımı oldukça kolay olan bu betiği de aşağıdaki resmi site adresinden inceleyebilir ve indirebilirsiniz.

Resmi sitesi: *http://www.zen-cart.com/*

Dersler: *http://tutorials.zen-cart.com/index.php*

Wiki: *http://www.zen-cart.com/wiki/index.php/Main_Page*

Destek Forumu: *http://www.zen-cart.com/forum/index.php*

PRESTASHOP



Güvenlik özellikleri çok iyi seviyede olup, kredi kartı sistemlerini de içerisinde barındırmaktadır. E-ticaret amaçlı olarak yazılmış olan **Prestashop** da ücretsiz bir script'tir.

İlk akla gelen özelliklerini bir çırpıda saymak istersek bunları; PayPal, Kredi Kart, Havale... vb sistemleri desteklemesi, kargo hesaplama özelliği, seo uyumluluğu, sonradan ilave özellikle kazandırabilmek için haricen modül eklenebilmesi, dil desteği ve görsel tema kalitesi olarak sayabiliriz.

Resmi sitesi: *http://www.prestashop.com/*

Demo sayfası: *http://www.prestashop.com/demo/*

Wiki: *http://www.prestashop.com/wiki/*

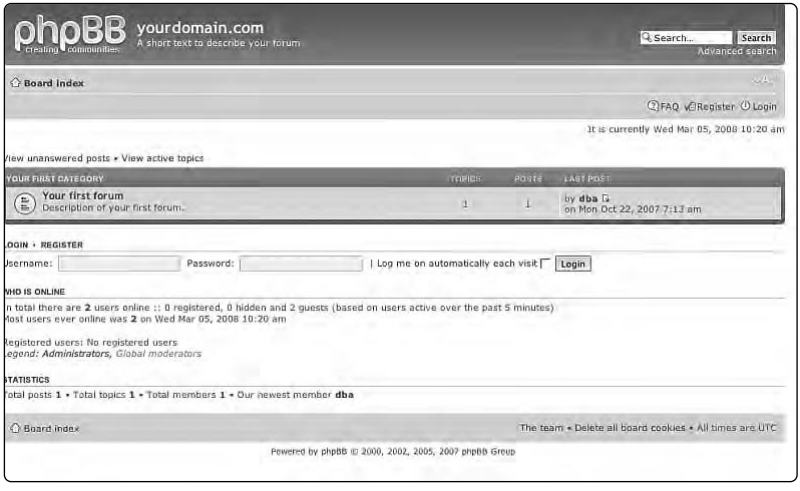
Destek Forumu: *http://www.prestashop.com/forums/*

FORUM SİSTEMLERİ

Çoklu tartışma, bilgi ve fikir alışverişi platformu olarak tarif edebileceğimiz forum sistemleri en çok ilgi gören siteler arasında yer almaktadır.

Bunların en fazla ilgi görenlerini aşağıda kısaca tanıtmaya çalıştık. Buradan yola çıkarak siz de kendi araştırmanızı yapabilirsiniz.

PHPBB3



phpBB dünyada açık kaynak kodlu forum (mesaj panosu) sistemleri arasında başı çeken birkaç sistemden bir tanesidir.

İsterseniz bazı özelliklerine şöyle bir göz atalım:

- Gruplara ve çoklu alıcıya gönderebilme özelliğine sahip özel mesajlaşma sistemi.
- Etkin şifre ve e-posta denetimli üye kayıt sistemi.
- Detaylı arama fonksiyonu
- Oy değiştirme imkanı veren detaylı anket ekleme özelliği.
- Kullanıcı grupları ve rütbeleri
- Yöneticiler için detaylı erişim izinleri tanımlayabilme
- Yeni profil alanı düzenleyebilme
- Konuları sık kullanılanlara ekleyebilme, istenildiğinde e-posta ile gönderebilme ya da yazdırabilme...

Forum kurmak istediğinizde denemeniz gereken bir sistem. İncelemenizi tavsiye ederim.

Resmi sitesi: *http://www.phpbb.com/*

Detaylı bilgi: *http://www.phpbb.com/support/*

Destek Forumu: *http://www.phpbb.com/community/*

MyBB

MyBulletinBoard (MyBB) PHP ile kodlanmış, **MySQL** veritabanı sistemi ile çalışan ücretsiz bir forum sistemidir. **Utf-8** desteğine ve **AJAX** ile hızlı cevap/düzenleme özelliğine sahiptir.

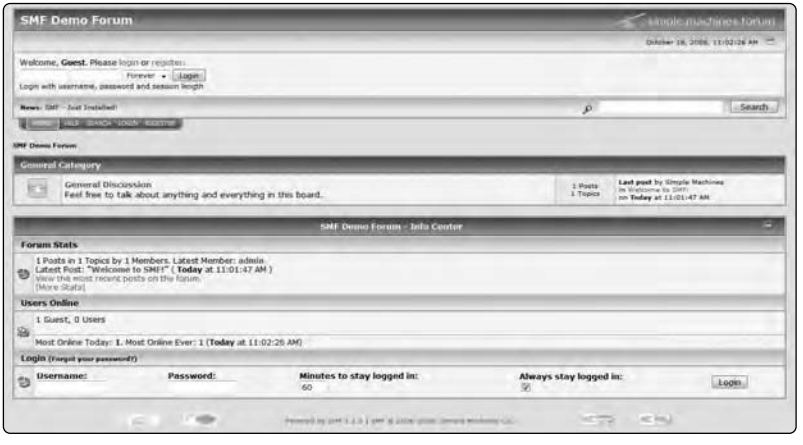
Mybb çok kullanışlı bir yapıya sahiptir. Forumu, sitenizin ihtiyaçlarına göre düzenleyebilme kolaylığı bunun en önemli sebebidir. Etkin forum ayarları ve içerik editörüyle de başarılı bir forum sistemidir.

Resmi sitesi: *http://www.mybboard.net/*

Wiki: *http://wiki.mybboard.net/index.php/Main_Page*

Destek Forumu: *http://community.mybboard.net/*

SMF



SMF (**S***imple* **M***achines* **F***orum*) hızıyla ve güvenli yapısıyla ön plana çıkan bir forum sistemidir. Şimdilerde 1.1.x ve 2.0 olmak üzere iki ayrı sürümde geliştirilmektedir.

SMF'nin geleceği olan 2.0'ın özelliklerine genel olarak bakalım:

- Üyelere, diğer üyelere ait görmek istemedikleri yeni mesajları göstermeme.
- Özel gruplar oluşturabilme. Buralara katılım için koşullar belirleyebilme.
- Üyelerin gelen mesajlarını arşivleyebilme ve gruplayabilmeleri.
- Ücretli üyelik ve bağış sistemi.
- **WYSING** editör.
- Otomatik forum bakımı.
- Arama motoru kayıtlarını inceleyebilme.
- Spam koruması.
- **PostgreSQL** ve **SQLite** veritabanı desteği.
- Gelişmiş önbellekleme sistemi.
- **Drop-Down** menü özelliği sayesinde temanıza uygunsa açılır menüler oluşturabilme.
- CSS3 uyumlu varsayılan (*default*) tema.
- Arama dostu yapı

sayabileceğimiz özelliklerden sadece birkaçıdır. Ücretsiz forum sistemleri içinde kurulum ve kullanım kolaylığı, esnek yapısı, tema ve modifikasyon desteğinin çokluğu... vb birçok sebepten ötürü SMF forumu için, mutlaka denemeniz gereken bir sistemdir diyebiliriz.

Resmi sitesi: http://www.simplemachines.org/

Türkçe destek:
http://www.simplemachines.org/community/index.php?board=76.0

Temalar: *http://custom.simplemachines.org/themes/*

Blog Sistemleri

Blog deyimi, **web** ve **log** kelimelerinin birleşiminden (*weblog*) doğmuştur. Fakat **weblog** şeklinde değil de daha ziyade **blog** şeklinde kullanıla geldi. Günümüzde **bloglar** (*günlük*) çok revaçta olan bir iletişim modeli haline gelmiş ve çeşitli türevleri ortaya çıkmıştır. Ayrıca blogların faaliyet konuları da çok çeşitlenmiş, ailevi, kişisel, edebi, ticari, sosyal katılım, teknik... vb birçok alanda başarılı pek çok örnek ortaya çıkmıştır. Biz burada, kısaca belli başlı blog sistemlerini tanıtarak bunlar hakkında bir fikir sahibi olmanızı sağlamaya çalışacağız.

b2evolution

b2evolution, bilindik tüm blog araçlarına sahip olmasının yanı sıra; çoklu dil desteği, birden çok blog oluşturabilme, gelişmiş dosya yönetim sistemi gibi özellikleriyle benzerlerinden farklılaşan , PHP ve mysql tabanlı ilginç bir blog sistemidir.

Resmi sitesi: *http://b2evolution.net/*

Bilgi ve belge: *http://manual.b2evolution.net/Main_Page*

Temalar: *http://skins.b2evolution.net/?blog=8&page=1&paged=1*

Destek Forumu: *http://forums.b2evolution.net/*

Nucleus CMS

PHP ile kodlanmış başarılı ve ücretsiz bir blog yazılımıdır. Sitenizde birden çok **skin** (*tema*) kullanmanıza olanak tanır. **RSS 2.0** ve **Atom** desteğine sahiptir. Çoklu yazar özelliği, eklentiler ile yeni özellikler kazandırılabilme... vb özelliklere sahip, hafif, esnek ve güvenli bir sistemdir.

Resmi sitesi: *http://nucleuscms.org/*

Wiki: *http://wakka.xiffy.nl/*

Demo: *http://demo.nucleuscms.org/*

Serendipity

Serendipity paket içerisinde birçok eklenti ve temaya ile gelen, **geribesleme** (*trackback*), RSS desteği... vb birçok özelliği bulunan, açık kaynak, ücretsiz, alternatif ve hızlı bir blog sistemidir.

Resmi sitesi: *http://www.s9y.org/*

Temalar: *http://spartacus.s9y.org/index.php?mode=template_all*

Destek Forumu: *http://board.s9y.org/*

WordPress



WordPress, en çok kullanılan blog sistemidir. 2001 yılına uzanan bir geçmişi olan WordPress, PHP ve MySQL altyapısı ile imal edilmiştir. Bu kadar yaygınlaşmasında, açık kaynak kodlu olması, kolay kurulum ve kullanımı ile **WordPress** kullanıcılarının ürettiği binlerce eklentiye ve temaya sahip olmasını ilk akla gelen etmenler olarak sayabiliriz.

Şimdi de bazı temel özelliklerini maddeler halinde ele alalım:

- Uzak sunucuda ve bilgisayarda (*localhost*) kurulup kullanılabilme. Kolayca sürüm yükseltebilme.
- G-zip opsiyonu ile bant genişliğinden (*bandwith*) tasarruf etme (*hostta apache mod_g-xip* kurulu ise).
- **RSS 1.0**, **RSS2.0** ve **ATOM** desteği ile blog okuyucularının içeriklere ve mp3'lere abone olabilmesi.
- Arama motoru dostu URL'ler ve etkin SEO yönetimi.
- WordPress temalarının ve dosyalarının dinamik bir biçimde Admin panelden düzenlenebilmesini sağlayan editör.
- Binlerce eklenti ve tema.
- Şifre korumalı içerikler.
- Oluşturulan içeriğin ileri ki bir tarihte otomatik olarak yayına girmesi.
- Uzun içerikleri sayfalama.
- Dosya ve fotoğraf yükleme ve bunları içeriklerde kullanma, istenildiğinde de bunları istenilen bir web sayfasının adresine link vererek bağlayabilme.
- İçerikleri kategoriler ve alt kategoriler şeklinde organize edebilme.
- Yayına hazır olmayan içerikleri çalışmalar bitene kadar taslak olarak kaydedip saklayabilme.

- Siteyi doğrudan masaüstünden güncelleyebilmeyi sağlayan masaüstü araçlarını kullanabilme.
- Gelişmiş arşivleme ve arama özellikleri.
- Yorumlarda spam'e karşı etkin mücadele.

Gördüğünüz gibi WordPress'in özellikleri saymakla bitmiyor ve bunlar sadece bir kısmı. Sitesinden daha detaylı bilgi alabilir ve dilerseniz kendi ihtiyaçlarınıza göre bir web günlüğünü kısa süre oluşturabilirsiniz.

Resmi sitesi: http://wordpress.org/

Wordpress ile yapılmış siteler: *http://wordpress.org/showcase/*

Destek Forumu: *http://wordpress.org/showcase/*

PORTAL SİSTEMLERİ

Türkçe'mize ana kapı, büyük kapı olarak tercüme edebileceğimiz portal sözcüğü, internet ortamında çeşitlilik arz eden içerikleri ve içerik türlerini bir arada barındıran (örn: haber portalları) siteler için kullanılmaktadır. Bu tarz; metin, ses, görüntü... vb farklı içerik materyalleri barındıran ve detaylı içerik yapısına sahip siteler yapabilmek için kullanabileceğimiz CMS sistemlerini de bu bölümde tanıtmaya çalışacağız.

ELXIS

Elxis, geçtiğimiz günlerde yeni bir sürüm duyurmuş olan, –2009.0 sürümü– faal bir geliştirici ekibe sahip dikkate değer portal sistemlerinden birisidir. Statik önbellek, otomatik dil algılama, tek tıklamayla indirme ve EDC yükleyici, doğrudan Admin panelden temaların CSS dosyalarını düzenleyebilme, içerik öğeleri yorum özelliği, Elxis'in yeni versiyonu olan 2009.0 Pandora'nın özelliklerden sadece bir kaçı...

Diğer yeni özellikler ise:

- İngilizce ve Yunanca'nın kurulumda gerekli dil olmaktan çıkması.
- Yeni 404 hata sayfası.
- FTP ayarlarını AJAX ile "Genel Ayarla"dan yönetme.
- Herhangi bir temaya ait CSS dosyasını doğrudan admin panalden düzenleyebilme.

- Kurulumda **htaccess.txt** dosyasının adının otomatik olarak **.htaccess** olarak değişmesi.
- Kurulumla gelen 4 ön yüz teması.
- Kurulumdan sonra **installation** klasörünün otomatik olarak kaldırılması.

Türkçeleştirilmesi tarafımdan yapılmış olan bu CMS, portal kurma projesi için ciddi bir alternatiftir. Biz burada sadece yeni versiyona ait bazı özelliklerini saydık. Sistemin temel özelliklerini resmi sitesinden öğrenebilirsiniz.

Resmi sitesi: *http://www.elxis.org/*

Türkçe Destek: *http://forum.elxis.org/index.php?board=45.0*

DRUPAL



Drupal, PHP tabanlı, açık kaynak kodlu, bir içerik yönetim sistemidir ve dünyada alanında en yaygın –Joomla ile beraber– sistemdir. Güçlü alt yapısı sayesinde kişisel günlük sitelerinden portal sitelerine kadar her türde siteyi **Drupal** ile yapabilmek mümkündür. İsterseniz hiç vakit kaybetmeden **Drupal**'in özelliklerine bakalım:

- Detaylı kullanıcı yetki seviyeleri oluşturabilme.
- İçerikleri detaylı bir şekilde sınıflandırabilme.
- Esnek ve özelleştirilebilir altyapı.
- Drupal'in yeteneklerini daha da geliştiren çok sayıda ve tamamı ücretsiz eklentiler.
- Arama dostu URL'ler.
- İndeksleyerek detaylı ve güçlü arama fonksiyonu.
- Görünümü içerikten bağımsız olarak değiştirebilme.
- Üye profillerine yeni bilgi alanları ekleyebilme.
- Kurulumla gelen standart içerik türlerinin yanı sıra değişik içerik türleri –haber, sunum...vb– oluşturabilme.
- Detaylı içerik yorum fonksiyonu.
- Taksonomi olarak adlandırılan özellik sayesinde içeriklerinizi farklı hiyerarşilerde ve kategorilerde oluşturabilme.
- Çoklu dil desteği.
- Detaylı web tabanlı raporlama sistemi.
- Varsayılan (kurulumla birlikte gelen-default) forum sistemi.
- Üyeler arası e-posta iletişimi.
- Performansı arttıran önbellekleme sistemi.
- Veriler ile verilerin sunumunu birbirinden ayıran (*node: düğüm yapısı*) katmanlı yapı.
- Üyelere kendi blog sayfalarını açma imkanı.

Saymakla bitiremediğimiz özellikleriyle gerçekten de **Drupal** bulunduğu konumu hak ediyor. Mutlaka inceleyin derim.

İşte linkler:

Resmi sitesi: http://drupal.org/

Dokümanlar: *http://drupal.org/support*

Destek Forumu: *http://drupal.org/forum*

ÖRNEK CMS İNCELEMESİ: JOOMLA

Joomla, açık kaynak kodlu, ücretsiz bir **İçerik Yönetim Sistemi**'dir. Kullanışlılığı fevkalade yüksek olan yönetim paneli ile, resim ve her türlü görsel malzemeyi kolaylıkla sisteme dahil edip düzenleyebilmenizi sağlayan yapısıyla, içeriklerinizi kolaylıkla hazırlayıp çok detaylı bir biçimde düzenleyebilmenizi sağlayan editör kullanımı ve kategorizasyon mantığıyla, esnek ve görsel kalitesi en üst düzeyde ayarlanabilen güçlü arayüz altyapısı ile, kısacası kodlama bilgisi gerektirmeden üst düzey çalışmalar ortaya koyabilmeyi mümkün kılmaktadır.

Joomla, tüm dünyada her geçen gün hızla yaygınlaşan bir yazılımdır. Joomla ile kişisel site, haber amaçlı veya genel içerikli portal, şirket sitesi, eticaret sitesi... vb bir çok uygulamayı dünyadaki ve ülkemizdeki pek çok örneğinde olduğu gibi üst düzeyde ve kolaylıkla yapmak mümkündür.

Joomla'nın mükemmel standart özellikleri yanında bir çok eklenti ile gücüne güç, yeteneklerine yetenek katabilir, ücretli ücretsiz yüzlerce şablon (tema) alternatifinden yararlanarak görsel kalitesi çok yüksek siteler hazırlayabilirsiniz.

TARİHÇE

Joomla CMS, 2005 senesinde ortaya çıktı. Joomla geliştiricileri, eski **Mambo CMS** takımda çalışırken 2005 senesinde Mambo'dan ayrılarak, Mambo çekirdek kodları üzerine **Joomla 1.0.x** serisini oluşturdular. Zamanla Mambo'dan farklılaşan Joomla, bugün artık milyonlara varan kullanıcı sayısına ulaşmış bulunmaktadır. Bu seri bu gün **1.0.15** sürümüne ulaşmış ve yerini aşağıda değindiğim yeni sürüme bırakmıştır.

Joomla.org sitesi üzerinden kullanıcıları ile buluşan Joomla, tüm dünyadan kod yazarlarının ve kullanıcıların da katkısıyla kolektif ve dinamik bir gelişim sürecine sahiptir. Bu dinamik süreç nihayaetinde Joomla **1.5.x** serisinin doğmasına yol açtı. **Joomla 1.0.x** serisi gibi Mambo kodlarını temel almayan bu seri, tamamen sıfırdan sağlam ve temiz bir kod altyapısı üzerinde geliştirildi.

Genel Terimler

Bileşen (Component)

Bileşenler, Joomla'nın temel unsurlarından birisidir. Site ziyaretçileri tarafından siteye kayıt, form, içerik...vb şekillerde veri girişi yapılacağı durumlarda veri tabanına tablolarını ekleyerek bu verileri kontrol eden, bileşen olarak adlandırdığımız bu programcıklar kullanılır. Kurumla gelen; reklam (*banner*), iletişim, haber beslemeleri (*newsfeed*), anket, arama ve web bağlantıları bileşenlerinin yanı sıra, Joomla'ya sonradan eklenebilen değişik işlevlere sahip birçok bileşen mevcuttur.

Modül (Module)

Modüller daha ziyade Joomla'nın ön yüzünde, site ziyaretçilerine değişik içerik gösterim şekilleri sunan programcıklardır. Tek başlarına programlanıp Joomla'ya yüklenerek kullanılabildikleri gibi, bileşenlere ait birtakım verilerin gösterimi için de kullanılabilirler.

Uyumlu Ek (Plug-in)

Plug-in'ler, sitenin ön yüzünde doğrudan gösterilmeyen, sitede kullanılacak verilerle ilgili birtakım kontroller için; site önyüzünde gösterilecek içerikler oluşturulurken, içeriğe birtakım işlevler ekleyebilmek için; bazı bileşenlerde yapılandırma, kontrol ve yayınlama işlevleri için ve bunlara benzer pek çok işlevi yerine getirmek üzere programlanabilen kod parçacıklarıdır diyebiliriz.

Tema-Şablon (Template)

Html, **CSS** ve **PHP** dilleri ile oluşturulmuş, ayrıca flash ve benzeri uygulamaların da entegre edilebildiği; esnek yapısıyla Joomla'nın görsel gücünü temsil eden; ön yüz şablonudur. Joomla'da sayfalarınızın bir bölümünü bir tema ile, diğer bir sayfanızı da bir başka tema ile gösterebilirsiniz. Bu yolla sitenizde çok sayıda tema kullanmanız mümkündür.

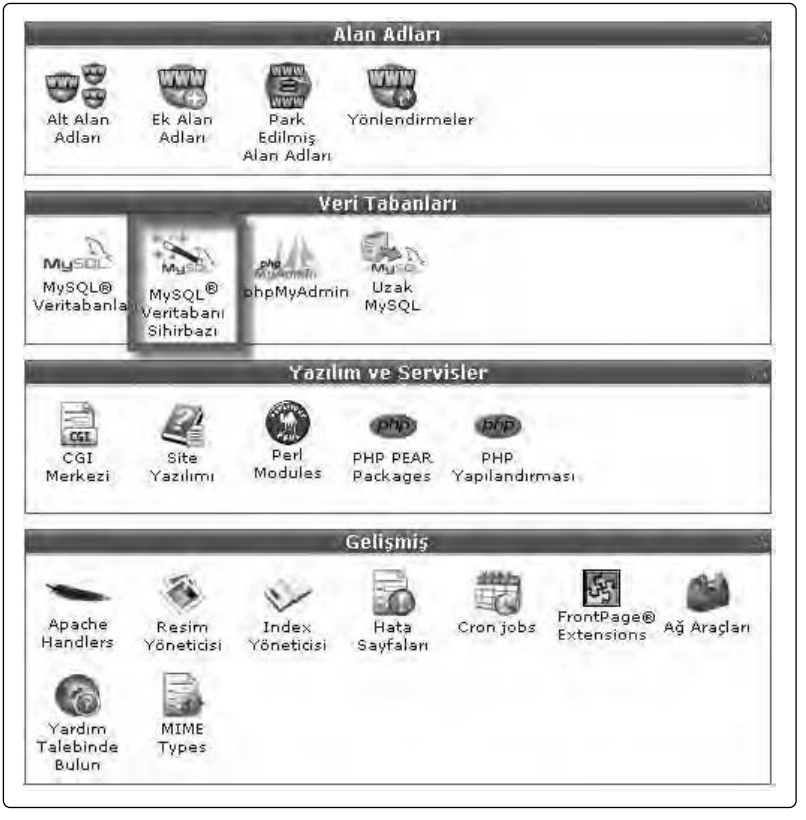
Pozisyon-Yerleşim

Pozisyon terimi, Joomla modüllerinizin, temanızda gösterilecekleri konumları ifade eder. Modül pozisyonu olarak ifade edilen bu konumlar, temaya eklenen kodlar ile oluşturulur. Joomla'da; left, right, top,

bottom... vb hazır tanımlı 27 adet modül pozisyonu bulunmaktadır. Bu tanımlı pozisyonlardan herhangi birini temanızı tasarlarken kullanabileceğiniz gibi, ayrıca kendiniz de 23 adet daha pozisyon tanımlayabilirsiniz.

CPANELDE VERİTABANI OLUŞTURMA

Öncelikle **www.siteniz.com/cpanel** yolu ile sitenin hosting kontrol panelinize ulaşıyoruz.

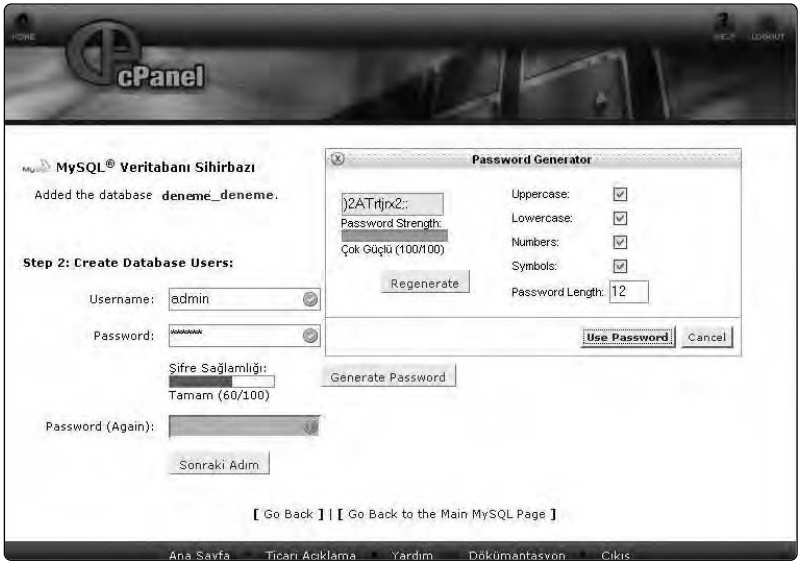


Daha sonra ise, MySQL® Veritabanı Sihirbazı ikonunu tıklayarak MySQL® Veritabanı Sihirbazı sayfasına ulaşıyoruz.



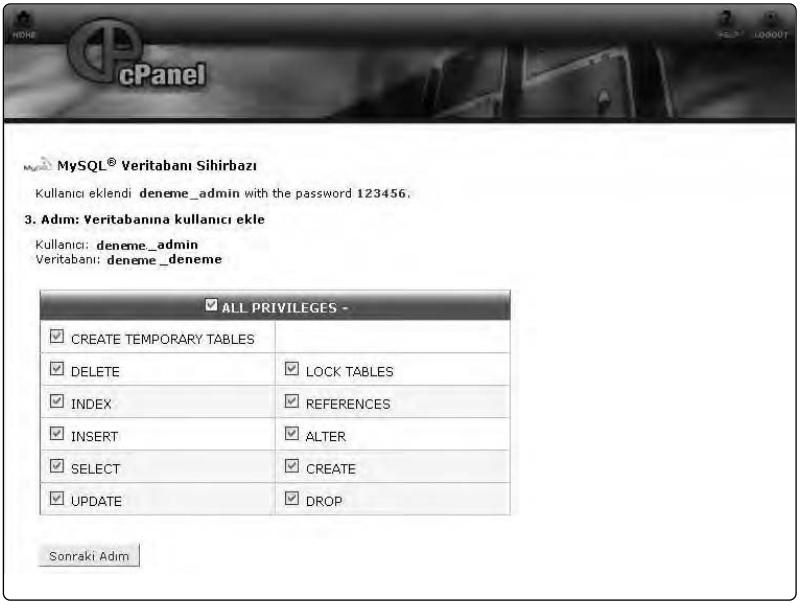
Veritabanımızı oluşturabilmek için ilk olarak, veritabanına vereceğimiz adı belirleyip **Yeni Veritabanı** kutucuğuna, bu adı yazıyoruz ve **Sonraki Adım** tuşu ile bir sonraki aşamaya geçiyoruz. Oluşturduğumuz veritabanın ismi, belirlediğimiz **veritabanı adı** (örneğimizde deneme) ve **cPanel** kullanıcı adından oluşur. (cPanel kullanıcı adını ve veritabanı adını deneme olarak belirlediğimizden, veritabanı adının tam şekli deneme_deneme şeklinde olacaktır.)

Bu ayrıntının üzerinde önemle durmamızın nedeni, ileride Joomla kurulumu esnasında veritabanı adını deneme olarak değil **cPanel** ön eki ile beraber **deneme_deneme** şeklinde girmemiz gerektiği içindir.



Resimde de gördüğümüz üçüncü aşamada ise oluşturduğumuz veri tabanına bir kullanıcı adı ve parola atayacağız. Belirlediğimiz veritabanı kullanıcı adına da **cPanel** ön eki eklenecektir. (Örneğimizde *deneme_admin*)

Burada dikkat edeceğimiz husus, sitemizin güvenliği açısından harfler, rakamlar ve işaretlerden oluşan güçlü bir şifre belirlemek olacaktır. Bu konuda dilersek **cPanel** bize yardımcı olmaktadır. **Generate Password** tuşu ile **Password Generator** penceresini açarak, buradan çok güçlü şifreler oluşturabiliriz.



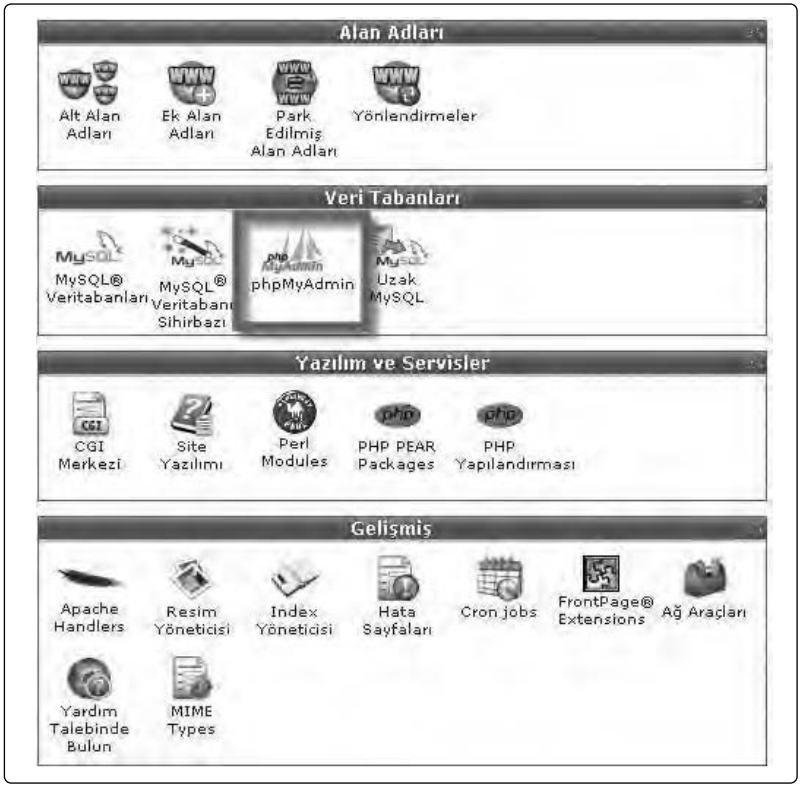
Daha sonraki aşamada ise, oluşturduğumuz veritabanı kullanıcısı için (örneğimizde admin) yetki tanımlaması yapacağız. Veritabanında her türlü işlemi yapabilmemiz için resimde gördüğünüz **ALL PRIVILEGES** kutucuğunu tıklayarak tüm işlem türlerini seçiyoruz. Aksi takdirde veritabanımızda seçilmeyen işlem türünde yetki sahibi olmayacağımızdan herhangi bir işlem yapmamız mümkün olmayacaktır.

Örnek vermek gerekirse, **CREATE** kutucuğu tıklanmamışsa veritabanı üzerinde yeni bir tablo oluşturmamız mümkün olmayacaktır. Bir veritabanına birden çok kullanıcı oluşturarak, her birine bu tablo yardımıyla değişik yetki düzeyleri atayabileceğimizi de hatırlatarak bir sonraki aşamaya geçelim.

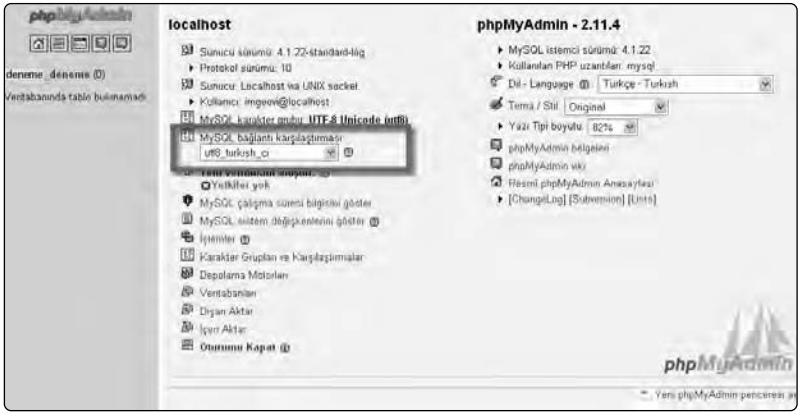
Bu sayfa bize, kullanıcı oluşturma ve yetkilendirme işleminin başarı ile tamamlandığını bildirmektedir.



Eğer yeni bir veritabanı ya da yeni bir veritabanı kullanıcısı oluşturmayacak isek, **Return to Home** linki ile **cPanel** anasayfasına dönebiliriz. Joomla kurulumuna geçebilmek için yapmamız gereken son biri işlem kaldı.



cPanel ana sayfasında **phpMyAdmin** ikonuna tıklayarak oluşturduğumuz veritabanını yönetmemizi sağlayan **phpMyAdmin** sayfasına gidiyoruz.



Buradan, **MySQL Bağlantı Karşılaştırması** değerini, **Joomla 1.5.x** Türkçe karakter seti olan **utf8_turkish_ci** olarak seçiyoruz...

Artık Joomla kurmaya hazırız...

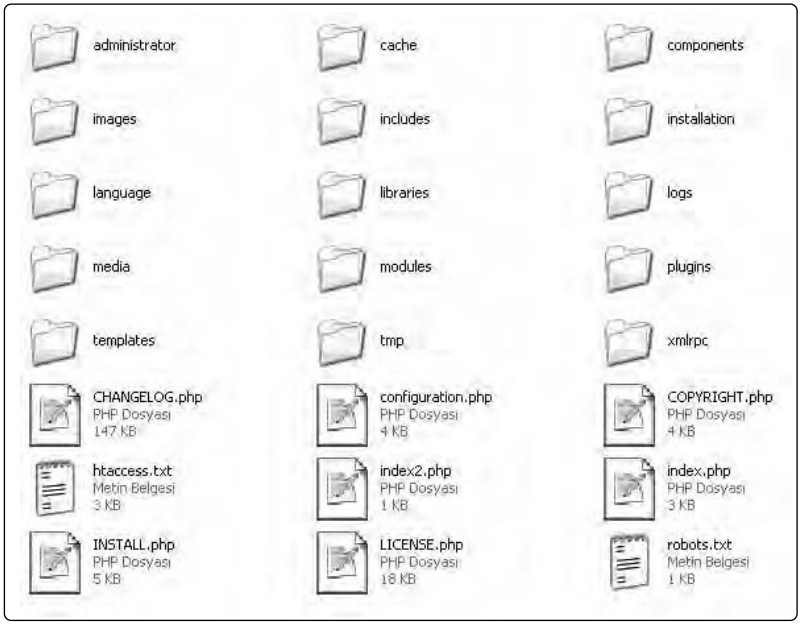
Uzak Sunucuda (cPanel) Joomla Kurulumu

Kurulum aşamalarına geçmeden önce kısaca, Joomla ile kuracağınız sitenizden azami verim alabilmek için gereken özelliklerden bahsedelim. Barınacağınız sunucunun Linux tabanlı bir sunucu olmasına, daha sonradan birçok eklenti yüklemeniz gerekeceğini de düşünerek, **gd2**, **imagemagick**, **domxml**, **cgi**, **zlib** özelliklerinin bulunmasına, sucuda **mysql5 PHP5** ve **Apache 2.2** sürümlerinin kullanılıyor olmasına dikkat edin.

Joomla kurulumu esnasında da göreceğiniz gibi; sunucuda Register **Globals** ve **Safe Mode** ayarlarının **KAPALI** olmaları, dikkat etmeniz gereken bir başka önemli husus olmaktadır.

Burada bahsettiğimiz bazı özellikler olmadan da Joomla kurup site hazırlayabilmeniz mümkündür. Fakat saydığımız bu özellikleri taşıyan yerli ve yabancı birçok host firması mevcut olduğundan, bu özelliklerin hiçbirinden taviz vermemenizi öneririz...

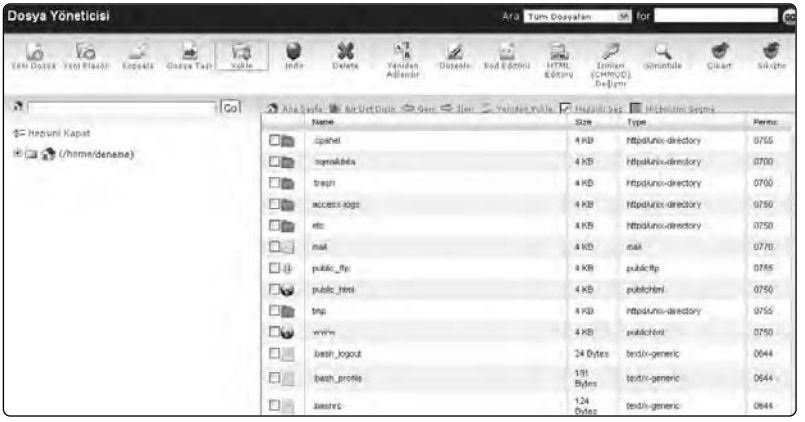
Öncelikle, **www.joomla.org/download.html** adresinden son sürüm Joomla'yı indirerek işe başlayalım. İndirdiğimiz **Joomla 1.5.x** kararlı sürüm paketini zip'ten çıkaralım.



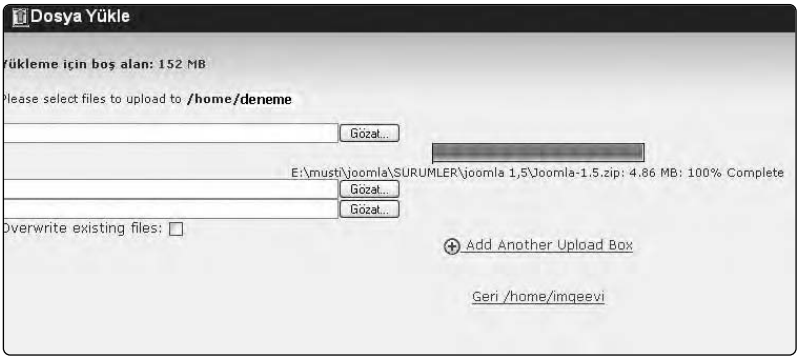
Daha sonra, **Joomla 1.5.x** klasörlerini ve dosyalarını bir **FTP** programı (*Filezilla*, *CuteFTP*... vb) ile sunucumuzdaki ana dizine (*public_html*) gönderelim.

Joomla'yı sunucuya göndermenin ikinci yöntemi ise; cPanel yoluyla zip'li olarak göndermektir. Bu ikinci yöntemde, dosyalar zip'li olduğu için, sunucuya birinci yönteme göre hem daha hızlı kopyalanacak hem de eksik dosya gönderimi ihtimali ortadan kalmış olacaktır.

Bunun için sunucumuzun **cPaneline** giriyoruz ve **Dosya Yöneticisi** ikonunu tıklıyoruz.



Resimde de görüldüğü gibi dosya yönetim sayfasına ulaşıyoruz. Sayfanın üst kısmındaki **Yükle** ikonuna tıklayarak yükleme sayfasına geliyoruz. **Gözat** tuşu ile bilgisayarımızdaki zip'li Joomla'yı seçip yükleme işlemine başlıyoruz.



Yükleme tamamlandıktan sonra dosya yönetim sayfasında zip'li dosyamızı görebileceğiz. Son olarak sayfanın üst kısmındaki **Çıkart** ikonuna ardından da açılan kutucukta **Extract File(s)** tuşuna tıklayarak dosyalarımızı zip'ten çıkarıyoruz.

Ve artık beklenen an geldi.



Tarayıcımızın adres satırına **www.siteniz.com** yazarak kuruluma başlıyoruz. Kurulumun bu ilk aşamasında sitemizi hangi dilde kuracağımızı seçeceğiz. Biz, tr-TR-Turkish (Türkiye) seçip sonraki tuşu ile devam ediyoruz.



Kurulumun ikinci aşaması olan Kontroller aşamasında sunucumuzun özelliklerini ve önerilen ayarların neler olduklarını görüyoruz.

Bunlardan configuration.php "Yazılabilir" ibaresinin karşısında "Evet" seçeneğinin olması, kurulumda oluşturulacak bilgilerin bu dosyaya oto-

matik olarak ekleneceği anlamına gelecektir. Joomla sitemizin genel ayarlarını barındıracak olan configuration.php dosyasının karşısında "Hayır" ibaresi var ise bu, dosyanızın yazılamaz olduğu anlamına gelmektedir. Bu durumda kullandığınız FTP programı ile ya da sunucunuzun Kontrol panelinden bu dosyanın **chmod** değerini değiştirmelisiniz.

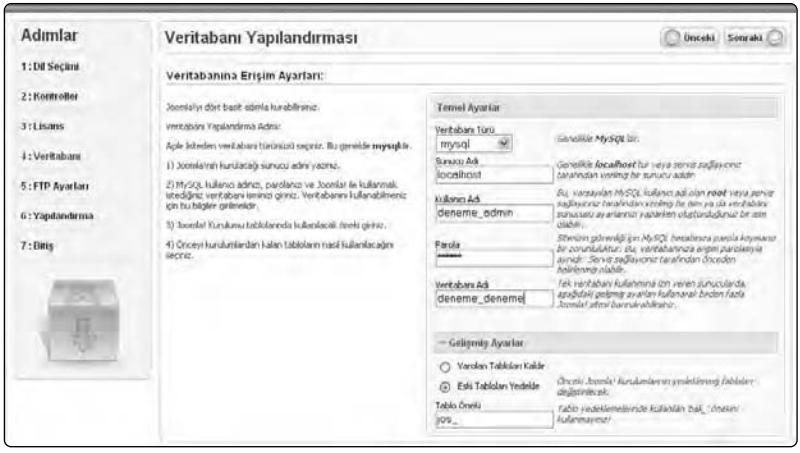
İkinci bir yöntem ise, kuruluma aynen devam etmek ve kurulumun son aşamasında size bir pencere içinde verilecek kodları kopyalayıp, Joomla ana dizininizde var olan **configuration.php-dist** dosyasını Wordpad gibi bir text düzenleme programında açtıktan sonra, içindeki tüm kodları silip, kopyalamış olduğunuz kodları yapıştırmak ve bu dosyayı **farklı kaydet** seçeneğiyle kaydedip ismini de **configuration.php** olarak değiştirmek olacaktır.

Önerilen ayarların hemen karşısındaki **Güncel** ibaresi, mevcut ayarları göstermektedir. Bu bölümdeki tüm ayarların önerildiği gibi olması, Joomla'nın daha sağlıklı çalışmasını sağlayacaktır. Ancak burada ilk etapta dikkat edeceğimiz husus, Güvenli Mod durumunun önerildiği gibi Kapalı olmasıdır. Aksi halde Joomla'nız düzgün kurulsa da çalışması problemli olacaktır.



Bu aşamayı da **Sonraki** tuşu ile geçip, Lisans sayfasına geliyoruz. Joomla Lisansı ile ilgili bilgilerin bulunduğu bu aşamadan sonra ise; Joomla sistemine ait veritabanı tablolarının kurulumu aşamasında, verita-

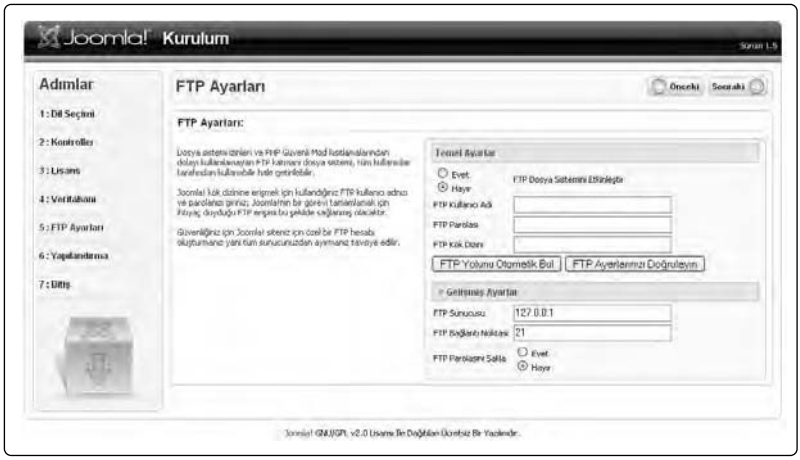
banımız ile ilgili bilgilerin girileceği **Veritabanı Yapılandırması** sayfasına geliyoruz.



Bu aşamada; veritabanımızın türünü, sunucu adını, veritabanı oluştururken belirlediğimiz veritabanı kullanıcı adını, parolayı ve veritabanı adını giriyoruz.

Eğer bu veri tabanını ilk kez kullanıyorsak, **Gelişmiş Ayarlar** bölümünde herhangi bir işlem yapmayacağız. Fakat, bu veritabanında daha önceden kullandığınız bir Joomla yüklemesi var ise ve mevcut tabloların silinmesini istiyorsak, **Varolan Tabloları Kaldır** radyo düğmesini, eski tabloları kaldırırken aynı zamanda bunların yedeklenmesini istiyorsak da **Eski Tabloları Yedekle** radyo düğmesini işaretlemeniz gerekecektir.

Joomla'nın genel veritabanı tablo ön eki **jos_**'tur. Eğer bunu değiştirmek isterseniz bu alandan istediğiniz tablo önekini belirleyebilirsiniz. Tavsiyemiz buraya 3 harfli ve Türkçe karakter içermeyen başka bir ön ek belirlemenizdir. Bu güvenlik açısından veritabanı yoluyla yapılan hack girişimlerinin sonuçsuz kalmasına –çoğu saldırgan haliyle burada standart ön eki hesaba katarak saldıracağı için– büyük katkı sağlayacaktır. Bu özellik aynı zamanda; farklı tablo ön ekleri belirleyerek bu veritabanına birden fazla Joomla ya da bir başka PHP betik kurmanıza olanak tanır.



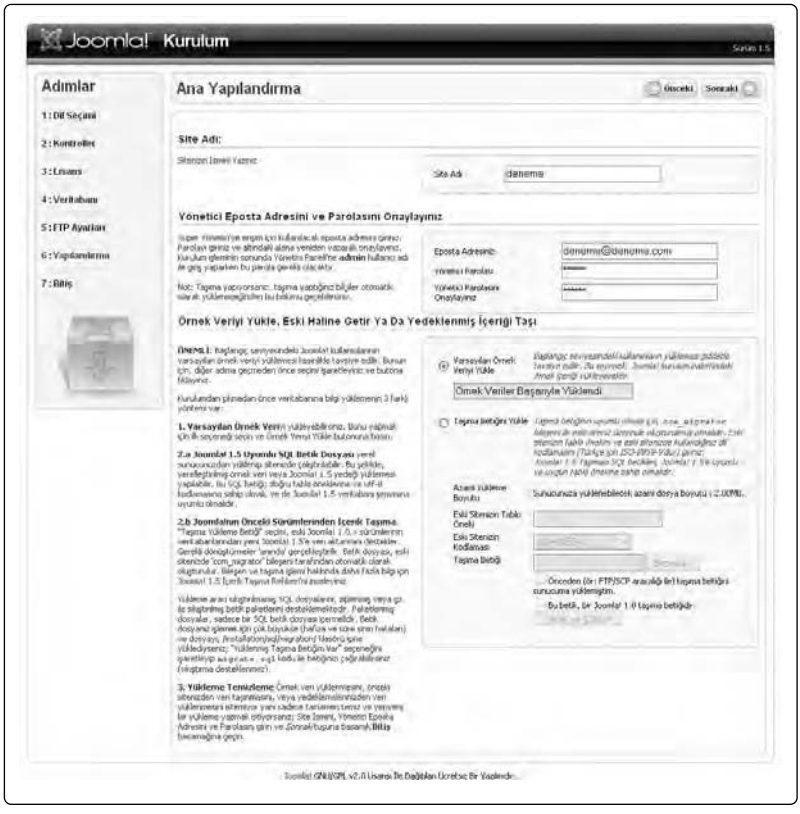
Dilerseniz şimdi de bir sonraki **FTP Ayarları** aşamasını inceleyelim.

İlk belirtmek istediğimiz husus, dilersek bu aşamayı **Sonraki** tuşu ile geçip kurulum işleminden sonra Joomla yönetim panelinden de bu bilgileri oluşturabileceğimizdir.

Eğer buradaki ayarları atlamak istemiyorsak, **Evet** radyo düğmesini işaretleyerek işe başlamalıyız. Gerekli bilgileri girdikten sonra herhangi bir soruna mahal vermemek için **FTP Ayarlarınız Doğrulayın** butonu ile doğrulama işlemi yapmanızı öneririz. Fakat bilgilerin doğruluğundan eminseniz doğrula işlemi yapmayabilirsiniz.

Alttaki Gelişmiş Ayarlar kısmında ise FTP sunucusunun ip adresi ve FTP işlevi için gerekli port numarası yer alır. Eğer girilen FTP parolasının sistem tarafından saklanmasını ve bu şekilde parolayı tekrar girmek zorunda kalmamak isterseniz **FTP Parolasını Sakla** seçeneğini "Evet" olarak seçmelisiniz.

Şimdi de mutlu sona ulaşmamız için geçmemiz gereken bir sonraki **Ana Yapılandırma** aşamasına geçelim.



Burada ilk olarak, sitenizin görünen adını (kullanıcıların tarayıcılarında en üstte görünecek ve sitenizi tanımlayacak isim); daha sonra ise yöneticinin yani sizin e-posta adresinizi ve Joomla yönetim panelinize girmek için kullanacağınız parolayı belirliyorsunuz.

Eğer ilk kez kurulum yapacaksanız Joomla'yı daha rahat kullanmanız için **Örnek Veriyi Yükle** butonunu tıklayarak Joomla'yı kullanmanızı kolaylaştıracak örnek verileri sisteminize yüklemenizi tavsiye ederiz. Fakat bunun bir zorunluluk olmadığını da hemen belirtelim.

Bu noktada üzerinde durulması gereken husus ise; bu kurulumu **Joomla 1.0.X** sürümlerinden **1.5.X** sürümüne geçmek için yapıyorsanız, **Migration** (*terfi*) işlemini kurulumun bu aşamasında yapmanız gerekti-

ğidir. Bunun için ilk olarak **Taşıma Betiğini Yükle** radyo düğmesini seçmeli, daha sonra da eski sitenize ait veritabanı önekini ve eski sitenizin karakter kodlamasını ilgili kutucuklara girmelisiniz.

Son olarak da **Migration** işlemiyle oluşturduğunuz dosyayı bilgisayarınızdan **Gözat** tuşu ile bulacaksınız. Artık tek yapmanız gereken **Yükle ve Çalıştır** butonuna tıklamak ve eski sürüme ait bilgileri kurmakta olduğunuz yeni sürüme aktarmak...

Ve yine ama bu sefer son kez sihirli **Sonraki** butonuna basıyoruz.



Ve işte karşımızda Bitiş çizgisi!

Son olarak, resimde gördüğünüz uyarıda istendiği gibi Joomla kurulum dosyaları ve klasörleri içinde bulunan **installation** klasörünün ya adını değiştireceğiz ya da bu klasörü tamamen sileceğiz.

Artık hazırız.

Önümüzde iki seçenek var. İstersek **Site** butonuna basarak sitemize; istersek de **Yönetici** butonuna basarak resimde de görülen Joomla! **Yönetici Girişi** sayfasına ulaşabiliriz.



Joomla! Yönetici Girişi sayfası.

JOOMLA GENEL AYARLAR

Joomla yönetim paneli girişi için, tarayıcınızın adres satırına site adresinizden sonra **/administrator** yazıp girebilirsiniz. (Örnek: *www.siteismi.net/administrator*)

Karşınıza ilk olarak aşağıdaki Yönetici giriş sayfası çıkacak. Burada, kurulum esnasında belirlediğiniz kullanıcı adını ve şifrenizi girerek yönetim paneline ulaşmış olacaksınız.



JOOMLA 1.5.X GENEL YAPILANDIRMA AYARLARI (SİTE SEKMESİ)

Bu bölümde yapılan ayarlar, ana dizindeki configuration.php dosyasında tutulur. Bu ayarları değiştirebilmesi için, **configuration.php** dosyasının yazılabilir olması gerekmektedir.

Genel Yapılandırma Ayarları'na: **Yönetim Merkezi>Site>Genel Yapılandırma** yoluyla ulaşılır.



Site Ayarları

Site Kapalı: Siteniz yapım aşamasındayken yahut bakıma aldığınızda siteyi ziyaretçilere kapatır.

Kapalı Uyarısı: Site kapalıyken, ziyaretçilere gösterilecek metin.

Site Adı: Sitenin, tarayıcılarda görüntülenecek ismi.

Varsayılan WYSIWYG Düzenleyici: İçerikleri girmek için kullanılacak yazı editörünü belirler.

Liste Uzunluğu: Admin panelinde, hangi sayfada olursa olsun, sayfalarda listelenecek öğelerin sayısını ayarlamaya yarar. Bu değeri 50 ya da 100 olarak ayarlayarak, ilgili öğeleri daha az sayfada görme imkanına kavuşmuş oluruz.

Besleme Uzunluğu: Beslemelerde gösterilecek içerik sayısı. En fazla 100.

Meta Veri Ayarları

Genel Site Meta Açıklaması: Sitenizin arama motorlarında görünecek tanıtım metnidir. Bu sebeple sitenizi en iyi ifade eder şekilde oluşturulmalı ve 20 kelime civarında olmalıdır.

Genel Site Meta Anahtar Kelimeleri: Site içeriğinizi ifade eden kelime ve kelime gruplarından oluşturulur. Kelimeler, (virgül) ile birbirlerinden ayrılır.

Başlık Meta Etiketi Göster: İçerik başlıklarının, meta etiketleri ile birlikte tarayıcıda gösterilmesi içindir. "Evet" olarak ayarlanması tavsiye edilir.

Yazar Meta Etiketi Göster: İçerik sahiplerinin adlarının içerik başlıkları ile beraber görüntülenmesi içindir.

SEO Ayarları

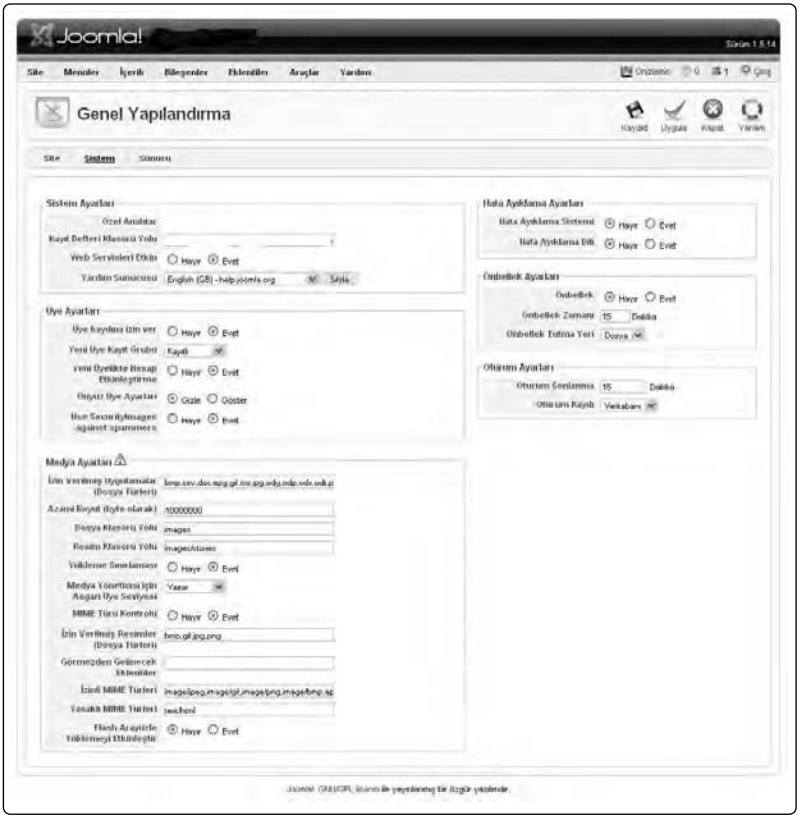
Arama Motoru Dostu Adresleme: Site URL'lerini anlamlı ve html uzantılı hale çevirir.

Apache mod_rewrite özelliğini kullan: Bu özellik sunucunuzda varsa **Evet** seçeneğini işaretleyip site URL'lerinizin belli bir şablona göre standart bir yapıya kavuşmasını sağlayabilirsiniz. Yani böylece her site

URL'si kafasına göre takılmaz. Tüm URL'ler aynı standartlara göre arama dostu hale getirilir. Bu özelliği aktif ettiğinizde; htaccess.txt dosyasının adını .htaccess olarak değiştirmelisiniz.

Adreslere son ek ekle: URL'lere son ek ekler. Örneğin, içeriklerinizin URL'lerine **.html** son eki ekler.

Joomla 1.5.x Genel Yapilandirma Ayarlari (Sistem Sekmesi)



Sistem Ayarları

Özel Anahtar: Kurulum aşamasında oluşturulur. Güvenlik için Admin girişlerini doğrulama amaçlıdır.

Kayıt Defteri Klasörü Yolu: Sistem kayıtlarının (*log*) tutulacağı konumu belirlemek içindir.

Web Servisleri Etkin: HTTP ortamında, XML ile RPC (uzak işlem çağrıları) işlemi yapabilmek için "Evet" seçilmelidir. Aksi takdirde bazı eklentileri sitenize kurmak ve çalıştırmak istediğinizde erişim hatası alırsınız.

Yardım Sunucusu: Admin paneldeki **Yardım** menüsünü kullanmak istediğinizde, kurum esnasında yüklenen **Help** dosyalarının mı yoksa uzak sunucu bağlantısı ile destek sitesinden mi yardım alınacağını belirlemek için kullanılır.

Üye Ayarları

Üye kaydına izin ver: Sitenize üye kaydı istemiyorsanız **Hayır** seçeneğini seçili hale getirin.

Yeni Üye Kayıt Grubu: Ziyaretçilerinizin, sitenize üye olduklarında hangi üyelik seviyesine atanacaklarını belirler. Kurulumda, **Kayıtlı** seçeneği aktiftir ve genelde lazım olan seçenek de bu seçenektir. Fakat bazı sitelerde sitelerin özelliği gereği, (örn: sadece davetiye usulü yazı yazmaları için davet edilen kişilerin üye olacağı bir site ise) kişilerin kayıt olur olmaz içerik ekleyebilmeleri ya da aynı zamanda içerikleri yönetebilmeleri için diğer seçenekler (yazar editör yayıncı) de seçilebilir.

Yeni Üyelikte Hesap Etkinleştirme: "Evet" olursa, üye olmak isteyenlerin e-posta adreslerine Joomla otomatik olarak e-posta gönderir. Gelen maildeki aktivasyon linkini tıklamadan üye olunamaz. Bu sayede kişilerin e-posta adreslerini doğrulamış oluruz.

Önyüz Üye Ayarları: Site üyelerinin, üye girişi yaptıklarında, kişisel ayarlarını düzenleyebilmelerine olanak verir.

Medya Ayarları

İzin Verilmiş Uygulamalar (Dosya Türleri): Yüklenmesine izin verilecek dosya türleri aralarına virgül koyularak buradan belirlenir.

Azami Boyut (byte olarak): Her bir yüklemede azami dosya boyutunu ifade eder. (byte cinsinden) 0 değeri kullanılırsa boyut sınırlaması olmadan yükleme yapılabilir.

Dosya Klasörü Yolu: Medya dosyalarının sunucuda tutulacağı dizin.

Resim Klasörü Yolu: Resimlerin bulunacağı alt klasörün yolu.

Yükleme Sınırlaması: Manager yetki seviyesinin altındaki kullanıcılara dosya yükleme sınırı getirebilmek için, "evet" seçilmelidir.

MIME Türü Kontrolü: Seçtiğiniz dosya türleri dışındaki dosya türlerinin yüklenmesini engellemek için, "Evet" seçili olmalıdır.

İzin Verilmiş Resimler (Dosya Türleri): İzin vermek istediğiniz resim dosya türlerini buradan belirleyin.

Görmezden Gelinecek Eklentiler: Özellikle yüklenmesini istemediğiniz dosya türlerini, aralarına virgül koyarak buraya girin.

İzinli MIME Türleri: İzin verilen MIME türlerini buraya girin.

Yasaklı MIME Türleri: Engellemek istediğiniz MIME türlerini de buraya girin.

Flash Arayüzle Yüklemeyi Etkinleştir: "Evet" seçerek bu özelliği de aktif etmeniz önerilir.

Hata Ayıklama Ayarları

Hata Ayıklama Sistemi: Dil, SQL tasarım hatalarını görmenizi sağlar. Fakat site yayında iken açarsanız bu hatalar site ön yüzünde de görüleceğinden "Hayır" seçilmelidir. Fakat sitenin yapım aşamasında ya da bakımda olduğu dönemlerde açık olması faydalı olacaktır.

Hata Ayıklama Dili: Dil hata ayıklaması içindir. Yukarıdaki açıklama bu seçenek için de geçerlidir.

Önbellek Ayarları

Önbellek: Önbelleklemenin (*cache*) etkin olup olmayacağını belirlemek içindir.

Önbellek Zamanı: Önbelleğin kaç dakikada bir yenileneceğini gösterir. Fazla uzun tutulmaması önerilir.

Önbellek Tutma Yeri: Önbellek kayıtlarının bir dosyada tutulacağını ifade eder.

Oturum Ayarları

Oturum Sonlanma: Buraya dakika cinsinden gireceğiniz değer, bir kullanıcının herhangi bir işlem yapmadan bağlı kalabileceği süreyi ifade eder.

Girdiğiniz süre dolduğunda üye bir işlem yapmak istediğinde, (sistem tarafından bu sürenin sonunda dışarı atıldığından) tekrar giriş yapması gerekir. Güvenlik açısından bu sürenin uzun tutulması önerilmez.

Oturum Kaydı: Çerezlerin veritabanında tutulması sayesinde kimlik saptamasının sürekli olarak yapılabilmesi sağlanmış olur.

Joomla 1.5.x Genel Yapılandırma Ayarları (Sunucu sekmesi)



Sunucu Ayarları

Temp Klasörü Yolu: Bazı geçici dosyaların tutulduğu klasördür. Kurulumda otomatik olarak oluşturulur. Kullanılabilir olması için yazılabilir olması gerekir.

GZIP Sayfa Sıkıştırma: Dosyalarınızı sıkıştırarak sisteminizin hızlanmasına ciddi oranda bir katkı sağlar. Sunucunuz destekliyorsa bu özelliği mutlaka aktif edin.

Hata Raporlama: İstediğiniz seçeneği kutucuktan seçebilirsiniz. "Sistem varsayılanı" uygun seçenektir.

Yerel Ayarlar

Zaman Dilimi: Bulunduğunuz bölgeye göre seçim yaparsanız, içeriklerinizin kaydında ve gösteriminde doğru saat dilimini kullanmış olursunuz.

FTP Ayarları

FTP Etkin: FTP'den yükleme yapmak istediğinizde, bu özelliği etkinleştirmelisiniz. Etkinleştirdiğiniz takdirde ise diğer alanları eksiksiz olarak ve doğru biçimde doldurmalısınız.

FTP Sunucusu: Doğru adresi, sunucu yönetim panelinizden öğrenebilirsiniz.

FTP Bağlantı Noktası (Port): FTP portu olan 21 olmalıdır.

FTP Kullanıcı Adı: FTP sunucunuza bağlanırken kullandığınız kullanıcı adı.

FTP Parolası: FTP sunucunuza bağlanırken kullandığınız şifre.

FTP Erişim Dizini: FTP dizin yolu.

Veritabanı Ayarları

Veritabanı Türü: Sunucunuzda yapılandırılan veritabanı türü, Bunu sakın değiştirmeyin.

Sunucu Adı: Genellikle localhost'tur.

Kullanıcı Adı: Veritabanınızı oluştururken belirlediğiniz kullanıcı adı.

Veritabanı: Veritabanınızı oluştururken belirlediğiniz veritabanı adı.

Veritabanı Ön Eki: Varsayılan değer **Jos_**'tur. Buradan değil, ama Joomla'yı kurarken kurulum esnasında başka bir ön ek de belirleyebilirsiniz. Veritabanı ayarlarınızı buradan değiştirmeye sakın kalkışmayın. Aksi takdirde sisteminizin veritabanına ulaşması mümkün olmayacaktır.

Posta Ayarları

Posta Aracı: Sitenizden yapılacak posta gönderimlerinde kullanılacak gönderim seçeneğidir. Varsayılan olarak "PHP posta işlevi" seçilidir. Ve bu şekilde kalmalıdır.

Gönderen adresi: Buraya e-posta adresinizi gireceksiniz.

Gönderen Adı: Buraya genellikle site adı yazılır. Fakat kendi adınızı da yazabilirsiniz.

NOT

Sendmail ve **SMTP** sunucunuz tarafından destekleniyor ise kullanılabilir. Bunları kullanacaksanız, sunucunuzun yöneticilerinden gerekli bilgileri alabilirsiniz.

Joomla'da İçerik Planlaması (Bölüm Kategori İçerik Mantığı)

Joomla'nın bu kadar beğenilen bir içerik yönetim sistemi olmasındaki en önemli faktörlerden birisi de, çok başarılı bir içerik planlama yapısına sahip olmasıdır.

Joomla, kademeli bir gruplama yöntemiyle içeriklerinizi en etkin bir biçimde tasnif etmenizi sağlar. Böylece site ziyaretçileri, site içerikleri arasında kaybolmadan aradıkları bilgiye kolaylıkla ulaşabilirler.

Bu aşamada dilerseniz, Joomla'nın **Bölüm-Kategori-İçerik** terimleriyle adlandırdığımız içerik yönetim yapılanmasını bir örnekle açıklamaya çalışalım.

Kitaplığınızdaki kitapları raflara rast gele koyduğunuzu var sayarsak; bir süre sonra aradığınız kitabı bulmanızın bir hayli zor olacağı açıktır. Oysa ki bu kitapları önce türlerine göre, daha sonra da aynı türden kitapları konularına göre tasnif ederek kitaplığınıza o şekilde yerleştirseniz; gerektiğinde aradığınız kitabı çok daha rahat bulabilirsiniz. (örneğin: önce Edebiyat, Bilgisayar... vb şeklinde türlerine göre sonra da Bilgisayar>İşletim sistemleri; Bilgisayar>Web tasarım kitapları... vb konularına göre tasnif edilebilir.)

Bu örneği Joomla'ya uyarladığımızda; Bilgisayar, Bölümü; Web tasarım kitapları, Kategori'yi; Bir web tasarım kitabı ise İçerik Öğesi'ni ifade etmektedir.

Burada belirtmemiz gereken bir diğer husus da; tek bir tasnif yönteminin söz konusu olmadığıdır. Joomla'da tasnif amacınıza göre, içeriklerinizin yapısına göre ya da herhangi bir başka kritere göre dilediğiniz gibi planlama yapabilirsiniz. Bizim örneğimizi ele alacak olursak; kitapları konularına göre değil de mesela yazarlarına göre de sınıflayabiliriz.

Bu noktada tavsiyemiz; öncelikle sitenizin ihtiyaçlarına göre iyi düşünülmüş sağlam bir içerik planlamasını kağıt üzerinde yaparak işe başlamanızdır.

Şimdi dilerseniz yandaki örnek içerik planlamasını inceleyelim ve buna göre sitemizde bu yapıyı birlikte oluşturalım.

Ekonomi Haberleri
Dünyadan
Haber 1
Haber 2
Yurttan
Haber 1
Haber 2
Spor Haberleri
Futbol
Haber 1
Haber 2
Basketbol
Haber 1
Haber 2

Örneğimizde; Ekonomi Haberleri ve Spor Haberleri olmak üzere iki bölümümüz, bölümlerimizin altlarında ikişer kategorimiz (Dünyadan, Yurttan, Futbol, Basketbol) her kategorinin altında ise içeriklerimiz (Haber 1; Haber 2) bulunmaktadır.

Artık; Joomla yönetim paneline gidip, bölümlerden başlayarak planlamamızı uygulamaya geçirebiliriz.

Bölüm Oluşturma

Yönetim Paneli>İçerik>Bölüm Yöneticisi yolu ile bölüm yönetimine ulaşıyoruz. Yeni tuşu ile bölüm oluşturma sayfasına giriyoruz. Bölüm adımızı (Ekonomi Haberleri) ilgili kutucuğa girip açıklama kısmına ise dilersek bölüm hakkında açıklama yazıyoruz.

Bölüm açıklaması olarak metin oluşturup, oluşturduğumuz metni, metin editörü sayesinde istediğimiz gibi düzenleyebiliriz. Dilersek buradan bölüme resim de ekleyebiliriz.

Ayrıca **Bölüm Erişim Seviyesi** olarak üç ayrı seçeneğimiz olduğunu da belirtelim. **Genel Yayın** seçeneğini tercih ettiğimiz takdirde tüm ziyaret-

çiler bu bölümü görebilirler. **Kayıtlı** seçeneğini tercih ettiğimiz takdirde ise bu bölümü sadece sitemizdeki kayıtlı üyeler göreceklerdir. **Yönetici** seçeneğini tercih ettiğimizde ise oluşturduğumuz bölümü (Bizim örneğimizde Ekonomi Haberleri) sadece site yöneticileri görebileceklerdir.

Son olarak işlemler bittiğinde kaydederek çıkmayı unutmuyoruz...

Kategori Oluşturma

Aynı işlemleri; **Yönetim Paneli>İçerik>Kategori Yöneticisi** linkini tıklayarak gerçekleştirip kategorimizi oluşturuyoruz. Birinci işlemden tek farklı nokta, Bölüm kutucuğundan, oluşturmakta olduğumuz kategorinin (Dünyadan), hangi bölümün altında (Ekonomi Haberleri) yer alacağını seçmemiz olmaktadır.

İçerik Oluşturma

Artık içeriklerimizi (Haber1) oluşturduğumuz bölüm ve kategorilere göre girebiliriz. İçerik oluşturma konusunu da ayrı başlık altında inceleyeceğiz.

Bu noktada Joomla içerik planlamasının can damarı olan menü öğelerinden de bahsetmemiz gerekmektedir. Çünkü oluşturduğumuz bu içerik planlamasını şu an biz site yöneticileri sitenin yönetim panelinde yapılandırdık ve sadece yönetim panelinde görebilmekteyiz.

İçeriklerimizin site ön yüzünde, bu planlamamıza göre yer almasını ise menüler vasıtası ile sağlıyoruz. Menü konusu başlı başına bir konu olduğundan, bu konuyu ayrı bir başlık altında inceleyeceğiz.

Takip ettiğiniz gibi yukarıda önce bölüm, kategori, içerik mantığını anlattık. Ardından da örnek bir bölüm ve kategori oluşturduk. Şimdi ise oluşturduğumuz bu bölüme ve kategoriye bağlı bir içerik oluşturacağız.

İlk olarak, yönetim paneli üst menüdeki **İçerik>İçerik Yöneticisi** linkinden aşağıdaki resimde gördüğünüz **İçerik Yöneticisi** sayfasına ulaşalım. Resimde de gördüğünüz gibi, sayfa boş.

NOT

Joomla'nın bir güzelliği de yeni kullanıcılara kolaylık olması için kurumda örnek veri yükleme özelliğidir. Bu sayede yeni kullanıcılar yönetim paneli kullanımını ve işin mantığını daha kolay kavrayabilmektedirler. -örnek verinin nasıl yüklendiği kurulum aşamasında anlatılmıştı- Biz, bu aşamada karışıklığa sebebiyet vermemek için örnek verileri sildik.

Genel İçerik Ayarları

Bu noktada içerik oluşturmaya başlamadan önce içeriklerimizde geçerli olacak genel içerik ayarlarının neler olduğuna bakalım. Bunun için, sayfamızın üst kısmındaki tuşlardan Ayarlamalar tuşuna tıklayalım. Karşımıza aşağıdaki pencere gelecek...

Şimdi tek tek bu ayarları inceleyelim:

Yetkilendirilmemiş Bağlantıları Göster: Yetkisiz kullanıcılara -ziyaretçilere- içerik linklerinin gösterilip gösterilmeyeceği belirleyebilmeniz içindir. "Evet" seçerseniz, sitenizi ziyaret edenler içeriklerin linklerini görebilirler. Fakat ziyaretçilerin sitenize üye olmaları sizin için önemli ise bu ayarı "Hayır" olarak seçmeniz önerilir.

İçerik Başlığını Göster: İçeriklerin başlıklarının gösterilip gösterilmeyeceğini belirleyebilmeniz içindir.

İçerik Başlığı Bağlantısı: İçerik başlıklarının linkli olmasını ve tıklandıklarında içeriğin gelmesini istiyorsanız "Evet" seçeneğini tercih etmelisin.

Giriş Metnini Göster: İçeriklerin giriş kısmının gösterilip gösterilmeyeceğini belirleyebilmeniz içindir.

Bölüm İsmi: İçeriğin dahil olduğu bölümün isminin gösterilip gösterilmeyeceğini belirleyebilmeniz içindir.

Bölüm Başlığı Bağlantısı: Sayfada gösterilen bölüm başlığına o bölüme dahil olan içeriklerin listesine giden bir bağlantının eklenip eklenmeyeceğini belirleyebilmeniz içindir.

Kategori Başlığı: İçeriğin dahil olduğu kategorinin başlığının gösterilip gösterilmeyeceğini belirleyebilmeniz içindir.

Kategori Başlığı Bağlantısı: Sayfada gösterilen kategori başlığına o bölüme dahil olan içeriklerin listesine giden bir bağlantının eklenip eklenmeyeceğini belirleyebilmeniz içindir.

Yazar İsmi: İçerikte yazar ismini "Göster" ya da "Gizle"

Oluşturulduğu Tarih: İçerikte içeriğin oluşturulma tarihini "Göster" ya da "Gizle"

Güncelleme Tarihi: İçerikte son güncellenme tarihini "Göster" ya da "Gizle"

Dolaşım özelliğini göster: İçerikler arasında geçiş özelliğini (öncekisonraki) "Göster" ya da "Gizle"

Devamını oku... Bağlantısı: İçerikte devamını oku linkini "Göster" ya da "Gizle"

İçerik Değerleme/Oylama: İçerikte oylama özelliğini "Göster" ya da "Gizle"

Simgeler: "Göster" derseniz; yazıcı, pdf, e-posta için simge; "Gizle" derseniz metin gösterilir.

PDF Simgesi: İçeriklerde PDF simgesinin gösterilip gösterilmeyeceğini belirleyebilmeniz içindir.

Yazdırma Simgesi: İçeriklerde yazıcı simgesinin gösterilip gösterilmeyeceğini belirleyebilmeniz içindir.

E-posta simgesi: İçeriklerde e-posta simgesinin gösterilip gösterilmeyeceğini belirleyebilmeniz içindir.

Tıklamalar: İçeriklerde içeriğin tıklanma sayısının gösterilip gösterilmeyeceğini belirleyebilmeniz içindir.

Her bir besleme öğesi için gösterilecek olan: Besleme öğelerinin sadece giriş metninin mi yoksa metnin tamamının mı gösterileceğini buradan belirleyebilirsiniz.

Filtreleme Seçenekleri: Siteniz, kullanıcıların gireceği özel HTML kodları vasıtası ile saldırıya maruz kalabilir. Filtreleme sitenizi bu saldı-

rılardan korumak için geliştirilmiş bir yöntemdir. Filtreleme özelliği size HTML kodları üzerinde daha fazla kontrol sunar.

Sitenizin gereksinimlerine göre bu filtreleme özelliği ile HTML kodları üzerinde çok sıkı ya da daha düşük düzeyde bir kontrol oluşturmayı tercih edebilirsiniz. Burada herhangi bir ayarlama yapmazsanız, varsayılan ayarlar geçerli olacaktır ki, bu da yaygın kullanılan tipteki ataklar için yeterli olacaktır.

Filtreleme Türü: Varsayılan Kara Liste, `applet`, `body`, `bgsound`, `base`, `basefont`, `embed`, `frame`, `frameset`, `head`, `html`, `id`, `iframe`, `ilayer`, `layer`, `link`, `meta`, `name`, `object`, `script`, `style`, `title`, **xml tag'larını** ve `action`, `background`, `codebase`, `dynsrc`, `lowsrc` **özelliklerini** içerir. Yani yukarıda da belirttiğimiz gibi baya kapsamlı ve yeterli bir varsayılan listesi mevcut.

Html Kullanma seçeneğini tercih ederseniz, içerik kaydedilirken içerikte kullanılan tüm tag'lar silinecektir.

Etiketleri Filtrele: Filtrelenmesini istediğiniz ilave etiketler varsa, bunları aralarına virgül koyarak bu alandan girebilirsiniz.

Özellikleri Filtrele: Filtrelenmesini istediğiniz ilave özellikler varsa, bunları aralarına virgül koyarak bu alandan girebilirsiniz.

İçerik Ekleme

Evet, artık içeriğimizi eklemeye hazırız. Bu aşamada, **İçerik Yöneticisi** sayfasında iken **Yeni** tuşuna tıklayarak aşağıdaki resimde gördüğünüz **Yeni içerik** sayfasını açıyoruz.



Sayfayı incelediğinizde de göreceğiniz gibi, adeta bir ofis programında çalışıyormuş kadar rahatlıkla her türlü görsel özelliği ihtiva eden içerikleri bu sayfadan **TinyMCE** editör aracılığıyla oluşturup yayına verebileceksiniz.

Sayfa Bilgileri

Başlık: İçeriğin başlığı.

Rumuz: Sef özelliğini genel ayarlardan açtığınızda, buraya gireceğiniz rumuz, sayfanın URL'sinde kullanılacaktır. Dilerseniz bu alanı boş bırakabilirsiniz. Boş bırakırsanız, içeriği kaydederken Joomla bunu otoma-

tik olarak düzenleyecektir. Dilerseniz de kendiniz girebilirsiniz. Gireceğiniz rumuz küçük harflerden oluşmalı ve kelimeler arasında boşluk bırakılmamalı. Boşluk yerine – (çizgi) işareti kullanılmalıdır.

Yayınlandı: İçeriği yayınlar ya da yayından kaldırır.

Ön Sayfa: İçeriğin anasayfada gösterilip gösterilmeyeceğini belirler.

Bölüm: İçeriğin dahil olacağı bölüm.

Kategori: İçeriğin dahil olacağı kategori.

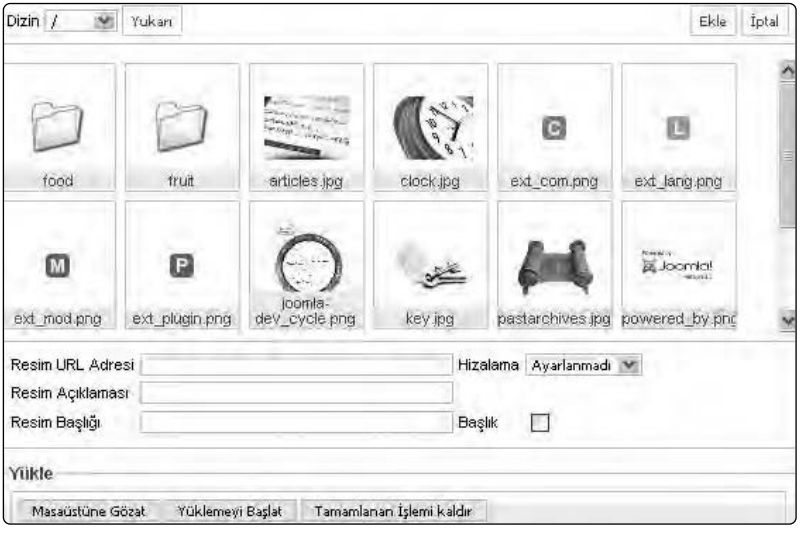
NOT

Joomla'da içerikler muhakkak bir bölüm ve kategoriye bağlı olmak zorunda değildir. Çeşitli nedenlerle (Ani bir duyuru, ana sayfada sabit bir içerik ... vb) kategorize etmek istemediğiniz içerikleri sabit içerik olarak tasnif etmek için Bölüm ve Kategori kutucuklarında "Kategorilenmemiş" seçeneğini tercih etmeniz yeterli olacaktır.

Resim, Ek Sayfa, Devamını Oku... tuşları ve İşlevleri



Şimdi de dilerseniz yazı editörünün hemen altında gördüğünüz bu üç tuş ve işlevlerini inceleyelim.



Adından da anlaşılacağı gibi içeriklere resim eklemek içindir.

Resimler için Joomla ana dizinde bulunan **images>stories** dizini kullanılır. Tabi buraya siz resimlerinizi daha iyi organize edebilmek için başka alt dizinler de açabilirsiniz. İşte tüm bu dizinleri bu kutucuktan bulup bu sayfada ekrana getiriyoruz.

Dizinimizi seçince o dizindeki resimlerin önizlemeleri bu pencerede görüntülenir. Ve siz içeriğe eklemek istediğiniz resmi seçip, pencerenin sağ üst köşesindeki **Ekle** tuşu ile içeriğinize ekleyebilirsiniz. Bir resmi seçtiğinizde **Resim URL Adresi** alanında resmin konumunu görebilirsiniz. Bir alt kutucuktan resme açıklama, **Resim Başlığı** kutucuğundan ise fare ile resmin üzerine gelindiğinde görünecek açıklamayı girebilirsiniz. Hemen sağındaki kutucuktan ise resmin hizalama işlemini yapabilirsiniz.

Eğer kullanmak istediğiniz resim henüz hostunuza yüklü değilse öncelikle **Masaüstüne Gözat** tuşu ile resmi bilgisayarımızdan bulup, Yüklemeyi **Başlat** tuşu ile de yüklüyoruz.

Ek Sayfa

İçerikleri sayfalara bölmek için, fare imlecini sayfanın bölüneceği kısmın bittiği yere tıklayıp yazı editörünün altındaki **Sayfa Sonu...** tuşuna basalım. Açılacak **pop-up** pencerede **Sayfa Başlığı**'nı ve **İçeriklerin diğer isimlerinin tablosu**'nu (Sayfa-2, Sayfa-3) doldurup kaydedelim. Bu işlemi istediğim kadar tekrar ederek içeriğimizi bir çok sayfaya ayırabiliriz.

Devamını Oku...

Uzun bir içeriğin tamamını anasayfada göstermek hiç de hoş olmayacağından ve içerikler aşağı doğru uzayıp gideceğinden, içeriklerin **giriş kısımlarını** (*intro*) anasayfada gösterip diğer kısımlarını içeriğin kendi sayfasında göstermek için, fare imlecini anasayfada görünecek giriş kısmının bittiği yere tıklayıp yazı editörünün altındaki **Devamını Oku**... tuşuna basalım. İmleci konumlandırdığımız yerde kırmızı bir çizgi belirecektir. Böylece, bu çizginin altında kalan içerik kısmını anasayfada göstermemiş olacağız.

TinyMCE editör



Şimdi de Joomla'nın varsayılan editörü olan **TinyMCE** editörünü kısaca tanıyalım.

Birinci satırda: Metin düzenleme tuşları yer almaktadır. Kalın, italik, harflerin altını ve üstünü çizme tuşları, ayrıca metin hizalama tuşları, yazı stili ve font tipi seçimleri de bu satırdan yapılmaktadır.

İkinci satırda: Madde imleri, ileri geri tuşları, resim, link, HTML ekleme gibi tuşlar yer alır.

Üçüncü satırda: Daha çok metin biçimlendirme tuşları –tablo ve CSS özellikleri ile ilgili tuşlar– yer almaktadır.

Çoğuna ofis programlarından aşina olduğunuz bu tuşları zamanla deneyerek çok daha etkin bir biçimde kullanabileceksiniz.

Şimdi de sayfanın sağ tarafındaki ayarlara geçelim. İçerik ayarlamaları kısımından; Yazar adını ve isterseniz yazara rumuz belirleyebilirsiniz. Ayrıca **Erişim Seviyesi**'ni de (Genel Yayın, Kayıtlı, Yönetici seçeneklerinden herhangi birini) yine buradan belirleyeceksiniz. Bir de dilerseniz **Yayını Bitir** kısmından içeriğin yayından kaldırılacağı bir tarih belirleyebilirsiniz. Belirlediğiniz tarih geldiğinde içerik otomatik olarak yayından kaldırılacaktır.

Gelişmiş Ayarlamalar bölümden ise, oluşturduğunuz içeriğe has parametreleri belirleyebilirsiniz. Bu içeriğe has olarak görüntülenmesini istediğiniz bir parametre yoksa bu kısma hiç dokunmaya bilirsiniz. Bu takdirde varsayılan olarak sunulan **Geneli Kullan** seçeneği geçerli olacaktır. Bunun anlamı, bu içerikte genel içerik ayarlarında belirlediğiniz parametrelerin geçerli olacağıdır.

Meta Veri Bilgileri alanını doldurmak opsiyoneldir. Dilerseniz boş da bırakabilirsiniz. Fakat arama motorlarında buraya gireceğiniz veriler içeriğinizin daha net sınıflandırılmasını sağlayacağından, dolayısıyla da önemli bir avantaj sağladığından bu alanları doldurmanızı özellikle öneririz.

Tanım: Meta açıklaması. İçeriğinizi ifade eden bir cümle olmasına özellikle dikkat edin.

Anahtar Kelimeler: Keyword dediğimiz içeriğinizle alakalı olan anahtar kelimeleri aralarına virgül koyarak bu alandan giriniz.

Botlar: İçerik indeksleyen programlar olan robotlar için özel bir keyword belirleyebilirsiniz.

MENÜLER VE MENÜ YÖNETİMİ

Öncelikle bu konuda kullandığımız genel tabirlerden söz ederek konumuza giriş yapalım. Menü tabirini, içinde birçok menü öğesi bulunan bir menü öğe grubu olarak tarif edebiliriz. (Örn: Ana Menü, Diğer Menü, Üye Menüsü) Menü öğesi ise, menüyü oluşturan her bir öğeye verilen addır. (Örn: Anasayfa, ...vb)

MENÜLER

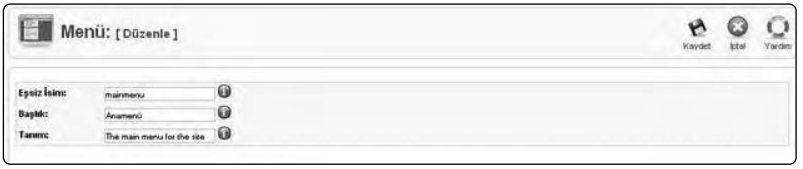
Bu girizgahtan sonra, **Menüler/Menü Yöneticisi**'ni tıklayarak menü yöneticisi sayfasını açalım ve menülerimizi oluşturma ve yönetme işlemlerini öğrenmeye başlayalım.



Yukarıdaki resimde, kurulumla gelen (Anamenü, Üye Menüsü...vd) menüleri görüyorsunuz. Bu sayfa bize, buradaki menülere bağlı menü öğeleri hakkında, (yayında, yayın dışı, çöpte...vd) birtakım bilgiler sunmaktadır. Burada, menü isimlerini tıklayarak ya da üstteki **Düzenle** tuşuna basarak menü düzenleme sayfasına girebilirsiniz. Ayrıca **Menü Öğeleri** sütunundaki ikonları tıklayarak o menüye bağlı menü öğelerinin listesini gösteren ve menü öğelerinin tüm ayarlarının yapıldığı **Menü Öğesi Yöneticisi** sayfasına ulaşabilirsiniz.

Yeni bir menü oluşturmanız gerektiğinde de yine buradan üstteki **Yeni** tuşu ile açacağımız sayfadan menünüzü oluşturabilirsiniz.

Şimdi **Başlık** sütunundan **Anamenü** adlı menüye tıklayarak menünün düzenleme sayfasına girelim.



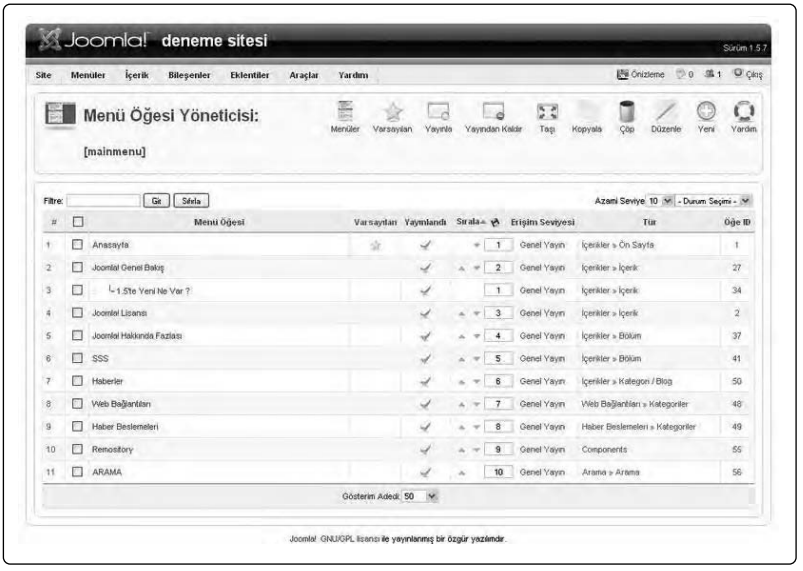
Eşsiz isim: Joomla'nın bu menüyü tanımlayacağı isimdir. Kodlar içinde yer alacağından, boşluk ve Türkçe karakter içermemelidir.

Başlık: Menü yöneticisi sayfasında, sitemizde, admin panel "Menüler" menüsünde... vd gerekli olan yerlerde menümüzü temsil eden addır.

Tanım: Menümüzün açıklaması.

MENÜ ÖĞELERİ

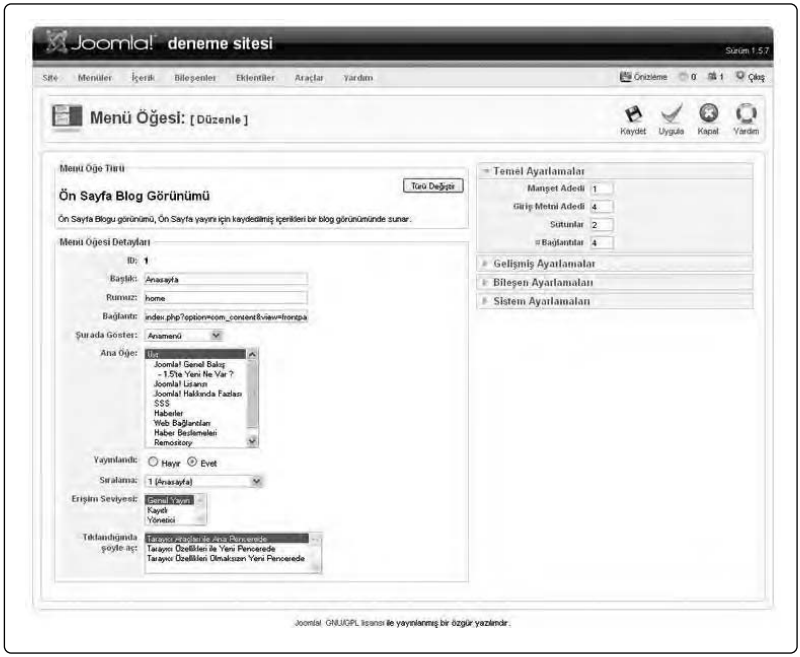
Menüler>Anamenü'yü tıklayarak **Anamenü** adlı sitemizde kullanılan temel menüye girelim.



Bir önceki sayfadaki resimde **Anamenü**'ye ait menü öğelerini görmektesiniz. Resimden de anlayacağınız gibi bu sayfa bize menü öğeleri hakkında öncelikle bilgi veren (Menü öğeleri, yayın durumu, kullanıcı erişim seviyesi, menü öğesinin türü... vd) bir sayfadır. Asli görevleri ise, (söz konusu görevleri sağ üstteki simge-tuşlar ile yapar) yeni menü öğesi oluşturmak, mevcut bir menü öğesini yeniden düzenlemek ya da yayından kaldırmak, bir menü öğesini çöpe göndermek ya da başka bir menüye taşımak... vd görevlerdir.

Bu noktada; her sitede bir ana sayfa menü öğesinin bulunması gerektiğini, bu öğenin o menüye ait (genellikle anamenü) menü öğeleri listesinde ilk sırada olması gerektiğini ve sitenin ana sayfası olarak kullanılabilmesi için varsayılan olarak (üstteki Varsayılan tuşu ile) işaretlenmesi gerektiğini belirterek, menü öğelerini daha yakından incelemeye geçebiliriz.

İncelemek için **Anasayfa** menü öğesini seçtik. Menü öğesinin adına tıklayarak (bu işlemi üstteki Düzenle simgesi ile de gerçekleştirebiliriz) öğenin düzenleme sayfasına girelim.



Dilerseniz bu aşamada, sayfamızı detaylı olarak inceleyelim.

Menü Öğe Türü: Joomla'da, çok sayıda menü öğe türü mevcuttur. Bunları müstakil bir başlık olarak inceleyeceğimiz için burada detaylarına girmiyoruz. Yukarıdaki resimden bizim menü öğemizin (Anamenü) menü öğe türünün "Ön Sayfa Blog Görünümü" olduğunu görüyoruz. Gerektiğinde menü öğemizin türünü **Türü Değiştir** tuşunu kullanarak değiştirebiliriz.

ID: Menü öğemizin, bu menü öğesine özel kimlik numarasıdır.

Başlık: Menü öğemizin adı.

Rumuz: Menü öğemizin site URL'sinde yer alacak adıdır. Burada birden çok kelime kullanabiliriz. Fakat, kelimelerin arasında boşluk değil – (tire işareti) koymalı ve Türkçe karakter kullanmamalısınız. Dilerseniz bu kısmı boş da bırakabilirsiniz. Boş bırakırsanız, Joomla buraya otomatik olarak bir rumuz atayacaktır.

Bağlantı: Menü öğesinin URL'si. Bu kısım Joomla tarafından otomatik olarak oluşturulur.

Şurada Göster: Bu menü öğesinin hangi menüde yer alacağını buradan seçiyoruz.

Ana Öğe: Bu menü öğesi, bir ana öğe olacaksa bu kutucuktan "Üst" ibaresi seçili olmalıdır. Eğer alt menü Joomla buraya otomatik olarak, hangi menü öğesinin alt menü öğesi olacaksa, bu kutucuktan o menü öğesini seçmelisiniz ki, oluşturacağımız menü öğesi, seçtiğiniz menü öğesinin alt menü öğesi olsun.

Yayınlandı: Menü öğemizi, menümüzde yayına vermek ya da yayından kaldırmak içindir.

Sıralama: Menü öğemizin, menüde kaçıncı sırada olacağını bu kutucuktan belirliyoruz.

Erişim Seviyesi: Menü öğemize kimlerin erişebileceğine buradan karar veriyoruz. "Genel" seçilirse, menü öğemizi sitemize giren herkes görebilir. "Kayıtlı" seçilirse menü öğemizi sadece siteye giriş yapan üyeler

görecektir. Mesela üye Menüsü oluştururken bu özellikte menü öğeleri oluşturacağız. "Özel" tercih ettiğimizde ise menü öğemiz sadece yöneticilere açık olacaktır.

Tıklandığında şöyle aç: Menü öğemiz bir ziyaretçi tarafından tıklandığında, menü öğemizin bağlı olduğu içerik sayfasının tarayıcıda açılış tarzını buradan seçiyoruz.

Menü Öğe Parametreleri

Hemen sayfanın sağ tarafına göz attığımızda ise, menü öğemize ait parametreleri görmekteyiz. Burada yer alan parametreler her menü öğesi türü için birtakım farklılıklar içerse de çoğunlukla benzer olduğundan biz yine Anamenü örneğimize devam edeceğiz.

Temel Ayarlamalar

Manşet Adedi: Anasayfada gösterilecek manşet içerik -tüm içerik alanını enine kullanan- adedini buradan belirleyebilirsiniz.

Giriş Metni Adedi: Anasayfada giriş metni kısmı gösterilecek içerik adedini ifade eder.

Sütunlar: Giriş metinlerinin kaç sütunda gösterileceğini belirler.

Bağlantılar: Anasayfada içeriklerin altında kaç tane içeriğin linkinin gösterileceğini belirler.

Olayı daha iyi anlamak için bir örnek vermek gerekirse; örneğin anasayfamızda 1 manşet, 2 sütunda 2 haber ve en altta da 4 haberin linki olsun istiyorsak, yukarıdaki kutucuklara sırasıyla, 1-2-2-4 sayılarını girmemiz gerekir.

Gelişmiş Ayarlamalar

Kategori Sırala: Öğelerin kategorik sıralanma şeklini belirler. Varsayılan olarak bir altta açıklayacağımız birincil sıralamayı kullanma seçeneği aktiftir.

Birincil Sıralama: Öğelerin birbirleri arasında hangi sıralama ölçütü ile sıralanacağını belirler. Açılır kutudan istediğinizi seçebilirsiniz. "Varsayılan" seçeneğinin kalması "En yeniler önce" seçeneğinin uygulanaca-

ğı anlamına gelmekte bir başka ifade ile en bilindik ve tutarlı sıralama ölçütü olmaktadır.

Multi Column Order: Çift yada daha fazla sütun sayısı seçildiğinde içeriklerin sıralama düzenini belirler. Örneğin, **Aşağıya** (*Down*) radyo düğmesini tıkladığımızda üçüncü içerik ikinci içeriğin altında, **Karşı** (*Across*) radyo düğmesini tıkladığımızda ise üçüncü içerik ikinci içeriğin yanında yer alacaktır.

Sayfalandırma: Sayfanın altında yer alan ve diğer sayfalara geçişi sağlayan linkli sayfa numaralarının gösterilip gösterilmeyeceğini belirler.

Sayfalandırma Neticeleri: Toplan sayfa sayısı gösterimi.

Besleme Bağlantısı Göster: İçeriklerde, besleme (RSS) bağlantılarının gösterilip gösterilmeyeceğini belirler.

Bileşen Ayarlamaları

Buradaki ayarlar, "genel içerik ayarları" bölümünde anlattığımız ayarlardır. Her bir öğe için ayrı ayrı ayar yapmaya uğraşmak yerine, genel içerik ayarlarını kullanmanızı öneririz. Fakat bazı durumlarda herhangi bir öğenizin, site genelinde kullandığınız ayarlardan farklı olarak özel ayarlamalara gereksinim gösterdiği durumlarda buradaki kısmı kullanmanız gerekecektir.

Sistem Ayarlamaları

Sayfa Başlığı: Sitenizin tarayıcılarda görünecek başlığıdır.

Sayfa Başlığını Göster: Sayfa başlığını **Göster/Gizle**.

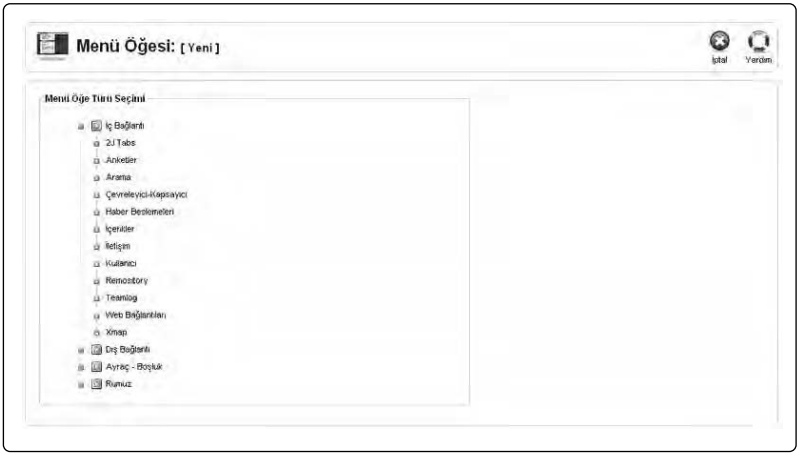
Sayfa Sınıf Son Eki: Genel tasarımınızdan farklı olarak sayfanıza farklı tasarım özellikleri kazandırmanız gerekirse, bu sayfaya özel oluşturduğunuz CSS kodlarını bir clsss eki ile çağırmanızı sağlar.

Menü Resmi: Menülerinize resim eklemenizi sağlar. Açılır pencereden resim seçebileceğiniz gibi ebatlar menünüze uygun istediğiniz bir resmi images>stories klasörüne atarak da kullanabilirsiniz.

SSL Uygunluğu: Varsayılan olarak "Yok say" işaretlidir. SSL kullandığınız takdirde bu seçeneği aktif edin.

Menü Öğe Çeşitleri

Öncelikle, menü öğemizi hangi menüde oluşturacaksak, (Örn: Anamenü) denetim masasında Menü/menü (örn: Anasayfa) yoluyla o menünün Menü Öğesi Yöneticisi'ne girelim. Üstten **Yeni** simgesine tıkladığımızda karşımıza aşağıdaki **Menü Öğe** sayfası gelecektir.



Buradan seçimimizi yaparak menü öğemizi oluşturacağız. Şimdi hep birlikte bu sayfadaki menü öğe türlerini tanıyalım.

Resimde de görebileceğiniz gibi ilk ana menü öğe türümüz, **İç Bağlantı** menü türü olmaktadır. Bu grupta içerikleri menüye bağlamak için kullanacağımız türler, kurulumla gelen eklentiler ve bizim sonradan Joomla'ya eklediğimiz eklentiler yer almaktadır. (Siz Joomla'ya yeni bir eklenti yüklediğinizde otomatik olarak burada listelenecektir.)

Biz bu bölümde, içerikle ilgili olanları göreceğiz. Kurulum ile gelen eklentilerle alakalı olanlara ise konu bütünlüğünü sağlayabilmek için, kitabımızın kurulumla gelen eklentileri incelediğimiz bölümlerinde değineceğiz. Zaten bu kısmı bitirdiğinizde olayın özünü de kavramış olacağınızdan birçoğunu kendiniz de inceleyerek rahatlıkla çözebilirsiniz.

Sözü daha fazla uzatmadan, listedeki İçerikler linkine tıklayarak devam edelim. Linke tıkladığımızda karşımıza aşağıdaki ağaç şeklindeki yapı gelecektir.



Resimden de anlaşılacağı üzere, içerik öğe türüne bağlı bir çok alt tür bulunmaktadır.

İÇERİKLER

Arşivle/Arşivlenmiş İçerik Listesi: İçeriklerinizi arşivlemek içindir.

İçerik/Tek İçerik Görünümü: Bir içeriğin tamamını müstakil olarak gösterir.

İçerik/İçerik Ekleme Alanı: Üyelerin içerik eklemesini sağlayan öğe türü'dür.

NOT

İçerik eklemesini istediğiniz üyelere **Author** ya da daha üst bir yetki vermeyi unutmayın.

Kategori/Kategori Blogu Görünümü: Bu tür, bir içerik kategorisini blog görünümünde görüntülemek içindir.

Kategori/Kategori Listesi Görünümü: Bu tür ise; bir içerik kategorisini liste halinde gösterir.

Ön Sayfa/Ön Sayfa Blog Görünümü: Ön sayfada yayınlanan tüm içerikleri blog görünümünde yayınlamak için kullanılır.

Bölüm/Bölüm Blogu Görünümü: Bu tür, bir içerik bölümünü blog görünümünde görüntülemek içindir.

Bölüm/Bölüm Görünümü: Bir içerik bölümünü bölümün içerdiği kategorilerin listesi şeklinde görüntülemek içindir.

Dış Bağlantı-Dış bağlantı öğe türü: Başka bir web sitesinin her hangi bir sayfasına bağlantı yapan bir menü öğesi oluşturmak içindir.

Ayraç-Boşluk: Bu menü öğe tipi, menüde bir boşluk ya da ayraç oluşturur. Uzun menüler için düşünülmelidir.

Rumuz: Bu menü öğe tipi, başka bir menü öğesine bağlantı oluşturur. Bunun amacı; iki ya da daha fazla menü öğesini özdeş –aynı ayarlara sahip– olarak kullanabilmektir. Öğelerin ayarları, bağlantı verilen öğeninki gibi olacaktır.

Üye Yönetimi

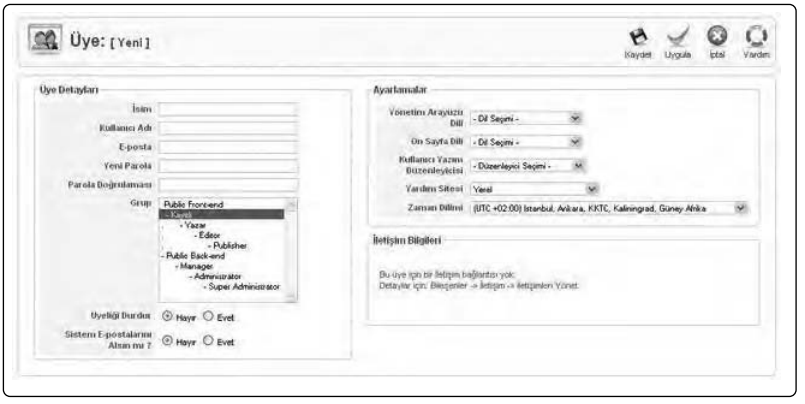
Üye Yönetimi ve Üyelik Ayarlamaları

Y**önetim paneli>Site>Üye Yöneticisi** yoluyla üye yönetim sayfasına ulaşıyoruz.



Resimde de gördüğünüz gibi bu sayfada sağ üstteki simgeleri kullanarak yeni üye ekleyebilir, mevcut bir üyenin ayarlarına yeniden düzenleyebilir, istersek üyeyi silebiliriz.

Bu sayfada **Yeni** tuşuna basarak yeni üye ekleyeceğimiz sayfaya geçelim.



Bu sayfa şu an yeni üye eklemekte olduğumuzdan boş. Gerekli bilgileri girip kaydettiğimizde tahmin edeceğiniz gibi bu üyeye özel yönetim sayfası olacaktır. Üyelerin yönetimini, her bir üyeye özel, adeta bir kartoteks gibi düşünebileceğiniz bu sayfadan yapacağız.

Üye Detayları

İsim: Üyenin ismi.

Kullanıcı Adı: Üyenin kullanıcı adı.

E-posta: Üyenin e-posta adresi.

Yeni Parola: Üyenin şifresi.

Parola Doğrulaması: Şifrenin doğruluğunu teyit için tekrar girileceği alan.

Grup: Üyenin yer alacağı grup. Bir başka ifadeyle üyenin erişim seviyesi.

Site arayüzü: Bu yetki seviyesindeki kullanıcılar, tüm işlemlerini site arayüzünden yaparlar. **Yönetim** (*Admin*) paneline giremezler.

Kayıtlı (Registered): Sitenize üye olan kullanıcılar.

Yazar (Author): İçerik ekleyebilme yetkisi olan üye grubu.

Düzenleyici (Editor): İçerik ekleyebilmekle beraber aynı zamanda da yazarların ekledikleri içerikleri düzenleyebilme yetkisine de sahiptirler.

Yayıncı (Publisher): İçeriklerin yayını ve düzenlenmesi ile ilgili tüm yetkilere sahiptir.

Site Yönetici paneli: Bu üye grubu sitenin Admin paneline girme yetkisi olan gruptur.

Manager: İçerik sorumlusu olan yönetici grubudur.

Administrator: Bazı yetkileri (Süper Administrator hesaplarını yönetme yetkileri yoktur. Site temasını ve site dilini değiştiremez. Üyelere site adına toplu e-posta atamaz.) kısıtlanmış, panel işlevlerini yönetmeye yetkili yönetici grubudur.

Super Administrator: Sitede tam yetki sahibi olan ve tüm fonksiyonlara kısıtlamasız erişebilen en üst düzey yönetici grubudur.

Üyeliği Durdur: Bu bölümden üyenin üyeliğini askıya alabilirsiniz. Bu takdirde üye, bu bilgilerle siteye giriş yapamayacaktır. Daha sonra dilerseniz üyeliği tekrar aktif edebilirsiniz.

Sistem E-postalarını Alsın mı?: Üyenin sistem tarafından gönderilecek e-postaları alıp almayacağını buradan belirleyebilirsiniz.

AYARLAMALAR

Yönetim Arayüzü Dili: Üyemizin erişim seviyesi yönetim paneline girme izni olan bir erişim seviyesi ise buradan yönetim paneli dilini ayarlayabilirsiniz. Müdahale etmediğiniz takdirde varsayılan admin panel dili geçerli olacaktır.

Ön Sayfa Dili: Üyenin sitenin ön yüzünde kullanacağı dili ayarlayabilirsiniz. Müdahale etmediğiniz takdirde varsayılan site dili geçerli olacaktır.

Kullanıcı Yazım Düzenleyicisi: Üyenin sistemde yüklü yazı editörlerinden hangisini kullanacağını belirleyebilirsiniz. Müdahale etmediğiniz takdirde varsayılan site editörü geçerli olacaktır.

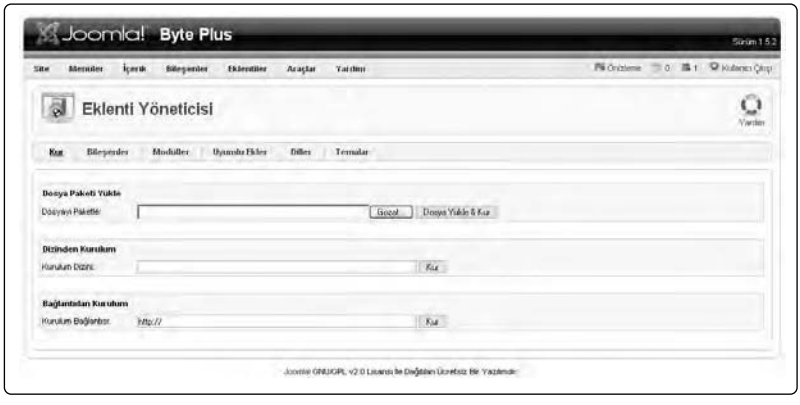
Yardım Sitesi: Sistemle ilgili yardım bilgilerine ulaşma kanalı.

Zaman Dilimi: Üyenin bulunduğu bölgeye göre zaman dilimi ataması yapabilirsiniz.

Tüm işlemleri bitirip **Kaydet** tuşu ile üyemizin kaydını tamamlayalım. Artık nur topu gibi bir üyemiz oldu.

Eklenti ve Tema Yöneticisi

Kur/Kaldır İşlemleri



Joomla 1.5 sürümünde **Joomla 1.0**'dan farklı olarak tüm eklentiler için ayrı ayrı yükleme alanları yoktur. Hepsi, yukarıdaki resimde gördüğünüz sayfadan yüklenir. Sistem yüklenen sıkıştırılmış dosyanın içeriğinin tipini (bileşen, modül, plug-in ...vd) paketin içindeki xml dosyası vasıtasıyla ayırt eder ve ilgili yere yükler. Söz konusu eklenti yükleme işini üç farklı şekilde yapmamız mümkündür.

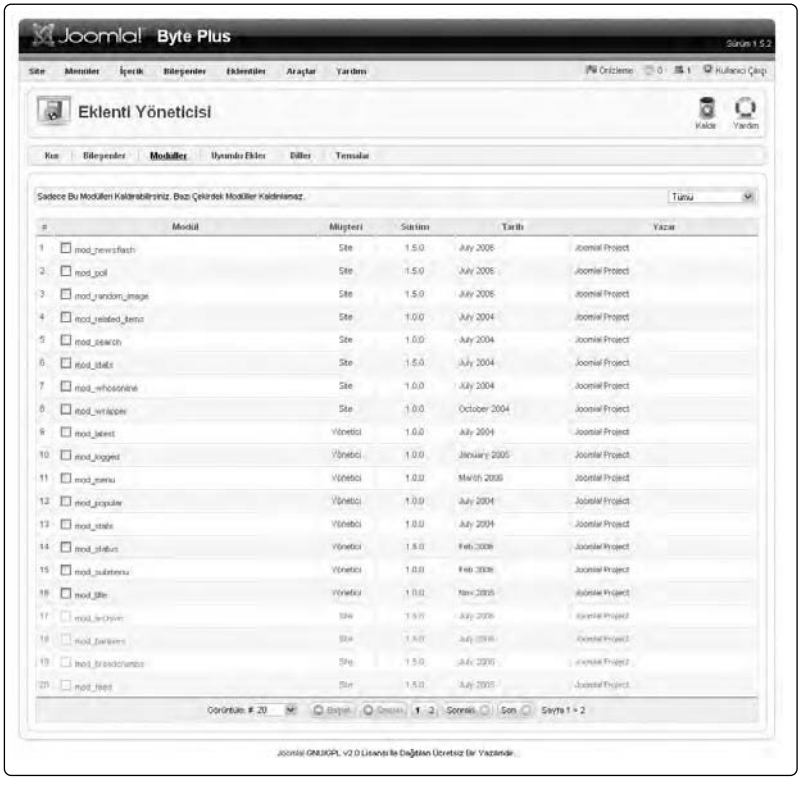
Paket Yükle: Bu yöntem en yaygın kullandığımız yükleme şeklidir. Bilgisayarımızda sıkıştırılmış (zip'li) halde bulunan eklenti paketini **Gözat** tuşu ile sisteme gösterir, ardından da **Dosya Yükle & Kur** tuşu ile yükleme işlemini gerçekleştiririz.

Dizinden Kur: Yüklemek istediğimiz eklentiyi sunucumuza yükleyip, konumunu "Kurulum Dizini" kutucuğuna girerek eklenti yüklemek, kullanabileceğimiz ikinci seçenektir.

URL Adresten Kur: Üçüncü bir alternatif olarak, herhangi bir uzak sunucuda zipli halde bulunan eklentinin konumunu, "Paket URL Adresi" alanına girerek (örn: *http://www.siteismi.com/com_deneme.zip*) kurulumu gerçekleştirebiliriz.

İşin **Kur** kısmını bitirdikten sonra şimdi de **Kaldır** kısmına bakalım. Yukarıdaki resimde de gördüğünüz gibi **Eklenti Yöneticisi** sayfasında Kur sekmesinin haricinde, **Bileşenler Modüller**, **Uygulama Ekleri**, **Diller** ve **Temalar** sekmeleri bulunmaktadır. Her bir sekmede o eklenti türüne ait sistemde yüklü bulunan eklentiler listelenmektedir.

Örnek olarak **Bileşenler** sekmesini tıklıyoruz ve aşağıdaki resimde gördüğünüz sistemde yüklü olan bileşenlerin listesini gösteren sayfa ile karşılaşıyoruz.



Resme dikkatle bakınca, bazı bileşenlerin listede soluk renkte yer aldıklarını ve kaldırmak için seçilemediklerini görüyoruz. Bunun anlamı bu bileşenlerin sistemin sağlıklı çalışabilmesi için sistemde mutlaka bulunması gereken unsurlar olduklarıdır. Bu sebeple de silinmelerine izin verilmez. Diğer bileşenlerden ise, gerekli gördüğümüz her hangi bir bileşeni silmek için, başında bulunan radyo düğmesini seçili hale getirip, sayfanın sağ üst kısmındaki **Kaldır** simgesine tıklamamız yeterli olacaktır.

Diğer eklenti türlerinde de işlemler aynı olduğundan hepsini tek tek örneklendirmeden bu kadar açıklama ile yetiniyoruz.

TEMA YÖNETİMİ

Joomla'nın görünen yüzü demek olan temalar; esnek yapılarıyla size, sitenizin görünümünü istediğiniz gibi oluşturabilme imkanı verir. Ayrıca bir çok Joomla tema sitesinde yüzlerce Joomla teması sunulduğundan, istediğiniz siteyi oluşturabilmek için alternatif sıkıntısı çekmezsiniz. Hatta tema seçeneği bu kadar çok olunca, seçim yapmakta bir hayli zorlanabilirsiniz.

Biz, örnek olarak en güzel ücretsiz **Joomla 1.5** temalarından biri olan **ja_purity** temasını seçtik. İlk olarak, **Eklentiler>Kur-Kaldır** yoluyla **Eklenti Yöneticisi** sayfasına gidiyoruz ve **Dosya Paketi Yükle** bölümünden **Göza Dosya&Yükle Kur** tuşu ile bilgisayarımızdaki sıkıştırılmış tema dosyasını seçiyoruz. Daha sonra tuşu ile temamızı yüklüyoruz

Daha sonra, **Eklentiler>Tema Yöneticisi** yolu ile sitemizde yüklü olan temaların listesinin bulunduğu **Tema Yöneticisi** sayfasına gidiyoruz. Yüklemiş olduğumuz **ja_purity** temasının solundaki radyo düğmesini tıklayıp sağ üstteki **Varsayılan** tuşunu tıklayarak **ja_purity** temasını sitemizin geçerli teması yapıyoruz.

Bu sayfadaki ikinci tuş olan **Düzenle** tuşu ise adından da anlaşılacağı gibi, temamızla ilgili düzenlemeleri yapacağımız bölümü ifade etmektedir. Temamızı seçip, **Düzenle** tuşuna tıklayarak, aşağıdaki resimde gördüğünüz tema düzenleme sayfasına geçelim.



Düzenleme sayfasının sağ tarafındaki **Özellikler** bölümünden temanızla ilgili birçok düzenlemeyi kolaylıkla yapabilirsiniz.

Ayrıca yine bu sayfadaki **HTML Düzenle** ve **CSS Düzenle** tuşları ile gerekli olduğunda temanızın **index.php** dosyasında ve **CSS** dosyasında kod düzenlemeleri yapabilirsiniz.

Modüllerin, temamızın hangi bölgelerinde gösterileceğini belirleyebilmek için, Joomla temalarının kod yapılarında **Modül Pozisyonu** olarak tabir ettiğimiz konumlar tanımlanmaktadır. Dolayısıyla sitemizde kullanacağımız modüllerin sitemizin hangi bölgelerinde gösterileceğini belirleyebilmek için kullandığımız temanın modül pozisyonlarının adlarını ve temadaki konumlarını bilmemiz gerekir. Tema düzenleme sayfasındaki bir başka tuş olan **Önizleme** tuşunun görevi de bu olmaktadır.

Joomla 1.5.x Temalarındaki Ana Dosyalar

Tema kurulum ve ayarları ile ilgili genel bilgilerden sonra, şimdi de bir Joomla temasını oluşturan temel dosyaların neler olduğuna bakalım.

templateDetails.xml

Kısaca, temamızı yüklememizi sağlayan dosya olarak tarif edebiliriz. XML formatındaki bu dosya, doğru bir şekilde yapılandırılmadığı takdirde tema kurulumunu sağlıklı bir şekilde yapmak mümkün olmayacaktır.

index.php

Bu dosya için, temanın Joomla sistemiyle bağlantısını sağlayan ana dosyadır diyebiliriz. İçeriği ise **PHP** ve **(X)HTML** konutlarından oluşur.

template_thumbnaiil.png

Temanın ön izleme resmidir. Genellikle `140x90 pixel` ölçüleri kullanılır. Bu resim, Joomla yönetim panelinde sisteminize yüklü temalar arasında seçim yaparken ve isteğe bağlı olarak sitenizin ön yüzünde tema seçme modülünü yayınlayıp, site ziyaretçilerinize tema seçme hakkı tanıdığınızda görüntülenecek olan resimdir.

css/template.css

Temanın **stil** (*style*) tanımlamalarının bulunduğu CSS komutlarını içeren dosyadır.

index.php

index.php dosyası için, temanın sunum kurallarını belirleyen ana dosyasıdır, diyebiliriz. **PHP** ve **(X)HTML** betiklerinin kombinasyonundan oluşur. Dosyamızın içerdiği temel (her temada olması gereken) satırlar tek tek bakacak olursak:

```
<?php//nodirectaccessdefined ('_JEXEC')ordie ('Restrictedaccess');?>
```

Dosyamıza dışarıdan, doğrudan erişimi engellemek içindir. Güvenlik açısından mutlaka dosyamızın başında olması gerekir.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
```

```
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns=http://www.w3.org/1999/xhtml xml:lang="<?phpec-
ho$this->language;?>" lang="
<?phpecho$this->language;?>" >
```

Yukarıdaki satırlar temamızın XHTML ve XML bilgilerini içermektedir.

```
<head>
<jdoc:include type="head" />
```

buradan itibaren temanın haeder bölümünde yer alan kodlar başlıyor.

```
<title>Welcome to the Frontpage</title>
```

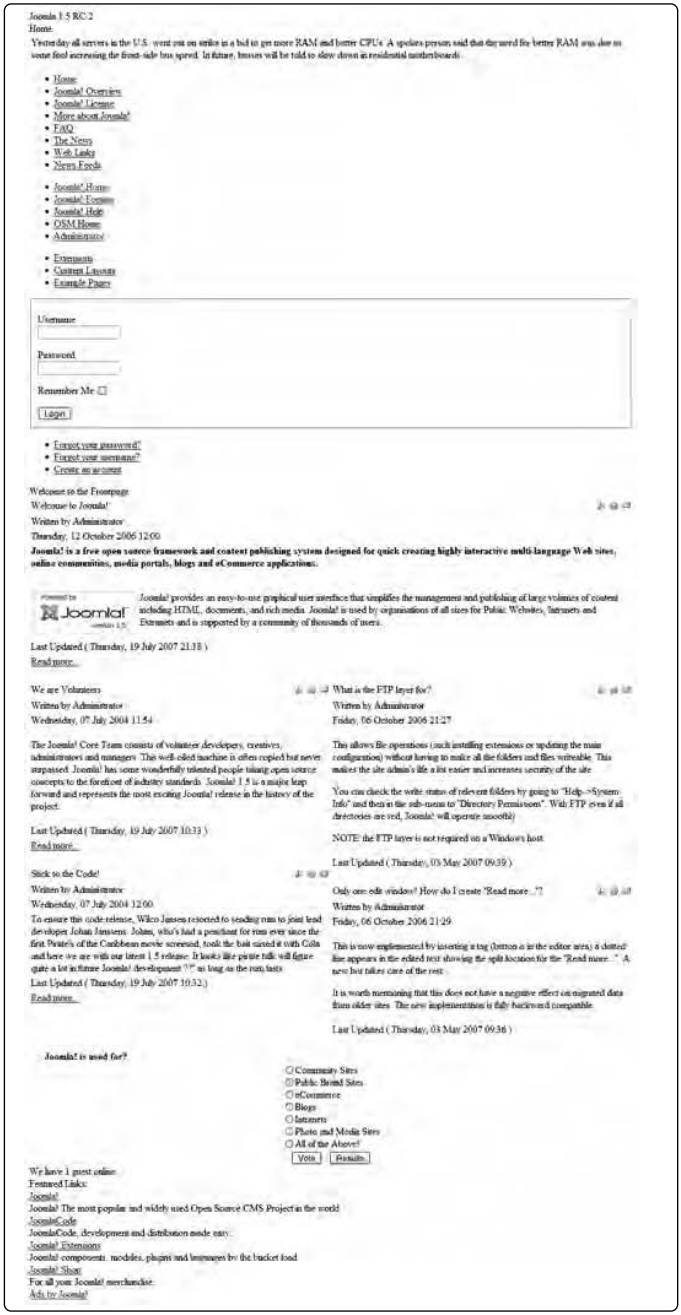
başlık bilgileri...

```
<meta name="description" content="Joomla! - the dynamic portal engi-
ne and content management system" />
<meta name="generator" content="Joomla! 1.5 - Open Source Content
Management" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="robots" content="index, follow" />
<meta name="keywords" content="joomla, Joomla" />
```

Yukarıdaki kısım ise; temamızın meta bilgilerinin bulunduğu bölümdür.

```
<link rel="stylesheet" href="<?phpecho$this->baseurl?>/temp-
lates/system/css/system.css" type="text/css" />
<link rel="stylesheet" href="<?phpecho$this->baseurl?>/temp-
lates/system/css/general.css" type="text/css" />
<link rel="stylesheet" href="<?phpecho$this->baseurl?>/temp-
lates/temanızın_adı/css/template.css" type="text/css" />
```

Temada kullanılan CSS dosyaları... Yeri gelmişken index.php dosyamız bittiğinde (CSS dosyasını hazırlamadan evvel) temamız, aşağıdaki resimde de göreceğiniz gibi tasarım unsurlarından yoksun tabiri caizse çıplak bir tema olacak.


Joomla 1.5 RC 2
Home
Yesterday all servers in the U.S. went on strike in a bid to get more RAM and better CPUs. A spokes person said that the need for better RAM was due to some fool increasing the front-side bus speed. In future, busses will be told to slow down in residential motherboards.
Home
Joomla! Overview
Joomla! License
More about Joomla!
FAQ
The News
Web Links
News Feeds
Joomla! Home
Joomla! Forums
Joomla! Help
OSM Home
Administrator
Extensions
Content Layouts
Example Pages
Username
Password
Remember Me
Login
Forgot your password?
Forgot your username?
Create an account
Welcome to the Frontpage
Welcome to Joomla!
Written by Administrator
Thursday, 12 October 2006 12:00
Joomla! is a free open source framework and content publishing system designed for quickly creating highly interactive multi-language Web sites, online communities, media portals, blogs and eCommerce applications.
Joomla! provides an easy-to-use graphical user interface that simplifies the management and publishing of large volumes of content including HTML, documents, and rich media. Joomla! is used by organizations of all sizes for Public Websites, Intranets and Extranets and is supported by a community of thousands of users.
Last Updated (Thursday, 19 July 2007 21:18)
Read more...
We are Volunteers
Written by Administrator
Wednesday, 07 July 2004 11:54
The Joomla! Core Team consists of volunteer developers, creatives, administrators and managers. This well-oiled machine is often copied but never surpassed. Joomla! has some wonderfully talented people taking open source concepts to the forefront of industry standards. Joomla! 1.5 is a major leap forward and represents the most exciting Joomla! release in the history of the project.
Last Updated (Thursday, 19 July 2007 10:33)
Read more...
What is the FTP layer for?
Written by Administrator
Friday, 06 October 2006 21:27
This allows file operations (such installing extensions or updating the main configuration) without having to make all the folders and files writeable. This makes the site admin's life a lot easier and increases security of the site.
You can check the write status of relevent folders by going to "Help->System Info" and then in the sub-menu to "Directory Permissions". With FTP even if all directories are red, Joomla! will operate smoothly.
NOTE: the FTP layer is not required on a Windows host.
Last Updated (Thursday, 03 May 2007 09:39)
Stick to the Code!
Written by Administrator
Wednesday, 07 July 2004 12:00
To ensure the code release, Wilco Jansen resorted to sending rum to joint lead developer Johan Janssens. Johan, who's had a penchant for rum ever since the first Pirate's of the Caribbean movie screened, took the bait mixed it with Cola and here we are with our latest 1.5 release. It looks like pirate talk will figure quite a lot in future Joomla! development 'YP' as long as the rum lasts.
Last Updated (Thursday, 19 July 2007 10:32)
Read more...
Only one edit window? How do I create "Read more..."?
Written by Administrator
Friday, 06 October 2006 21:29
This is now implemented by inserting a tag (button is in the editor area) a dotted line appears in the edited text showing the split location for the "Read more..." A new bot takes care of the rest.
It is worth mentioning that this does not have a negative effect on migrated data from older sites. The new implementation is fully backward compatible.
Last Updated (Thursday, 03 May 2007 09:36)
Joomla! is used for?
Community Sites
Public Brand Sites
eCommerce
Blogs
Intranets
Photo and Media Sites
All of the Above!
Vote
Results
We have 1 guest online
Featured Links:
Joomla!
Joomla! The most popular and widely used Open Source CMS Project in the world.
JoomlaCode
JoomlaCode, development and distribution made easy.
Joomla! Extensions
Joomla! components, modules, plugins and languages by the bucket load.
Joomla! Shop
For all your Joomla! merchandise.
Ads by Joomla!

Bunu özellikle gösterdim ki, CSS dosyasının (dolayısı ile CSS 'in) tasarım gücünü bir kez daha görmüş olalım.

Kaldığımız yerden devam edelim.

```
<!—[if lte IE 6]>
<link href="/templates/<?phpecho$this->template?>/css/ie-
only.css" rel="stylesheet" type="text/css" />
<![endif]_
```

Browser farklılıkları için CSS kullanımı (IE 6) Tabi, bir tane ieonly.css dosyası oluşturmanız lazım.

```
<body>
<?phpecho$mainframe->getCfg('sitename');?><br />
<jdoc:include type="module" name="breadcrumbs" />
<jdoc:include type="modules" name="top" />
<jdoc:include type="modules" name="left" />
<jdoc:include type="component" />
<jdoc:include type="modules" name="right" />
</body>
```

Temanın gövde kısmı. **Modül** pozisyonları ve **mainbody** içerik alanı tanımlamaları...

```
<jdoc:include type="component" />
```

Yukarıdaki kod satırı, sitemizin mainbody alanını, yani içeriklerimizin görüneceği alanı ifade eder.

```
<jdoc:include type="modules" name="top" />
<jdoc:include type="modules" name="left" />
<jdoc:include type="modules" name="right" />
```

Bunlar da temamızda oluşturduğumuz modül pozisyonları. Tabiki bunlar, top, left, right ile sınırlı değil. Gerektiği kadar modül pozisyonu ekleyebilirsiniz.

Bunu formule edecek olursak:

```
<jdoc:include type="modules" name="modülünadınıburaya"
style="istersek modüle has stil tanımlayabiliriz. Bu kısım
opsiyoneldir." />
```

şeklinde formüle edebiliriz.

Kullanılabilecek stiller: table, horz, xhtml, rounded, none .

table: Sütun olarak görüntülenme.

horz: Yatay olarak görüntülenme.

xhtml: div ile **css** tanımlaması kullanılarak görüntülenme

rounded: Yuvarlatılmış, oval ...vb uygun formatta görüntülenme.

none: Stilsiz ve başlıksız çıplak görüntülenme

Tabi bu temayı giydirebilmek, bir başka ifade ile CSS dosyamızda oluşturacağımız stilleri kullanabilmek için ve dahi temayı bir şekle şemale sokmak için; ana içerik alanını ve modülleri div tag'ları içine alıp, birer id veriyoruz.

Örnek

```
<div id="icerik"><!— İçerik alanı başı —>
  <jdoc:include type="component" />
</div><!— @İçerik alanı sonu —>
```

Bu noktada üstünde durmak istediğimiz bir önemli husus da; `mosCountMoules` fonksiyonunun kullanımı olacak... Genel kullanım şekli aşağıdaki gibidir. Burada, temamıza "user4" modül pozisyonu eklemek istediğimizi varsayıyoruz.

```
<?phpif($this->countModules('user4')):?>
   <div id="deneme">
            <jdoc:include type="modules" name="user4" />
   </div>
<?phpendif;?>
```

Bu kod, **user4** pozisyonuna modül atamadığınız durumlarda, bu modül pozisyonunun temada görünmemesi sağlar.

Resmi sitesi: http://www.joomla.org/

Türkçe Destek:

http://forum.joomla.org/viewforum.php?f=51&sid=14b20a3fce34f86d-1a1507036479885c

SON SÖZ

Bilgilerin ve düşüncelerin yazıya, yazıların okuyucuyla yazar arasında köprü kuran bir kitaba dönüşmesi gerçekten heyecan verici bir duygu.

Uzun ve yorucu bir çalışmanın ardından sizinle buluşmak gerçekten bana tüm yorgunluğumu unutturdu.

Kitapta yer alan konularla ilgili soru sormak istediğinizde, kitabın sitesi olan **www.webtasarimrehberi.net** adresinden ya da **www.kodlab.com** adresindeki **Yazara Sor** bölümünden bana ulaşarak sorularınızı yöneltebilirsiniz.

Ayrıca kitapla ilgili yapıcı eleştirilerinizi de mutlaka **www.webtasarimrehberi.net** adresinde bekliyor olacağım. Yapacağınız eleştiriler kitabın sonraki baskılarında mutlaka değerlendirilecektir.

Bu kitapla başlayan dostluğumuzun daha nice güzel paylaşımlarla devam etmesi temennisiyle...

MUSTAFA AYDEMİR