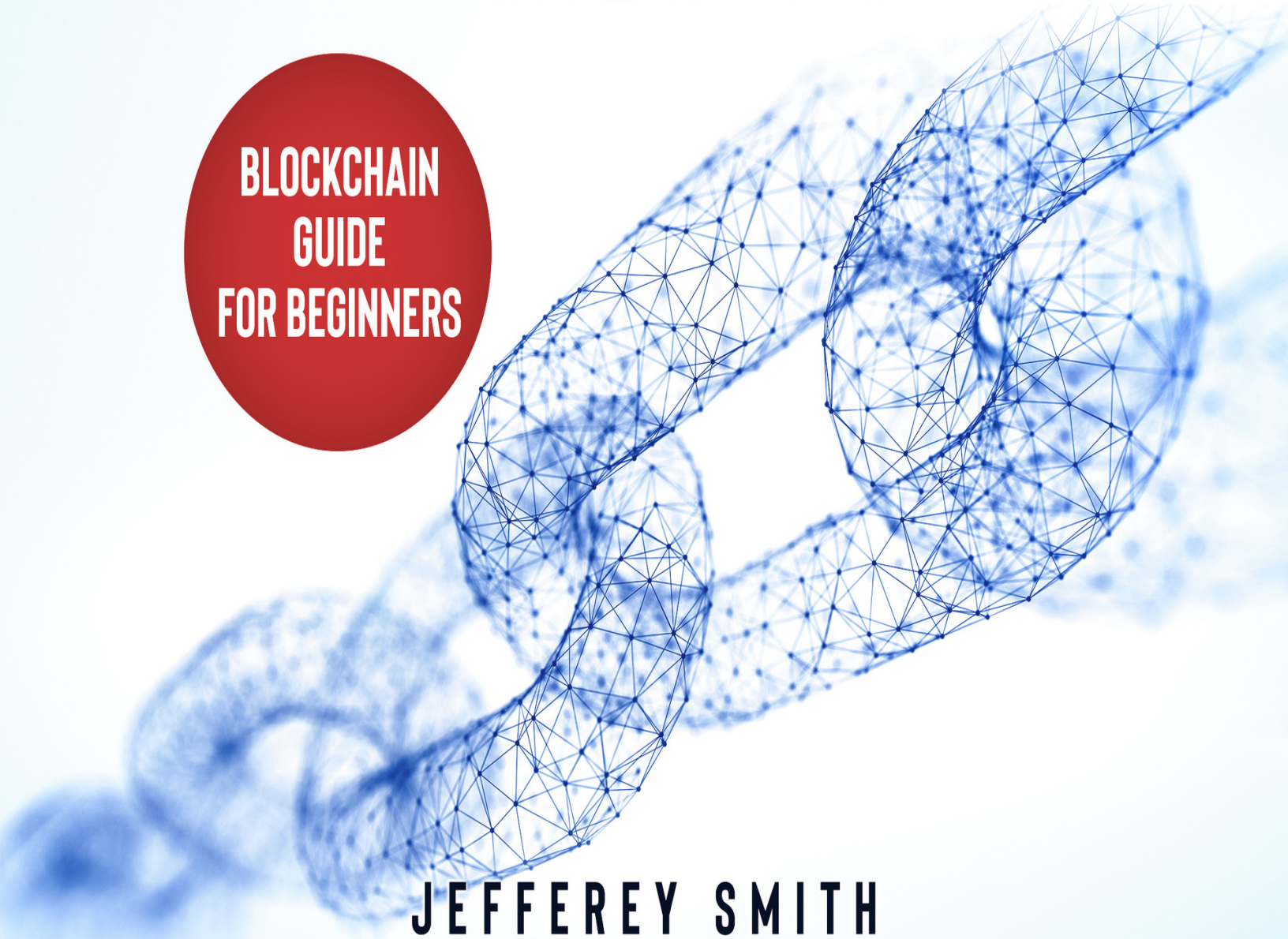
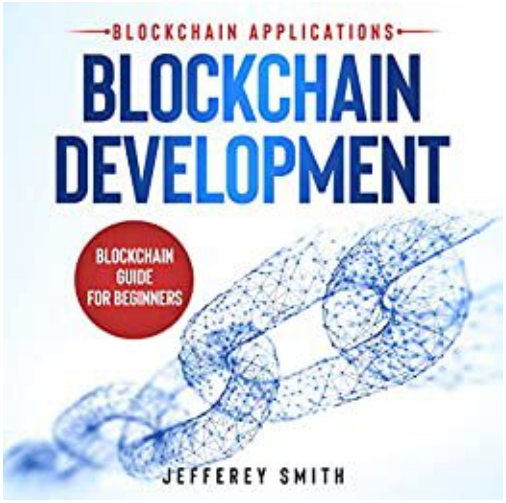
BLOCKCHAIN APPLICATIONS

BLOCKCHAIN DEVELOPMENT

BLOCKCHAIN GUIDE FOR BEGINNERS



JEFFEREY SMITH



Blockchain Development– Blockchain Applications

Blockchain Guide for Beginners

Jefferey Smith

© Copyright 2022–All rights reserved.

The content contained within this book may not be reproduced, duplicated or transmitted without direct written permission from the author or the publisher.

Under no circumstances will any blame or legal responsibility be held against the publisher, or author, for any damages, reparation, or monetary loss due to the information contained within this book, either directly or indirectly.

Legal Notice:

This book is copyright protected. It is only for personal use. You cannot amend, distribute, sell, use, quote or paraphrase any part, or the content within this book, without the consent of the author or publisher.

Disclaimer Notice:

Please note the information contained within this document is for educational and entertainment purposes only. All effort has been executed to present accurate, up to date, reliable, complete information. No warranties of any kind are declared or implied. Readers acknowledge that the author is not engaged in the rendering of legal, financial, medical or professional advice. The content within this book has been derived from various sources. Please consult a licensed professional before attempting any techniques outlined in this book.

By reading this document, the reader agrees that under no circumstances is the author responsible for any losses, direct or indirect, that are incurred as a result of the use of the information contained within this document, including, but not limited to, errors, omissions, or inaccuracies.

Table of Contents

Introduction
- Blockchain Development
  - Blockchain Solution Architect
  - Blockchain Quality Engineer
  - Blockchain Project Manager
- Blockchain Applications

Chapter 1: Blockchain and Cryptocurrency
- Blockchain at Its Inception
  - Blockchain Today
- How Does a Blockchain Work?
- Different Types of Blockchains
  - Private Blockchains
  - Public Blockchains
  - Consortium Blockchains
- What Is Cryptocurrency?
  - Cryptocurrency Key Features
  - Characteristics of a Good Cryptocurrency

Chapter 2: Blockchain Programming
- Blockchain Programming Fundamentals
- Best Programming Languages for Blockchain
  - C++
  - Python
  - JavaScript
  - Solidity
  - GOLang
- How to Become a Blockchain Programmer
- Types of Blockchain Developers
  - Core Blockchain Developer

Blockchain Software Developer
Chapter 3: Blockchain Finance
The Benefits of Blockchain in Finance
The Possible Drawbacks of Blockchain in Finance
How the Digitization of Financial Instruments Could Impact Finance
Payments
Asset Management
How Does Blockchain Impact Capital Markets
Chapter 4: Bitcoin and Blockchain
What Is Bitcoin?
What Is Bitcoin Mining?
Is It Safe to Buy?
The Ins and Outs of the Bitcoin Blockchain
Chapter 5: Blockchain Security
Blockchain Security Basics
Permissioned Blockchains
Permissionless Blockchains
The Role of Miners in Blockchain
How Blockchain Security Averts Double Spending
Blockchain Security Risks
Blockchain Security in Practice
Conclusion
A Step-by-Step Guide for Beginners
References

Introduction

When human beings lead a more nomadic life, there was no concept of ownership. There was no need to own anything because everything they needed to survive they either gathered or produced for themselves. And because of the nomadic nature of their lifestyle there was no need to own land or livestock. They moved around and lived off the land.

As time went on, they started to settle and grow their own crops. Once they started doing this there became a need to own the land that their crops grew on. There also was a need to own the apparatus they used to tend to these crops. Thus the concept of ownership was born.

As they continued to farm and produce fruits and vegetables, they would end up with more produce than they actually needed. They began to barter with each other. If one had more tomatoes than they needed and wanted spinach, they could exchange some of their tomatoes for some of someone else's spinach.

Eventually, they felt the need to record these transactions so they started writing to keep track of ownership. We can look at these writings as the first *ledgers*.

While they were moving in the right direction, they had a problem with trust. There was no way for them to verify the information contained in the ledgers because not everybody had access to them.

When banks eventually came around, people also had to trust them to keep their money safe, and that didn't always end well. There have been seven financial crises from the Great Depression to the Great Recession.

Some people started to realize that we needed a way forward that didn't see us relying heavily on other people. They saw that we needed a way to transact without fear or the constant threat of human error. Thus, Blockchain was born.

The overarching theme with Blockchain and Cryptocurrency is that you don't have to put your trust in an individual or a central authority to keep your assets safe. You don't even have to worry about breaches of contract

or the integrity of any other party. Blockchain has the potential to ensure that most, if not all industries run more efficiently.

Blockchain Development

Blockchain Development is a novel career path. Because of this there are many opportunities you can take advantage of. A blockchain developer builds Decentralized Applications or DApps for short. DApps are programs that operate on a peer-to-peer network of computers as opposed to a single computer.

On a daily basis, a developer uses different programming languages to execute the code that designs user-interfaces, features and structures for blockchain applications. They are also responsible for putting security measures in place to avoid any attempts at hacking.

There are many self taught programmers and developers operating in the tech world today. It is possible to become a blockchain developer without a degree, but you would have to work extremely hard and possess the right skill set. If you plan on going about becoming a developer the traditional way, you will need:

- At the very least, a Bachelor’s degree in computer science or IT.
- Analytical expertise. A key part of development is working out what the customer needs and finding ways to achieve the best results with whatever infrastructure they have.
- Coding skills. A thorough knowledge of various coding languages would be advantageous because it takes more than one to build a blockchain network.
- Foundational knowledge of blockchain protocols. A protocol is an array of practices for sharing data from device to device. The only way devices can share information is if there is an aforementioned agreement stating how to structure it and how it will be sent and received. If there was no protocol, computer A could be sending data in 32-bit increments while computer B is expecting to receive data in 64-bit increments. The most prominent set of protocols is OSI or

Open Systems Interconnection. It is a set of instructions on how all network interactions should be carried out between computers. Internet protocols you may recognize are: IP, HTTPS, and DNS.

Not every blockchain developer is going to have the same job specs because there are different positions one can hold, namely:

Blockchain Solution Architect
Solution Architects are responsible for everything technical during the sales process. They work closely with the sales head in devising solutions. They then lead a team of developers and eventually provide the solution to the customers.

Blockchain Quality Engineer
Quality Engineers supervise the development of blockchain applications and ensure that they are up to standard. Quality engineers are required to keep abreast of the newest blockchain features and developments.

Blockchain Project Manager
Project managers are more like overseers. They monitor and evaluate resources, supervise security protocols and transaction administration systems, compose distinct features and manage any operational issues on a blockchain network.

Blockchain Applications

When people think of the Blockchain, they almost immediately think of cryptocurrency, but it has other capabilities that could potentially involve *Fiat money*. Fiat money is tangible money. Fiat is Latin for ‘let it be done’. Money only has value because the government says so but it isn’t backed by anything.

Applications include:

- Money transfers

By cutting out the middleman, transferring money from person to person works out cheaper and is much quicker. It’s convenient when transferring money across countries. Somebody in the United States can send money to the Dominican Republic in minutes.

- Financial exchanges

The bulk of cryptocurrency exchanges on the blockchain used to happen from person to person. Now, different organizations are starting to offer decentralized cryptocurrency exchanges. These organizations have more control and can feel safe knowing that their transactions are secure.

- Lending

Smart contracts can be used by lenders to administer guaranteed loans. Processing these loans is faster and less expensive because Smart contracts have features that automatically set off service payments and release of funds. This also means that they can offer better interest rates.

- Insurance

Insurance has always been murky. Customers have never really been sure of what the fine print says and have found themselves in precarious positions because of this. Smart contracts allow for transparency between insurance providers and customers. No customer would be able to claim twice for the same incident and they could actually receive payouts faster.

- Real estate

Real estate involves a considerable amount of paperwork. Personal information needs verification and transfer documents need to be checked thoroughly before a sale goes through. Blockchain technology can minimize paperwork while still offering security and ensuring quicker transactions.

- Storing personal information

It is possible for any entity storing personal information online to experience a breach in security. Storing this information on a public ledger offers more security because it’s harder to penetrate the blockchain system.

- Storing data

The issuing of state benefits would substantially improve if identification information was stored on a blockchain.

- Voting

If identity information was stored on a blockchain, people would be able to cast their votes more easily. Nobody would be able to vote twice and votes

wouldn’t be interfered with. It would also encourage more people to vote by being more convenient.

- State benefits

Storing identification information on a blockchain could aid in the dispersion of government benefits like Medicare. Blockchain’s infrastructure would reduce fraud and operation costs. People would also receive their benefits a lot faster.

- Storing and sharing medical information

Storing medical records on a blockchain could assist in patients receiving the best medical care. Doctors would receive the latest and most accurate information on their patients thus avoiding misdiagnosis and wrongful treatment.

- Royalties

If there were up-to-date records on all film and music files spread over the internet, there would be a lot less piracy and artists’ royalties would be paid in full and on time.

- NFTs

Non-fungible tokens are a way for people to own the rights to anything that can be stored as data online, but it’s most prevalent in digital art. Putting NFTs on a blockchain can make certain that only a singular piece of digital art exists. It makes investing in art easier for people because there’s no need to worry about storage and preservation.

- Supply chain and logistics tracking

Keeping track of articles as they traverse a supply chain network can help supply chain and logistics departments work better together. If all the data is stored on a blockchain network then it can’t be modified and it can be accessed easily by the relevant parties.

- Securing *Internet of Things* networks

The *Internet of Things* or IoT, is a system of interlinked computers that have UIDs or Unique Identifiers that allow them to transmit data over a network. It works without human interaction. It’s convenient, but susceptible to digital crime. It makes use of a centralized server, which makes it easier for

hackers to access. The blockchain interface can enhance security by keeping login credentials on a decentralized network.

- Gambling

The blockchain network could potentially be very beneficial, especially for gamblers. If a casino were to conduct its business on a blockchain there would be a lot more transparency. All dealings would be recorded on the blockchain which means that patrons would always be sure that their games were fair and that they would be paid out what they are owed. Use of a blockchain would also mean that personal information wouldn’t need to be shared and players could gamble anonymously.

Chapter 1: Blockchain and Cryptocurrency

Blockchain at Its Inception

The concept behind ‘Blockchain’ technology was first outlined in 1991 by research scientists Stuart Haber and W. Scott Stornetta. They proposed a computationally practical method that added time stamps to digital documents, so that they could not be backdated or altered. The system was to use an encrypted chain of blocks to store the time-stamped documents. In 1992, Merkle trees were included in the design. A Merkle tree is a structure used by computer applications to organize data. This made it possible for multiple documents to be collected and stored in one block. After its development, the technology was not used and the patent ended up lapsing in 2004.

In that same year, a computer scientist named Harold Finney launched a system called RPoW or Reusable Proof of Work. The system worked by collecting non-exchangeable *Hashcash* based Proof of work tokens and then generating an RSA. It is a signed token that could be passed on from person to person. Hashcash is a proof-of-work model and RSA is a technique used to encrypt a public-key used for secure data transmission especially over the internet. When you are spending tangible money, it’s hard to spend the same coin or note twice, because the transaction is happening in the presence of one or more people. Double spending was a term coined after the rise of bitcoin, it describes the possibility of spending the same unit of cryptocurrency more than once. RPoW solved this problem by keeping records of the ownership of tokens and registering them on a server that allowed users to validate their authenticity immediately.

In 2008, a Whitepaper was posted to a cryptography mailing list by someone, or a group of people, using the pseudonym *Satoshi Nakamoto*. The paper presented a decentralized, peer-to-peer, electronic cash system based on the Hashcash POW algorithm, called Bitcoin. (Binance Academy, 2018). It proposed the use of a decentralized peer-to-peer protocol that could trail and validate any and all transactions.

On the 3rd of January 2009, the first Bitcoin block was mined by Satoshi Nakamoto. It yielded him a reward of 50 Bitcoins. The first ever Bitcoin transaction took place on the 12th of January when Nakamoto sent Harold Finney 10 Bitcoins.

Blockchain Today

In 2013, a programmer named Vitalik Buterin expressed that Bitcoin needed a scripting language for building decentralized applications. He then started developing a new Blockchain-based distributed computing system called Ethereum. A distributed computing system has numerous pieces of software on several computers but run as one system. The computers in the system can either be in close proximity to each other sharing a local network, or they can be dispersed in different areas and connected by a Wide-area network. Ethereum has a scripting feature called *Smart Contracts*. Smart Contracts are scripts that are implemented on the Ethereum Blockchain. They can be used to make transactions if specific conditions are met. They are written in specific coding languages and arranged into *Bytecode*. Bytecode is source code that has been simplified for a software interpreter to understand. The Ethereum Virtual machine then reads and executes the transactions. The EVM is often called a virtual CPU because it is not tangible and is run on hundreds of machines worldwide.

Developers can produce apps that run inside the Ethereum Blockchain. These apps are called DApps (Decentralized Applications). There are hundreds of apps running on the Ethereum blockchain, including various social media platforms. Blockchain technology is becoming more conventional and is even used in applications outside of the cryptocurrency space.

How Does a Blockchain Work?

Blockchain is an apt name for this system. Like the name states, a blockchain is a chain of blocks that contain information. It is a distributed ledger that is open to anyone. A ledger is a bookkeeping system that contains all of a company’s financial data. It records each and every

transaction that happens from day one. It also contains the account information that company's use to prepare their financial statements.

Once data has been recorded inside a block it is virtually impossible to change. Each block has three components; the data, the hash and the previous block's hash. The data depends on what kind of block it is. The hash works the same way a fingerprint does in that it can't be duplicated and you can use it to identify a single block and all it's content.

If a piece of data in the block changes, so does the hash. Hashes are useful because they help in identifying any changes that have happened within a block. The hash of the previous block is what links all the blocks together like a chain. This is what makes a blockchain so secure.

If we had three blocks in one chain then block three would contain block two's hash and block two would contain block one's hash. The first block only has its own hash so it's called the Genesis block.

In the event that the second block is interfered with, its hash will change and that would invalidate all the blocks that come afterward because block three would not contain a valid hash. While a blockchain is inherently secure, computers are developing and increasing their capabilities on a daily basis so it's not impossible for blocks to be interfered with.

To abate any interference, blockchains have something called Proof of Work (PoW). PoW works to slow down the building of new blocks. It's helpful because if you were to meddle with block two, you would have to do the proof of work for every block that comes after it.

In order to manage the chain, blockchain uses a Peer-to-peer network. A peer-to-peer network is where two or more computers are able to share files without having to connect to one server. As soon as a new user joins the network they have full access to the entire blockchain and all its data. The node is used to verify the data. A node can either be a redistribution point, a communication endpoint or a connection point. In a blockchain it is a connection point. A node can register information, process it and then send it to other nodes.

If a new block is added it gets sent to all the computers on the network. The node verifies the block and then adds it to the blockchain. All the nodes in the network create consensus by *agreeing* on which blocks are valid and

which are not. If a block has been interfered with it gets rejected by all the nodes in the network.

Because blocks are constantly changing, Smart Contracts were developed. They are programs that are kept in the blockchain. They automatically exchange coins if certain conditions are met.

Different Types of Blockchains

Private Blockchains

Access to Private blockchains is invitation based. Users can be validated by a rule set by the central administrator of the network. Only some users can have access to the network and of those few, there are limits on the kinds of transactions they can perform. These kinds of blockchains use a Proof of Authority consensus approach (PoA). PoA is part of a consensus algorithm that gives a small group of users on a blockchain the authority to validate transactions on the network. PoA is mostly used within secure business environments to control access and record keeping because the transaction data is not usually visible to them.

Public Blockchains

Public blockchains are all about engagement and transparency. Transaction consensus is not centralized, so anybody can take part in authenticating network transactions and the software code is open source. This means it is available to the public. Public blockchains are decentralized through crypto-economics to promote cooperation throughout a network. There isn’t a sole entity controlling the network and there is no central point of failure. The consensus algorithms used in a public blockchain are PoW and Proof of Stake (PoS). PoS is an algorithm that processes transactions and creates new blocks in a chain. It is used to validate any data entries and keep the database secure.

Consortium Blockchains

No one ever talks about Consortium blockchains because they are not widely used. They can be described as a combination of both public and private blockchains because they have a ‘semi-permissioned’ approach. Consortium blockchains’ participants are usually known and have been

authorized by a central authority to participate in the consensus. This kind of blockchain is semi-decentralized while still offering some level of control. Transaction data is also kept private. Consortium blockchains make use of all three consensus algorithms; PoW, PoS, and PoA.

What Is Cryptocurrency?

Most people know what cryptocurrency is, but not many understand it. Until around ten years ago, all forms of modern currency were controlled and regulated by banks and governments. This means that the regulatory authority was centralized, and that restricted the capabilities of tangible money and credit cards. While modern banking has come a long way in recent years, it has its issues. When two parties are exchanging money, it's possible for there to be a single point of failure. The bank can have a glitch in the system or one party's transfer limit could have been reached.

Cryptocurrency removes all the problems of modern banking because there is no single point of failure. A single point of failure happens when a vital part of a system ceases to function, thus causing the entire system to fail. Crypto is a virtual currency that is used as a medium of exchange. It works in a similar way to any other currency, it just isn't tangible and operates using cryptography. Cryptography is a way to protect information using code, so that only the relevant parties have access to it.

Cryptocurrency Key Features

- Cryptocurrency is decentralized

  Banks and governments control the economic system in the outside world. In the digital realm, there are no third-parties and transactions can be processed and authenticated via an open network that doesn't belong to an entity.

- Cryptocurrency is immutable

  When something is immutable, it cannot be changed.

- Cryptocurrency is anonymous

There is no need for users to identify themselves when trading cryptocurrency because there is no third-party. The network processes, authenticates and documents all transactions on the blockchain.

- Cryptocurrency has a finite supply

  When most cryptocurrency is created, it has a predetermined amount to be mined out of the system. Other digital currencies may not have an actual limit, but they do control the amount of new coins that can be generated on a yearly basis.

- Transactions are irreversible

  Because there is no management by any central authority, there isn't a body or an organization that can intervene when any kind of mistake is made. So if you were making a transaction and you happened to enter the wrong details and it gets sent through, it cannot be reversed and there is no organization that you can reach out to for support.

Characteristics of a Good Cryptocurrency

- Security

  Anybody who has assets they can't actually touch wants to be sure that they are safe at all times. Cryptocurrencies stay in a digital 'wallet'. A wallet is essentially just a long line of letters and numbers. It can also be called a *private key* and it allows its owner to withdraw coins. There is a shorter line of numbers linked to the private key that allows a user to accept deposits. This number is called a public key. When two wallets share a transaction, that data is entered into a block which is then added to a ledger. Once all of this information reaches the ledger it cannot be changed.

- Scalability

  When a cryptocurrency is scalable, it means it can confirm a large number of transactions, usually per second. When trading cryptocurrency, a user wants to know that all transactions are going to happen fairly quickly and without fail. Because of innovations in technology there are apps that offer speedy transactions of normal

money like Paypal. This means that cryptocurrencies need to be processed at a speed of no more than 3 seconds at a time.

- Usability

  Like most things, cryptocurrency needs to be easy to navigate. It can be daunting for a new user or for somebody who is considered a technophobe. A well developed cryptocurrency has a lot of complicated things going on in the background but should always appear to be easy on the user's end. Also, it is always best to use cryptocurrencies that you can transact with via a mobile app and not on a web page.

- Demand

  The more demand there is for any given cryptocurrency, the more valuable it becomes. When there is enough interest in the market and the price of a certain coin rises, it will be beneficial to users.

- Finite supply

  Most, if not all cryptocurrencies have a limited amount of coins in existence. The number of coins available is determined upon inception and was calculated by mathematics. Some cryptocurrencies release all their coins at once and others release them bit by bit in an attempt to keep the price stable. In the long run, a good cryptocurrency is determined by its adherence to good economic practices.

Chapter 2: Blockchain Programming

Since the beginning of time, data capturing and sharing has always had a human element. There has always been a middleman present between you and your information. For example, a retailer is the middleman between you and a wholesaler.

In that instance, the retailer becomes the central authority because your fresh produce is controlled by them. The retailer determines how many you can get and at what price. The middlemen have all the power and you have to just trust that they are putting your needs first and conducting business ethically.

If you have ever watched movies online then you have made use of a BitTorrent file-sharing protocol. It is a peer-to-peer protocol that allows one to access, download and share copies of files that are located on various computers. No single computer controls a single file. In that way, the central authority has been totally done away with.

Blockchain works the same way because it is completely decentralized. It completely cuts out the middleman and you aren’t forced to put all your trust into one entity.

Blockchain Programming Fundamentals

- Consensus

  Because blockchain is decentralized, there are millions of nodes all over the world transmitting information. Not all nodes are the same though, some are connected to fast computers and some may be connected to slow computers. The slower computers will receive the data after the faster ones, but there still needs to be one blockchain that consists of the same information. Consensus protocols assist all the nodes by adjusting the data intermittently to ensure that at the end of the day they all have the same information and the blockchain is accurate.

- Hash functions

  A Hash is a function that takes a lengthy numerical value and compresses it. It is then used to identify the data in a specific block on a chain.

- Distributed ledger technology

  DLT uses cryptography to store information accurately and securely. Once the data has been stored, it cannot be changed and it can only be accessed through the use of a *key* or a cryptographic signature.

- Smart contract development

  Smart Contracts are programs that run on a blockchain. They are triggered by the terms of an agreement between two users. Essentially they are lines of code that state the terms of the agreement.

- Cryptography

  Cryptography is taking plain text and converting it into code or taking code and converting it into plain text. its main function is to keep and share data but only be legible to those it is meant for. It is also used to authenticate users that want access to a specific network.

Best Programming Languages for Blockchain

C++

C is a programming language that uses a procedural programming model and was originally developed to write operating systems. Attempting to develop a large-scale application in C proved difficult for a lot of programmers because there is no concept of classes and objects.

In 1985 Bjarne Stroustrap released C++. C++ is an augmented C language. It is an Object-Oriented programming language, which means that it combines data and functions to make objects. This is helpful when developing apps because the code can easily be reused in other programs without having to be rewritten.

C++ took all the good elements of C, like its productivity, pliability and security and combined them with its object-oriented approach to create the

main language for blockchain programming.

Here are some reasons why it's usually a blockchain programmer's first choice:

- C++ is one of the most fully fledged programming languages out there and it is improved upon regularly. Its functions are optimized because of its analytical methods and debuggers.
- It has the ability to segregate different lines of code and encapsulate them so that they do not collide.
- C++ is capable of polymorphism. This means it can perform multiple tasks separately. This aids in the performance of the digital ledger in its entirety.
- It allows for multi-threading. Threads are a bunch of instructions that are supposed to be carried out at the same time. In blockchain development there will be some tasks that need to run concurrently and C++ allows for this.
- C++ allows for better memory usage, which is essential because there's a lot of information being passed around in the blockchain system.

Python

Python is an easy-to-learn programming language. Its structure is based on a simple premise: Simplicity, readability, and minimalism. (Farr, 2018). It has garnered support from developers old and new because it has a fairly limited syntax which makes it easier to read.

JavaScript

While JavaScript is well-known for adding interactivity to websites, it also possesses capabilities that are beneficial to blockchain. It can be used to construct an uncomplicated but fortified blockchain that can protect against hackers and any other unsavory characters. One thing it doesn't have though, is the SHA256 hash function. SHA stands for Secure Hashing Algorithm. Hashing is the process of receiving an input of any length and producing an output of fixed length. 256 refers to the length of the output. When using the SHA256 function, regardless of the length of the input, the output will always be 256 bits long.

Solidity

The Ethereum blockchain is superior by way of speed and ease of use. This is because it makes use of Solidity. It is an object-oriented and effective programming language that is mainly used to implement Smart contracts. It is also called the *curly bracket language.* Solidity merges JavaScript's code and functions along with C++'s classes. This makes it the most intricate coding language. Despite that, it is still user friendly and there is a lot of information available on how to use it.

GOLang

GoLang ('Go' for short) is a programming language that was developed by Google in 2007. It was only released for use by the public in 2012. On inception, the aim was for it to take easy-to-use syntax from JavaScript and combine that with C++'s tried and tested security features. GoLang is run inside of an operating system which means it is able to handle many parts of a block chain at the same time. Ethereum's Software Development Kit is written in GoLang so there are several blockchain apps that use this language.

How to Become a Blockchain Programmer

With this book, you will gain a theoretical knowledge of blockchain technology, its benefits, drawbacks and its place in the ever-changing world. But beyond that you need technical skills that will allow you to do the practical work.

You need to have an understanding of some major concepts, like:

- Consensus
- Hash functions
- Distributed ledger technology

A data structure is the model you follow to sort and store data so that it's easy to find and amend later. It could also describe a stack of data values, how they relate and what functions you can apply to them. There are two simple data structures that blockchains combine:

- The linked list data structure

Linked lists are organized into straight lines. The different components are linked with pointers to the first node. The first node is referred to as the head and it has two parts. One half of the node contains data and the other half makes reference to the next node. They are all structured the same way and always have something to refer back to.

- The stack

  The stack is also a straight data structure. Its principle is; Last in, First out. So the last component in the stack is always taken out first. The first pointer is called ‘top.’ It’s how a programmer will keep track of the first component in the stack. Once you add another component to the stack, the top's value increases by one and becomes two so that it always appears on top.

Types of Blockchain Developers

Blockchain Developers are responsible for building and improving upon all Blockchain-related applications. They develop Decentralized applications and Smart contracts. They also design the architecture and protocols for a blockchain. So developers make secure digital transactions possible by generating systems and applications that are able to record and store data, while preventing external changes and compromise.

Core Blockchain Developer

Core developers are responsible for the architecture of the Blockchain. They are in charge of designing the protocols needed to operate a blockchain solution. They also design consensus protocols and make big decisions.

In order to be an effective core developer, you have to have a full understanding of Blockchain technology and its architecture. You have to know how hash, DLT and Consensus functions work. You also have to have a firm grasp on data structures and coding.

These are some of their main duties:

- Designing blockchain features that enhance functionality

- Designing blockchain and consensus protocols
- Mapping out security for the network
- Quality assurance

Blockchain Software Developer

Software developers put the architecture designed by core developers into practice. They generate decentralized applications in line with blockchain architecture and protocol. They deal with the front and back end development of these applications and they oversee the stack running the applications.

These are some of their main duties:

- Use core developer designs to implement smart contracts
- Develop the front and backend of decentralized applications
- Make sure that the applications are running as they should
- Make sure that blockchain integrates well with other apps and services.

Chapter 3: Blockchain Finance

The Benefits of Blockchain in Finance

Blockchain is a major innovation in the finance industry. It has the potential to reduce fraud, lower transaction times and manage any risks that may occur being an interconnected global financial system. The benefits are:

- Transparency

  It has standards in place that keep all users in mind and all processes are shared. All transactions can be viewed via a node or by making use of blockchain explorers. Explorers are websites that let you see transactions happening in real time.

- Trust

  The transparency and fixed nature of the ledger makes it easy for all users to cooperate and be productive.

- Privacy

  Businesses can be selective of the data they share in a business network. They have the choice to keep privileged information private but ensure that all parties have access to the information they need which is in line with transparency but still allows for a level of confidentiality.

- Superior performance

  On a daily basis, hundreds if not thousands of transactions can be made and recorded on one network. From time to time there is a spike in network activity but it has been built to sustain that.

- Security

  The blockchain framework eliminates single points of failure and middlemen, so there are less people involved in the assessing and recording of a business's data. The Ethereum blockchain has

specialized application code that is fortified. It is almost impossible to hack.

The Possible Drawbacks of Blockchain in Finance

A lot of the transactional tasks that have to be performed by individuals and companies these days are cumbersome. Blockchain has the potential to improve on a lot of those things just by it being automated. It’s important to note though, that the bigger the network gets, the more impractical it becomes to maintain.

The concept of blockchain is rooted in democracy because there is no central authority. Things are also more secure because there is no single point of failure. When a company builds its own blockchain, they are the only ones who have reasons to maintain it. In this way, nodes could become centralized and that is contrary to the purpose of a blockchain.

There is a great level of trust on the Bitcoin blockchain because Bitcoins are mined on it and everyone can see what happens to them from the moment they are mined to the time they end up in a person’s wallet. In the finance industry however, the assets being traded and recorded exist outside of the blockchain. Nobody can be 100% certain that all the information being logged on a blockchain is completely accurate and without mistakes. This could be a potential trust issue.

Lastly, Blockchains are completely digital and use up an enormous amount of computing power. One day there may not be enough computer power in the world to sustain them.

How the Digitization of Financial Instruments Could Impact Finance

Blockchain technology works very well in the world of cryptocurrency and has the potential to revolutionize financial services across the board.

Digitization takes the benefits of blockchain and applies them to the financial services sector. The way things work in the financial market will change and new standards will be set, bringing worthwhile changes at every turn.

Financial instruments in a digital format would offer several benefits. Some of which include:

- Credibility and scarcity: When data is in a digital format, it can be easily verified which makes it easier to trust.
- Programming capacity: Integrity and effectiveness would be on a whole new level because of tamper-proof software and Smart contracts.
- Efficient processes: Mechanized processes take the efficiency of operations up a notch by reducing waiting times and cutting out intermediaries. Auditing and reporting can be monitored while it's being done, so there’s less room for error.
- Economic benefits: Automation means doing away with a lot of redundant processes in order to achieve the same result. It brings infrastructure and operation costs down and increases liquidity.
- Market reactivity: Digital securities depict ownership of an asset. Should there be a need to alter or review this information it can be done way faster than with standardized securities. Other financial instruments can also be converted into a digital format to suit the needs of different investors.
- New markets and products: New products include fast, safe and adjustable transferal of assets, subdivided ownership of tangible assets and more

The effects of these benefits on operations would be major. Governance systems would be transparent and business models would improve. There would also be minimal operation costs that would free up some capital and increase liquidity.

There are a few companies that have slowly started implementing blockchain technology but they’re still in the experimental phase. If they implemented it and used it at its full capacity it could improve the following:

Payments

Sending money from one country to another the normal way takes time and ends up costing a lot of money in fees. Each party that has a hand in getting the money from point A to B receives a fee. If more people made use of blockchain applications this process would be faster, cheaper, and more secure.

- There would be more than one way to pay. Customers would be able to send Fiat money or cryptocurrency.
- The chances of falling victim to fraud would be significantly lower because of facilities like Know Your Customer (KYC). KYC is a set of standards that was created to protect financial institutions against fraud and money-laundering. There are several steps involved to ascertain customer identity, be sure of the nature of a customer's activities and ensure that wherever the money's coming from, it's legal. They also work to assess the risk of money-laundering with each customer.
- Smart contracts allow for two parties to enter into an agreement and for that information to be stored in the blockchain.

Asset Management

Real estate funds, Private equity firms and venture capital companies have to try and keep up with the ever-changing regulations in their sector. It's a task to stay compliant. Blockchains host of financial services can help them manage their assets and Blockchain security could further safeguard them.

Asset management can be improved through:

- Tokenizing securities and offering more liquidity. Tokenization is turning data like account numbers into an arbitrary string of characters called a token. The token becomes like a reference to the original data but does not expose the actual value of it.
- Confidential transactions having bespoke privacy settings
- Initiating a decline in human error in shareholder voting

- Better administration with more clarity for investors

How Does Blockchain Impact Capital Markets

Capital market is a blanket term that describes all the ways individuals, businesses and government trade financial instruments. They trade through the stock, bond, currency and foreign exchange markets. Most of the world's biggest markets are located in Hong Kong, London, New York and Singapore. Capital markets have suppliers and entities that make use of funds. Suppliers include retirement funds and companies that have surplus cash. The funds disbursed on capital markets are used mostly by people looking to purchase assets and governments that need financing for operating costs and infrastructure expenditure.

The markets are divided into two categories:

- Primary markets

  When a company makes an Initial Public Offering (IPO) , it happens in the primary market. When stakeholders acquire securities on the market, the company supplying the securities hires an underwriter to review it and produce an outline that states the price and other details of the securities being supplied. Everything pertaining to the primary market operates under strict regulations. Companies are required to file statements with the Securities and Exchange Commission (SEC) and other agencies before they can go public.

- Secondary markets

  Secondary markets are divided into Auction and Dealer markets. The auction market is where buyers and sellers gather and state the prices they are willing to buy and sell their securities for. One example is the New York Stock Exchange. Smaller investors trade through the dealer markets because trade mostly happens electronically.

  Companies have a number of hurdles in the market. The regulations are stringent which makes it take longer for them to get to market and interest rates are volatile, which threatens their liquidity.

Blockchain improves the way companies are able to operate in the Capital market by removing the single point of failure through decentralization and expediting capital market ventures by ensuring the efficacy of all processes.

Chapter 4: Bitcoin and Blockchain

What Is Bitcoin?

Bitcoin is an intangible form of currency that exists on the internet. It is not centralized and therefore operates outside the parameters of local authorities and banks.

There is a public ledger that documents all Bitcoin transactions and stores the data pertaining to them on servers known as nodes. These servers are located all over the world. Ownership of bitcoins can be verified across all servers.

Bitcoin is kept in a virtual wallet and that can be accessed from a computer with an internet connection. Bitcoins can be divided by up to 7 decimal places. This means that you can buy or transact with as little as a thousandth or a hundred millionth of a Bitcoin. A thousandth is called a *milli* and a hundred millionth is called *satoshi*.

What Is Bitcoin Mining?

Because you cannot actually touch Bitcoin and it doesn't get printed out like fiat money, it's easy to be confused on where it comes from and how it's 'mined'.

Simply put, they are mined out of the system. It is referred to as mining because, like other natural resources, there is a predefined amount of Bitcoin available. Technically, mining is adding the data from transactions onto a blockchain. Miners are vital to the network because they maintain the integrity of all payments and receipts. They use powerful computers to solve complex mathematical problems every ten minutes. Once the problem is solved, a block of all the latest confirmed transactions is added to the chain. Miners are rewarded a block of bitcoin for every problem they solve. The more miners there are, the faster transactions can be validated.

The mathematical problem or 'puzzle' that miners have to solve is called Proof of Work (POW).

Is It Safe to Buy?

Because of Blockchain technology, having Bitcoin is safer than leaving your money in the bank. It’s no secret that banks tend to be corrupt and have had several banking catastrophes. Bitcoin cannot be forged and unless you share your wallet’s password, it's virtually impossible for you to lose track of it.

From the outside looking in, it may appear hard to buy Bitcoin, but it’s easier when you start looking into it. All you need is an internet connection, some ID and a cryptocurrency exchange account. On top of the exchange account, you should download an app that allows you to have a personal cryptocurrency wallet. This way, you can pay for your Bitcoin with a debit or credit card. Bitcoin is also attainable at niche ATMs and through Peer-to-peer exchanges. P2P forms part of a decentralized network. You can swap and share information and data with no intermediaries. You can engage in anonymous exchanges of currency in a safe and secure way.

The Ins and Outs of the Bitcoin Blockchain

The bitcoin blockchain is an open ledger that is basically run by a network of unknown people. Historically, security models are based on concealing information and keeping people out. Bitcoin's security model is based on letting people in. It works because of self-interest. Each user is so focused on gaining that they aren’t worried about what another user is doing. People can access shared and updated records without having to communicate with one another.

If you are in possession of Bitcoin, you’re essentially in possession of a *private key.* A private key is one half of a digital signature that signifies that you own a Bitcoin. The other half of the key is public and is stored in the blockchain. Each block carries numerous transactions. Every transaction refers to a transaction that came before it.

When you want to make a purchase using Bitcoin, your request is broadcast and that request is received by all the computers on the peer-to-peer network. The owners of these computers are called miners. Miners collect requests and put them into a block. They then run the latest block and its predecessor through hash functions. Each miner has ten minutes to solve a cryptographic puzzle. Once it’s solved, it gets added as the latest block at

the end of the chain and the miner broadcasts it to their peers who verify it. The first miner to solve the puzzle gets rewarded with some Bitcoin.

The main advantage of having miners is that it beefs up security. The puzzles that the miners solve increase in complexity each time. So, every time a new block is added on to the chain it acts as a barrier for the blocks before it. Trying to tamper with a block is not easy. If a person wants to tamper with one, they have to be fast enough to change the information in every block that comes after it too. This all has to happen before any updates are made to the chain.

Because every miner is out to protect their own interests, the blockchain will always remain secure and functional.

Chapter 5: Blockchain Security

Blockchain is on two sides of the same coin, in that it has innovative practices that reinforce existing cybersecurity, but just like any other electronic entity, it needs securing of its own.

The CIA triad is a security model designed to protect information. The three principles are Confidentiality, Integrity and Availability. It aims to protect an organization's data, devices and networks.

The confidentiality principle ensures that all information goes undisclosed. This means that only approved users are granted access to view and change information.

The integrity principle relies on the confidentiality principle being functional. As long as information is safe and untainted, integrity remains intact.

The availability principle ensures that all devices work at full capacity so that the relevant parties have access to any and all information whenever they need it.

Blockchain has taken the CIA triad, implemented and in some instances, improved it.

In blockchain, we can look at the heavy encryption as adhering to the confidentiality aspect. Anything on a computer can be hacked, but the blockchain encryption is near impossible to break through. It would take a lot of time and energy to penetrate, so the information in the blocks is safe.

The verification of transactions within the blockchain framework speaks to the integrity principle. We can be certain that the information stored has not been interfered with.

Blockchain Security Basics

The fact that all the data and functions in the blockchain network are not concentrated in one place makes it easier to verify transactions. There are multiple computers in a single distributed ledger. For the chain to be

compromised, one person would have to assume control of at least 60% of those computers.

The largest blockchain networks allow members of the public access. On the surface that may seem like a bad idea, but having so many computers on one network boosts security. The more people there are on the network, the easier it is to verify transactions. Also, the more computers there are on the network the harder it becomes to gain unsanctioned access to any blocks on the chain.

Permissioned Blockchains

Permissioned Blockchains are private distributed ledgers that can only be accessed with pre-authorization. Not all users have far ranging control of all aspects in the blockchain though, there are ledger administrators who are in control of the access control layer of the ledger. They are also subject to verification checks but are the only people who are authorized to grant other users access. They are mainly used by companies that have a lot of sensitive information, like banks. Ripple is a permissioned blockchain that was developed by the banking industry to cut transaction costs.

The key characteristics of a permissioned blockchain are:

- There is limited transparency
- There is no anonymity
- There is also no central authority, but ledger administrators can allow or deny access.

Advantages of a Permissioned Blockchain:

- Provides fixed signatures that encrypt all transactions, making them more secure
- All users in the network are completely anonymous
- Because there are no third parties, there isn’t a long waiting period between transactions
- Information is not stored in a central archive so it's easily accessible from any location at any time

Disadvantages of a Permissioned Blockchain:

- The security of a permissioned network relies heavily on the integrity of its users
- There is an element of censorship, because ledger administrators can restrict transactions

Permissionless Blockchains

Permissionless Blockchains are open to the public. Anybody with a computer can transact on these. All data is stored on several computers around the world and users are pretty much anonymous. Most cryptocurrencies use these networks.

The key characteristics of a permissionless blockchain are:

- Complete transparency on all transactions
- Anonymity
- Operates without a central authority

Advantages of a Permissionless Blockchain:

- Less prone to cyber attacks. A hacker would have to gain control of at least 51% of the network to be able to do any damage
- Uncensored. Because there are so many nodes participating in the network, there cannot be a central authority

Disadvantages of a Permissionless Blockchain:

- Even though we've praised blockchain networks for being completely anonymous, there are some drawbacks. If by chance, somebody was to successfully commit fraud, it would be difficult to track them down. Also, people have been reported to use the blockchain network for unlawful activities.
- Bitcoin relies heavily on the PoW system, and while it ensures security, it consumes a lot of energy and ends up being quite costly.
- The block size on a permissionless blockchain is limited.

The Role of Miners in Blockchain

Mining has really taken off since Bitcoin arrived on the scene. Not only do they receive Bitcoin, but they add to the blockchain security protocol. Miners task themselves with validating all transactions and making sure

they adhere to the Bitcoin code. On the Bitcoin blockchain, miners present their Proof of Work which signifies why they either approve or deny a particular transaction. They get paid in Bitcoin for every transaction they successfully validate.

How Blockchain Security Averts Double Spending

Blockchain developers are cautiously optimistic about the steps they've taken to ensure visibility and security on different networks. But they are realistic about the fact that there is always a slim chance that some things can go unseen. Blockchains could fall victim to a double spending attack and a user could spend one unit of cryptocurrency a number of times before anything is picked up on the network.

On the blockchains of the various cryptocurrencies, each network has to reach a consensus on the order of a transaction to be able to confirm and share them in full view of the entire network.

The first ever cryptocurrency to solve the double spending problem was Bitcoin. Whenever a user would try and spend the same coin twice by sending it to two other users at the same time, both transactions would end up in a store for unconfirmed transactions. The first payment would be added to the coin's chain as a block. The second transaction wouldn't fit on the chain because it would be connected to the first and then the transaction would fail completely.

Blockchain Security Risks

Blockchain is a pretty solid technology but it has its vulnerabilities. Even though it's difficult, there are a few ways Hackers could compromise the system. These are some of the ways:

- Because the blockchain is decentralized there's a ton of information traveling through cyberspace. A talented hacker could gain access to data while it's on its way to the Internet Service Provider.

Unfortunately there is no alert for this, so attacks like this can go unnoticed.

- Wide-reaching public blockchains rely on a large number of computers for the purpose of mining. If a group of miners were to get together, they would have enough computing power to take over 50% of the network's mining capacity. This is called a 51% attack.
- Hackers can create an inordinate amount of false identities and crash a specific blockchain network. These are called Sybil attacks.
- An age-old practice for hackers is phishing. Unfortunately blockchain isn't immune. Hackers send well designed and constructed false emails requesting the login credentials of cryptocurrency wallet holders.
- The lack of human involvement was originally what sold a lot of people on Blockchains, but now it's receiving some mild criticism. People have cited the anonymity that the network allows as being the reason why it is used by unsavory characters like terrorist groups. It is said that they make use of blockchains and cryptocurrency to launder their money thus eluding banks and the long arm of the law.
- As previously stated, blockchains need a ton of power to operate. According to Morgan Stanley, the bitcoin network consumes the same amount of energy as 2 million homes in the United States. The mining, which is a huge part of the model, is responsible for a lot of that consumption. The worst case scenario would be that the price of coins decreases and the price of electricity increases. This would see the number of miner's decreasing heavily.

Blockchain Security in Practice

While there is a standard for how a blockchain is supposed to operate, not all blockchains are exactly the same. Some companies have amended their security measures, either to save on costs or to cater to their specific security needs.

Here are six examples:

- Mobilecoin

Mobilecoin is a cryptocurrency company located in California. They are generating a safe and easy to use cryptocurrency for companies that can't afford to implement ledger security procedures on their own. Mobilecoin replaces transaction service providers by keeping transaction information encrypted on both sides. Mobilecoin works with WhatsApp and Facebook Messenger.

- Coinbase

Coinbase, also based in California is a company that facilitates the acquiring and exchange of most, if not all cryptocurrencies. Every customer gets a digital wallet with a password. Coinbase relies solely on encryption and every employee is subject to exhaustive background checks to ensure the safety of their customers' cryptocurrency.

- J.P. Morgan

J.P. Morgan is a well-known financial institution in the United States. They have developed a version of Ethereum called Quorum. It caters to the specific needs of enterprises and uses blockchain technology to execute private transactions. They make use of Smart contracts on their network to ensure transparency and accuracy on all transactions.

- Lockheed Martin

Lockheed Martin is a defense contractor. A defense contractor is an entity that has an agreement with the United States to produce arms or take up arms for the national defense force. It is the first defense contractor to make use of blockchain security. They are collaborating with Guardtime federal to implement blockchain protocols in their software development, supply chain processes and engineering systems. They are using a blockchain network to keep track of their arms development systems.

- Cisco

Cisco is one of the largest tech companies, also located in California. They proposed the idea that blockchain complements the Internet of Things because there are no single points of failure and important information can be protected by encryption. If the Internet of Things adopts blockchain as it's preferred network, it could see a spike in Blockchain's profile.

- Hashed Health

Hashed Health is a healthcare innovation firm located in Tennessee. They want to spearhead the movement to get the healthcare industry to make use of blockchain technology. Hashed Health has various subsidiaries and each one of them utilizes different facets of blockchain. They have already worked with a number of hospitals and facilitated the building of various blockchain networks that store and give access to patient information.

Conclusion

Even though there has been tons of enthusiasm around Blockchain and how it shot up to the top spot in the tech world, there are analysts with pessimistic views. The biggest talking point is that it can turn pretty much any industry on its head. Blockchain promises to transform how we transact, how we enter into contracts and how we protect data. Analysts have likened Blockchain to the internet, stating that it could be more than twenty years before we see blockchain being a part of how businesses operate on a daily basis. If central authorities and businesses seriously invest in the infrastructure and rework how they operate, then we would be able to see more extensive use of blockchain technology.

We cannot predict the future, but the prospective advantages of blockchain are at least fascinating enough for there to be some sort of insurgence far into the future.

If you are just starting out on your software development journey, you can't be too hard on yourself if you don't grasp everything at lightning speed. If you learn with the end goal in mind, it will be a smoother process.

Having the aptitude to fully understand blockchain and its concepts will always be an advantage to you. You may find yourself working on ICO projects. Initial Coin Offering is to the cryptocurrency industry what an IPO is to the business world. Companies use ICOs to raise the funds necessary to create a new coin, application or service.

One of the main benefits of learning blockchain is that it's in high demand, so you will probably never be without work. If you choose not to have a career in the blockchain industry, you can always teach, it's an excellent way to make money.

It's fairly new, so only 0.5% of the population is using it right now. If you start learning it today, you will be at the forefront of its development by the time it has gained more traction.

The demand for Blockchain shot up almost 2000% and it became one of the most sought after skills in the tech industry. It is estimated that by 2024 the Blockchain market's net worth will be valued at $20 billion. Right now, a

significant number of banks are implementing Blockchain technology to improve the quality of their service and to tighten up their security.

The cities with the most openings are New York, San Francisco and San Jose and the median salary for a blockchain developer stands at $154,550 as of 2022. There is always room to grow in this field as the technology improves and advances. You should become a blockchain developer because:

- The demand for Blockchain developers far outweighs the supply. This means the job market is not saturated and probably will not be for a long time.
- The salaries are among some of the most competitive in the tech space right now. With the right amount of drive and experience, you could earn up to $200,000 a year.
- You get to be involved in revolutionizing the way the financial market operates.

A Step-by-Step Guide for Beginners

1. Determine your goal

Be sure of what you'd like to do with the new skills that you learn. Is it purely to make a positive contribution to the world? Most people are eager to learn because of the job prospects but you should know whether you want to be employed or to be doing the employing.

2. Learn the basics

This book is an excellent start, because you get an overview of blockchain's architecture, software, networks, systems and processes. Additionally, you can source a glossary to better understand the most important concepts.

3. Familiarize yourself with the process

Once you've got step 1 down, it only makes sense to delve deeper into things. It would be a good idea to get some practical experience by

acquiring a small amount of cryptocurrency. Being on the client-side may help you have more of an all-round understanding of things. Also, learning about the blockchain ecosystem will put you at an advantage.

4. Watch tutorials and learn how to code

Steps 1 and 2 have given you basal and practical knowledge so the next logical step is to start coding. If you do not have any coding experience, you should look into Solidity. It’s easy to learn and will help you with Smart contracts.

5. Gain an in depth knowledge of Smart Contracts

Smart contracts are a must learn! Once you’ve developed your coding skills it will be easier to get into.

6. Keep up to date with changes in Blockchain development

Just like in medicine, you never stop learning coding. It’s important to be up-to-date with what’s going on and learning all you can about any new developments. It may be helpful to subscribe to a few blogs.

7. Sharpen your skills

Once you feel confident enough to call yourself a developer, don’t be afraid to go back and tackle things that you may find difficult. Everything gets easier with time.

B

O

N

U

S

B

O

O

K

B

E

L

O

W

References

BDO. (2019, January). *Cryptocurrency: The top things you need to know.* BDO. https://www.bdo.com/insights/assurance/financial-reporting/cryptocurrency-the-top-things-you-need-to-know#:~:text=How%20are%20cryptocurrencies%20created%3F&text=Some%20cryptocurrencies%2C%20like%20Bitcoin%2C%20are

Binance Academy. (2018, October 30). *The history of blockchain | Explained for beginners.* [Video]. YouTube. https://www.youtube.com/watch?v=ZbHLNinXy9E

Bitcoin. (2009). FAQ - Bitcoin. Bitcoin. https://bitcoin.org/en/faq#general

Blockgeeks. (2018a, November 8). *What is cryptocurrency? A simple explanation.* [Video]. YouTube. https://www.youtube.com/watch?v=6Gu2QMTAkEU

Blockgeeks. (2018b, December 7). *What is hashing on the blockchain?* [Video]. YouTube. https://www.youtube.com/watch?v=IGSB9zoSx70

CareerKarma, & Aina, M. (2021, September 15). *How to become a blockchain developer.* CareerKarma. https://careerkarma.com/blog/how-to-become-a-blockchain-developer/:~:text=There%20are%20two%20types%20of,architecture%20of%20the%20blockchain%20system.

CompTIA. (2019). *How blockchain and cybersecurity work together | CompTIA.* [Video]. YouTube. https://youtu.be/kV72OVWhg6o

ConsenSys. (2020). *Finance | ConsenSys.* ConsenSys. https://consensys.net/blockchain-use-cases/finance/

Cope, J. (2002, April 8). *What's a peer-to-peer (P2P) network?* Computer World. https://www.google.co.za/amp/s/www.computerworld.com/article/2588287/networking-peer-to-peer-network.amp.html

Corporate Finance Institute. (n.d.). *Bitcoin mining - Overview, benefits, and requirements.* Corporate Finance Institute. https://corporatefinanceinstitute.com/resources/knowledge/other/bitcoin-mining/

Farr, A. (2018, December 31). *A beginner's guide to blockchain programming.* Simple Programmer. https://simpleprogrammer.com/beginners-blockchain-programming/#:~:text=As%20you%20might%20have%20deduced

Frankenfield, J. (2019a). *Bitcoin mining, explained.* Investopedia. https://www.investopedia.com/terms/b/bitcoin-mining.asp

Frankenfield, J. (2019b). *Distributed ledger technology.* Investopedia. https://www.investopedia.com/terms/d/distributed-ledger-technology-dlt.asp

Frankenfield, J. (2019c). *Initial coin offering (ICO).* Investopedia. https://www.investopedia.com/terms/i/initial-coin-offering-ico.asp

Frankenfield, J. (2019d). *Merkle tree.* Investopedia. https://www.investopedia.com/terms/m/merkle-tree.asp

Frankenfield, J. (2019e, August 11). *Proof of stake (PoS).* Investopedia. https://www.investopedia.com/terms/p/proof-stake-pos.asp

Frankenfield, J. (2021, September 23). *Peer-to-peer (Virtual currency).* Investopedia. https://www.investopedia.com/terms/p/ptop.asp#:~:text=Peer%2Dto%2Dpeer%20refers%20to

Garbade, D. M. J. (2019, August 2). *How to become a blockchain developer.* Medium. https://ledumjg.medium.com/how-to-become-a-blockchain-developer-dd0d4b83cc37

GeeksforGeeks. (2013, March 8). *Linked list | Set 1 (Introduction) - GeeksforGeeks.* GeeksforGeeks. https://www.geeksforgeeks.org/linked-list-set-1-introduction/

GeeksforGeeks. (2020, April 10). *RSA full form.* GeeksforGeeks. https://www.geeksforgeeks.org/rsa-full-form/#:~:text=RSA%20stands%20for%20Rivest%2C%20Shamir

Goel, A. (2021, December 6). *Blockchain programming: Step-by-step guide for beginners (with code).* Hackr. https://hackr.io/blog/blockchain-programming-beginners-guide

Hayes, A. (2019a). *Capital markets: What you should know.* Investopedia. https://www.investopedia.com/terms/c/capitalmarkets.asp

Hayes, A. (2019b). *How to buy bitcoin.* Investopedia. https://www.investopedia.com/articles/investing/082914/basics-buying-and-investing-bitcoin.asp

Hayes, A. (2022, January 20). *Blockchain explained.* Investopedia. https://www.investopedia.com/terms/b/blockchain.asp#toc-transparency

IBM. (2015, September 14). *What is distributed computing.* IBM. https://www.ibm.com/docs/en/txseries/8.2?topic=overview-what-is-distributed-computing

IEEE Spectrum. (2016, July 6). *The bitcoin blockchain explained.* [Video]. YouTube. https://www.youtube.com/watch?v=M_zCjjy59cg

ISC. (2020, October 21). *How blockchain job market is booming! | International Search Consultants.* International Search Consultants.

https://iscjobs.com/how-blockchain-job-market-is-booming/

Ivan on Tech. (2021, May 21). *What is Ethereum Virtual Machine (EVM)? Programmer explains.* [Video]. YouTube. https://www.youtube.com/watch?v=GPoze5RmDVU

Kids Kiddle. (n.d.). *Internet of Things facts for kids.* Kids Kiddle. https://kids.kiddle.co/Internet_of_things

Klepatch, J. (2018, November 1). *How to become a blockchain developer: Step-by-step plan.* Eat The Blocks. https://eattheblocks.com/how-to-become-a-blockchain-developer-step-by-step-plan/#step1

Levy, A. (2021, October 25). *15 applications for blockchain technology.* The Motley Fool. https://www.fool.com/investing/stock-market/market-sectors/financials/blockchain-stocks/blockchain-applications/

Marín, J. (2019, January 30). *Blockchain as a data structure.* Medium. https://medium.com/@juliomacr/blockchain-as-a-data-structure-3bd125d8ddda

McAfee. (n.d.). *What is tokenization vs encryption - Benefits uses cases explained | McAfee.* McAfee. https://www.mcafee.com/enterprise/en-us/security-awareness/cloud/tokenization-vs-encryption.html#:~:text=Tokenization%20definition

Nunns, J. (2016, June 20). *What is a node?* Tech Monitor. https://techmonitor.ai/what-is/what-is-a-node-4927877

Official, A. (2020, August 28). *Key characteristics of cryptocurrency and why do they matter to you.* Medium. https://medium.com/the-capital/key-characteristics-of-cryptocurrency-and-why-do-they-matter-to-you-5f33e483a40f

Petrov, C. (2019, March 11). *91+ blockchain statistics, facts, and trends for 2021*. TechJury. https://techjury.net/blog/blockchain-statistics/#gref

Programiz. (n.d.). *Stack data structure.* Programiz. https://www.programiz.com/dsa/stack

Richards, K. (2021, September). *What is cryptography? Definition from SearchSecurity.* SearchSecurity. https://www.techtarget.com/searchsecurity/definition/cryptography

Security Scorecard. (2021, September 1). *What is the CIA Triad? Definition and examples.* Security Scorecard. https://securityscorecard.com/blog/what-is-the-cia-triad

Sharma, T. K. (2018, October 3). *Advantages and disadvantages of permissionless blockchain.* Blockchain Council. https://www.blockchain-council.org/blockchain/advantages-and-disadvantages-of-permissionless-blockchain/#:~:text=It%20provides%20immutable%20signatures%20for

Simplilearn. (2018, July 26). *Cryptocurrency explained | What is cryptocurrency? | Cryptocurrency explained simply | Simplilearn.* [Video]. YouTube. https://www.youtube.com/watch?v=8NgVGnX4KOw

Simplilearn. (2019). *Blockchain in 7 minutes | What is blockchain | Blockchain explained simply | Blockchain |Simplilearn.* [Video]. YouTube. https://www.youtube.com/watch?v=yubzJw0uiE4

Simplilearn. (2021, July 19). *What is blockchain security and its top 6 examples | Simplilearn.* Simplilearn.

https://www.google.com/amp/s/www.simplilearn.com/what-is-blockchain-security-and-its-examples-article/amp

Simply Explained - Savjee. (2017). *How does a blockchain work - Simply Explained.* [Video]. YouTube. https://www.youtube.com/watch?v=SSo_EIwHSd4

Sofi Learn. (2021, November 15). *Curious about blockchain? Here's an overview of the technology.* SoFi. https://www.sofi.com/learn/content/what-is-blockchain-technology/

Solidity Lang. (n.d.). *Solidity — Solidity 0.8.11 documentation.* Solidity Lang. https://docs.soliditylang.org/en/v0.8.11/

Sparkes, M. (n.d.). *What is bitcoin and how does it work?* New Scientist. https://www.google.com/amp/s/www.newscientist.com/term/bitcoin/amp/

Tech Terms. (2018, January 23). *Bytecode definition.* Tech Terms. https://techterms.com/definition/bytecode

Techopedia. (2013, February 9). *What is a Torrent? - Definition from Techopedia.* Techopedia. https://www.techopedia.com/definition/5263/torrent#:~:text=A%20torrent%20is%20a%20file

The Economic Times. (2022, February 5). *What is cryptography? Definition of cryptography, cryptography meaning.* The Economic Times. https://www.google.co.za/amp/s/m.economictimes.com/definition/cryptography/amp

Tutorials Point. (2012). *Cryptography hash functions.* Tutorials Point. https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm

Valente, M. (2019, July 23). *What is proof of authority?* Coinhouse. https://www.coinhouse.com/learn/what-is-proof-of-authority/

Warholak, M. (2021, March 15). *What is the double spending problem with bitcoin?* SoFi. https://www.sofi.com/learn/content/double-spending/#:~:text=The%20double%20spending%20problem%20is

Wu, A. (n.d.). *Blockchain consensus: What is it and why do we need it?* [Video]. FutureLearn. https://www.futurelearn.com/info/courses/blockchain-cryptocurrencies/0/steps/76497

B

O

N

U

S



JavaScript for Beginners - JQuery

Learn JavaScript Quickly. Discover All You Need to Know

Jefferey Smith

© Copyright 2022-All rights reserved.

The content contained within this book may not be reproduced, duplicated or transmitted without direct written permission from the author or the publisher.

Under no circumstances will any blame or legal responsibility be held against the publisher, or author, for any damages, reparation, or monetary loss due to the information contained within this book, either directly or indirectly.

Legal Notice:

This book is copyright protected. It is only for personal use. You cannot amend, distribute, sell, use, quote or paraphrase any part, or the content within this book, without the consent of the author or publisher.

Disclaimer Notice:

Please note the information contained within this document is for educational and entertainment purposes only. All effort has been executed to present accurate, up to date, reliable, complete information. No warranties of any kind are declared or implied. Readers acknowledge that the author is not engaged in the rendering of legal, financial, medical or professional advice. The content within this book has been derived from various sources. Please consult a licensed professional before attempting any techniques outlined in this book.

By reading this document, the reader agrees that under no circumstances is the author responsible for any losses, direct or indirect, that are incurred as a result of the use of the information contained within this document, including, but not limited to, errors, omissions, or inaccuracies.

Table of Contents

Introduction
- Learn JavaScript Quickly
- What Is JQuery ?

Chapter 1: JavaScript for Beginners
- Syntax
  - JavaScript in Browsers
  - How to Get Started Writing JavaScript
  - Datatypes in JavaScript
    - Numbers
    - BigInt
    - Strings
    - Null
    - Undefined
    - Booleans

Chapter 2: JavaScript Programming
- Functional Programming
    - Types of Functions
    - Higher-Order Functions
    - What Are Conditional Statements?
- Object-Oriented Programming
- Procedural Programming
  - Fundamental Features of Procedural Programming
  - Applications of JavaScript Programming

Chapter 3: JavaScript Architecture
- Typical JavaScript Application Architecture
- Framework-Based Architecture
- JavaScript Engines
  - Web API’s

Event Loops
Advanced JavaScript Architecture
Chapter 4: JavaScript and JQuery
JavaScript Frameworks
Features of JQuery
Chapter 5: JavaScript for Kids
HTML and CSS
Important JavaScript Terms
Data Types in JavaScript
Variables
Arrays
Objects
Conditionals and Loops
Functions
Three Ways to Write Your First Line of Code
Conclusion
References

Your FREE Gift

I have something for you. It won't cost you a dime. This is a FREE 40-page PDF guide. I'd like you to have free copies with my compliments. You can grab Your copy by clicking on the following link and joining my mailing list.

FREE BOOKS https://imris-kdp-publishing.com/

Before we jump into this Book, I'd like to express my thanks. I realize there are many books on this subject that you could otherwise spend your time reading. You're taking a chance on me. I appreciate it. Giving you a FREE PDF copy of this amazing guide is my way of showing that appreciation. On that note, let's dig in. You're going to love what's coming your way in the following pages

Introduction

Programming is essentially telling a computer what to do, when to do it and how to do it. It is giving your computer a series of instructions to execute. Programming controls most electronic objects. From computers, to vending machines, to cellphones and most importantly, the web.

JavaScript has been coined the language of the web. This is because most, if not all websites use it to make their pages appear more dynamic. It is easy to learn and understand, plus it doesn't cost too much to update and upgrade. A lot of the software that's used is open source. There are also lots of websites with free activities you can use to practice.

It is a programming language that is rooted in text and it allows web pages to be interactive. It provides attractive elements that bring users to web pages. Common examples of JavaScript you use everyday are Amazon searchboxes, New York Times embedded videos or Twitter feed updates. Making use of JavaScript improves the user experience of a website by converting static pages to interactive or animated pages. So, in short, JavaScript adds behavior to web pages.

Most of the time, JavaScript is used inside web browsers and on web-based apps. Beyond that, it is used on servers, in software and on embedded hardware controls. Some examples of interactive features on web pages are: zooming in and out on an image, playing audio and video on a web page and showing animations.

If you are just starting out and your goal is to become a software developer, then you must be well versed in JavaScript. It is the most established programming language in the world. Once you are able to master it, you will be able to develop front and back-end software. JavaScript is omnipresent and is pre-installed in every contemporary web browser, so you don't have to go out of your way to study it.

There are many structures within JavaScript that developers can use when constructing mobile and web apps. *JavaScript frameworks* are bodies of code libraries that offer pre written code for everyday programming tasks and features. One such library is JQuery.

At first, JavaScript was only used in Browsers to build interactive web pages and it was referred to as *Toy language* by developers. That quickly changed when it started receiving a lot of support and getting investment from large companies like Google and Facebook. These days you can build full scale mobile apps, Real-time networking apps, Command-Line tools and Games.

Learn JavaScript Quickly

When embarking on your coding journey, you may have a constant feeling of having a hard time. Your progress can seem slow and it's easy to want to throw in the towel. Here is what I can recommend for you to learn expeditiously:

1. Appropriately emphasize the amount that will bring the greatest benefit.

Pinpoint the main components of JavaScript and study them more intensively. Do not feel the need to study everything equally. The Pareto principle states that in certain situations, 80% of the effects result from 20% of the causes. So, if you focus on 20% of the content in this book that will bring about 80% of the results you desire, you will be able to learn in record time. Key areas to focus on are Functions, Variables, Conditional statements, Objects and Arrays.

2. Be diligent.

Watching tutorials and reading is extremely helpful when learning something new. But, it would serve you better to devote more of your time to practicing. When you learn to ride a bike, you have to actively move your limbs in order to move forward. The same thing applies when learning JavaScript, you learn faster by writing code on your machine. In addition to this, dedicating your time is essential. Setting aside an hour to practice and write code at least five times a week will greatly improve your knowledge.

Challenge yourself, and build an actual project with JavaScript, no matter how small. Doing this will just cement everything you've

learned. You will also be able to identify what you might be struggling with and apply extra effort in those areas.

3. Learn how to troubleshoot errors.

When learning JavaScript you will come across bugs. Harnessing the skill to solve those problems will always be useful. It will make you sharp as a programmer and allow you to finish projects faster.

What Is JQuery ?

Many years ago, when programmers wanted to change the look and feel of the attributes on their web pages, they had to include lots of font tags and color attributes. This made their HTML pages become overloaded with code and it was very hard to maintain. This meant that all web pages had to be maintained and modified manually. After the invention of CSS they were able to change the appearance of websites without having to write endless lines of code for each page. When JavaScript came along, programmers began writing the JavaScript code into their HTML documents because they weren't well versed on it at the time. Eventually, they applied the CSS theory to JavaScript and JavaScript libraries were born. Among all the libraries in existence at the time, JQuery stood out and we still use it today.

When you google JQuery, all that comes up is that it is a JavaScript library, but there is more to it than that. It simplifies the scripting of HTML on the client-side. Its main objective is to make using JavaScript easier for you. It can also be used to make adding animation easier. It basically takes a whole lot of common tasks that need several lines of JavaScript to run and attaches them to form a single line of code.

What do client-side mean and server-side mean?

Client-side and server-side are also referred to as front end and back end even though they aren't entirely the same thing. Client side is where the application code runs and server-side is where the source code is stored.

Some of the most well known features of JQuery are:

- HTML manipulation

- DOM manipulation
- CSS manipulation
- Effects and Animation
- Ajax
- Utilities

JQuery became popular because there was a demand in the market for a tool to simplify the coding experience. JQuery is faster than its counterparts and is constantly being extended. It aids in writing User Interface function codes in as few lines as possible. It is compatible with all browsers and is up to date with their latest features.

Chapter 1: JavaScript for Beginners

Before you can begin learning how to code, there are some Prerequisites. You have to at least have a basic knowledge of HTML and CSS.

HTML stands for Hypertext Markup language. It is not considered a programming language. Like its name suggests, it is a Markup language that is used when structuring a web page. At its core, it is a very simple language that incorporates the elements applied to text to give it a distinct meaning. You can use these elements to wrap, markup or enclose components on your page and make them look or act a particular way. It can determine whether the text in your document comes in the form of a paragraph, has bullet points or comes in the form of a table. It can also embed pictures and videos onto your page.

When you go on to any website, a number of files are dished out to you. At a rudimentary level your website will present you with an HTML page and a script.js page. The HTML page is the bare bones of the web page. It tells the browser what to show and where everything goes. So if there needs to be a title at the top, a paragraph in the middle and a picture at the bottom, the HTML file is responsible for relaying that information to the browser. This is all well and good but it's very limiting because the page will always be static and the data cannot change. That's where JavaScript comes in. JavaScript is able to manipulate the appearance of the website by accessing the DOM. DOM stands for Document Object Model. It tells JavaScript what is on the page and then allows it to make changes to elements of the page. DOM makes it easier to read, access and obtain the content of an HTML document. If we break it down even further, the Document is the HTML file, the object is the tags and elements within the HTML file and the Model is the Layout or structure. Without DOM, JavaScript would not have a perception of any of the elements on a web page and would therefore not be able to change or manipulate them. The DOM information is available through an API. API stands for Application Programming Interface. It basically gives JavaScript an opening to control things on the HTML page. The API acts as the go between for JavaScript and HTML.

HTML syntax is simply the way HTML is written. HTML elements are structured a certain way; they are made up of the opening tag, the content

and the closing tag. The opening tag is where the element commences. Visually, it is the name of the element wrapped in opening and closing angle brackets. For example, when your element is a paragraph, the opening tag would look like this: . The content of the element goes in the middle, in this case, it would be the text in the paragraph. The closing tag marks where the element ends. It is similar to the opening tag, but it has a forward slash after the first bracket. In this case it would look like this: . If you forget to insert the forward slash or fail to include the closing tag altogether, you may not achieve the desired results.

Elements are split into two categories, namely: Block level and inline elements. Block-level elements create blocks that you can actually see on a page, they take up the full width of a web page by creating a line break. Inline elements take up only as much as they need and are able to add content to a page without any line breaks.

CSS or Cascading Style Sheets is a styling language used for describing the appearance of a document written in HTML. It is used in the formatting and layout of web pages. It is basically responsible for making your web page look really good. CSS syntax is pretty straightforward. The first part of the syntax is the selector, it is followed by opening and closing curly brackets. The brackets are used to show the start and the end of the style that is going to be applied to the selector. It can look something like this: selector { property1: value;} . Property refers to color, size or length, while value refers to the text or image the property is being applied to.

Syntax

In order to execute JavaScript, we have to put statements inside of *script* HTML tags on a web page. For example: <script> JavaScript code </script>. The script tags are made up of two properties:

- ➔ Language

  Defines the scripting language you are working with. The value will always be JavaScript.

- ➔ Type

Stipulates the scripting language being used and should always be set to ‘text/javascript’.

Your segment will look like this:

<script language = “javascript” type = “text/javascript”> JavaScript code </script> (*JavaScript - Syntax*, n.d.)

You have complete freedom to use spaces, tabs and newlines in your program, you can also format and indent your code in a way that is simple to read and understand for you, but JavaScript ignores these things. In JavaScript, simple statements are usually followed by a semicolon, but you have the choice to omit it if your statements are written on different lines.

If your statement looks like this:

<script language = “javascript” type = “text/javascript”>

<! - -

var1=20

var2=30

// -–>

</script> (*JavaScript - Syntax*, n.d.)

You can write it without semicolons. When your statement is written in a single line format and looks like this:

<script language = “javascript” type = “text/javascript”>

<! --

var1=20; var2=30;

// -–>

</script> (*JavaScript - Syntax*, n.d.)

Then you must make use of semicolons.

JavaScript is case sensitive, all variables and other identifiers must be written in all caps.

Comments in JavaScript are used to add notes to your code, or deactivate portions of the code without fully removing them. We produce comments by putting a forward slash before a single line or adding a forward slash

followed by an asterisk after several lines. Commenting on code is a core feature of a lot of programming languages.

Here is an example of how to use comments in JavaScript:

<script language = “javascript” type = “text/javascript”>

<! --

//comment

/*

*multi-line comment

*multi-line comment line 2

*/

//-- >

</script> (*JavaScript - Syntax*, n.d.)

When a user operates a web page, JavaScript and HTML interact, this is called an *Event.* That means that any and all actions performed on a web page are an event. Developers code responses to all events. So if you scroll over any of the windows on your browser, and you see the name of the website and web address, that is a response to an event. There are many event types, two of them being:

Onload - The Onload event gets activated when the page loads

Onmouseover - Is activated when the cursor moves on or around an element

JavaScript in Browsers

JavaScript is supported on all current browsers, but sometimes you may have to manually enable or disable it. This is how you manage it on each different browser:

Firefox:

- Open a new tab, select the address bar and then type in: ‘about: config’
- The warning page will appear, accept the risk and continue

- The list of configure options will appear in the browser
- Go to the search bar and type in javascript.enabled
- The toggle will appear and you will be able to switch between disabled and enabled as you please.

Chrome:

- Click on the menu in the top right hand corner
- Click on settings
- Go down to the end of the page and click on Show advanced settings
- Go to the privacy section and click on the content settings button
- There you will find the JavaScript section, you will have the option to either allow or disallow sites to run JavaScript.

Edge:

- Click the settings and more button
- Select settings
- Find the JavaScript section
- Choose whether to enable or disable JavaScript with the toggle

How to Get Started Writing JavaScript

There are three methods of writing JavaScript. The first, is to write it directly into the Browser. This is the easiest way because you don't need to set anything up and you can get started by trying out some simple comments. The downside is that if you refresh the page, the code will disappear and you will have to start from scratch. You can also only execute one command at a time.

If you are developing, you are going to need to be able to save your code and execute multiple commands at any given time. For that you are going to want to write JavaScript in a simple text file and then send it to the browser to execute. Browsers accept files with the extension .html. You can create HTML files with Notepad in windows and with Textedit on Mac. Once you

have written your code into the file, you need to change the format to HTML. You can achieve this by selecting the html extension when saving.

Even though you can write multiple lines of code and include various comments and variables, there are some disadvantages to using a simple text editor. You have to be extremely meticulous because they don't help you in writing code. Keywords are not highlighted, syntax errors cannot be detected and code completion is completely up to you.

Another way you can write your code is with Special Code Editors. Special Code Editors support various programming languages and have specialized features to help you write your code. They highlight different keywords and alert you when there is a syntax error. An editor that is available for free online is *Visual Studio Code*.

When you are coding, you can put your code anywhere on the file, but the most popular way to do it is either in the head or the body. We put script in the head section if we want it to run when a user clicks on something. It will look something like this:

```
<html>
        <head>
                <script type = "text/javascript">
                < ! - -
                function sayHello() {
                alert("Hi there testing")
                }
                //- - >
                </script>
                </head>
        <body>
                <input type = "button" onclick = "sayHello()" value = "Say Hello"
        </body>
```

<html> (*JavaScript - Syntax*, n.d.)

That code would produce a button, and once you click on that button there would be a pop up that says *Hi there testing*.

Datatypes in JavaScript

Data Types are the values that are supported by a programming language. In JavaScript you will work with six; Numbers, BigInt, Strings, Null, Undefined and Booleans.

Numbers

Before we can understand the numbers value, we need to have an idea of what ECMAscript is. ECMA is a firm that sets standards for technology. ECMAScript is one of those standards. ECMAScript carries specifications for scripting languages, numbers and BigInt are built into it. JavaScript keeps to the ECMAScript specification. This means that when we are actually coding we make use of JavaScript and not ECMAScript. The number type is used to represent integer values and floating point values. Integers do not have decimal points, and floating numbers do.

BigInt

BigInt is an abbreviation for Big Integer. An integer is a whole number without decimals. This data type was created for JavaScript in early 2020 because the number data type was limited. When the script encountered numbers higher than 2^53 - 1 they tended to lack accuracy thus affecting the efficacy of applications.

Strings

A string is a chain of one or more symbols. It could include anything from letters to numbers. Strings are primitive which means they cannot change.

Null

Null is the value shown when there is no object.

Undefined

A variable that can’t be identified by its characteristics becomes undefined in JavaScript. A lot of people are unsure what the difference between Null and Undefined is. Undefined is a variable that has no value, while null is an empty value assigned to a variable.

Booleans

Booleans only have two possible values. True or false. Boolean is usually used with conditional statements in JavaScript.

Once you have a firm grasp on the Syntax and all the jargon, you'll want to turn your attention to programming, because it's the fun part, but it is also going to determine how your apps and websites turn out.

Chapter 2: JavaScript Programming

The best way to get started with JavaScript programming, is having an understanding of programming paradigms. We should be able to differentiate between them. There are three paradigms; Functional programming, Object-oriented programming and Procedural programming. A paradigm is a set of concepts and standards that make up legitimate contributions to any given field. With regards to JavaScript, it's really just how you write and arrange your code.

Functional Programming

Functional programming in JavaScript In this context, it is creating pure functions that focus solely on the outcome and not the process. Pure functions are immutable, which means you can't change their state. They make use of conditional expressions and recursion when performing calculations. They do not rely on anything beyond their parameters, so they are predictable. If you have the same parameters, the outcome will always be the same. Recursion is a pattern that programmers use when a line of code can be split into several similar tasks but simpler.

Types of Functions

First class functions are treated as sole entities. They are independent. We can use them the same way we do data. Functional programming uses arguments, return values and variables as functions to create more intricate code. First class functions are relatively pliable so some Object Oriented languages incorporate them too.

Higher-Order Functions

Higher-order functions work on other functions and take them as arguments. They also return other functions as output.

Functional programming does not support loop statements or conditional statements like *if*.

What Are Conditional Statements?

Conditional statements look at the conditions in the JavaScript code and then make decisions based on that. Like the name suggests, a statement can either be true or false. The code only gets executed if the condition is true. There are four types of conditional statements:

- The if statement

  Carries out part of the code if the condition is *true*.

- The else statement

  Carries out the part of the code related to else if the condition is *false*.

- The else if statement

  Designates and tests a new condition if the first *if* condition is false.

- The switch-case statement

  Designates a number of substitute code blocks to be carried out based on specific conditions.

Object-Oriented Programming

JavaScript itself is not based in Object oriented language, but we can still use it when programming. While functional programming is centered around functions and logic, OOP or Object-oriented programming is not. It is a model that focuses more on objects. It is the process of organizing your data in classes and then in objects based on their class. In this case, an object is a data field with its own features and properties. OOP is good for bigger programs that require constant updating and maintenance. But that slows down their speed and results in them using a lot of memory.

Before anything, you want to start off with an understanding of data-modeling. Data modeling is when you take all the objects that you want to work with and compare them. If your object was cars, you would want to compare them by make, color and size.

The properties that make up Object-Oriented Programming are:

- **Classes**–Are the models used to create objects

- **Objects**–Are units that possess characteristics and can be organized by type
- **Methods**–Make JavaScript able to manipulate or change the data inside it
- **Attributes**–Are words we put in the start and end tags of html elements to control what the tag does

Objects

Objects are made up of characteristics. If one of the characteristics has a function, it's considered a method. If it has no function then it's just a property. A property can be any of the primitive data types, but they are usually variables located in objects.

The principles of OOP are:

- **Encapsulation–**Is a method used to hold vital information inside an object and prevent additional information from being exposed. It gives the other objects controlled access and aids in preventing any data corruption.
- **Abstraction–**Only shares pertinent attributes between objects and hides any unnecessary code
- **Inheritance–**Is the operation by which one class can obtain properties (like fields) from another
- **Polymorphism–**Is the process of representing one thing in different forms. In JavaScript it means that different objects can share the same attributes.

OOP is easier to manage and it can be likened to the real world so it's easier to grasp.

Procedural Programming

Procedural programming has a direct approach to organizing code. It aims to break a program down into procedures. In this case, procedures are functions that are conducted chronologically. It tells the computer to perform functions in sequence until the desired outcome is reached.

Fundamental Features of Procedural Programming

- **Predefined Functions**

  A Predefined function is a command identified by a name. They are incorporated in complex programming languages but they are stored in the JavaScript library and not the program itself.

- **Local Variable**

  A local Variable is stated in the main structure of a method and cannot exist outside of a certain field. Local variables can only be used in the method they are written in. If a programmer attempted to write them outside of the method, the code would fail.

- **Global Variable**

  Global variables are written outside of methods and can be used in all functions.

- **Modularity**

  Modularity is when two different systems are tasked with different actions but get grouped together to perform a singular task first. Each group would perform their tasks in sequence until completion.

- **Parameter Passing**

  Parameter passing is an operation used to pass guidelines to functions, subroutines or procedures. They can be executed through passing by value, reference, result and name.

Applications of JavaScript Programming

JavaScript can be found in nearly every aspect of software development. Some aspects include:

- **HTML Manipulation**

  JavaScript allows you to add, remove, and change the appearance of any element of HTML to suit the requirements of any given device.

- **User Notifications**

It is critical to double-check user inputs before sending it to the server, so JavaScript verifies those inputs on the front end.

- **Back-End Data Loading**

  Ajax library allows back-end data to load while other processing is happening. This ensures that there are no interruptions for the client.

- **Presentations**

  JavaScript lets you construct presentations to enhance a website's appearance.

- **Server Applications**

  The Server Application is pretty much the same as the client application because they both deliver and receive data. The only difference between them is that they connect to the server differently.

- **Client-Side Validation**

  Contributes to the overall user experience. It is an initial check that is run to identify any invalid data on the client-side so that the user can fix it.

A recurring term that we should expand on is the 'client'.

Client-Side and Server-Side are expressions used to define where the application code is executed. In a 'serverless' framework, the vendor distributes resources to all processes that happen on the side of the server. Cloud service providers still make use of servers to run code for developers, so the name can be misleading.

The **Client-Server Model** is a framework located on the web. It divides computers into two sections. The first section requests services (the client) and the second section serves the clients (the server). A server is contained within a high performance computer, but it is not a computer itself. Servers are programs that aid in client functionality. The client-server model is a constant cycle of requests and responses. It is used because it has a larger capacity and is therefore more reliable than a user device.

Now that you have an idea of what programming actually entails, you are ready to learn about the skeleton of an application and what goes into constructing it.

Chapter 3: JavaScript Architecture

Understanding JavaScript Architecture can be helpful when you are learning how to code. If you have a good understanding about how JavaScript works online, you are at an advantage.

There are three types of JavaScript architecture. Typical, Framework based and Advanced.

Typical JavaScript Application Architecture

Ordinarily, JavaScript applications use the bottom-up approach. The User Interface is put at the center of development at all times. The User Interface and the server link directly to the code from behind.

This JavaScript architecture is mostly used for simple programs like websites that don’t require intricate frameworks. It simply can’t keep up with more complex programs. It does allow you to interact with the screen, but it just does not have the capacity for a large scale application.

SPAs or Single page applications make use of typical JavaScript Architecture. They can be updated easily and they are still in demand today. SPAs can only be used in a web browser, so they rely a lot on JavaScript. SPAs tend to be faster than mobile apps because the script that was used to build them only needs to be loaded once. From that point on its only job is to send data between the user and the server.

Framework-Based Architecture

As time progressed, The capabilities of Typical application architecture were not able to keep up with the needs of programmers around the world. Thus Framework based Architecture was born. It is able to solve more complex problems because it makes use of either the MVC or the MVVM patterns.

MVC stands for *Model View Controller*. It splits larger applications into different sections that all have distinct functions. Each section is equipped to handle different aspects of an application. The **model** represents everything data related. It gets its commands from the controller and then carries out the user's input. The **view** controls everything User Interface related**.** It gets the input from the user and then presents the data from the model to the user. The controller is the connection between the two. It processes and manipulates data from either side and then produces an output.

There are different adaptations of MVC:

HMVC–Which stands for Hierarchical Model view controller. It is essentially the same thing as MVC, but with layers and more capabilities It takes MVC and repeats it in different parts of the application. Having it in layers does wonders for app development. You can easily reuse the code and not have to worry about constant maintenance.

MVA–Which stands for Model View Adapter. Usually data travels from the controller to the model, but with MVA the controller is the go-between and the Model and View must pass it. View and Model do not actually intersect. Here, the Controller acts as an adapter.

MVP–Which stands for Model View Presenter. This one is slightly different from MVC. The presenter is in charge of all the data in the application. It sends that data to the model to process, receives it back and then passes it onto the View.

MVVM–Stands for Model-View ViewModel and is like an amalgamation of MVC and MVP because it borrows from their patterns. Data and information are completely separate from the User Interface. The controller is replaced by the Viewmodel which connects the Model and the View. The Viewmodel updates the Model every time it receives input and then adapts that data before sending it off to the View.

The framework has a feature rich Runtime environment that renders and puts out the Application. We can liken the Runtime environment to a vessel. It consists of all the things we need to run JavaScript code. Namely:

- The Engine
- Web API's

- Event loops
- Call back queue
- Micro task queue

JavaScript Engines

Engines take the intricate code written in JavaScript and convert it into a code that the machine can understand. Each browser has its own engine. A few notable engines are V8, SpiderMonkey and JavaScript Core.

V8

No two engines are the same, they all run differently but ultimately do the same thing. In the V8 engine, everything starts in the **Parser**. The Parser checks for syntax and semantics. Syntax is the structure that the code comes in and Semantics refers to what certain words and symbols in the code mean. The parser then analyzes and breaks the code down to make it into **Abstract Syntax Tree(AST)**. AST is the representation of the source code in tree form. The AST then goes into the **Interpreter**. The Interpreter converts the AST into Byte Code. This is known as **ignition**. Sometimes code gets repeated up to one thousand times. When this happens, the **Profiler** checks for the code repetition and tries to optimize it. Once it gets the optimized code, it moves it on to the **Compiler**. The compiler in the V8 engine is called **TurboFan.** Turbofan releases the Byte Code in its most optimized form.

SpiderMonkey

SpiderMonkey was actually the first engine and it was created in 1995. The Mozilla foundation has kept it going and still makes use of it today. The engine converts the JavaScript code into ByteCode first and then sends it to the **Interpreter** and the **JIT Compiler**. JIT stands for Just in Time. The JIT Compiler optimizes the code and sends any unused code to the **Garbage collector**.

Web API's

API stands for Application Programming Interface. It is an interface that gives the User a set of functions, while concealing the underlying mechanism of that function. Web API's are not actually part of JavaScript. They are located in the browser, but they do give us access to the JavaScript Engine.

Web API's are comprised of:

- A set timeout–Allows you to execute a block of code after a certain amount of time (defined by you) has passed
- DOM API's - Are made up of the interfaces that define how each of the elements in HTML function. That includes any supporting types and interfaces they may rely on.
- Fetch–API Gives JavaScript access to manipulate requests and responses in the Call Stack
- Local storage–Lets JavaScript apps save important value pairs indefinitely
- Console–Registers any and all messages logged by JavaScript code
- Location–Allows users of the browser to share their Geographical location at any given time

Event Loops

JavaScript is a single threaded code. This means that it can only run one command at a time. These commands are synchronous because they run one after the other. This doesn't always work in programming so there are ways to make JavaScript asynchronous. Things like Call stacks help to avoid any blocking that might happen in a synchronous thread.

A Call stack is a mechanism used by the interpreter to keep track of functions and their place in any given script. It allows functions to run concurrently. When the script calls a function, the interpreter puts it in the call stack and then conducts the function.

A **Callback Queue** is like a safe house where an unexecuted line of script stays until it's time for it to run.

The Event Loop's job is to check the Call Stack and Callback Queue. If the Call Stack is empty, Event Loop will take the statement in the Callback Queue and push it to the Call Stack. It appears to be a pretty straightforward process, but there is something called the **MicroTask Queue.** The MicroTask queue is pretty much a carbon copy of the Callback Queue but it has a higher priority, which means that any functions in it will be executed first.

MPAs or Multi Page applications are better built with Framework based architecture. They are obviously larger than Single Page Applications and take more time to build. Because it's data heavy, a lot of information has to go back and forth between the browser and the server. This makes it slower than an SPA. Some developers use AJAX to lighten the load, but it's still not as widely used because it's simply too much to maintain.

Advanced JavaScript Architecture

Even though Framework Architecture has more capabilities than typical architecture, it still has its limits when it comes to constructing larger applications. The only way it could grow to accommodate larger apps would be through constant maintenance and development. MVC and MVVM patterns are convenient and allow for developers to reuse code, but they present problems in the framework. The User Interface became compromised because the controller was corresponding with the server while manipulating the view. The need for a more feature rich architecture arose, then Advanced JavaScript Architecture came to be. In this architecture the User interface and server correspondence work completely separately, so it is easy to keep up with any demands that larger apps might have.

Data computation and User Interface are completely separate in this instance. The User Interface is no longer at the center, so the app becomes easier to manage and use.

Universal Applications

Formally known as Isomorphic applications, they provide faster browser connection because there is less code. They work on both the client and

server side, ensuring much easier engagement and less work for the programmer. The one downside is that they are quite hard to debug.

Chapter 4: JavaScript and JQuery

Let's go through some of JavaScript's most important features to give you a better idea of how it works. The first feature is Variables. Variables are like storage containers for values. Data can be placed in these containers and then you can refer to it by naming the container. Every worthwhile programming task needs a variable. If values were unable to change, your webpage would be less interactive. You must declare your variable before you can use it in a JavaScript program. The keyword for variable (as seen in Chapter One) is **var**.

```
<script type= "text/javascript">
        <! --
        var age;
        var race;
        // -- >
</script>
```

You can also declare different variables with the same keyword. It would look like this:

```
<script type= "text/javascript">
        <! --
        var age, race;
        // -- >
</script>
```

(*JavaScript - Syntax*, n.d.)

Variable initialization happens when we store a value in a variable. You can add values at any point in time. We have created a variable called **race**, so we can add the value 'black' to it at a later point and we can add the value for age at the time of initialization. It would look like this:

```
<script type= "text/javascript">
        <! --
        var age = "39"
            var race;
            race= black;
        // -- >
</script>
```

(*JavaScript - Syntax*, n.d.)

The *var* keyword only needs to be used once, at the time of initialization.

A variable's scope is the area of your program where it is defined. In JavaScript, there are only two:

- **Global Variables**–Can be defined anywhere in your JavaScript code.
- **Local Variables**–Is only visible within the scope of the function in which it is defined.

Local variables take precedence over global variables with the same name within the body of a function. When a local variable or a function parameter has the same name as a global variable, the global variable is effectively hidden. For example:

```
<html>
        <body onload = checkscope();>
        <script type = “text/javascript”>
         <! --
        var myVar = “global”; // Global variable
        function checkscope( ) {
        var myVar = “local”; // Local variable
        document.write(myVar);
        }
        // —->
              </script>
      <body>
<html>
```

(*JavaScript - Syntax*, n.d.)

There are rules to naming your variables. The names of your variables are case sensitive, so **race** and **Race** are two different variables. Variable names cannot start with a number. They either have to start with a letter or an underscore. If your variable was named 345age, it would be invalid. But if it was named _345age it would be valid.

There is a list of JavaScript reserved keywords that cannot be used as a variable name. The list is as follows:

abstract	boolean	break	byte	case	catch
char	class	const	continue	debugger	default
delete	do	double	else	enum	export
extends	false	final	finally	float	for
function	goto	if	implements	import	in
instanceof	int	interface	long	native	new
null	package	private	protected	public	return
short	static	super	switch	synchronized	this
throw	throws	transient	true	try	typeof
var	void	volatile	while	with	

(*JavaScript - Syntax*, n.d.)

The lion’s share of applications work because of an interaction between a client and a remote server. A client refers to the end user device. The client requests data from the server, the server receives and processes the request and then the client gets the data back in a readable format. For the most part, we do need the client server model, but with JavaScript we can avoid it completely. JavaScript helps to reduce the traffic by validating forms without input from the server.

The JavaScript Engine is a program that translates source code into a language the computer understands. All browsers today come with a JavaScript engine and each engine has two components: A MemoryHeap and a Call Stack. A Memory Heap is where memory is allocated at any given time. A Call Stack processes information added to it from the script.

JavaScript Frameworks

Frameworks for JavaScript enable the language to perform at its best with minimal setup. They give developers the building blocks they need to create JavaScript applications. The building blocks are a set of code libraries. The libraries generate code that evokes specialized functionality for the sort of app being developed. The framework establishes the structure of the whole application. There are several JavaScript frameworks out there, but the most popular ones are :

Node.js	Node.js is a cross platform, open-source, back-end JavaScript runtime environment that makes use of the V8 engine to execute JavaScript code outside of a web browser.
Vue.js	Vue.js is an open-source JavaScript framework that creates user interfaces and single page apps. It uses a model-view viewmodel (MVVM) architectural pattern.
AngularJS	AngularJS was an open-source, front-end framework for constructing single-page apps based on JavaScript.
Ember.js	Ember.js is an open-source, JavaScript client-side framework for constructing web apps. It enables the development of client-side applications by offering a comprehensive solution that includes data management and application flow.
React	React is a front-end JavaScript toolkit for creating user interfaces using UI components. React can be used as a foundation to create single-page or mobile apps.

(*JavaScript - Syntax*, n.d.)

Features of JQuery

HTML Manipulation

Elements in an HTML document can be manipulated because JQuery allows the user access to the DOM.

DOM Manipulation

JQuery gives you access to the DOM's API so that you can make any necessary changes to your HTML document. You can add and subtract and rearrange elements as you please. By doing this, your web page can display updated information without you having to refresh the page a number of times.

CSS Manipulation

The JQuery library has lines of code that can be used to control and change elements of CSS and DOM.

Effects and Animation

JQuery has an animate function that makes it easier to add different effects and animations to your web page. It also lets you animate your HTML elements by manipulating the CSS. Examples of these effects include: show, hide, fade in, fade out and slide.

Ajax

Ajax stands for Asynchronous JavaScript and XML. It lets you put content from a server or a database onto your website without having to refresh the site every time. If you are visiting a website that is constantly receiving data, it would be really inconvenient to have to refresh the page each time.

Utilities

Utilities are software programs that optimize a computer's performance. Some utility programs are good for keeping your computer virus free and others just make your desktop look good.

JQuery can be considered legacy code because it was introduced in 2006. It has stood the test of time and can be used in most applications that exist today. JQuery is used on most, if not all wordpress sites. There are a whopping 455 million of them as of 2022. It teaches you a lot about JavaScript and is the easiest coding library to use.

Chapter 5: JavaScript for Kids

It may be hard to believe, but there was a time when there was no internet! And before that, no computers either! It's probably really hard to imagine a world without the two, but people used to exist and get by with fewer electronics.

Before the invention of cellphones, people used to communicate via telephone or by writing letters and sending them through the post. Before online shopping, people had to physically go to the shops and kids had to have their friends in the room in order to play TV or computer games together.

With time, all those things changed. Things became faster and easier to do, and we have computers to thank for that. A computer is a gadget that takes different kinds of information and processes it, then gives us an answer. It works the same way as a calculator. You tell the calculator the sum you want it to work out and it gives you an answer.

Computers work with Operating systems. An operating system is a form of software that basically runs the computer. It controls the hardware by taking information in and giving information out. It is also responsible for the connection between us, the users, and the computer.

These days, most people use computers to get on to the internet. The internet is a network that connects millions of computers around the world. There are so many websites that we can't even count. The thing that makes these websites look cool and do amazing things is JavaScript.

JavaScript is a language that we use to tell the computer what to do and how to do it. When you type something on your keyboard or scroll around a page with your mouse, the web page you are using responds to that. That's because of JavaScript.

As amazing as JavaScript is, you can't use it on its own to build your website. Websites are built using three languages. HTML, CSS and JavaScript.

HTML lets you put content on your webpage. So all the things you can see, like text and pictures.

CSS takes the content on your page and changes how it looks. So with CSS you can make your text and pictures big or small. You can change colors that appear on your page too !

And lastly, JavaScript tells the content on your page how to behave. That means that it can make your text or pictures move around, and it can even add animation.

HTML and CSS

HTML is really easy to learn. It’s where some of the most successful programmers and software developers started, and who knows? You could be the next Mark Zuckerberg!

HTML stands for Hypertext Markup Language. Hypertext describes the words or pictures on your web page that are hyperlinks. A hyperlink is something you can click on that takes you either to a different page or to a different part of the page you are already on. Markup Language is a collection of letters called tags that tell our web page how it should look.

HTML can be written in the notepad app on your computer.

In order to create a tag, we put them inside brackets that look like this <>. Some of the most common tags are <h> for heading, <p> for paragraph and <img> for images.

Tags can’t be alone, they always need something in between them. The thing we put inside the tag is called *content.* The content is what actually shows up on your web page. We can think of the tags like ‘the boss’ because the content will always do what the tag says. Here’s how:

At the top of your web page, you’re going to want to have a heading. Your main heading is going to be called <h1> so that the computer knows that it goes at the very top of your web page. Your heading can be whatever you want it to be, but for today it’s going to be ‘I am smart’. Once you put it inside the tags it’s going to look like this: <h1>I am smart</h1>. The two tags show where your heading should start and stop.

CSS stands for Cascading Style Sheets. Cascading means things that happen in a sequence, or one after the other. If we think of it like a

waterfall, styles can fall from one sheet to another. This makes it possible for lots of different styles to appear on the same HTML sheet and then on a webpage. Style sheets are files that have font and styling settings.

Important JavaScript Terms

We can write JavaScript from scratch, but we don't have to. There are coding libraries that exist to make it easier to write script. A coding library is a collection of code that has already been written. Most coding libraries are open source, which means they are free to download and use. An example of a coding library is JQuery.

In JQuery's collection of code you will find:

Variables–We can liken a variable to a case filled with something. Variables hold one word or number at a time and can be sorted into one of three parts; type, name and value.

Keywords–Keywords are words with special meanings that have been set aside by programmers.

Functions–Are words within the script that we can use to get a task done

Parameters–Sometimes functions need to have a bit of extra information in them so they can perform properly. That extra information is called a parameter. Sometimes they are called *arguments* too.

Arrays–Arrays are like a bigger box for variables. They take variables that are alike and group them together.

Data Types in JavaScript

A language is something people use to communicate. All Languages have words that are unique to them. The same goes for programming languages. They each compose their data a certain way. JavaScript has Primitive values and Objects.

Primitive Values are forms of data that cannot change. In JavaScript there are seven:

- Number type

The number type is used to represent all numbers. It includes integers and floating point values. Floating point values are numbers that have fractions.

- BigInt type

  BigInt is short for Big Integer. An integer is a whole number without fractions. 5 is an integer but 5.5 is not. This data type was created for JavaScript in early 2020 because there was a problem with the number data type. It didn’t have a number big enough for programmers to use, so in some situations they had problems when coding.

- Null type

  Null is the value we use to show that there is no object. In real life, Imagine if you really wanted a glass of milk. You walk over to the fridge, grab a carton of milk and start to pour into your glass, but nothing comes out. Your carton of milk has a null value because it’s empty.

- Undefined type

  We can compare Undefined to something that is unclear. A variable that can’t be identified by its characteristics becomes undefined in JavaScript. A lot of people are unsure what the difference between Null and Undefined is.

- Symbol type

  A symbol is the newest primitive data type, it got added to JavaScript recently. Symbols have a unique value and cannot be recreated. We use them when we want to assign a specific value to an object so that it doesn’t clash with another object.

- String type

  A string is a chain of symbols. It could include anything from letters to numbers.

- Boolean type

  A Boolean is a value that can represent a true or a false statement. We usually use it with conditional statements when coding.

Objects are values that have their own characteristics and can be organized by type. We assign characteristics to an object by using **variables**.

Variables

The word variable means something that doesn't stay the same, something that can change at any time. In JavaScript, Variables hold on to values that can be changed at any time. Because of this, variables cannot have the same name. Variables have declarations and definitions. A Declaration is the part of the code where the variable is before any action takes place. The definition is where the variable is placed once it has a value and a place in the code.

Arrays

In JavaScript, an Array is one variable that we can use to keep different elements. Keeping several items of the same kind together makes it easier to work out where each element is by adding an offset to a base value. A real life example would be if a teacher had all her students sitting at their desks. The teacher would be able to pinpoint any student by knowing what number desk they're sitting at. It makes coding easier, because we can have access to a whole list of elements just by typing in one word.

Objects

Objects are words in your script that have characteristics and can be organized by type. Those characteristics are usually variables determined by the programmer.

Conditionals and Loops

Conditional statements work out what is happening in your JavaScript code and then use boolean phrases to decide on the outcome. So the outcome on a conditional statement will either be true or false. There are four types of conditional statements:

- The if statement

Runs a certain part of the code if the condition is *true*

- The else statement

  Runs the part of the code related to *else* if the condition is *false*

- The else if statement

  Gives and tests a new condition if the first *if* condition is *false*

- The switch-case statement

  Gives a number of substitute code blocks to be carried out based on specific conditions

Loop statements help to keep your code short and sweet. Sometimes, you are going to have to perform the same action a number of times on your web application. If loop statements didn't exist, a programmer would have to write the same line of code over and over again. There are three types of Loop statements:

- The while loop

  The while loop is straightforward. The while loop carries out a conditional statement over and over again if the statement is true. As soon as the statement becomes false, the loop stops running.

- The for loop

  The for loop is a bit more complex. It only runs a specific number of times and it's made up of three parts:

  - The Loop Initialization

    This gets your code ready to receive a starting value. This happens before the loop starts.

  - The Test Statement

    Tests if a condition is true. If it is true, then the loop will run. If it's not true, then the loop will not run.

  - The Iteration Statement

    Lets you increase or decrease the number of times your loop runs

- The for..in loop

As we read before, objects have properties. The for..in loop's purpose is to loop through an object's characteristics.

Functions

Functions are used to wrap a specific piece of your code so you can reuse it without having to type it out more than once. Sometimes a developer needs the same action to take place on different parts or pages of the same website. Functions can be broken down into three parts:

Function Statement

The function statement reveals a function. The function gets saved and only works when it is called for.

Function Expression

A function expression does not give the name of the function. The name can be left out of the line of code and create an *anonymous function.*

Function Declaration

Names a function with its special parameters

Three Ways to Write Your First Line of Code

The first way you can practice writing some code is to write it directly into the Browser. It's really easy because you don't need to set anything up and you can start off by trying out some simple comments.

- First, open your web browser and find the search box.
- Type in *about:blank* and click enter–Your browser tab should be empty.
- Click on the menu and choose *tools* or *more tools* depending on your browser.
- Choose *Developer tools*–A developer tag should appear.
- Click on *Console.*
- Type your first line of code!

Once you have gotten used to writing and using some basic HTML tags and content, you are going to need to be able to save your code and you might want to use lots of commands at the same time. For that you have to write your code in a simple text file and then send it to the browser so you can view it. Browsers accept files with the extension .html. You can create HTML files with Notepad in windows and with Textedit on Mac. Once you have written your code into the file, you need to change the format to HTML.

On Windows:

- Open Notepad
- Define the document type by typing in <!DOCTYPE html>
- Type in some tags and content e.g.:

  <h1>I AM SMART</h1>

   I can code!
- Click on file in the top left corner
- Click on Save as
- Look for: Save as type
- Click on the drop down menu and choose the .html file extension.
- Save your document

On Mac:

- Open Textedit
- Go to the very top of the page and click on preferences
- Choose the plain text option
- Define the document type
- Type in some tags and content
- Click save and type .html at the end of your document name
- Save your document

You have to be very careful, because you will be writing code all by yourself.

Another way you can write your code is with Special Code Editors. Special Code Editors know lots of programming languages and they have special features to help you write your code. They highlight different keywords and let you know when there is a syntax error. An editor that is available for free online is *Visual Studio Code*.

Now that you have a basic knowledge of HTML and JavaScript, challenge yourself and do even more research. By the end of your coding journey you should be able to create beautiful websites, build games and construct really cool mobile apps.

Conclusion

When you're learning something new it's not advisable to jump in at the deep end. What you have here is a ten thousand foot view of JavaScript.Having a theoretical knowledge of it will always be beneficial to you and it's something nobody can take away from you. Both theoretical and practical knowledge are important when learning how to code, but having purely practical knowledge can be limiting. You need to have an understanding of the concepts behind the code you've set out to learn. Problem solving while coding will come easily to you, because you will have an in-depth knowledge of how everything works from the bottom up.

The software development field is constantly growing and changing, so knowing the fundamentals of JavaScript means that you will always be able to keep up with any updates.

Whenever we say *JavaScript is everywhere*, we're not just saying that, it is literally everywhere! It's not just something we use on web browsers. It can also be used to program electronic toys, wifi connected coffee machines and mobile apps.

According to the 2021 StackOverflow developer survey, JavaScript is used by 94.5% of all websites.

As far as career opportunities go, the possibilities are endless. The tech industry is always hiring and is one of the only career fields that is employee centric. They understand that employee satisfaction leads to productivity. You are constantly learning and also very likely to move up the ladder fairly quickly because there's always room to grow.

You can become a front-end, back-end or full stack developer.

A **front-end developer** uses HTML, CSS and JavaScript to build websites from scratch. It's an exciting job because every website is different, so the chances of doing the same thing everyday are slim. Front end developers need to ensure they are up to date with all the changes happening in the field, so if there is a downside it's that they need to constantly do research.

A **back-end developer** works on the server-side. Back end development is the management of the server and the database of any website. They create

the environment that makes it possible for front-end development to exist.

A **full stack developer** works on both the client-side and the server side. It's like being in the middle. They are responsible for everything that happens when the front-end and back-end meet. Full stack developers can design software and then debug it as well. Full stack developers can work on small programming projects alone. Building a website for an individual can be done by one person.

There is a big demand in the Job market for JavaScript developers because not all people who are employed in the tech industry have an in-depth knowledge of it. The average basic salary of a JavaScript developer in the United States is approximately $111,823 per annum.

There are a lot of self-taught JavaScript developers working today. With time, discipline and the will to learn, you could be one too!

Your FREE Gift

You can grab Your copy by clicking on the following link and joining my mailing list.

FREE BOOKS https://imris-kdp-publishing.com/

References

A coding Nomad. (2020, November 18). *Event loop explained in 3 minutes.* [Video]. YouTube. https://www.youtube.com/watch?v=5YcMKYTZZvk

akshitsaxenaa09. (2021, September 20). *Introduction to Javascript engines.* Geeks for Geeks. https://www.geeksforgeeks.org/introduction-to-javascript-engines/

Aryan, A. (n.d.). *Functional programming in JavaScript*. Toptal Engineering Blog. https://www.toptal.com/javascript/functional-programming-javascript#:~:text=Is%20JavaScript%20a%20functional%20programming

Atnyla. (n.d.). *What are the applications of Javascript programming?* Atnyla. https://www.atnyla.com/qanswer/what-are-the-applications-of-javascript-programming-/10/772

Automation Step by step. (2020, November 26). *What is DOM | Document Object Model beginner tutorial.* [Video]. YouTube. https://www.youtube.com/watch?v=ipkjfvl40s0

Bhatia, S. (2021, January 8). *What is procedural programming? Key features of procedural programming*. Hackr. https://hackr.io/blog/procedural-programming

Brewster, C. (n.d.). *JavaScript frameworks: What are they and how do they work?* Trio. https://trio.dev/blog/javascript-framework

Code With Ania Kubow. (2020, May 14). *setTimeout explained in 5 minutes!* [Video]. YouTube. https://www.youtube.com/watch?v=z9lJb4D3nJY

Cogan, B. (2010, May 18). *HMVC: An introduction and application.* Code Envato Tuts+. https://code.tutsplus.com/tutorials/hmvc-an-introduction-and-application--net-11850

DataFlair. (2019, February 27). *JavaScript architecture - Architecture for different applications*. DataFlair. https://data-

flair.training/blogs/javascript-architecture/

Dorian Develops. (2020, September 18). *Don't waste your time learning jQuery (2021 and beyond).* [Video]. YouTube. https://www.youtube.com/watch?v=ZEEiHzMejRw

Elliott, E. (2019, January 10). *Higher order functions (Composing software).* Medium - JavaScript Scene. https://medium.com/javascript-scene/higher-order-functions-composing-software-5365cf2cbe99#:~:text=A%20higher%20order%20function%20is

Envisage Digital. (2020). *60+ WordPress market Share & CMS usage statistics* (2021). Envisage Digital. https://www.envisagedigital.co.uk/wordpress-market-share/

freeCodeCamp. (2017, March 13). *For loops - Beau teaches JavaScript.* [Video]. YouTube. https://www.youtube.com/watch?v=24Wpg6njlYI

FrontEnd NE. (2016, July 11). *Large scale JavaScript application architecture - Andrew Humphreys | February 2016.* [Video]. YouTube. https://www.youtube.com/watch?v=1n4qWEEa8mI

Geeks for Geeks. (2017, October 27). *Introduction to arrays* - GeeksforGeeks. Geeks for Geeks. https://www.geeksforgeeks.org/introduction-to-arrays/

Gillis, A. S., & Lewis, S. (2019). *What is object-oriented programming (OOP)? - Definition from WhatIs.com.* Search App Architecture. https://searchapparchitecture.techtarget.com/definition/object-oriented-programming-OOP

Hack Reactor. (2018, October 18). *What is JavaScript used for?* Hack Reactor. https://www.hackreactor.com/blog/what-is-javascript-used-for

Hague, K. (n.d.). *5 great reasons to consider a job in tech.* The Muse. https://www.themuse.com/advice/5-great-reasons-to-consider-a-job-in-tech

Hartman, J. (2020, January 11). *JavaScript variable: Declare, assign a value with example.* Guru 99. https://www.guru99.com/using-

variables-in-javascript.html

Javascript. (2021, December 12). *Recursion and stack.* Javascript. https://javascript.info/recursion

JavaScript Tutorial. (n.d.). *JavaScript functions*. JavaScript Tutorial. https://www.javascripttutorial.net/javascript-function/

Java T point. (n.d.). *What is jQuery.* Java T point. https://www.javatpoint.com/what-is-jquery

Junior Developer Central. (2020, January 23). *Should you learn jQuery in 2020.* [Video]. YouTube.. https://www.youtube.com/watch?v=bGHG6XKs6tE

Karate & Padawan Vlogs. (2015, April 13). *jQuery basics - What is jQuery*? [Video]. YouTube. https://www.youtube.com/watch?v=U4UZVbQ8md0

Krossing, D. (2017, April 18). 1: *How to get started with AJAX | AJAX tutorial for beginners | Learn AJAX | PHP | JavaScript*. [Video]. YouTube. https://www.youtube.com/watch?v=tNKD0kfel6o

Krossing, D. (2017, April 27). 7: *How to animate elements using jQuery - Learn jQuery front-end programming.* [Video]. YouTube. https://www.youtube.com/watch?v=fW7rwCpD0FY

Lagutin, V. (2021, September 15). *How to learn JavaScript a little faster.* Free Code Camp. https://www.freecodecamp.org/news/how-to-learn-javascript-a-little-faster/

Martin, M. (2021, December 18). *What is functional programming? Tutorial with example.* Guru 99. https://www.guru99.com/functional-programming-tutorial.html#2

MDN Contributors. (2021, July 15). *Browser Console*. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Tools/Browser_Console

MDN Contributors. (2021, July 20). *undefined.* MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/undefined

MDN Contributors. (2021, October 7). *Call stack - MDN web docs glossary: Definitions of web-related terms* | MDN. Developer Mozilla.

https://developer.mozilla.org/en-US/docs/Glossary/Call_stack

MDN Contributors. (2021, October 13). *Using fetch*. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

MDN Contributors. (2021, November 10). *Function expression - JavaScript | MDN*. Developer Mozilla. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/function

MDN Contributors. (2022, January 21). *JavaScript basics*. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics

Mohammed, A. (2020, May 11). *Does theoretical knowledge matter?* Linked In. https://www.linkedin.com/pulse/does-theoretical-knowledge-matter-mohammed-zubair-ahmed#:~:text=Importance%20of%20theoretical%20knowledge%3A

MuleSoft. (n.d.). *What is an API? (Application programming interface)*. MuleSoft. https://www.mulesoft.com/resources/api/what-is-an-api#:~:text=API%20is%20the%20acronym%20for

Neoteric. (2016, December 2). *Single-page application vs. multiple-page application*. Medium. https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58

Norton, D. (2020, March 20). *How to save a text document as an html file (using TextEdit on Mac)*. [Video]. YouTube. https://www.youtube.com/watch?v=koIiDBOpqkE

Obaseki, N. (2020, December 29). *localStorage in JavaScript: A complete guide*. LogRocket Blog. https://blog.logrocket.com/localstorage-javascript-complete-guide/#:~:text=localStorage%20is%20a%20property%20that

Parish, A. (2021). *JavaScript in the browser console (Creative Coding)*. Creative-Coding.decontextualize.com. https://creative-coding.decontextualize.com/browser-console/

Pavlutin, D. (2017, April 15). *7 tips to handle undefined in JavaScript.* Dmitri Pavlutin Blog. https://dmitripavlutin.com/7-tips-to-handle-undefined-in-javascript/#1-what-is-undefined

Pavluti, D. (2020, September 22). *Everything about null in JavaScript.* Dmitri Pavlutin Blog. https://dmitripavlutin.com/javascript-null/#1-the-concept-of-null

Programiz. (n.d.). *Getting started with JavaScript*. Programiz. https://www.programiz.com/javascript/get-started

Programming with Mosh. (2018, March 29). *Object-oriented programming in 7 minutes | Mosh.* [Video]. YouTube. https://www.youtube.com/watch?v=pTB0EiLXUC8

Rahmat, A. (2020, January 30). *JavaScript null vs undefined.* Medium. https://levelup.gitconnected.com/javascript-null-vs-undefined-2acda986f79f

Ryan. (2020, December 7). *JavaScript for kids | A beginner's guide to getting started in programming.* ID Tech. https://www.idtech.com/blog/javascript-for-kids

Sharma, R. (2021, March). *How do you use the 80/20 Pareto principle in your life?* Quora. https://www.quora.com/How-do-you-use-the-80-20-Pareto-principle-in-your-life

Smartcode. (2021, October 5). *Number data type in JavaScript | Part 12 | JavaScript | #SmartCode.* [Video]. YouTube. https://www.youtube.com/watch?v=sHoX1oLaO1o

Steele, C. (2019, January 11). *The complete guide to JS symbols ES6.* [Video]. YouTube. https://www.youtube.com/watch?v=4J5hnOCj69w

Team, D. (2019, February 25). *Pros and cons of JavaScript - Weigh them and choose wisely!* DataFlair. https://data-flair.training/blogs/advantages-disadvantages-javascript/

Tech World with Nana. (2020, May 1). *Where to write JavaScript | Where to execute JavaScript code JavaScript tutorial #3.* [Video]. YouTube. https://www.youtube.com/watch?v=5N7Tq9e1U_0

TheTechCave. (2016, May 20). *The client server model | Clients and servers.* [Video]. YouTube. https://www.youtube.com/watch?v=L5BlpPU_muY

The Net Ninja. (2015, August 15). *JavaScript tutorial for beginners #11 - Booleans in JavaScript.* [Video]. YouTube. https://www.youtube.com/watch?v=B4ZCFdrBmbE

Tutorials Point. (n.d.). *JavaScript - if...else statement*. Tutorials Point. https://www.tutorialspoint.com/javascript/javascript_ifelse.htm

Tutorials Point. (n.d.). *JavaScript - Syntax.* Tutorials Point. https://www.tutorialspoint.com/javascript/javascript_syntax.htm

Tutorials Point. (2019). *MVC framework - Introduction - Tutorialspoint.* Tutorials Point. https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

Tutorial Republic. (n.d.). *How to create an HTML page - Tutorial Republic*. Tutorial Republic. https://www.tutorialrepublic.com/html-tutorial/html-get-started.php

Tutorials Teacher. (n.d.). *CSS manipulation using jQuery.* Tutorials Teacher. https://www.tutorialsteacher.com/jquery/jquery-css-manipulation

Tyson, M. (2021, April 1). *What is functional programming? A practical guide.* InfoWorld. https://www.infoworld.com/article/3613715/what-is-functional-programming-a-practical-guide.html

Udondian, N. (2021, August 9). *Hypertext markup language (HTML) syntax.* Medium. https://medium.com/@nseabasiudondian_56413/hypertext-markup-language-html-syntax-3d1e68366363

Ukpaiugochi. (2021, June 3). *How JavaScript works: Modularity and reusability with MVC.* Medium. https://blog.sessionstack.com/how-javascript-works-writing-modular-and-reusable-code-with-mvc-16c65cbd9f64

W3 Schools. (n.d.). *JavaScript function statement.* W3 Schools. https://www.w3schools.com/jsref/jsref_function.asp

Web Dev Simplified. (2018, August 22). *Learn CSS in 20 minutes.* [Video]. YouTube. https://www.youtube.com/watch?v=1PnVor36_40&t=352s

Webstack. (2020, August 28). *JavaScript tutorial #7 | Symbol data type es6 | Symbols in JavaScript es6.* [Video]. YouTube. https://www.youtube.com/watch?v=c7aJyJeBOxQ

What do client-side and server-side mean? | Client side vs. server side | Cloudflare. (n.d.). Cloudflare. https://www.cloudflare.com/learning/serverless/glossary/client-side-vs-server-side/

B O N U S



Design Your Own Crochet Projects!

Crochet Projects for Adults

Magnus D'Jango

© Copyright 2022-All rights reserved.

The content contained within this book may not be reproduced, duplicated or transmitted without direct written permission from the author or the publisher.

Under no circumstances will any blame or legal responsibility be held against the publisher, or author, for any damages, reparation, or monetary loss due to the information contained within this book, either directly or indirectly.

Legal Notice:

This book is copyright protected. It is only for personal use. You cannot amend, distribute, sell, use, quote or paraphrase any part, or the content within this book, without the consent of the author or publisher.

Disclaimer Notice:

Please note the information contained within this document is for educational and entertainment purposes only. All effort has been executed to present accurate, up to date, reliable, complete information. No warranties of any kind are declared or implied. Readers acknowledge that the author is not engaged in the rendering of legal, financial, medical or professional advice. The content within this book has been derived from various sources. Please consult a licensed professional before attempting any techniques outlined in this book.

By reading this document, the reader agrees that under no circumstances is the author responsible for any losses, direct or indirect, that are incurred as a result of the use of the information contained within this document, including, but not limited to, errors, omissions, or inaccuracies.

Table of Contents

Introduction
- Fast Crochet Projects
  - Crochet Scrunchie
- Crochet-Along Projects
- Useful Crochet Projects
  - Flower Pot Hanging Basket

Chapter 1: Beginner Crochet Projects
- Simple Crochet Projects
  - Beginner Scarf
- One Skein Crochet Projects
  - One Skein Soap Holder
- Scrap Yarn Crochet Projects
  - Scrap Yarn Rug
- Finger Crochet Projects
  - Cozy Blanket

Chapter 2: Small Crochet Projects
- Can Cozies
- Flower Bookmark

Chapter 3: Knitting Projects
- Easy Knitting Projects
  - Foldover Beanie
  - Knitted Infinity Scarf

Chapter 4: Fun Crochet Projects
- Crochet Headband
- Crochet Snowflake Ornament
- Granny Square

Chapter 5: Cute Crochet Projects
- Crochet Owl

Conclusion
References
Image References

Your FREE Gift

I have something for you. It won't cost you a dime. This is a FREE 40-page PDF guide. I'd like you to have free copies with my compliments. You can grab Your copy by clicking on the following link and joining my mailing list.

FREE BOOKS https://imris-kdp-publishing.com/

Before we jump into this Book, I'd like to express my thanks. I realize there are many books on this subject that you could otherwise spend your time reading. You're taking a chance on me. I appreciate it. Giving you a FREE PDF copy of this amazing guide is my way of showing that appreciation. On that note, let's dig in. You're going to love what's coming your way in the following pages

Introduction

Do you love to crochet, or wish you knew how but aren't sure where to start? Do the different tools, accessories and yarn types overwhelm you? Don't let that intimidate you! Full of small, simple and quick crochet projects, this book is a perfect way for beginners to come into their own as crocheters. Good luck, and happy crocheting!



Fast Crochet Projects

Sometimes we're looking to do a crochet project that's quick and easy, rather than committing lots of time and energy to a larger project, and there's nothing wrong with that! Here is a fast project that you can do in an hour or less!

Crochet Scrunchie

There is so much to love about this crochet scrunchie pattern. It can be made with countless color and texture combinations, so feel free to get creative and try it out with all your favorite yarns.

Materials:

- Any bulky 5 or medium weight yarn
- H 5.0 mm hook
- Yarn needle
- Scissors
- Plain hair tie

Step 1: Attach the yarn to the hair tie by using an SC. Continue around the hair tie until it is covered, about 12 SC. SL ST into first SC to complete.

Step 2: DC approx. 150 times around. This can depend on the size of your hair tie and the type of yarn being used. The yarn should take on a "scrunched up" look.

Step 3: SL ST into the top of the first DC of that round.

Step 4: Fasten off the yarn and weave in the loose ends using a yarn needle. Your scrunchie is complete!

Crochet-Along Projects

You may be asking yourself: What *is* a crochet-along?

Crochet-alongs, commonly referred to as CALs, are essentially a larger group of crocheters that are all working together on the same, big project at the same pace. Many CALs are offered for crocheters of all skill levels, so that even absolute beginners can participate.

CALs are a great way to connect with the crocheting community and share ideas. People host CALs on many social media platforms and websites, so if this sounds like something you'd find interesting, finding one to join is only an internet search away.



Useful Crochet Projects

Crocheting isn’t just for creating cute and fun projects; they can be useful, too! Whether you’re looking to make something new or spruce up something you already have, there is a crochet project out there that’s for you. Here’s a great one for you to try.

Flower Pot Hanging Basket

This beautiful yet simple pattern will add a little flare to your plain potted plants. Try using a color that matches the rest of your home decor!

Materials:

- Tapestry needle
- Size G crochet hook
- Worsted weight yarn (This yarn works best, but feel free to experiment with other types, too!)
- Flower pot
- Ceiling hook

Step 1: Make a magic ring, chain 2 and make 10 DC in the ring. Join the first DC.

Step 2: Chain 3 and DC in first, chain 2 then skip a stitch. [2 DC in next chain, skip a stitch, and repeat] around. Join in space between the first DC and chain 3.

Step 3: Chain 3 and DC in first space, chain 1 [DC, chain 1, DC in next space, chain 1, repeat] around. Join (20 DC).

Step 4: Chain 3 and DC in first space, chain 1 [DC in next space, chain 1, DC and chain 1 and DC in next, repeat] around. Join chain 1.

Step 5: 2 SC around in each space.

Step 6: [Chain 7, skip four stitches, SL ST to join next stitch, repeat] around, join with bottom of first chain 7. There should be 12 chain spaces.

Step 7: SL ST up to middle of first chain space, chain 7 and join with SL ST to middle of next chain space. Do not join.

Step 8-11: Continue to chain 7 and join with the SL ST to the middle of the next chain space.

Step 12: Reach the hook over to make 2 DC over each joining point and 3 SC around each chain space. Join with the first DC.

Step 13: Chain 50, skip four chain spaces and join with SL ST. Repeat twice to create the 3 hanging handles.

Step 14: SC in each stitch around the rim of the basket (or along handles, if you want).

This design fits a 3x4x4 pot, but can stretch out to fit larger pots. If you want longer/shorter handles, do more/less chains in step 13.

Chapter 1: Beginner Crochet Projects

In this chapter, you'll find some crochet projects that are perfect for those that are new to the craft of crocheting. These projects don't require a ton of materials and can be done in a short amount of time, making for great practice. You may even have something beautiful to show for it once you're finished!

Simple Crochet Projects

There are many simple crochet projects out there that are perfect for beginner crocheters. Typically only using two or three different stitches, they have simple instructions and are a great way to practice your crocheting skills.

Beginner Scarf

This scarf is super beginner-friendly; it uses only one stitch, has simple instructions that are easy to follow, and works up fast.

Size: Approx. 6.5" width x 65" length

Materials:

- Any super bulky 6 yarn (less than two skeins will be used)
- 9 mm crochet hook
- Scissors
- Yarn/tapestry needle
- Measuring tape

Step 1: With your bulky yarn, CH 11.

Step 2: CH 1, in the second CH from the hook, SC, SC once in every CH across (10 total).

Step 3: CH 1, turn your work over, SC in the first stitch (not the CH 1), SC once in every stitch across (10 total).

Step 4: Repeat step 2 until you have 101 rows (1o times).

Step 5: Fasten yarn off with scissors, weaving in any loose ends in the scarf with a yarn needle.

One Skein Crochet Projects

If you're in the mood to crochet something but only have one skein of yarn, don't worry! One skein can go a long way. There are countless projects you can do even if your inventory is running a little low, and can be the perfect way to get rid of that one random skein you have lying around.

One Skein Soap Holder

These elegant looking soap bags can fit a full bar of soap and help prevent soap from staying wet and soggy. You can use any color you'd like!

Size: Approximately 7" x 6 ½" long

Materials:

- One skein of any 100% cotton yarn (this type is best to use if the item will be getting wet!)
- H 5.0 mm crochet hook
- Tapestry needle
- Scissors

Row 1: SC in the second chain from the hook, [chain 1, skip next stitch, SC in next space] continue between [] and end with a SC in the last stitch.

Row 2: Chain 2, HDC in the first SK ST from the first row [chain 1, skip next stitch, HDC in next space] ending with a HDC in the last stitch.

Row 3: Chain 2, in the first SK ST from the last row [chain 1, skip next stitch, SC in next space] continue between [] and end with a SC in the last stitch.

Rows 4-17: Repeat rows 2 and 3.

Row 18: SC, then in the first chain space, SC, chain 2, TR, DC. [In the next chain 1 space, SC, chain 2, TR, DC] Repeat between the [] until the end.

Be sure that there is still enough yarn so that it can go around the side and bottom of the soap holder, leaving about 1o inches at the end to be weaved. Once you're sure that much length is left, cut the yarn.

To join the edges of the soap holder, fold it lengthwise with the right sides together. By weaving the yarn through the loops on the end of your stitches, sew up the side and bottom. Weave in the yarn tale with the yarn needle and cut off any extra. Then turn the holder inside out.

To crochet the tie closure: Chain 60 and SL ST into each chain. Use the yarn ends to help you pull the tie through the holes that were made by the HDC and the SC. Then weave them in and cut off the excess. Weave the tie through the top holes made by the HDC and SC.

Scrap Yarn Crochet Projects

Most projects lead to leftover yarn, which can build up quickly. Luckily, there are lots of projects that are perfect for using up all those scraps and still result in a beautiful creation!



Scrap Yarn Rug

This easy crocheted rug is the perfect way to use up all the yarn scraps you have lying around. Using various colors make this rug vibrant and eye-catching, and it follows simple steps that make it a great beginner project.

Size: 22 x 32 inches

This size is perfect for a doorway rug. If you want it longer or shorter, simply alter the number of chains made at the beginning.

Materials:

- Leftover yarn scraps (about 40 oz. worth, give or take)
- 7 mm crochet hook (any larger hook will work)

Step 1: Holding two strands of yarn together, CH 54.

Step 2: SC in second CH from the hook each stitch to end and turn (53).

Step 3: CH 1, SC in each stitch across (53).

Step 4: Repeat step 3 until the rug measures 32” long. Fasten the yarn off and weave in the end.

And you’re done! This rug makes a great welcome mat or bath mat, and the color combinations can be endless!

Finger Crochet Projects

Finger crocheting is exactly what it sounds like—it’s just like normal crocheting, but you use your finger instead of a crochet hook. It’s a great way to practice the mechanics of each stitch and is for pros and newbies alike!

Cozy Blanket

This big, chunky blanket is super cozy and super easy to make.

Materials:

- Chunky wool yarn (about 10 skeins)

Chunky wool yarn tends to be a bit expensive. For this project, regular yarn will still do the trick, though you'll need to use four or five strands at once.

Step 1: Create a slip knot by crossing the tail of the yarn over the yarn you're working with, insert your fingers into the loop, and pull the working yarn through.

Step 2: Pull the working yarn through the loop to begin chaining. Make 26 chains.

Step 3: Turn the work and insert a finger into the second chain to pull up the working yarn. Place each loop on the thumb and pointer finger and pull the working yarn through, making the first SC. Continue to SC each chain. At the end of the row, CH 1, and turn the work.

Step 4: Place your finger under the first 'V' stitch to pull through the working yarn. Pull it through by inserting a finger and thumb into each loop. Continue with SC for the rest of the row.

Step 5: CH 1 at end of the row, turn, and SC the row. Repeat SC until your blanket reaches the desired length.

Chapter 2: Small Crochet Projects

Maybe you don’t have much time to spare, or maybe you’re running low on yarn. Maybe you’re just plain out of patience. No matter what the reason, there are endless amounts of small crochet projects that can be done quickly and use a limited amount of yarn. This chapter will cover some favorites, picked for their simplicity and accessibility for even the newest crocheters.

Can Cozies

These cozies are a great beginner project because they’re small, simple, can be completed quickly, and can be made in an endless number of color combinations.

Size: 3” x 4”

Materials:

- Worsted weight (CYC 4) cotton yarn in two colors (you can use other cotton or acrylic yarn, if desired)
- H 5 mm crochet hook
- Tapestry needle

Step 1: Take your first color and make a magic circle.

Step 2: Work 6 SC into the magic circle. Pull the loop and tighten. Join with SL ST in first SC. (6)

Step 3: CH 1, make 2 SC in each stitch around. Join with SL ST in the first SC. (12)

Step 4: CH 1, SC in first stitch, SC in the next 2 stitches. *2 SC in next stitch, SC in next 2 stitches* repeat from * around. Join by using SL ST in the first SC. (24)

Step 5: CH 1, SC in first 3 stitches, 2 SC in next stitch, *SC in next 2 stitches, 2 SC in next stitch* repeat from * around. Join by using SL ST in the first stitch. (30)

Step 6: CH 1, work in back loops, SC in each stitch around. Join by using SL ST in the first stitch. (30)

Step 7: CH 1, skip the first stitch, SC in the next stitch, *CH 1, skip next stitch, SC in next stitch* repeat from * around. Last stitch should be a SC. Join by using SL ST into chain space created at the beginning of the round. (30)

Step 8: CH 1, SC in the first CH-1 space, CH 1, skip the next stitch, *SC in next CH-1 space, CH 1, skip the next stitch* repeat from * around. It should end with a CH 1. Join into the first SC with SL ST.

Step 9-14: Repeat steps 7 and 8 with your main color. Switch to your second color at the end of the 14th round.

Step 15: Repeat step 7 with the second color.

Step 16: Repeat step 8 with the first color.

Step 17: Repeat step 7 with the second color.

Step 18: Repeat step 8 with the first color.

Step 19-21: Repeat by alternating steps 7 and 8 with the second color.

Step 22: Fasten off and weave in the ends. Congrats, your can cozy is finished!

Flower Bookmark

If you’re an avid reader (or even if you’re not), you know the importance of having a good bookmark. This beautiful flower bookmark can be made in any color and will give your book a touch of botanical elegance.

Size: 11” long (including leaves) x 1.5” wide

Materials:

- F 3.75 mm crochet hook
- Cotton yarn or other acrylic yarn (something not too thick)
- Tapestry needle
- Scissors

Step 1: To begin making the flowers, CH 4 and join using SL ST to form a ring.

Step 2: CH 3 and work 11 DC into the ring's center. Join and close the round using SL ST.

Step 3: CH 3, in the first space work 2 DC, then a SL ST. *Skip the next space, then work 3 DC and a SL ST all in the same space. Repeat from * until you have six petals. Join and close the round with a SL ST and fasten off.

Step 4: To begin the longer leaf, CH 4 and join using SL ST to form a ring.

Step 5: Work 3 DC, 4 SC, 1 TR, 4 SC, and 3 DC into the center of the ring. Join with SL ST to close the round. CH 10 without fastening off and insert the hook into a stitch between the petals. Wrap yarn around the hook and bring both loops through from the flower and from the leaf. Fasten off.

Step 6: To begin the shorter leaf, CH 4 and join using SL ST to form a ring.

Step 7: Work 3 DC, 4 SC, 1 TR, 4 SC, and 3 DC into the center of the ring. Join with SL ST to close the round. Without fastening CH 6, insert the hook into a stitch between the petals. Wrap yarn around the hook and bring both loops through from the flower and from the leaf. Fasten off.

Step 8: Weave in all the ends and you're all finished!

Chapter 3: Knitting Projects

Like crocheting, knitting is a method of stitching yarn together, just in a different style. Knitting consists of using long needles to form loops, keeping the stitches on the needles. If you're feeling confident in your crochet skills, then you may be ready to undertake a knitting project.



Easy Knitting Projects

The following knitting projects are ideal for beginners. Like crochet patterns, most knitting patterns use abbreviations that help make them easier to read and avoid clogging up the steps with too many words. Here are some common knitting abbreviations that you might find when reading patterns:

K–knit

P–purl

YO–yarn over

K2tog–knit two stitches together

P2tog–purl two stitches together

Foldover Beanie

This beanie requires the most basic knitting skills and makes a great project for any beginner knitters.

Size: 20” in diameter (but can be stretched out)

Materials:

- Size 8 (5.o mm) knitting needles
- Set of four size 8 double-pointed knitting needles
- Any wool or wool blend yarn (100 grams)
- Yarn needle

Step 1: To start the base of the hat, cast on 80 stitches using circular knitting needles. P1, K4 to the end of the row. Join for knitting in the round. Knit 4 inches of K4, P1 rib.

Step 2: Start knitting K1, P4 rib. Knit 5 inches in the following order:

- **Row 1:** *K1, P1, P2tog, P1* repeat from * until the end of the row. When finished, you should have 64 stitches on your needles.
- **Rows 2-3:** Knit K1, P3 rib
- **Row 4:** *K1, P1, P2tog* repeat from * until the end of the row. When finished, you should have 48 stitches on your needles.
- **Rows 5-6:** Knit K1, P2 rib
- **Row 7:** *K1, P2tog* repeat from * until the end of the row. When finished, you should have 32 stitches on your needles.
- **Rows 8-9:** Knit K1, P1 rib
- **Row 10:** K2tog until you reach the end of the row. There should be 16 stitches left.
- **Rows 11-12:** Knit until the end of the row.
- **Row 13:** K2tog until you reach the end of the row. There should be only 8 stitches left.

Step 3: Cut the yarn and weave the long tail through the 10 remaining stitches. Pull it tight and weave it into the end inside of the beanie. Your beanie is now complete!

Knitted Infinity Scarf

Infinity scarves have never quite gone out of style, and with good reason. They're fashionable, super comfy, and make great accessories. And if you've got some knitting skills, then why not make your own?

Size: Approximately 10" x 62"

Materials:

- Three colors of any bulky yarn (color 1, 2, and 3)
- 11/8 mm circular needles
- Slightly larger needle (for weaving ends)

Step 1: Using color 1, cast on 30 stitches and knit rows 1-8.

Step 2: Use color 2 to knit rows 9 and 10.

Step 3 (row 11): On the right side still using color 2. K1, YO2, *K1, YO2,* repeat between * across to the last two stitches, K1, YO2, K1.

Step 4 (row 12): On the wrong side. K1, drop 2 YO off the needle, *K1, drop 2 YO off the needle,* repeat between * across until the last stitch, K1.

Step 5: Knit rows 13 and 14.

Step 6: Using color 1, knit rows 15-22.

Step 7: Using color 3, knit rows 23-26.

Step 8 (row 27): K1, YO2, *K1, YO2,* repeat between * across to the last two stitches, K1, YO2, K1.

Step 9 (row 28): K1, drop 2 YO off the needle, *K1, drop 2 YO off of the needle,* repeat between * across until the last stitch, K1.

Step 10: Knit rows 29 and 30.

Step 11: Switch back to color 1. Knit rows 31-40.

Step 12: Switch to color 2. Knit rows 41-44.

Step 13 (row 45): K1, YO2, *K1, YO2,* repeat between * across to the last two stitches, K1, YO2, K1, turn.

Step 14 (row 46): K1, drop 2 YO off the needle, *K1, drop 2 YO off the needle,* repeat between * across until the last stitch, K1, turn.

Step 15: Switch to color 1. Work evenly using the garter stitch until the piece measures 62” long, or whatever your desired length is. Bind it off and join the ends using a mattress stitch. Your infinity scarf is complete!

Chapter 4: Fun Crochet Projects

Any experienced crocheter will tell you that crocheting isn't only relaxing and therapeutic, but also a lot of fun. Most crochet projects have room for variation and invite you to use your imagination. These projects are all highly customizable and result in something gorgeous, no matter what your skill level is.

Crochet Headband

This simple crocheted headband uses only two simple stitches, making it a great option for beginners while leaving lots of room for variation for those advanced crocheters looking to challenge themselves. Once you've mastered this pattern, try combining two different colors!

Size: 18" x 2" (To adjust the size, simply add more or less rows.)

Materials:

- Any color size 3 yarn or worsted weight yarn (30-50 yards will be needed)
- J/10 6 mm hook
- Tapestry needle
- Stitch markers

Step 1: Make a slip knot on the hook, CH 9.

Step 2: In the first chain from your hook, place a stitch marker. SC in the third chain from the hook, [CH 1, skip next chain, SC in next chain] repeat between [] three times, turn. 7 stitches.

Step 3: CH 1, [SC in the next chain, 1 space, CH 1] repeat between [] three times, work SC into marked stitch, turn. 7 stitches.

Step 4: Repeat step 2. At the end of the row, work the last SC into the turning chain of the previous row. Repeat until the headband reaches 18 inches, or whatever your desired length is.

Step 5: Cut the yarn, making sure that there is 10 inches left on the tail. Thread the tail into a tapestry needle and stitch the ends together on the wrong side. Then weave the ends in and turn it back to the right side. Your headband is now ready to wear!

Crochet Snowflake Ornament

These cute little snowflakes are small, simple to make, easy to customize, and a great way to decorate the Christmas tree. And there's nothing quite as fulfilling as knowing you made them yourself!

Materials:

- Cotton or acrylic yarn
- 3 mm crochet hook
- Scissors

Step 1: CH 5, 1 SL ST in the first chain to form a ring.

Step 2: CH 1, 11 SC in chain ring, 1 SL ST in first stitch.

Step 3: CH 2, *1 DC in the next SC, CH 6, 1 DC in the next SC* repeat between * five times, 1 DC in the next SC, CH 6, 1 SL ST in the second chain at the beginning of the round.

Step 4: CH 1, *1 SC in between the next 2 DC, 4 SC in chain space, CH 3, SC in third chain from the hook, 4 SC in the chain space* repeat between * five times, 1 SC between the next 2 DC, 4 SC in the chain space, CH 20, 4 SC in the chain space, 1 SL ST in the chain from the beginning of the round.

Step 5: Fasten off. Your snowflake is ready to be hung!

Granny Square

This classic granny square pattern is a known beginner's project because it uses the most basic crochet stitches. There are countless variations on this pattern and can serve as a base for larger projects like blankets and bags.

Once you master this one, use your imagination and branch out; the possibilities with these are truly endless.

Materials:

- Small amount of any desired yarn
- H 5.0 mm hook
- Stitch markers
- Scissors

Step 1: CH 4. Insert hook into the first chain stitch and make a SL ST. This should join the chain stitches into a circle.

Step 2: CH 3.

Step 3: 2 DC into the center ring. CH 3.

Step 4: Work 3 DC into the center ring. CH 3.

Step 5: Work 3 DC into the center ring. CH 3.

Step 6: Work 3 more DC into the center ring. CH 3. You should now have four ‘granny clusters.’

Step 7: SL ST into the top of the first CH 3 to join the round and create a square shape.

Step 8: CH 4.

Step 9: 3 DC, CH 3, 3 DC, CH 1 in the next CH-3 space.

Step 10: Use SL ST on CH 3 of the beginning chain stitches to join.

Step 11: CH 3.

Step 12: 2 DC, CH 1 worked into the CH-1 space below the previous round.

Step 13: 3 DC, CH 3, 3 DC, CH 1 in the next CH-3 corner space.

Step 14: 3 DC, CH 1 worked into the next CH-1 space.

Step 14: Repeat steps 13 and 14 around to the beginning chain. Join with SL ST to the top of the first CH-3.

Step 15: CH 4. 3 DC, CH-1 worked into the next CH-1 space.

Step 16: 3 DC, CH 3, 3 DC, CH 1 worked into the next CH-3 corner space.

Step 17: Repeat steps 16 and 16 in the remaining CH-1 and CH-3 spaces. 2 DC in the last CH-1 space.

Step 18: Use SL ST to join to the top of the first CH-3. Your granny square is complete! If you want it to be larger, simply continue making rounds until the square reaches the desired size.

Chapter 5: Cute Crochet Projects

Unlike many of the other patterns in this book, the pattern in this chapter doesn't have a function or serve a particular purpose—other than being adorable, of course! This cute pattern is relatively simple and results in a cute creation that makes a great decoration or gift for a loved one.



Crochet Owl

This sweet and sleepy looking owl looks beautiful and uses simple stitches, making it a great beginner craft. This pattern requires you to use some minor sewing skills and stuffing as well.

Size: Approx. 4" tall

Materials:

- 1.80 mm hook
- Worsted weight yarn (4 ply) in six colors
- Polyester fiberfill
- Yarn needle
- Scissors

BODY

Step 1: CH 2, 6 SC in the second chain from the hook. Do not join. (6)

Step 2: Increase stitches in each stitch around. (12)

Step 3: *SC in next stitch, increase stitches* repeat between * around. (18)

Step 4: *SC in next two stitches, increase stitches* repeat between * around. (24)

Step 5: *SC in the next three stitches, increase stitches* repeat between * around. (30)

Step 6: SC in the next two stitches, increase stitches, *SC in next four stitches, increase stitches* repeat between * five times, SC in next two stitches. (36)

Step 7: *SC in the next five stitches, increase stitches* repeat between * around. (42)

Step 8: SC in the next three stitches, increase stitches, *SC in the next six stitches, increase stitches* repeat between * five times, SC in the next three stitches. (48)

Step 9: *SC in the next seven stitches, increase stitches* repeat between * around. (54)

Step 10: SC in the next four stitches, increase stitches. *SC in the next eight stitches, increase stitches* repeat between * five times, SC in the next four stitches. (60)

Step 11: SC in each stitch round. (Makes rows 11-20)

Step 12: *SC in the next 12 stitches, decrease stitches* repeat between * around. (56)

Step 13: SC in each stitch around. (row 22)

Step 14: SC in the next six stitches, decrease stitches, *SC in the next 12 stitches, decrease stitches* repeat between * three times, SC in the next six stitches. (52)

Step 15: SC in each stitch around. (row 24)

Step 16: *SC in the next 11 stitches, decrease stitches* repeat between * around. (48)

Step 17: SC in each stitch around. (rows 26-27)

Step 18: SC in the next five stitches, decrease stitches, *SC in the next 10 stitches, decrease stitches* repeat between * three times, SC in the next five stitches. (44)

Step 19: SC in each stitch around. (rows 29-30)

Step 20: Fold the work in half, matching the stitches on the front with the stitches on the back. Working through both layers, SC in the next 21 stitches. Secure the end. The mouth is now finished.

*Note: Stuff the body firmly with the polyester fiberfill as you work it up.

WING

Step 1: CH 2, 6 SC in the second chain from the hook. Do not join. (6)

Step 2: Increase stitches in each stitch around. (12)

Step 3: *SC in the next stitch, increase stitches* repeat between * around. (18)

Step 4: *SC in the next two stitches, increase stitches* repeat between * around. (24)

Step 5: *SC in the next three stitches, increase stitches* repeat between * around. (30)

Step 6: SC in the next two stitches, increase stitches, *SC in the next four stitches, increase stitches* repeat between * five times, SC in the next two stitches, SL ST in the next stitch. (36)

Step 7: Finish off and leave a long end for sewing. Fold the work in half and sew between round 19 and 20 of the body using three stitches.

Repeat steps 1-7 to make the second wing.

EYES

Step 1: CH 2, 6 SC in the second chain from the hook. Do not join. (6)

Step 2: Increase stitches in each stitch around. (12)

Step 3: *SC in the next stitch, increase stitches* repeat between * around. (18)

Step 4: *SC in the next two stitches, increase stitches* repeat between * around. (24)

Step 5: SC in the next stitch, increase stitches, *SC in the next three stitches, increase stitches* repeat between * five times, SC in the next two stitches, SL ST in the next stitch. (30)

Step 6: Finish off and leave a long end for sewing. Sew between rounds 17 and 25 of the body, leaving a width of one stitch.

Repeat steps 1-6 to make the second eye.

MOUTH

Step 1: CH 2, 3 SC in the second chain from the hook. (3)

Step 2: CH 1, turn work, SC in the same stitch, increase stitches, SC in the next stitch. (4)

Step 3: CH 1, turn work, SC in each stitch across. (4)

Step 4: Finish off and leave a long end for sewing. Sew between rounds 15 and 18 of the body.

Your owl is complete! If you want to challenge yourself, you can add details into the eyes or on the front of the owl.

Conclusion

The patterns in this book are only a glimpse into the crocheting world and all it has to offer. They are simple, functional, and fun, all great for practicing the mechanics of crocheting and becoming familiar with abbreviations and equipment.

Hopefully this book has left you feeling confident in your crocheting skills, and the only place you can go from here is up. With hundreds of thousands of patterns out there for mittens, food, and everything in between, the possibilities are truly endless. Good luck as you embark on your crocheting adventure!



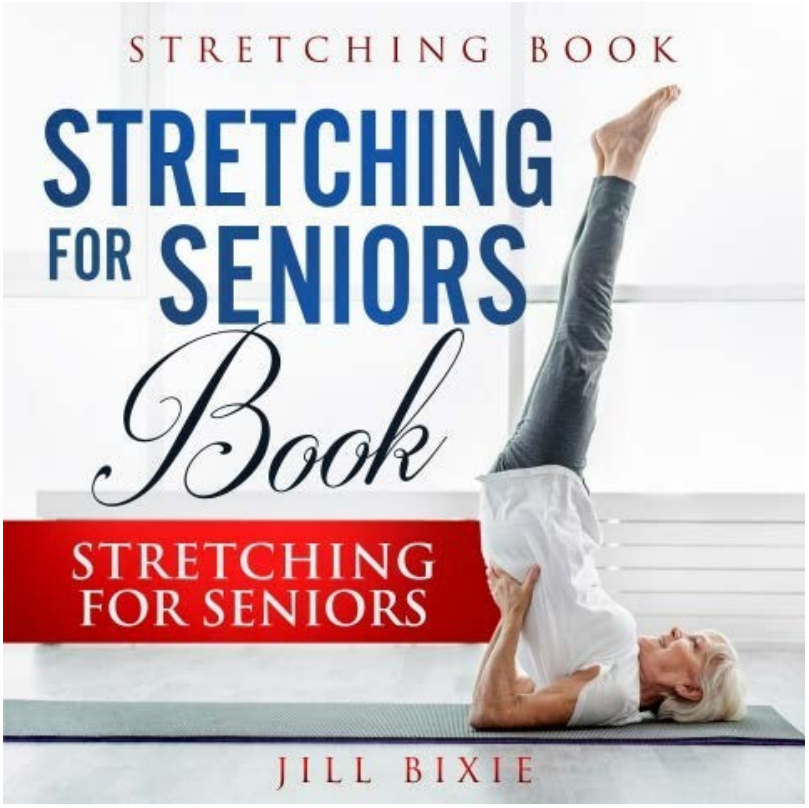
Your FREE Gift

You can grab Your copy by clicking on the following link and joining my mailing list.

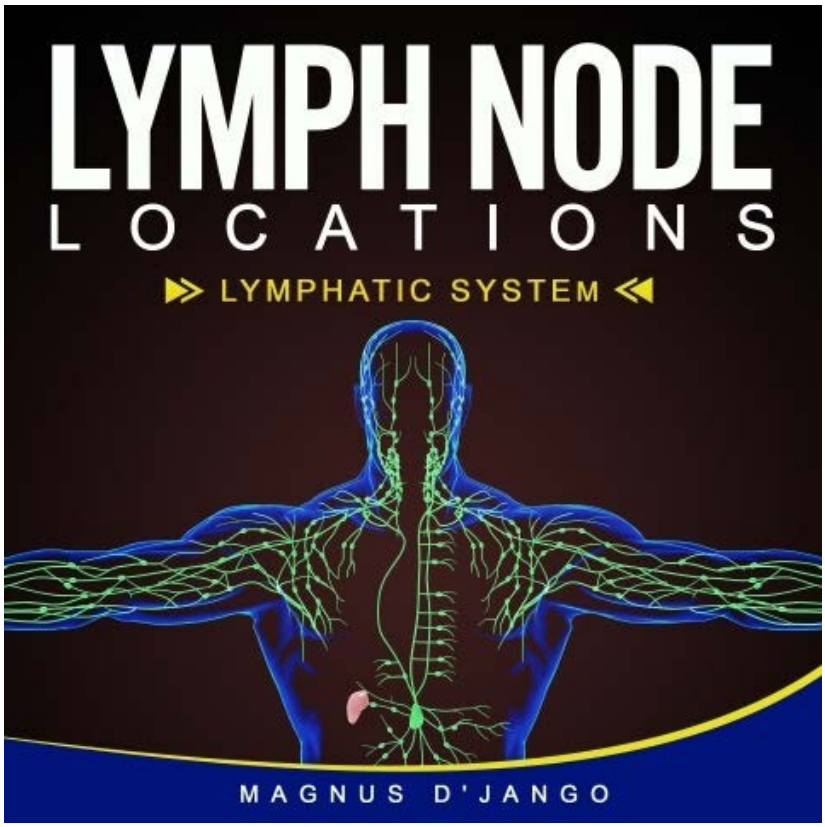
FREE BOOKS https://imris-kdp-publishing.com/

BONUS BOOK BELOW

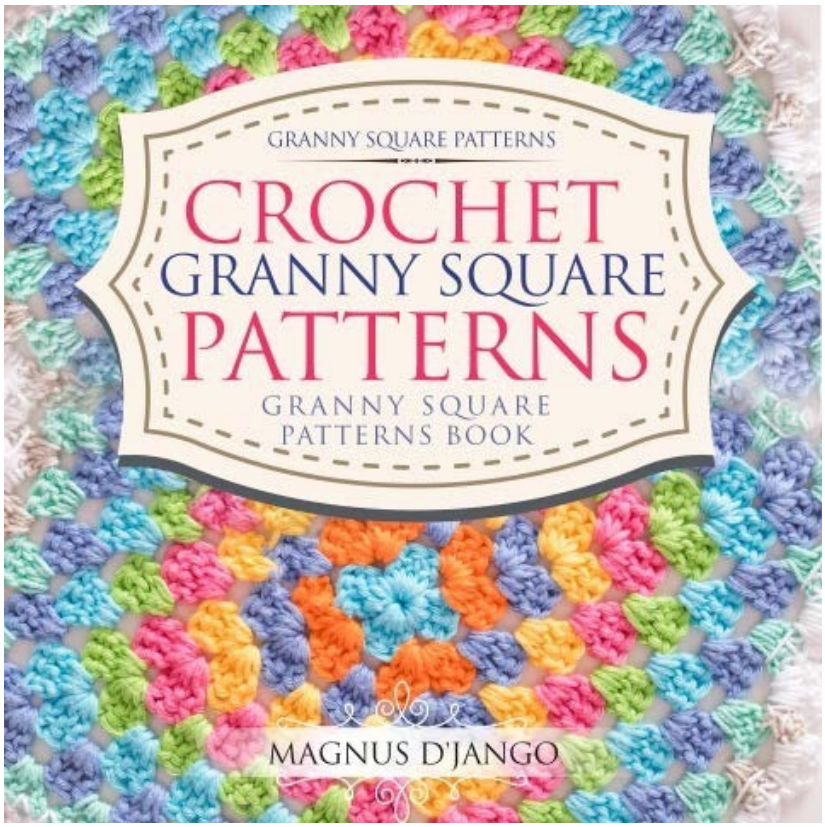
Our Other Titles



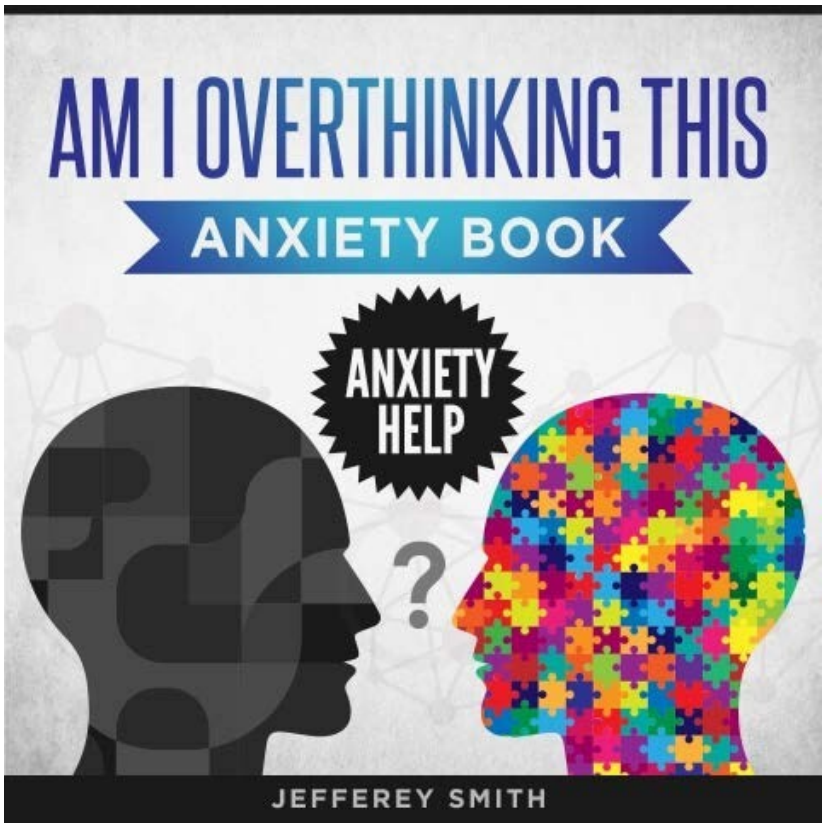
CLICK HERE NOW TO GET YOUR AUDIOBOOK



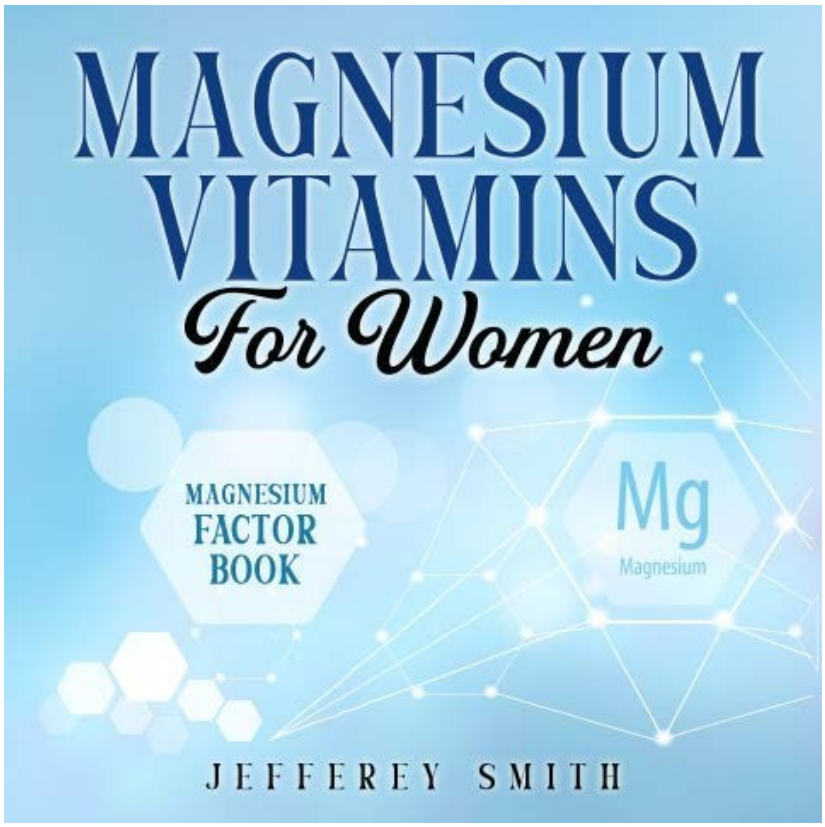
CLICK HERE NOW TO GET YOUR AUDIOBOOK



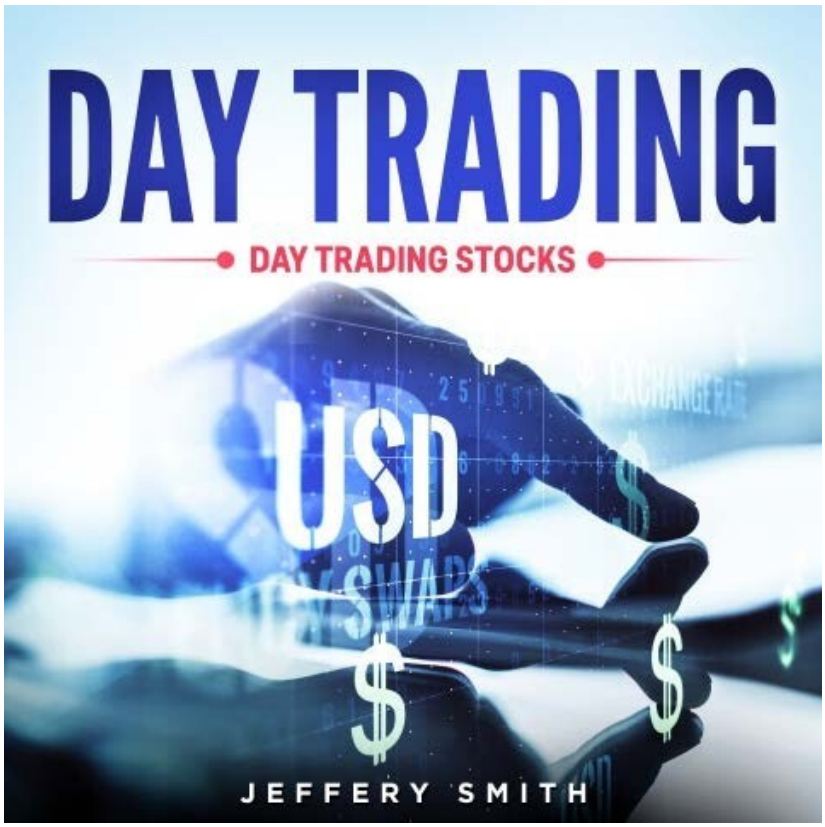
CLICK HERE NOW TO GET YOUR AUDIOBOOK



[CLICK HERE NOW TO GET YOUR AUDIOBOOK]



CLICK HERE NOW TO GET YOUR AUDIOBOOK



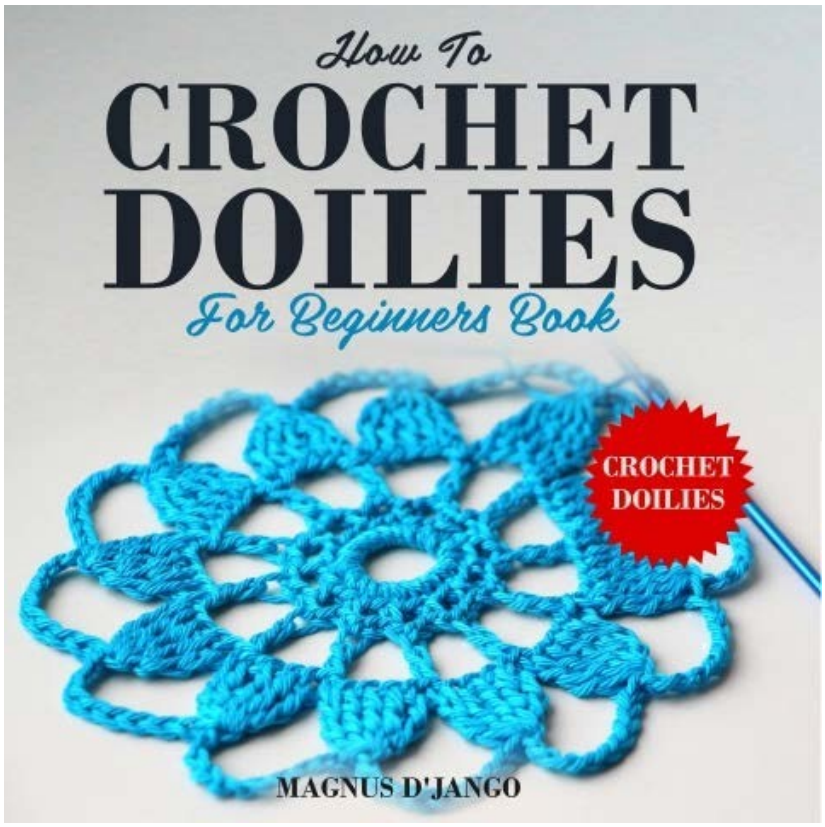
CLICK HERE NOW TO GET YOUR AUDIOBOOK



[CLICK HERE NOW TO GET YOUR AUDIOBOOK]



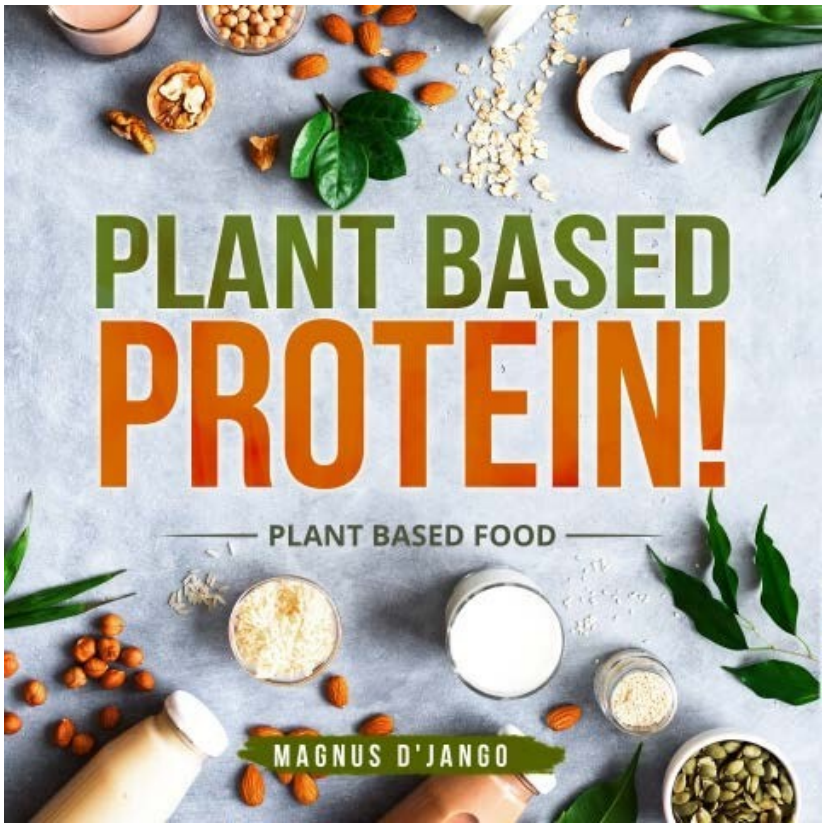
CLICK HERE NOW TO GET YOUR AUDIOBOOK



CLICK HERE NOW TO GET YOUR AUDIOBOOK



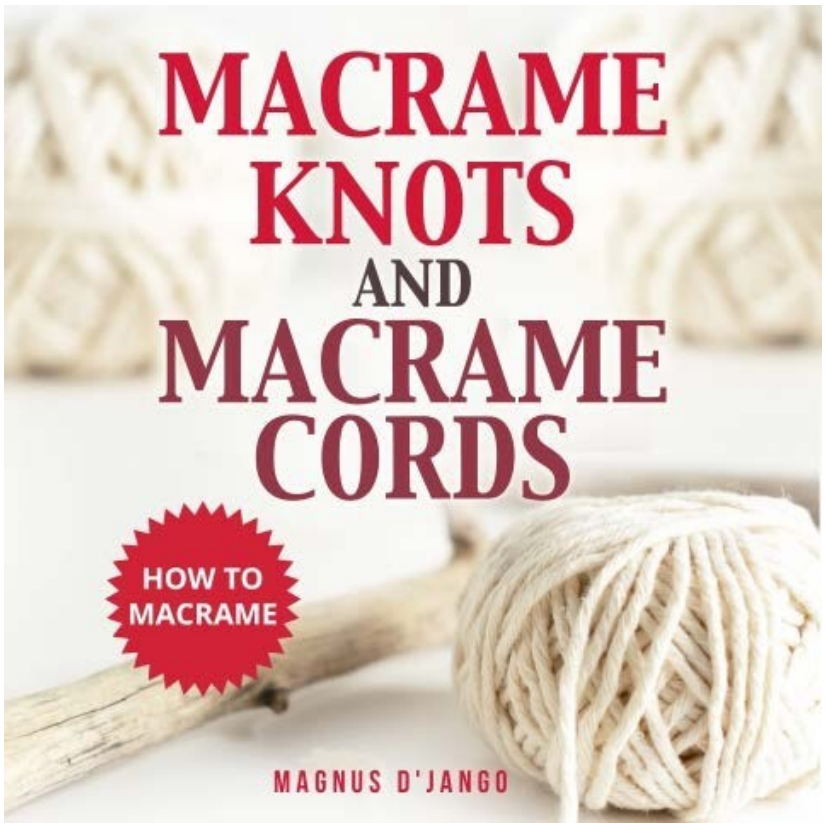
CLICK HERE NOW TO GET YOUR AUDIOBOOK



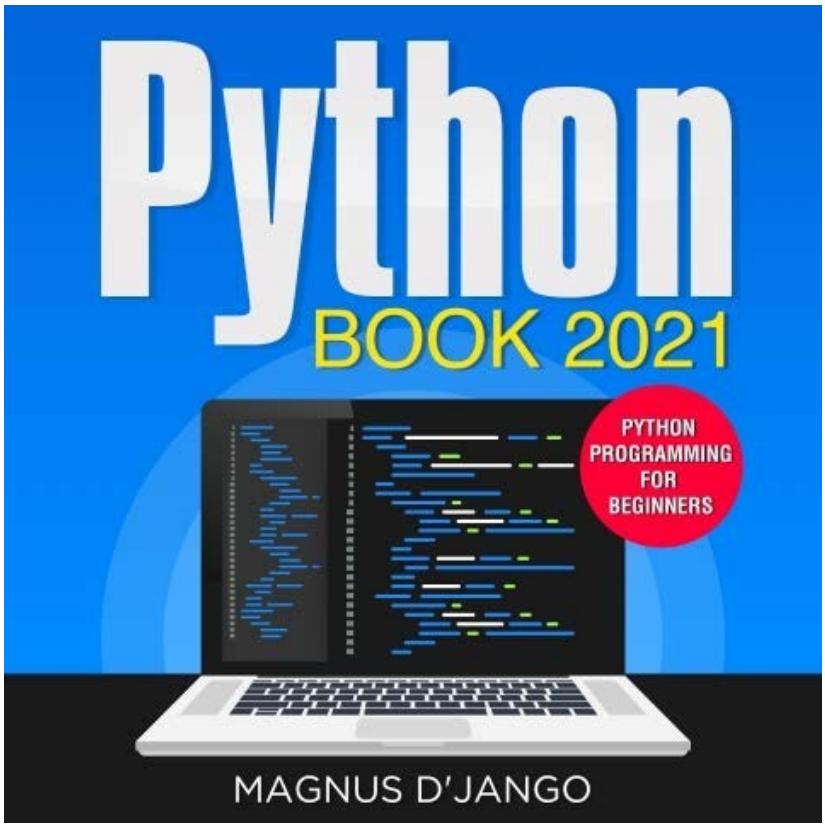
CLICK HERE NOW TO GET YOUR AUDIOBOOK



CLICK HERE NOW TO GET YOUR AUDIOBOOK



CLICK HERE NOW TO GET YOUR AUDIOBOOK



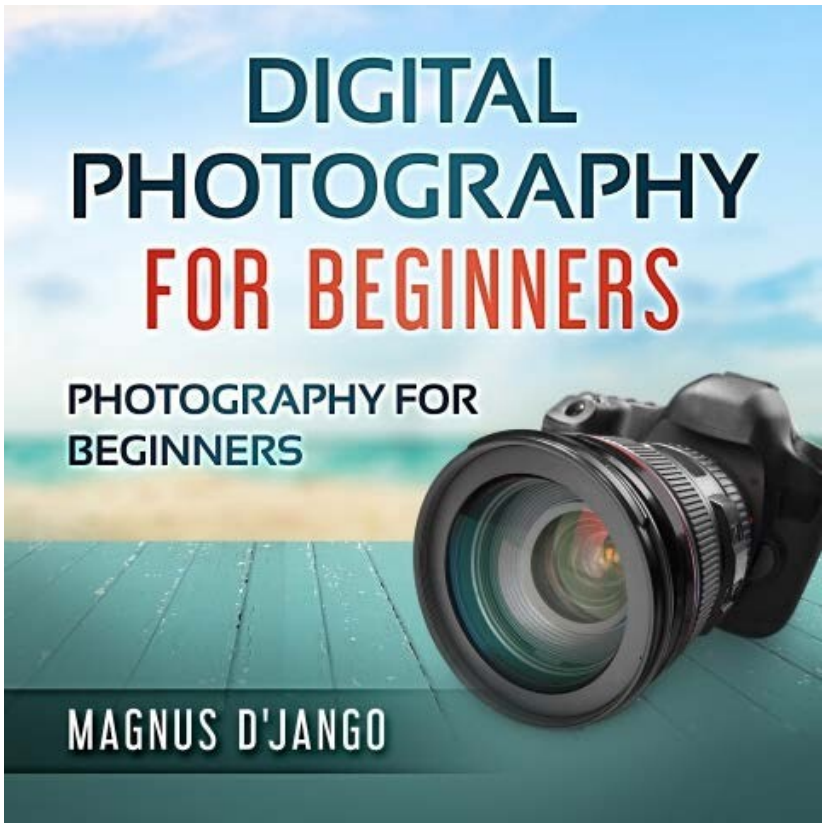
CLICK HERE NOW TO GET YOUR AUDIOBOOK



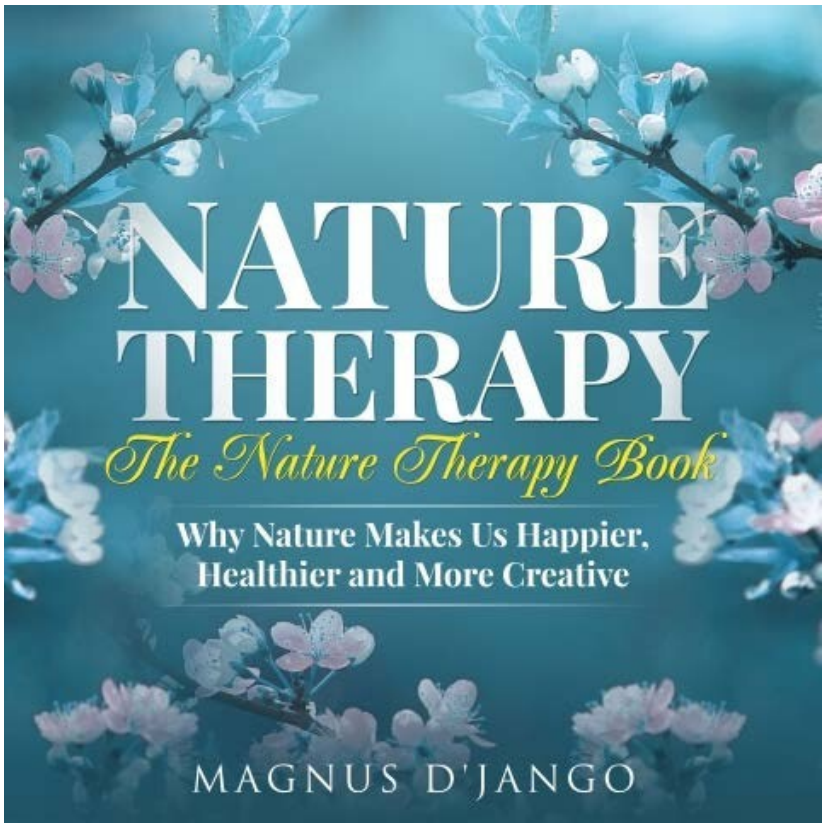
CLICK HERE TO GET YOUR AUDIOBOOK



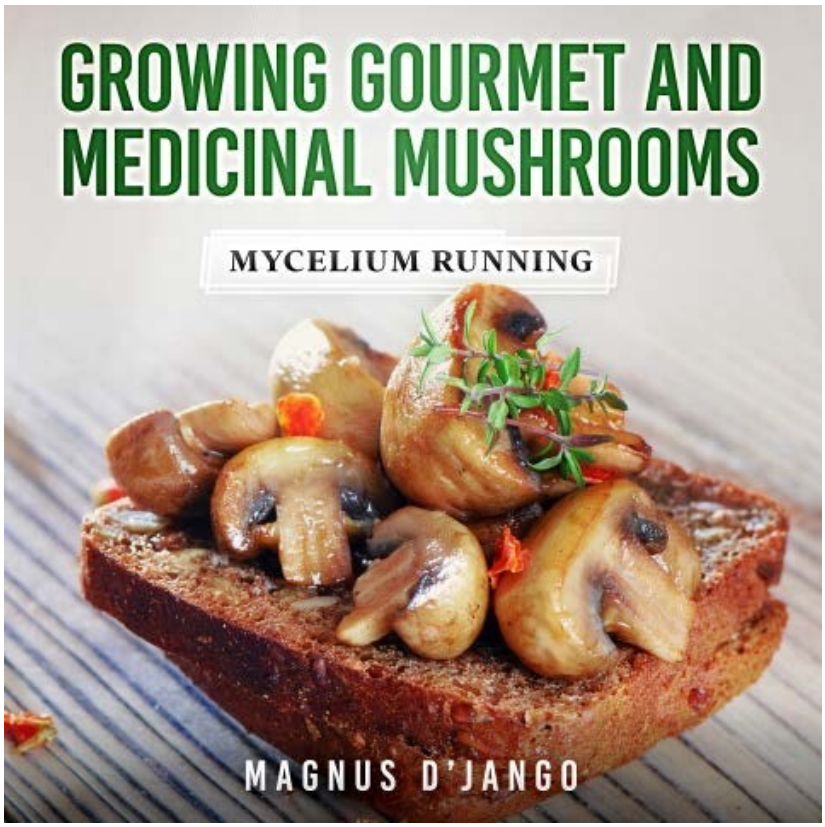
[CLICK HERE TO GET YOUR AUDIOBOOK]



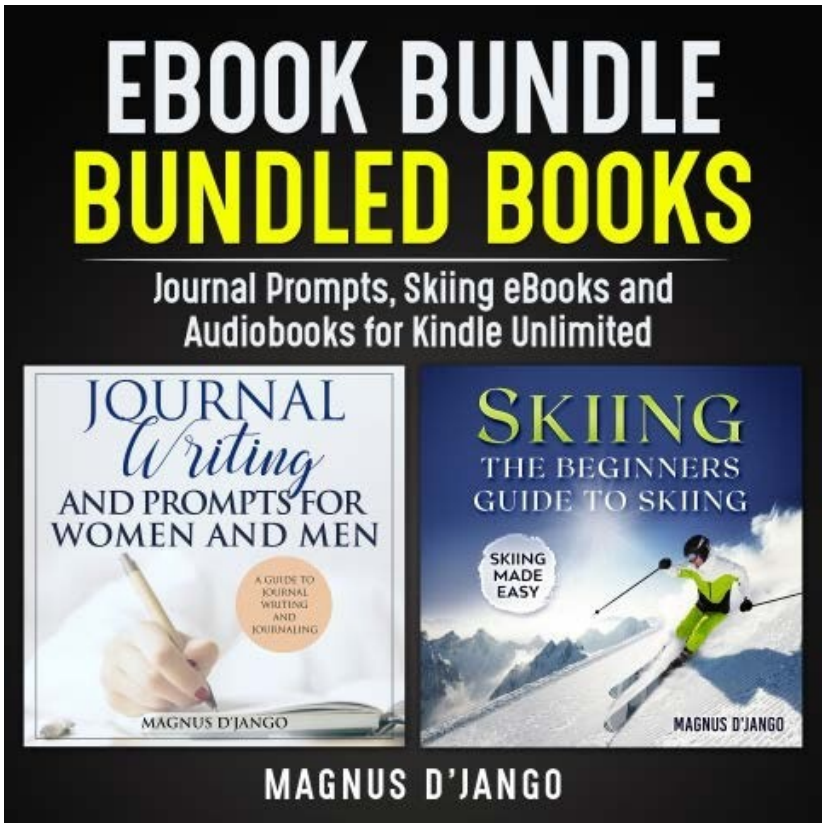
CLICK HERE TO GET YOUR AUDIOBOOK



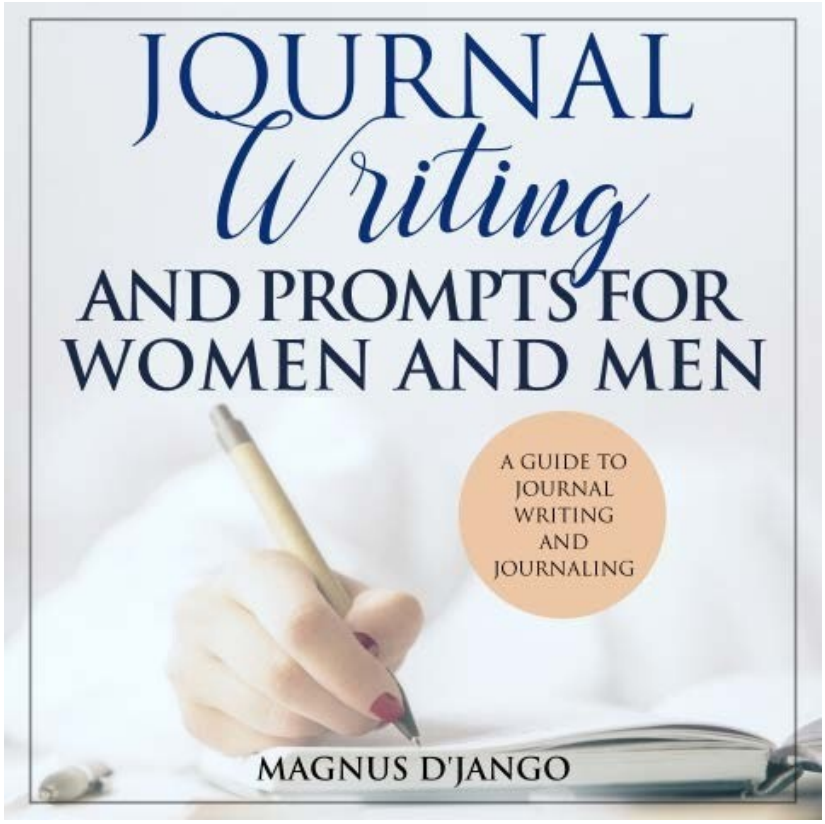
CLICK HERE NOW TO GET YOUR AUDIOBOOK



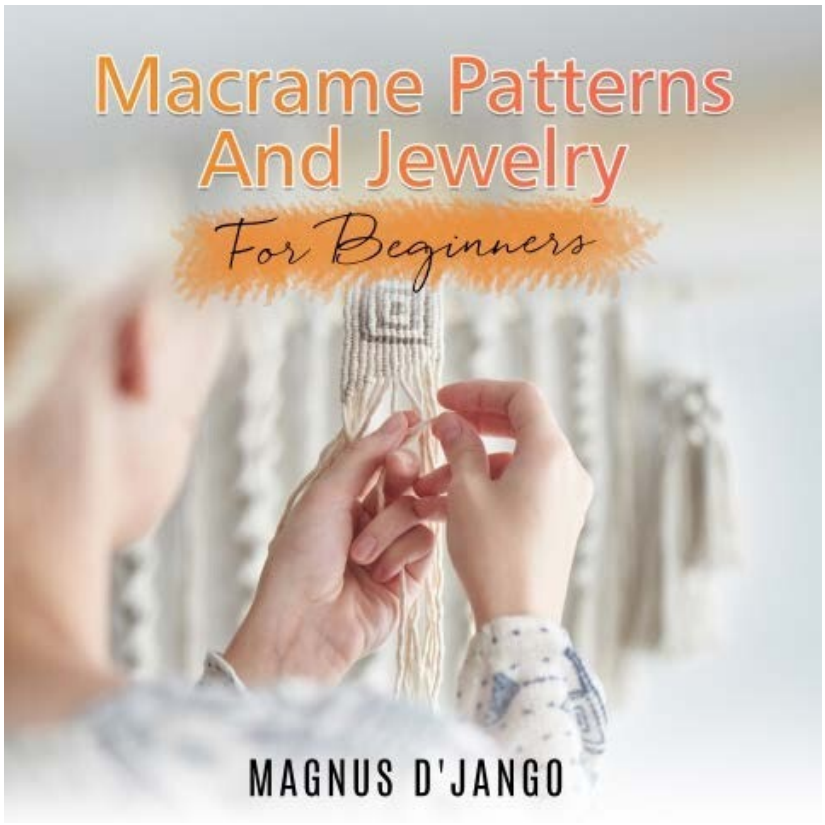
CLICK HERE NOW TO GET YOUR AUDIOBOOK



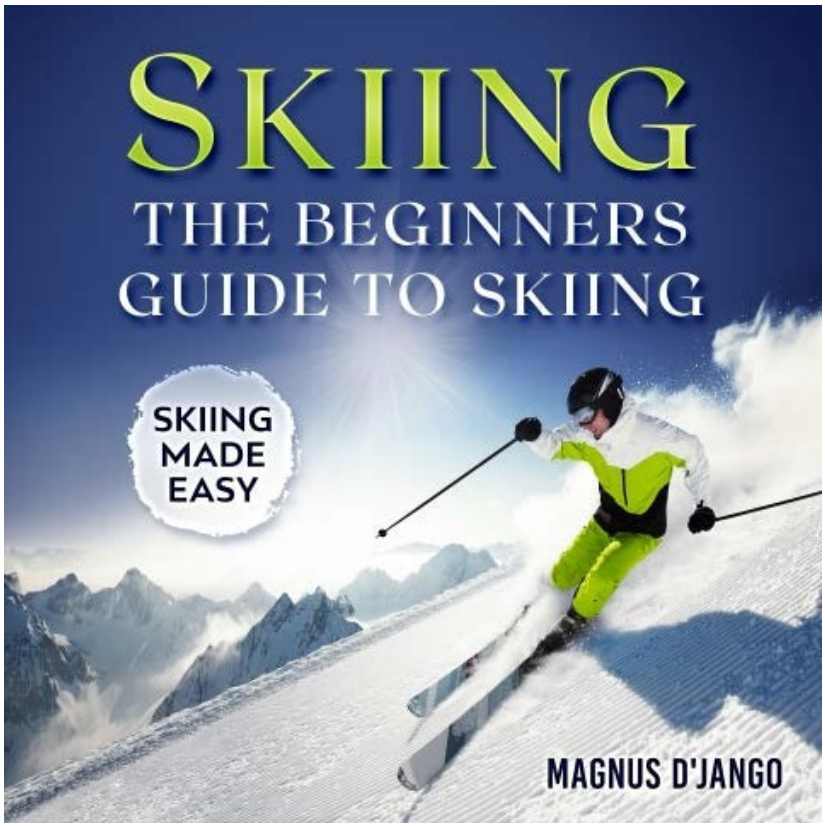
[CLICK HERE NOW TO GET YOUR AUDIOBOOK]



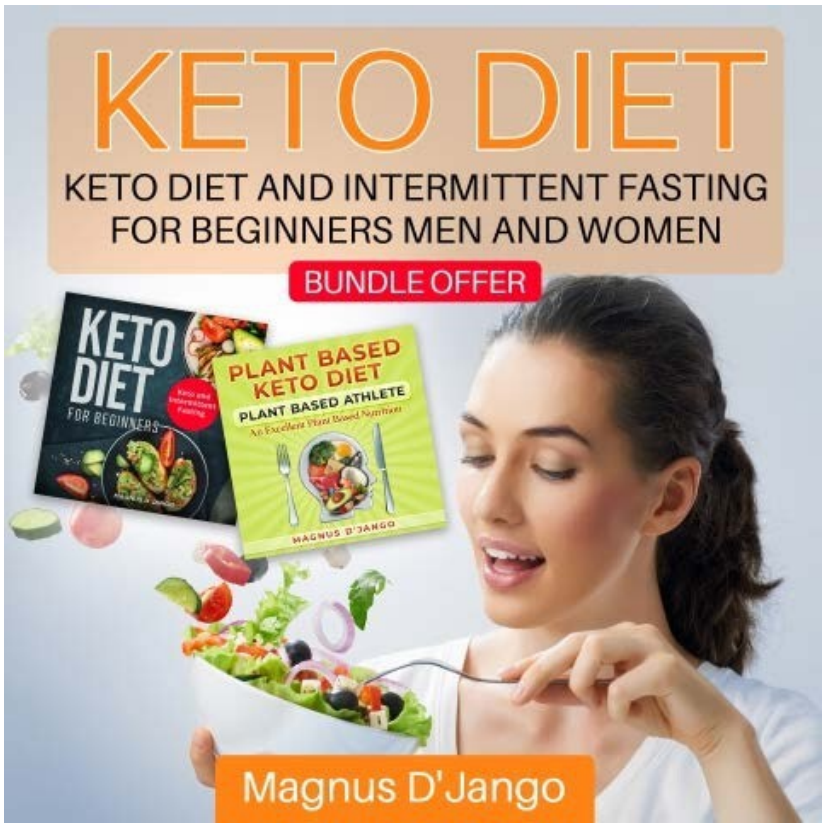
CLICK HERE NOW TO GET YOUR AUDIOBOOK



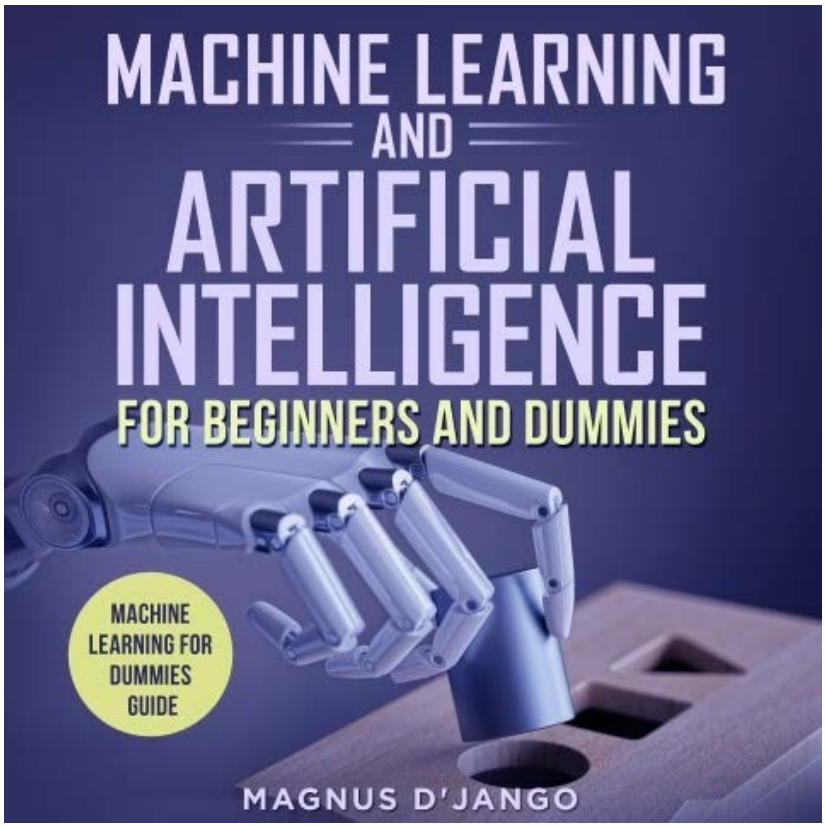
CLICK HERE NOW TO GET YOUR AUDIOBOOK



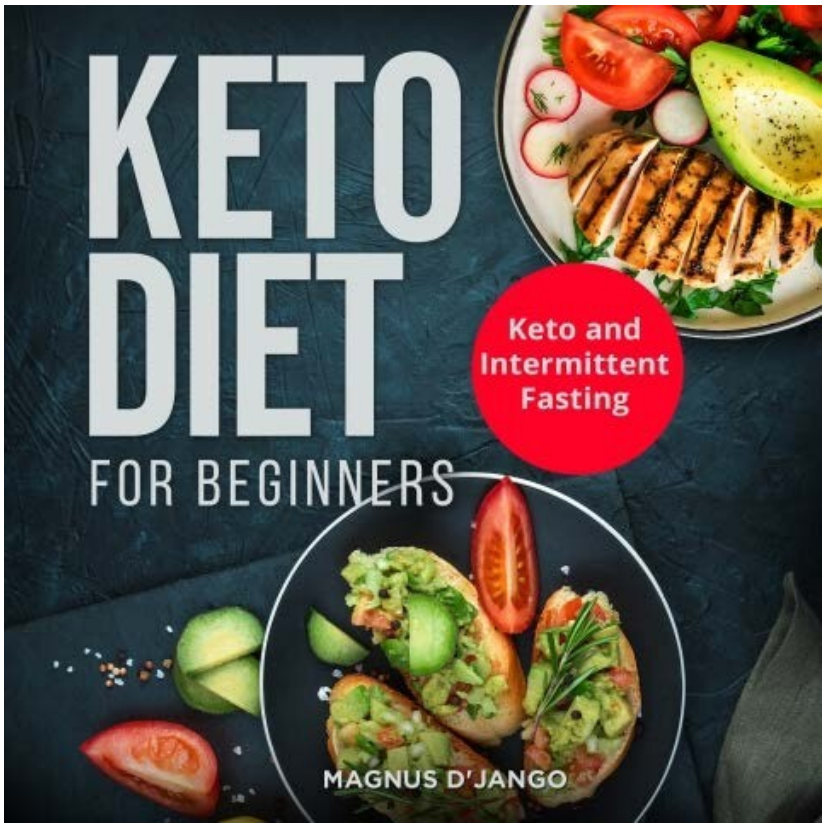
CLICK HERE NOW TO GET YOUR AUDIOBOOK



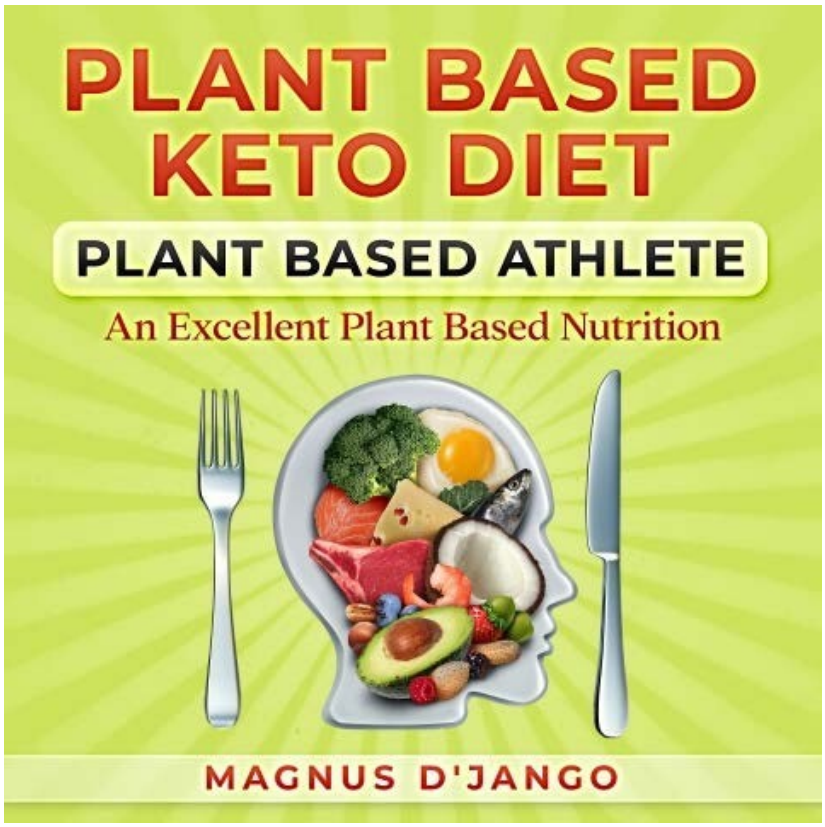
[CLICK HERE NOW TO GET YOUR AUDIOBOOK]



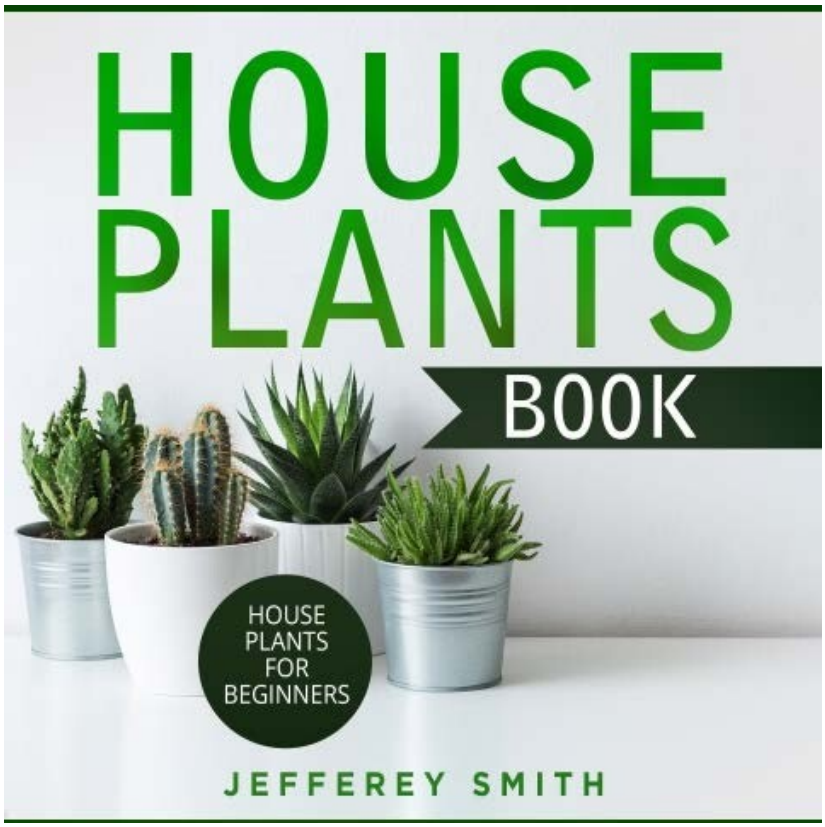
CLICK HERE NOW TO GET YOUR AUDIOBOOK



[CLICK HERE NOW TO GET YOUR AUDIOBOOK]



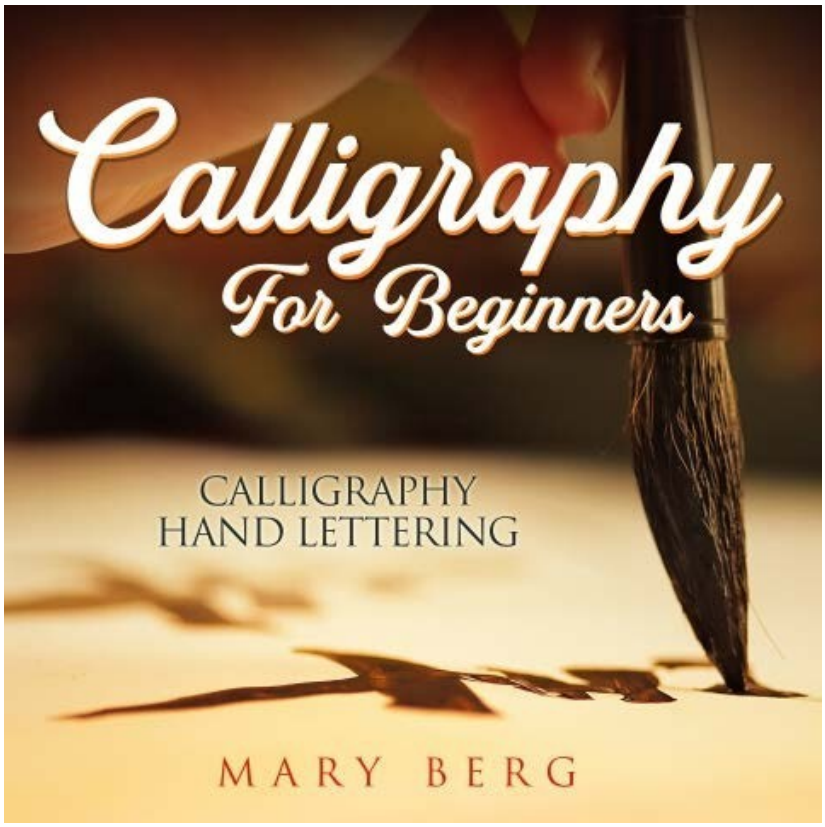
CLICK HERE NOW TO GET YOUR AUDIOBOOK



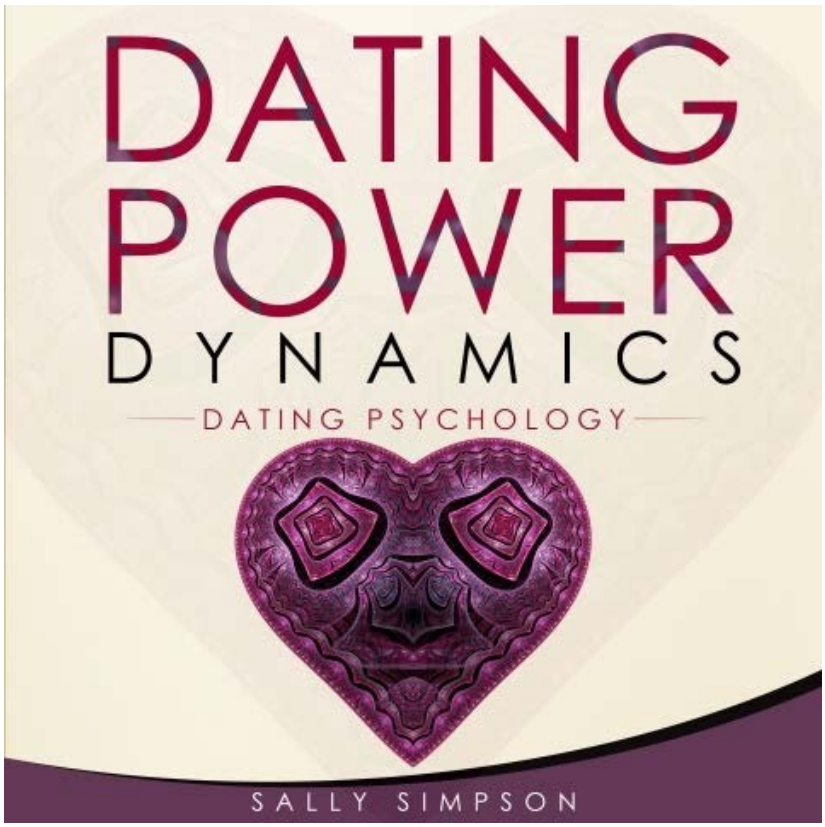
[CLICK HERE TO GET YOUR AUDIOBOOK NOW]



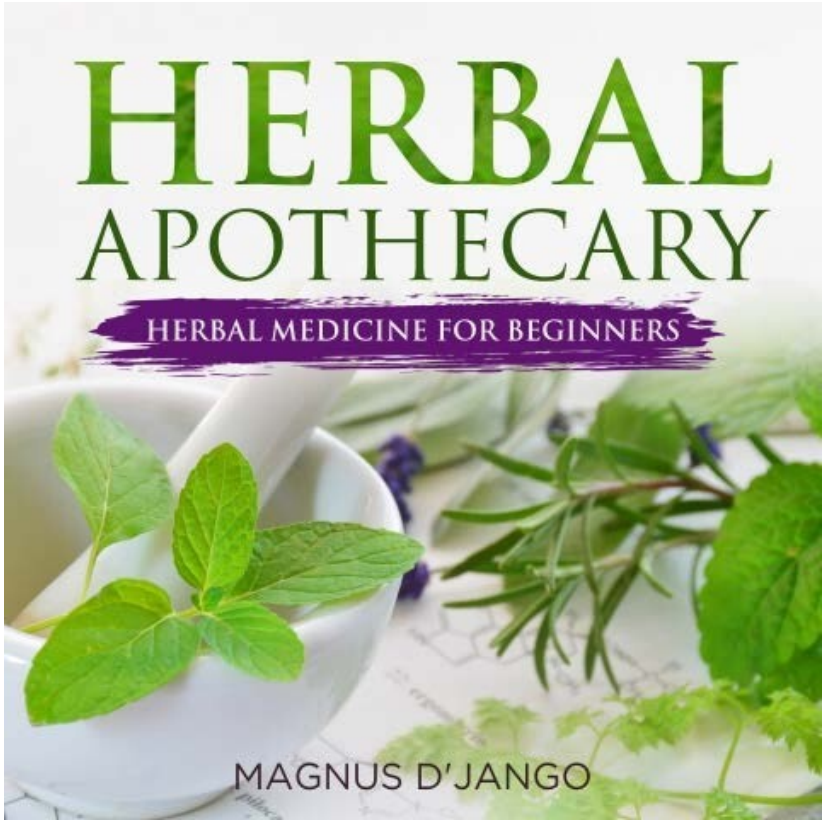
CLICK HERE TO GET YOUR AUDIOBOOK NOW



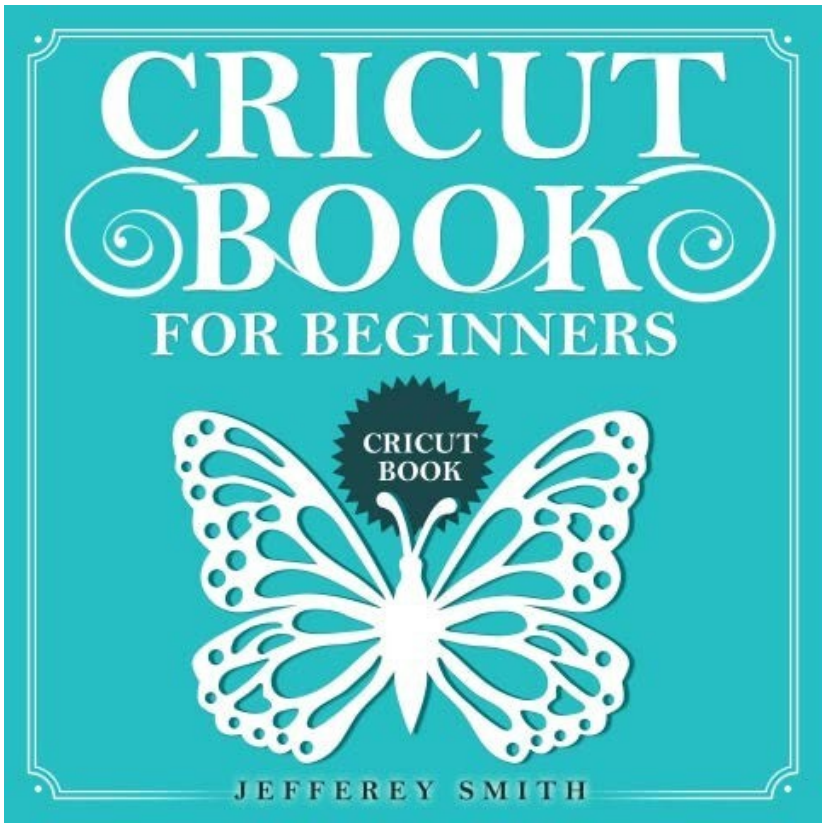
CLICK HERE TO GET YOUR AUDIOBOOK NOW



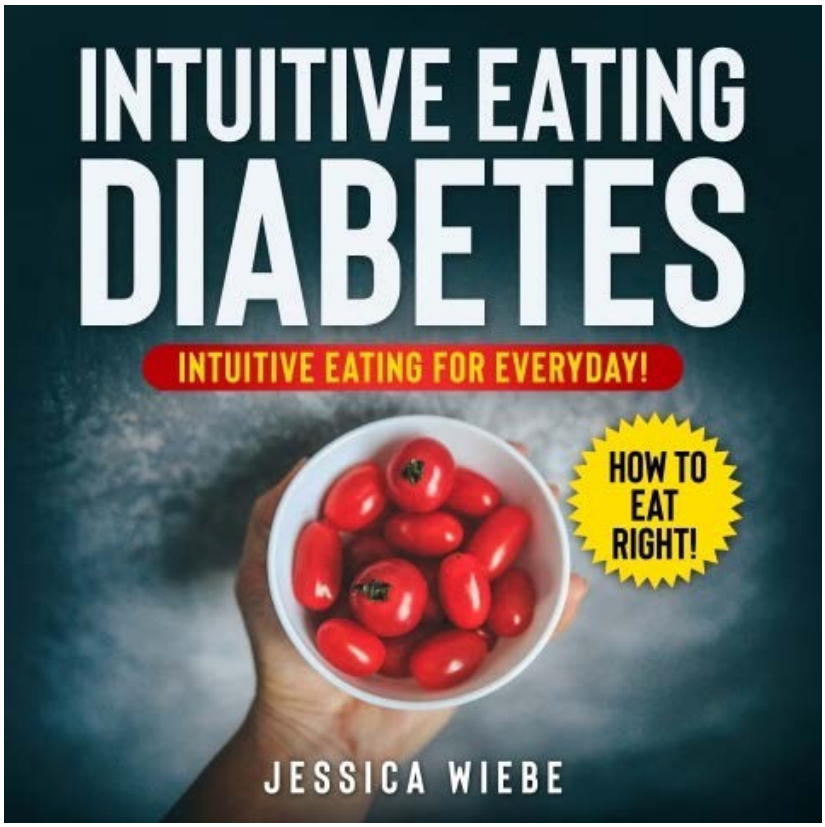
CLICK HERE TO GET YOUR AUDIOBOOK NOW



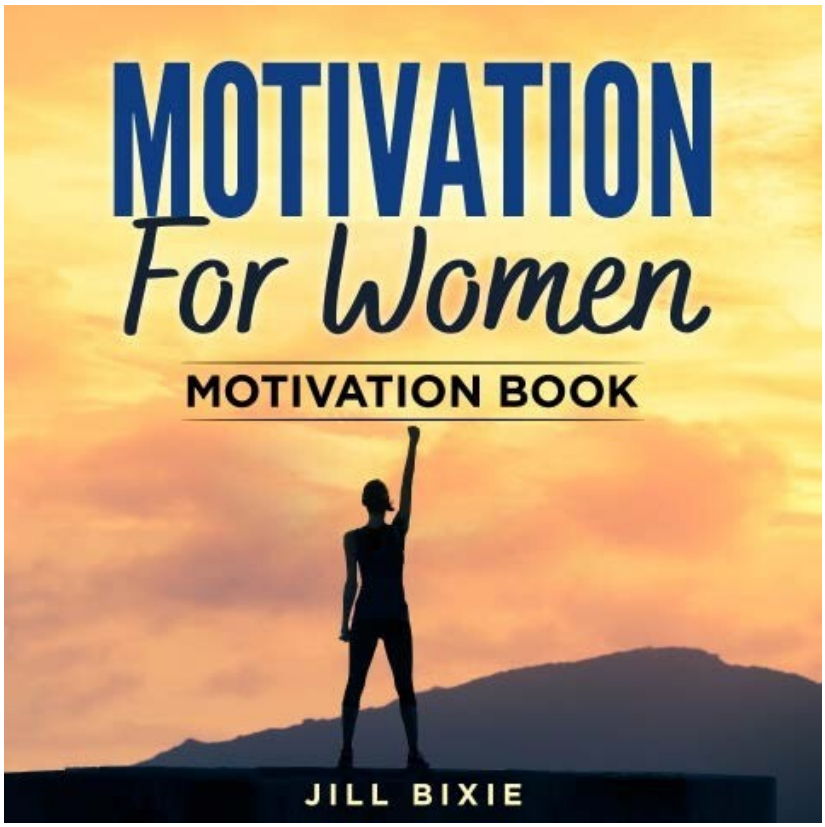
CLICK HERE NOW TO GET YOUR AUDIOBOOK



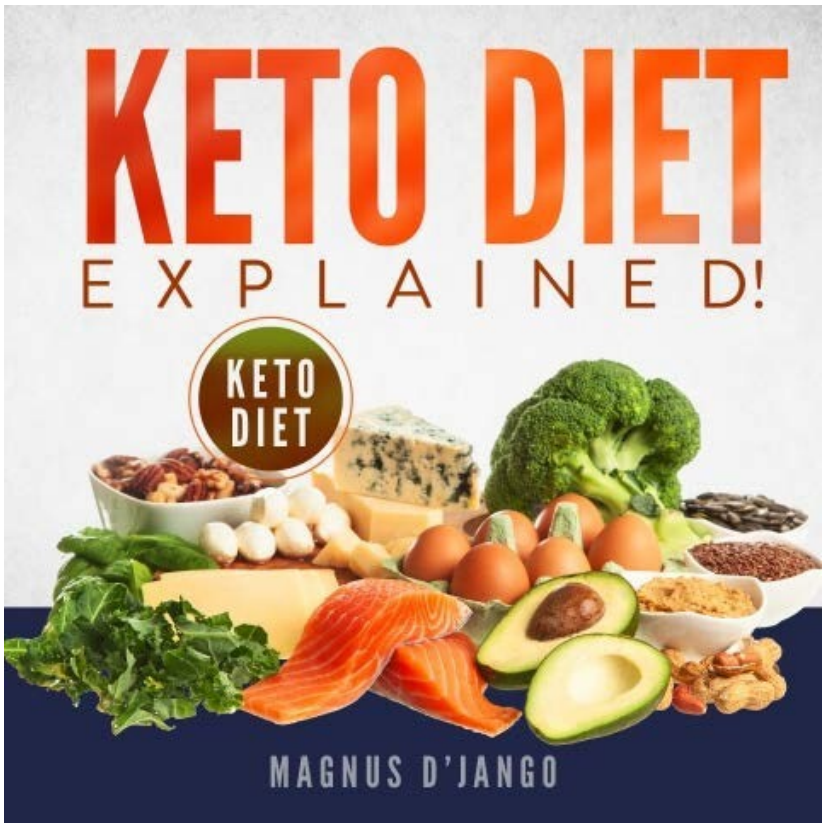
CLICK HERE NOW TO GET YOUR AUDIOBOOK



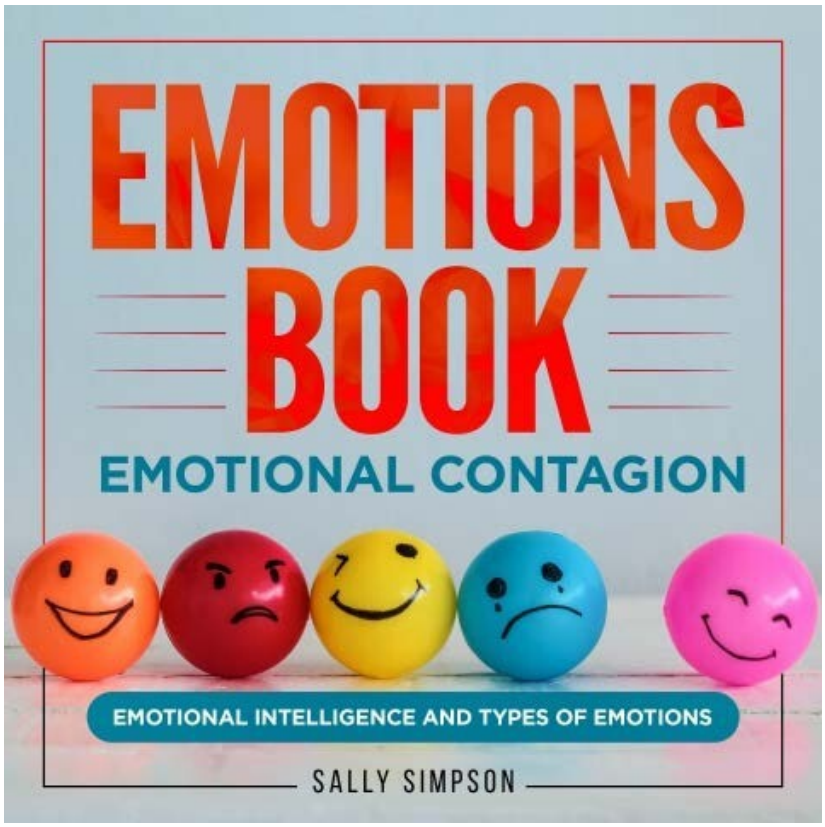
CLICK HERE NOW TO GET YOUR AUDIOBOOK



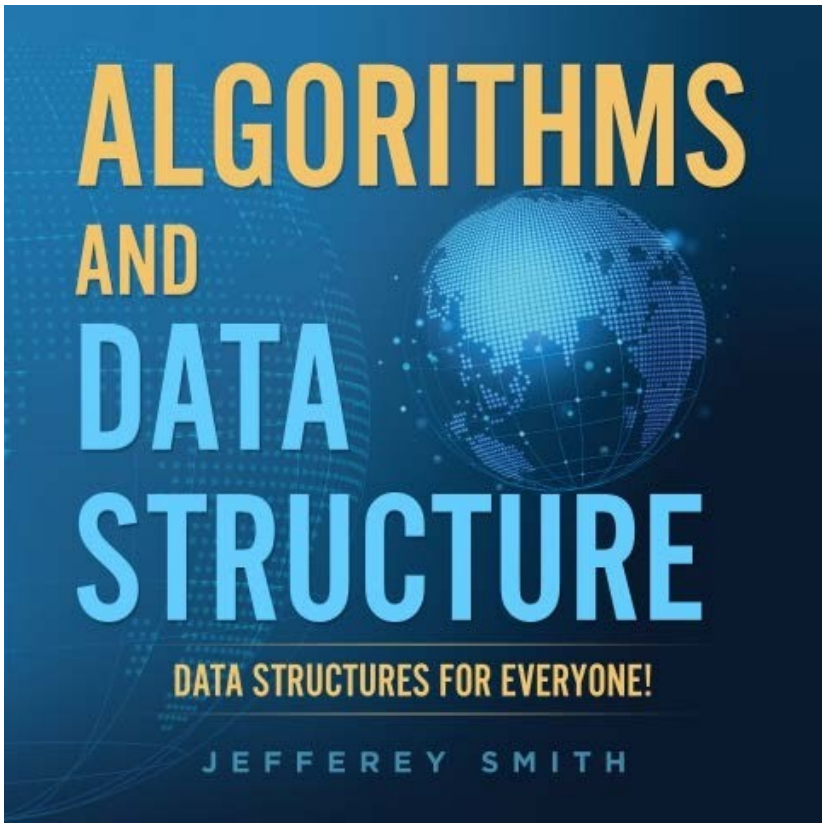
CLICK HERE TO DOWNLOAD YOUR AUDIOBOOK



CLICK HERE NOW TO DOANLOAD YOUR AUDIOBOOK



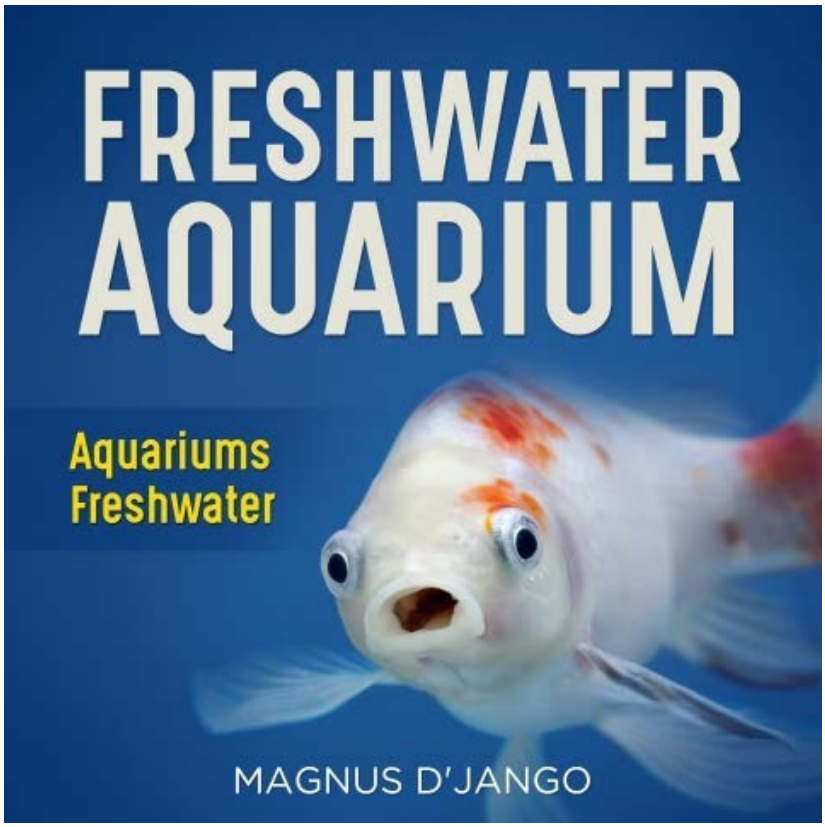
CLICK HERE NOW TO DOWNLOAD YOUR AUDIOBOOK



CLICK HERE NOW TO GET YOUR AUDIOBOOK.



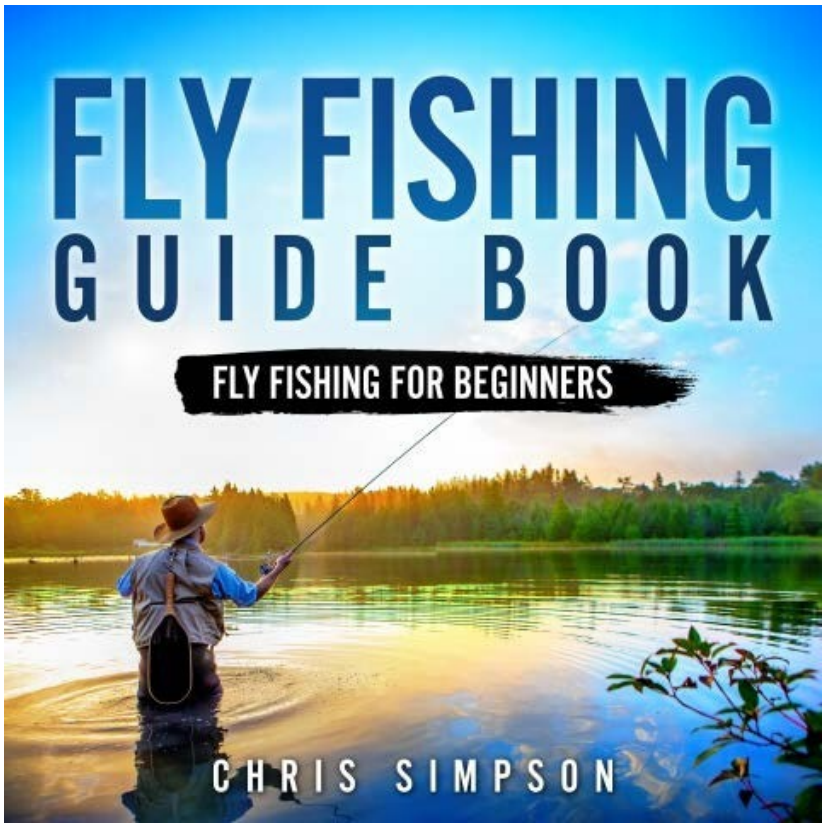
CLICK HERE NOW TO GET ACCESS TO YOUR AUDIOBOOK



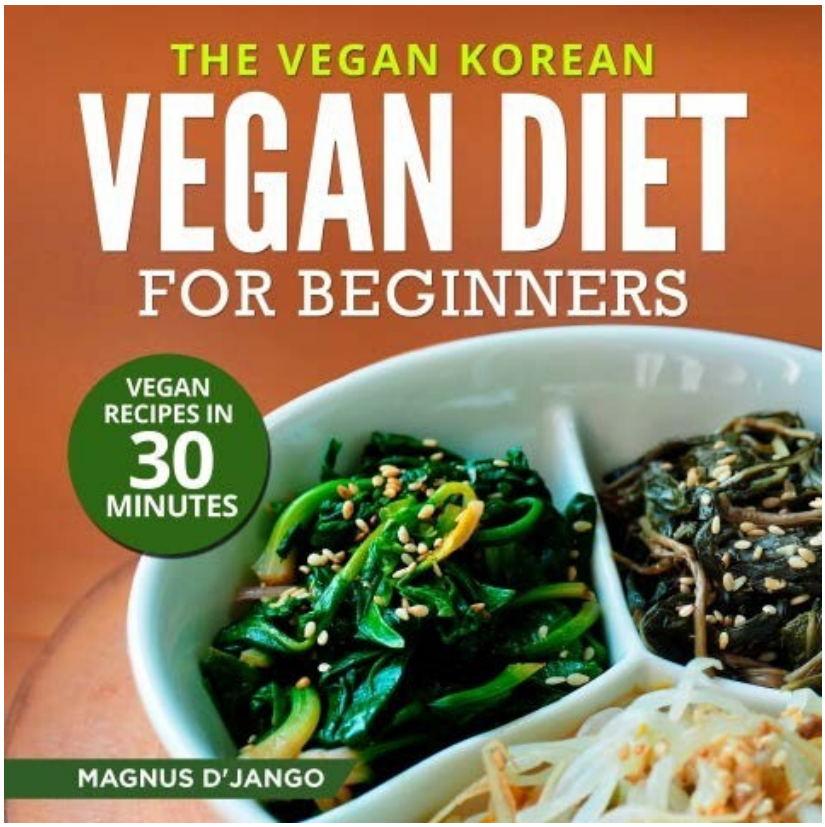
CLICK HERE NOW TO GET ACCESS TO YOUR AUDIOBOOK



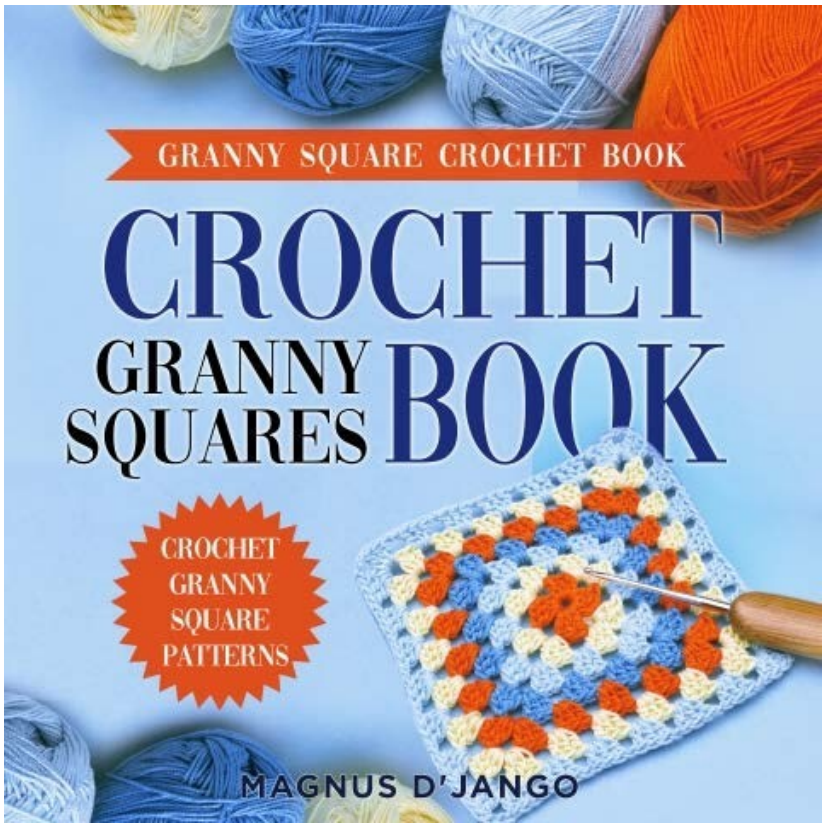
CLICK HERE NOW TO GET ACCESS TO YOUR AUDIOBOOK



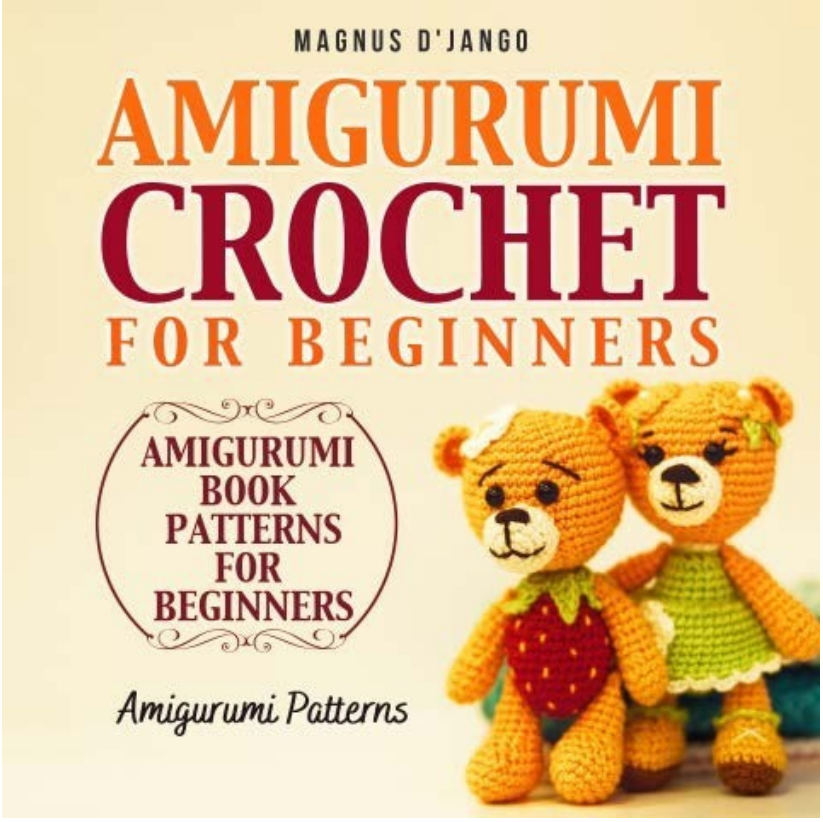
CLICK HERE NOW TO GET ACCESS TO YOUR AUDIOBOOK



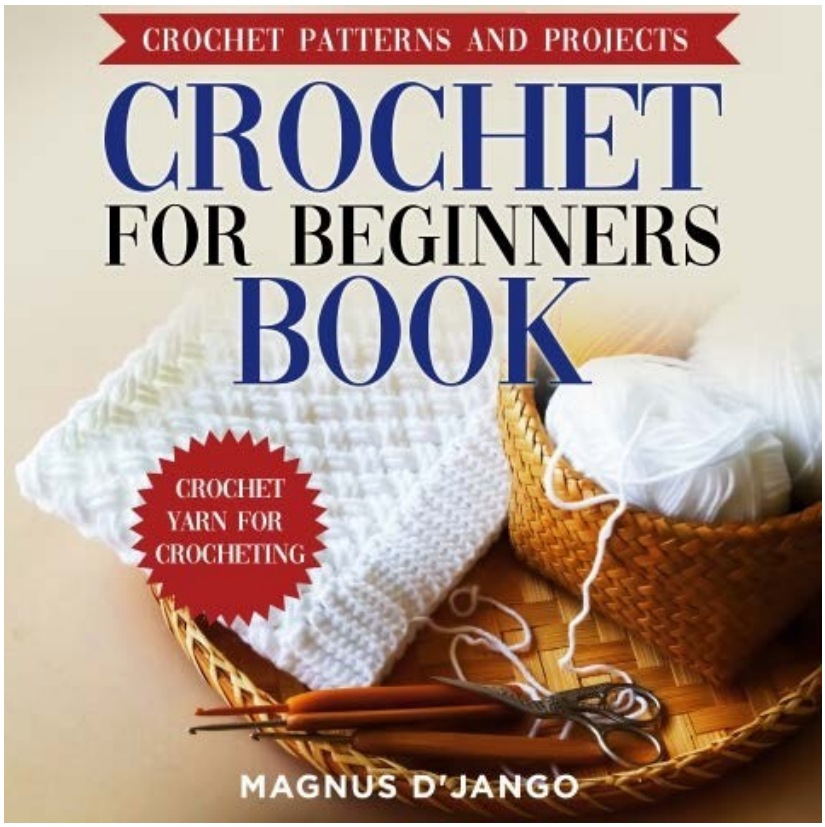
CLICK HERE NOW TO GET ACCESS TO YOUR AUDIOBOOK



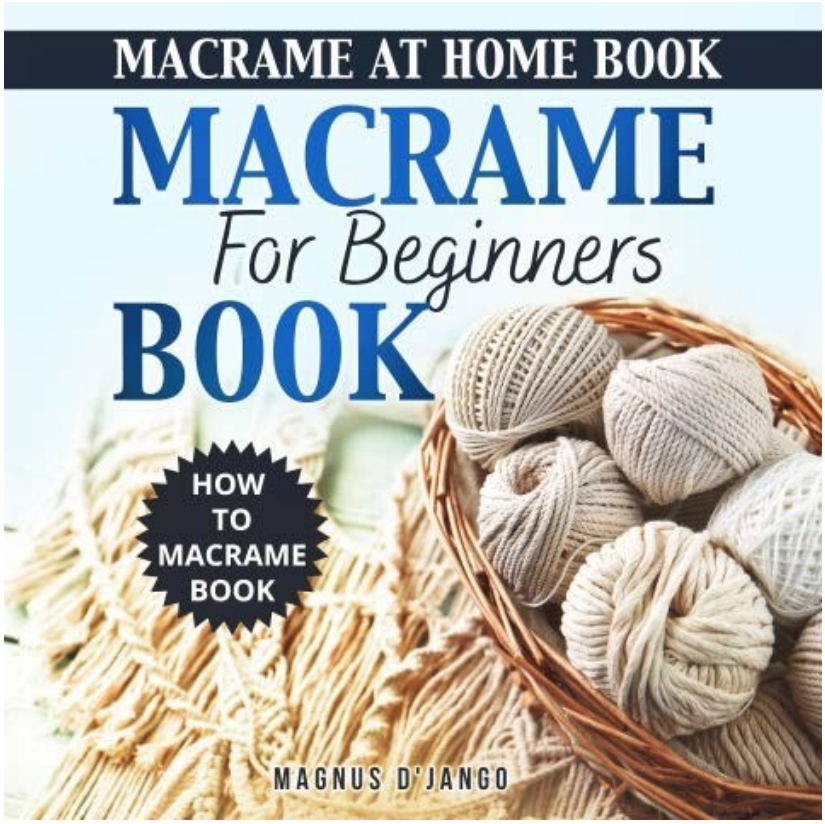
CLICK HERE TO GET ACCESS TO YOUR AUDIOBOOK



CLICK HERE NOW TO GET ACCESS TO YOUR AUDIOBOOK



CLICK HERE NOW TO GET ACCESS TO YOUR AUDIOBOOK



CLICK HERE NOW TO GET ACCESS TO YOUR AUDIOBOOK

References

Alexandra. (2012, November 16). *The new crochet: Christmas decoration (+ pattern)*. New Crochet. http://newcrochet.blogspot.com/2012/11/christmas-decoration.html

Cagle, K. (2021, March 7). *How to crochet a scrunchie*. Easy Crochet. https://easycrochet.com/crochet-hair-scrunchie-pattern/

Cagle, K. (2021, December 19). *How to crochet a scarf for beginners*. Easy Crochet. https://easycrochet.com/crochet-scarf-for-beginners/

C, M. (n.d.). *Lansing stashbuster infinity scarf & cowl free knitting pattern*. Stitch & Hustle. https://www.stitchandhustle.com/free-patterns/lansing-stashbuster-infinity-scarf-cowl-free-knitting-pattern

Dickerson, J. (2016, April 14). *Free crochet pattern...Botanical bookmark!* Fiber Flux Blog. https://www.fiberfluxblog.com/2016/04/free-crochet-patternbotanical-bookmark.html

Fukurou Crafts. *Fukuroucrafts: Pattern crochet cute owl doll, □□□ □□□□□□ □□□□□□ □□□ □□□□□□□ □□□□□ □□□□ □□□□□□*. (2015, February 17). Fukurou Crafts. https://fukuroucrafts.blogspot.com/2015/02/crochet-owl-doll-pattern-crochet-owl.html?m=0

Griggs, J. (2016, December 1). *You've seen these cozy blankets, now make one for yourself!* Crafty House. https://crafty.house/finger-crochet-diy/

Lucy. (2017, January 6). *Basic mustard beanie – Things we do blog*. Thethings-We-Do. https://www.thethings-we-do.com/basic-mustard-beanie-2/

patternprincess1. (2021, July 27). *Simple soap holder crochet pattern*. Pattern Princess. https://patternprincess.com/simple-soap-holder-crochet-pattern/

Solovay, A. (2021, June 25). *Make a sweet and simple crochet headband with this free pattern.* The Spruce Crafts.

https://www.thesprucecrafts.com/simple-crochet-headband-pattern-978852

Stearns, S. (2020, April 25). *Classic granny square pattern*. Sarah Maker. https://sarahmaker.com/crochet-a-granny-square/

Stearns, S. (2021, August 14). *Crochet can cozy*. Sarah Maker. https://sarahmaker.com/crochet-can-cozy/

Wilson, L. (2018, March 15). *Crochet scrap yarn rug*. Traverse Bay Crochet. https://traversebaycrochet.com/2018/03/15/easy-simple-beginner-crochet-rug-free-pattern-scrap-yarn/

Zimmerman, S. (2015, March 31). *Crochet flower pot hanging basket*. Repeat Crafter Me. https://www.repeatcrafterme.com/2015/03/crochet-flower-pot-hanging-basket.html#_a5y_p=3602077

Image References

Amilovesdesign. (2021, August 12). Yarn Threads Crochet. *Pixabay*. https://pixabay.com/photos/yarn-threads-crochet-hobby-craft-6541847/

Edz Norton. (2021, May 10). [Silver and Gold Rings on Brown Wooden Table]. *Unsplash*. https://unsplash.com/photos/9rStyyFGKQA

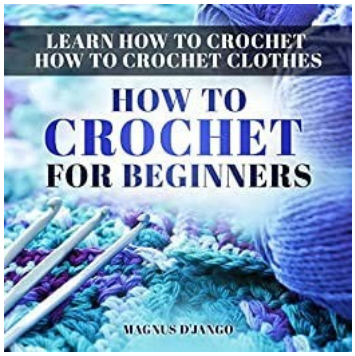
HBieser. (2015, March 14). Lake Constance Tree Promenade. *Pixabay*. https://pixabay.com/photos/lake-constance-tree-promenade-673103/

Lela. (2016, November 27). Toddler Sitting on Person's Lap While Person Sitting Also on Tree Trunk. *Pexels*. https://www.pexels.com/photo/toddler-sitting-on-person-s-lap-while-person-sitting-also-on-tree-trunk-295208/

MrsBrown. (2017, November 19). Toys Play Child. *Pixabay*. https://pixabay.com/photos/toys-play-child-beast-2962776/

Surene Palvie. (2018, August 22). Assorted-Color Yarn Lot. *Pexels*. https://www.pexels.com/photo/assorted-color-yarn-lot-2767689/

BONUS BOOK



Learn How To Crochet - How To Crochet Clothes

How To Crochet For Beginners

Magnus D'Jango

© Copyright 2021 - All rights reserved.

The content contained within this book may not be reproduced, duplicated or transmitted without direct written permission from the author or the publisher.

Under no circumstances will any blame or legal responsibility be held against the publisher, or author, for any damages, reparation, or monetary loss due to the information contained within this book, either directly or indirectly.

Legal Notice:

This book is copyright protected. It is only for personal use. You cannot amend, distribute, sell, use, quote or paraphrase any part, or the content within this book, without the consent of the author or publisher.

Disclaimer Notice:

Please note the information contained within this document is for educational and entertainment purposes only. All effort has been executed to present accurate, up to date, reliable, complete information. No warranties of any kind are declared or implied. Readers acknowledge that the author is not engaged in the rendering of legal, financial, medical or professional advice. The content within this book has been derived from various sources. Please consult a licensed professional before attempting any techniques outlined in this book.

By reading this document, the reader agrees that under no circumstances is the author responsible for any losses, direct or indirect, that are incurred as a result of the use of the information contained within this document, including, but not limited to, errors, omissions, or inaccuracies.

Table of Contents

Introduction

How To Crochet Clothes Crochet For Beginners

Basic Stitches

Chapter 1: How To Crochet Visually Online Options

Chapter 2: How To Crochet Hats

Chapter 3: How To Crochet A Baby Sweater Materials Construction Assembly

Chapter 4: How To Crochet Jewelry Friendship Bracelet

Crochet Flower & Chain Conclusion

References

Your FREE Gift

I have something for you. It won’t cost you a dime. This is a FREE 40-page PDF guide. I’d like you to have free copies with my compliments.

You can grab Your copy by clicking on the following link and joining my mailing list.

FREE BOOKS

https://imris-kdp-publishing.com/

Before we jump into this Book, I'd like to express my thanks. I realize there are many books on this subject that you could otherwise spend your time reading. You're taking a chance on me. I appreciate it. Giving you a FREE PDF copy of this amazing guide is my way of showing that appreciation.

On that note, let's dig in. You're going to love what's coming your way in the following pages.

Introduction

Crochet is an art form that is full of variety and potential. With any number of stitches, one can create blankets, baskets, clothes and jewelry. Many crafters report that crochet is faster than knitting, and this appeals to many who might be considering picking up the hook for the first time. Crochet has long been considered easier to learn than knitting since, at its most basic, it only requires one hook and a ball of yarn. Crochet is a hobby that doesn't discriminate. Anyone can do it. Men, women, young people, and older folks have all adopted crochet as a hobby. You don't need a lot of money to crochet, either. All you need is a hook and yarn, as many wonderful items have been made with a simple aluminum hook and sturdy acrylic yarn.

Crochet, as we know it at least, developed during the 16th century. Historical sources claim that crochet may be found in Italy as far back as the 1500's, but because fiber-based archaeological finds do not survive the ravages of time very well, it is difficult to find evidence of crochet in Europe before 1800.

One place that crochet made a real difference was in Ireland during the Great Potato Famine. Once Queen Victoria bought crocheted Irish lace from women in Ireland, everyone who was anyone had to have it too. Irish lace took off and the art of crochet saved many families in Ireland from further poverty and likely death by starvation. Schools were formed, teachers trained and scattered throughout the island, and crafters began designing their own patterns. As personal finances improved, people began to emigrate to America (Marks, 2019). Neighbors began to share patterns and it wasn't long before crochet took off in North America. Necessity is the mother of invention and it wasn't long before crochet made a mark in clothing, housewares and jewelry.

But what's with that hook? You might be surprised to know the humble crochet hook has been made of a wide range of materials. Aluminum is the most common material, the most sturdy, and the most cost effective for beginners. Historically the hook was made out of whatever crafters had on hand such as fossilized mammoth bone, porcupine quill, celluloid,

agate, ivory and bone. Very small hooks that measured 2mm or less were, and still are, made from steel. These days, handles can be covered with soft foam for increased comfort of crafters with joint pain. Handles have also been ornately carved, adding a unique and personal touch to a crafter's tool.

The crochet hook is a simple tool made up of a few important sections. We refer to the entire tool and the tip of it with the same word, "hook." The pointed tip is used to poke into stitches, or into the center of circles. This is the leading edge of your tool. It is the hook that captures the yarn. The yarn then sits in the throat, or the depression right behind the hook. The body of the crochet hook is made up of the shaft, the thumb rest, and the handle. To make things even more interesting, there are two kinds of crochet hooks: in-line and tapered. An in-line hook has a throat that is flatter and deeper than a tapered hook. If you find one kind too difficult to work with, try the other. There's no point in using a tool you don't like!

Hooks range from tiny to quite large! Hook size refers to the size of the head and the throat. Generally, crochet hooks are all the same length. The size of a hook is measured in millimeters, and directly impacts the size of your stitches and the size of whatever you are trying to make. While you are still getting used to the feel of the hook, it's recommended that you start out with a mid-sized hook, rather than one that is very small or too large and cumbersome. The best yarn to start with is a worsted weight acrylic yarn thick enough to be easy to work with and durable in the washing machine and dryer. This will stand up very well to the friction produced by pulling back stitches if the need arises. And as anyone who has ever picked up a crochet hook can tell you—occasionally pulling back stitches is a fact of (crochet) life! If you study the label on the yarn you have in mind (or the details of the yarn if you're shopping online), you'll find a recommended hook size that corresponds to that yarn.

How To Crochet Clothes

Many crafters decide they'd like to create their own clothes, but they want more choices than were available back in the days of the granny square

vest. While those vests can be quite attractive, there's nothing wrong with variety. It can be great fun to decide what to make, once you know who you want to make the garment for! Are you crocheting for an adult or a child? Or perhaps a pet? If you're crocheting for an adult, will your garment be for a man or a woman? We'll discuss some of the many options available, discuss stitches that might lend themselves more readily to a garment and a few other considerations you may want to think about before starting.

Crochet For Beginners

As with all other crafts, you'll want to learn the basics first and build your skills up from there. Most of the challenge of learning this wonderful craft will be in building your muscle memory. Holding the hook, manipulating the yarn, and learning how your item should look will all become easier with time. You need to practice these new skills as you learn them. Once you master one stitch, then the next, making one charming item after another, you'll build your skillset and your confidence. You've got this!



There are many hooks available on the market today, made from a wide range of materials. Bamboo, aluminum, nickel and plastic are among the

most sought-after. Some even have foam grips designed to help crafters with arthritis grip the hook more comfortably. It really comes down to trial and error, but you want to choose a hook that will allow the yarn to glide smoothly over the hook and shaft and not make you work harder than you have to. For this reason, many beginners start with an aluminum hook. Thankfully, hooks are relatively inexpensive, so one doesn’t need to break the bank to find a hook they like. Hooks are measured around the shaft, using millimeters. The millimeter measurement is then used to denote the hook's size—for example, a 4mm hook measures 4mm around the shaft. The number of millimeters also corresponds to a letter. For example, a 4mm is also a G/6.

Yarn is another material swimming with choices. There are so many to choose from that it can be overwhelming! Acrylic is a common choice, as is wool. Yarn is also made from alpaca, buffalo, bamboo, cotton, wool blends, Angora, and silk. You’ll want to start with a yarn that feels good in your hands. The more pleasant a yarn feels as it slides between your fingers, the more pleasure you’ll get as you learn. This is another material that you may want to experiment with to really get a good idea of your preferences. Yarn comes in

different thicknesses, commonly referred to as “weights.” Generally speaking, the higher the number assigned to a weight, the thicker it is. So a size 3 yarn would be a good weight to create baby items. A fine size 1 yarn would be ideal to make intricate wraps and shawls.

Basic Stitches

All directions in this book are written using US crochet terminology.

Chain (ch)

All chains start with a slipknot. Holding the yarn that leads to the ball in your left hand and your hook in your right, maneuver your hook under the yarn so that the yarn comes over the hook from back to front. Twist the hook so it captures the yarn and pull through the loop. You should have one loop on the hook. Repeat. After you’ve done a few of these, what you’ve created begins to resemble a chain: Hence the name of the stitch.

Foundation Row (FR)

A series of chain stitches into which all successive stitches are worked.

Slip Stitch (ss)

Start with one loop of yarn on the shaft of your hook. Push the tip of your hook under the “arms” of a previously worked stitch and swing the hook under the yarn. Twist to catch the yarn, and pull back toward you. There are two loops of yarn on your hook. Pull the left loop through the right, and you’ll have one loop on your hook!

Single Crochet (sc)

Start with one loop of yarn on your hook. Insert the tip of your hook into a previously worked stitch and swing the hook under the yarn. Capture the yarn and pull the hook toward you. There are two loops on the shaft of your hook. Catch the yarn again and pull through both loops on the shaft, leaving one loop on your hook.

Double Crochet (dc)

Start with a single loop on the shaft of your hook while holding the piece in your other hand. Position the hook so it passes under the yarn, capture the yarn, and twist so the yarn is wrapped around the shaft of the hook. You now have two loops of yarn on the shaft of the hook. Then push the hook into the middle of a previously worked stitch, capture

the yarn with your hook, and pull it back through the stitch. You should now have three loops on your hook. Capture the yarn with your hook and pull it through the first two loops. You now have two loops on your hook. Capture the yarn again and pull through both loops. You now have one loop on your hook and you've made a double crochet stitch.

Whip stitch (for sewing up)

Hold your two layers of fabric (in this case, crocheted pieces) with the two public sides, or "right sides" together. Thread a darning needle with yarn and push the needle through both pieces, from front to back. Bring your needle to the front and once again, push it from front to back. This is one whip stitch. Move your needle slightly to the right and repeat the steps. Do this for the length of the two pieces of fabric. The smaller and neater these stitches are, the less noticeable they'll be.

Chapter 1: How To Crochet Visually

There are a few different ways to learn a new craft, and one thing you need to keep in mind is that we do not all learn the same way. Some of us can read instructions, grasp the intended lesson and carry out the instructions we've just read. Others among us need to sit with a person who carries out the instructions and shows us by example, sometimes repeatedly. We learn best with visual and auditory components when we're trying to learn something new. We're the ones who would be best served by a real teacher. Once upon a time, everyone's grandmother or mother knew how to knit or crochet. It was easy to find someone to teach a new crafter. These days, that's not the case.

So how do we find someone to teach us how to crochet if there's no one in our family that can pass on the knowledge? Many libraries hold fiber craft evenings, and a call to your local librarian would quickly tell you if that was an option for you. If not, place a call to a nearby senior center. You might be able to find someone who would gladly share their crochet skills in exchange for a little company and conversation. You might just make a new friend this way, too! If that scenario isn't an option for you, visit a local yarn shop, referred to in crafting circles as LYS. Some shops have a bulletin board with names and telephone numbers of folks willing to teach newbies what they know. Many yarn shops offer lessons in either group or private settings. Again, a simple phone call will provide you with the information you seek. Sometimes, community colleges offer continuing education classes in a wide variety of hobbies. Your local college will have both a course guide and someone that could answer your questions. Another option would be an advertisement in your local newspaper or a job board. Is there a crochet guild in your town or city? They may have members that could teach you to crochet, or put you in contact with someone who could.

Online Options

Perhaps none of the options above appeal to you, or you don’t have easy geographic access to crocheters in your community. Not to worry, you still have options! If you have a phone that can access the internet, or if you have a computer or can borrow a friend’s, there are still ways to learn how to crochet. A simple query of your favorite search engine can reveal crochet websites, craft instruction blogs, or crochet guilds that have an online presence. Often designers will post tutorials in an effort to attract new fans. A simple online search might just demonstrate you have more choices than you had initially believed! You will

find more than one video sharing website, and on those, thousands of tutorial videos that can propel you along the path of being a proficient crocheter.

Chapter 2: How To Crochet Hats

You'll want to decide early on who you'll be crafting for. Sizing is more important with crochet than with knitting since the fabric you'll create has less stretch. We'll assume you're able to measure the head you intend to crochet for. You'll want to measure both circumference (the total measurement around the head, at temple height) and the height (from the ear to the top of the head, or crown). The measuring tape should be snug, but neither too tight or too loose. There are averages, and we'll list these below. Be aware, however, that one size does not fit all!

Baby (6 months +): circumference = 16" height = 6"

Child & youth: circumference = 18"-20" height =

7.25" - 7.50" Adult (small): circumference = 21"-22"

height = 8" - 8.50" Adult (large): circumference =

24"+ height = 9.25"

You want a hat to be snug, not so loose that it falls off or so tight that it pops off the wearer's head. This snugness is called "negative ease" and it's important to the fit of garments—from hats to socks to sweaters.

One significant element to the fit of the hat is your yarn. You'll want to choose a yarn that has some elasticity, so cotton alone is not a great choice for this. There are a number of acrylic yarns on the market that are sturdy, soft and have enough elasticity to be a good choice for a hat. As an added bonus, you can simply throw the hat in the washer and dryer if it gets dirty! The best choice is a yarn that's a little thicker, a worsted weight yarn for example, and a lighter color. You don't want to start off with a dark yarn that you might find difficult to see. Save the darker shades for after you're a little more experienced.

We're going to presume that you've chosen to use an aluminum hook for ease of use. For a hat crafted from a worsted weight yarn, you'll usually want to use a size H-8 hook. There are two ways to hold a hook, and

either way can feel foreign at first. Your hands will need to build “muscle memory,” so be patient with yourself. It might take some time to feel comfortable holding the hook, but try both holds and see which feels better for you.

The first hold is called a “knife hold” because you hold the hook as you would a steak knife.

The second option is called a “pencil hold” because the hook is held like a pencil when writing.

Okay, ready to move on? You’ll be creating a “preemie” sized hat with an approximate circumference of 10” and an approximate 4” height. Size, as long as it’s not too large, is

not as important for preemies, because they will grow into—and if all goes well, quickly

out of—the hat!

All crochet work starts with a slipknot, and the same is true here. Start with a slipknot and pull the tail of the yarn slightly to tighten it, but not so much that it can’t move on the shaft of the hook.

FR - Your first row of stitches is called the foundation row. In this example, which produces a preemie-sized hat, your foundation row will be four stitches. Holding the tail end of your slipknot, wrap the yarn that leads to your ball around the throat of the hook, from back to front. Then pull the hook toward you through the loop of the slipknot. You’ve done your first stitch! Now repeat that stitch four more times. It will look and feel awkward at first, but the beauty of crochet is that even if you somehow unravel your stitches, it’s easy to re-create them. As we’ve said before, practice, practice, practice.

Tip - *you don’t want to pull your stitches too tightly in crochet, as this makes maneuvering the hook through your stitches more work than it needs to be.*

Now, make a slip stitch to join the two ends of the row together. You should have a loose circle of stitches, with one loop on the hook.

R1: Chain 3 (ch 3). Next, we’ll make a double crochet stitch. Loop the yarn over the end of your hook, go through the circle, capture the yarn

and pull it back through the ring. Capture the yarn again and pull through two loops on your hook. Capture the yarn again and pull through all of the loops on your hook. Complete ten more double crochet stitches, putting 3 dc in each stitch from the foundation row. Your first round will consist of 12 dc stitches, including the chain of 3 stitches that began the row (which counts as your first dc stitch). End the round by making one slip stitch (sl st) to join the first and last stitches of the round. This is the end of your first round.

As you work, you'll need to be aware of the number of stitches you have so that you can shape your hat correctly. This is most easily done by marking the beginning of your row with a marker. A bobby pin or stitch marker does the job well, or even a piece of yarn in a contrasting color tied into the first stitch of the second row.

Now you're ready for the second round.

R2: Chain 3, then do 2 double crochet (dc) in each stitch of the circle. When you come all the way around, do a slip stitch (sl st) to the first dc to join it all together. Remember your chain 3 counts as your first dc of the round. You should have 24 stitches.

R3: Chain 3 again, then do 2 double crochet into the next stitch, then 1 dc in the stitch after that. Repeat this process all the way around your circle, alternating between putting 2 dc and 1 dc in each stitch until you're back at the beginning. Slip stitch (sl st) to your

first dc to join. The changeup in dc stitches provides the shaping. Now you'll have 36 stitches.

R4: This one's a little different, so pay attention. Chain 3 as you did in earlier rounds, do 2 dc into the next st and then 1 dc in each of the next 8 stitches.

Repeat the pattern of 2 dc into one stitch followed by 1 dc into the next 8 stitches four times. You will now have 40 stitches. Sl st to your first dc to join.

R5-11: This round is a bit simpler. Ch 3, then do 1 dc in each stitch, all the way around. As with all your other rounds, sl st to the first dc to join. You should still have 40 stitches.

R12: Work one entire round of double crochet, end with a slip stitch and weave in your ends securely.

You're done!

Chapter 3: How To Crochet A Baby Sweater

There is a great deal of choice when it comes to baby sweaters, but the main things to keep in mind are comfort and fit. One will directly influence the other, of course. If the sweater doesn't fit the child, they won't be comfortable. An uncomfortable baby will let you know by fussiness or crying, and you don't want that! If you're going to make baby clothes with crochet, there are some measurements you need to know. Keep in mind, however, that these are only averages. Babies vary in size!

If in doubt, it's a good rule of thumb to make the garment a little bigger. Babies will always grow, so a little more room is a good thing.

In handcrafted garments, whether made by knitting or crochet, one factor that plays into how your garment will fit is called "ease." A garment that has "negative ease," such as socks, will have a smaller dimension than the body part wearing it. If we stick with our sock example, this means that the sock is slightly smaller than the foot it's intended for. This is what keeps handmade socks from sliding down off the foot. In contrast, "positive ease" is larger than the body part wearing it. Going back to our baby sweater—2" of positive ease will give the baby room to wriggle and move in the sweater. You don't want a sweater for a baby to have negative ease!

The following pattern is for a sweater that will fit most newborn babies. The sweater should have two inches of positive ease, giving lots of room for the baby to be comfortable.

Abbreviations used in this pattern:

- **ch** = chain
- **ch #** = chain # stitches, # is the specified number of stitches
- **rep** = repeat
- **sc** = single crochet
- **sl st**= slip stitch

- **sp** = space
- **st(s)** = stitch(es)

Materials:

335 yards of worsted weight yarn of your choice. Check the yardage measurement on the ball of yarn to make sure you are purchasing enough; you may need to purchase two or even three balls, if they are sold in shorter yardage amounts. Whichever yarn you choose, the baby's parent(s) will likely be grateful if the garment can be thrown in the washer and dryer.

I- 9/5.5 mm

hook Four

buttons, 1/2"

A yarn or tapestry needle, for sewing up

Coilless safety pins or locking stitch markers (you'll understand when the sweater gets sewn up)

Construction

Back

Foundation Row (FR): Chain 27 stitches

Next: single crochet in 2nd chain from the hook, sc in each chain stitch to end. You should have 26 sts; turn work.

Next: ch 1, sc in each st to end—26 sts; turn work.

Repeat last row until the piece measures 8" high and fasten off.

Fronts - Make 2 of

these! FR: Chain 14

stitches.

Next: Single crochet in 2nd chain from the hook, sc in each chain stitch to the end. You should have 13 sts. Turn your work.

Next: Chain 1, sc in each stitch to the end. You should still have 13 sts. Turn work.

Repeat the previous row until the piece measures 8” long. Fasten off and set aside.

You’re getting there! Now make the other front, following the same instructions. You’ve got this!

Sleeves - Make 2 of these as well!

FR: Chain 22 stitches.

Next: 1 single crochet in 2nd chain from hook, sc in each chain st to the end. You should have 20 stitches. Turn your work.

Next: Chain 1, sc in each st to end. There should be 20 stitches. Turn the work.

Repeat the previous row until the piece measures 6-¾” high. Fasten off and set aside.

Assembly:

To put all your pieces together, you’ll sew the front and back shoulder seams together first. Then the upper arms get sewn to the shoulder seams. Here’s how we’ll do that.

The front and back pieces were made from the bottom up, so the chain edge is your bottom of all pieces. Lay the back and one of the front pieces together, “right” sides together. (The beauty of single crochet is that it has no “right” or “wrong” side, so it doesn’t matter which face you show to the world!) Thread your tapestry needle with about 10” of yarn and beginning at the sleeve end of one shoulder, whip stitch across to about an inch away from the neck opening. Repeat on the other side of the back with the other front piece.

Now we’ll sew the sleeves on.

Your chain edge on the sleeves is the cuff hem, and where you fastened

off is the shoulder. Fold the sleeve in half. The side that is currently facing out will eventually become the inside of the sleeve. Now take the sleeve and center the fold at the top of the sleeve on one side of the garment. Half of the sleeve top will be sewn to the back and half to the front. Pin these pieces together with the coilless safety pins (or locking stitch markers) that I mentioned in the materials list.

Now thread your tapestry needle with about 15” of yarn. Whip stitch the top of the sleeve to the body pieces. Repeat on the other side of the sweater with the other sleeve.

Next we’ll sew up the sides. All your pieces should be “inside out,” or “wrong side out.” Thread your needle once more with approximately 20” of yarn and whip stitch the side of the sweater (the back and one front piece) from hem to armpit, then continue sewing the two sleeve edges together from armpit to wrist.

Tip - *If you find 20” of yarn too much, not to worry! Use half that length. But if you choose this route, sew from the armpit to the hem, then down the sleeve from underarm to cuff.*

Now repeat on the other side, and turn the sweater right side out.

To keep the collar flat, fold the collar edges down from the corner and sew in place.

Now space your buttons evenly along the edge of one of your front pieces. If this sweater is intended for a boy, the buttons should be on the right front piece. If the sweater is for a girl, the buttons will go on the left front piece. You won’t need a button band for buttonholes. The buttons should slide through the stitches on the opposite piece.

You did it! Well done!

Chapter 4: How To Crochet Jewelry

It’s no surprise that crochet can be found in jewelry as well. Crafters of all ages and skill levels can produce wearable art. Below are a couple of examples.

Friendship Bracelet

This bracelet is easy to make, versatile and can be made in a variety of colors. It’s made with crochet thread, which is a specially formulated thread used for crochet. You’ll need a length of crochet thread and a B-1/2.25 mm crochet hook. Depending on the recipient, you shouldn’t need more than 9-10” of thread. The average 8 year old girl’s wrist is about 6,” while an adult woman’s wrist can be up to 8” in circumference. You could measure the recipient’s wrist to be certain of how much crochet thread you’ll need.

R1: Leaving a long tail, chain 3, work 1 sc into the 2nd chain, then again in the next 2 chains. Chain 1. Turn the work.

R2: Work 1 sc in each of the stitches. Chain 1, then turn your work. You should still have 3 stitches.

R3: Work 1 sc in each of the 3 sts, chain 1, turn work.

R4: Repeat R3 until your bracelet measures the length you want. Fasten off, leaving another long tail. Now you can use your long ends to fasten your bracelet around your friend’s wrist!

Crochet Flower & Chain

For this project, you’ll need:

- US G-6/4mm crochet hook

- a heavy bead or button for the center of your flower
- approximately 6-10 yards of any sport weight yarn you like best

Chain 82. Now chain 8 more for your flower. Put a stitch marker here to easily find where the stitches for the flower start.

Turn your work and put a slip stitch into the 8th chain from the hook. Chain 3, turn the work and slip stitch into the center of that ch 8 ring you just made. Repeat 4 times. This will create five small flower petal bases around the edge of the center circle.

Chain 3, turn your work, and put 3 dc stitches into the space made by one of the ch-3 from the previous step. Chain 3 more, and sl st to the same space where you worked the dc. Now you have your first full flower petal.

Now repeat the pattern to [ch 1, sl st to next chain 3 sp, ch 3, 3 dc in same ch-3 sp, ch 3, sl st to same ch-3 sp] 4 times. This will create the other petals for your flower.

Chain 82 for the other side of the necklace. Now fasten the heavy bead or button in the center of the flower.

Voila! You’ve made a necklace!

Conclusion

Crochet is a hobby that is fun, accessible, and full of potential. It can be done by anyone who can hold a hook and yarn, of almost any age, of any income, and certainly of any gender. There's never a fear of plateau or boredom either, as there's always something more to learn, or a new pattern to try, or a new yarn that might be perfect to bring your project to life.

But crochet can bring more to your life than simply an answer to boredom. It has also been shown to be good for your mental health and improve memory and concentration. As you craft each stitch, you need to be aware of where your hook is going, how the yarn is behaving, and how the overall project is coming together. Mindfulness at its best!

The repetitiveness of the actions has its own benefits as well, once the muscle memory has been developed. It's this predictability that helps to lower your blood pressure and provide a sense of calm. When we're stressed, our bodies produce an excess of cortisol which can be disastrous to our circulatory system. So it's in our best interests to regulate it as much as possible. Crochet can help do that.

Once you're comfortable with the basic stitches used to complete the projects in this guide, you can go on to learn the treble stitch, the half double crochet, the shell stitches, or perhaps even make a blanket with the garnet stitch.

Hopefully you can see the potential of crochet as a pastime and a link to the past. You can clothe your loved ones, keep them warm, and make some pretty nifty jewelry as well. From the foundation stitches, you can build your crochet skills and move onto making socks, mittens, blankets, adult sweaters, and so much more! There are a number of items that can benefit your pets too. Did you know that you could crochet a chicken sweater? It's true! It's not just dogs and cats that can be well-dressed. Many crafters are also turning their skills to making baskets and beds for animals in shelters as well. The world of crochet is full of potential!

May your hook never snag and your yarn never tangle.

Your FREE Gift

You can grab Your copy by clicking on the following link and joining my mailing list.

FREE BOOKS

https://imris-kdp-publishing.com/

ACKNOWLEDGEMENTS

Special Thanks to our ARC Team.

- CarolAnn Williams