

The Swift Programming Language Guide (5.8) - (SPLG)

2023-09-14 - A Software Developer self-study Guide: by the SudoKoders - AWS & KSM
- with Data Structures & Algorithms and Math for Computer/Data Science.

I: Swift Basics

- [] 01) Types, Operations, Expressions, Constants & Variables
- [] 02) Control Flow (Conditionals)
- [] 03) Functions.
- [] 04) Optionals

II: Collection Types

- [] 05) Arrays, Dictionaries & Sets
- [] 06) Advanced Control Flow
- [] 07) Advanced Functions
- [] 08) Closures.
- [] 09) Strings

III: Creating (your own) Types - (Enums, Structures, Classes, Actors & Protocols)

- [] 10) Enumerations. (Do Kodeco Mini Exercises & Challenges after 14) Methods)
- [] 11) Structures & Classes
- [] 12) Structures
- [] 13) Properties
- [] 14) Methods
- [] 15) Classes
- [] 16) Inheritance
- [] 17) Initialization
- [] 18) Deinitialization
- [] 19) Type Casting
- [] 20) Nested Types
- [] 21) Protocols
- [] 22) Equatable, Comparable, & Hashable
- [] 23) Generics
- [] 24) Structs & Classes (BNRG) - building Xcode projects

IV: Advanced Swift

- [] 25) Access Control, Code Organisation & Testing
- [] 26) Custom Operators, Subscripts (docs.swift), & Keypaths
- [] 27) Extensions
- [] 28) Pattern Matching
- [] 29) Error Handling
- [] 30) Optional Chaining
- [] 31) Encoding & Decoding Types
- [] 32) Memory Management, ARC & Memory Safety
- [] 33) (Value vs Reference) Types & Semantics
- [] 34) Property Wrappers
- [] 35) Protocol-Oriented Programming
- [] 36) Protocol Extensions
- [] 37) Advanced Protocols & Generics
- [] 38) Opaque & Boxed Types
- [] 39) Advanced Operators
- [] 40) Concurrency
- [] 41) Macros
- [] 42) Result Builders
- [] 43) Regex

V: Language Reference

- [] 44) Language References

VI: Writing Applications

- [] 45) Command line Utilities
- [] 46) Apple (iOS, macOS, iPadOS, watchOS, tvOS, visionOS, and CarPlay) - Apps
Linux (Ubuntu, CentOS, Amazon Linux (AWS), Red Hat) and Windows 10 - Apps

1. >>> Index: eBooks & Documentations to Download.

Chapters of the 3 eBooks below, have been combined for the Language Guide on page 1.

> 1. Kodeco - Swift Apprentice (\$60) - eBook -> <https://www.kodeco.com/books/swift-apprentice>

> 2. Big Nerd Ranch Guide - Swift Programming 3e - Ward (\$32) 2020 - eBook -> https://www.amazon.com/Swift-Programming-Ranch-Guide-Guides-ebook/dp/B08PVXFKFN/ref=tmm_kin_swatch_0?_encoding=UTF8&qid=1680308910&sr=8-4

>. 3. Docs.Swift - Swift Programming Language (5,9) (free) - eBook -> <https://docs.swift.org/swift-book/documentation/the-swift-programming-language/thebasics/>

Docs.Swift PDF appdissected version -> <https://www.appdissected.com/wp-content/uploads/2022/11/SwiftProgrammingLanguage57.pdf>

Note: This is Swift version (5.7), so expect changes with new updates to Swift, with time. New topics may get added to these eBooks, which may differ to those in this guide.

Which Format is best - ePub vs PDF vs MOBI (Amazon Kindle) for eBooks ?

- ePub - Best for eBooks - (if possible) - always download your eBooks in this format.
- MOBI (Kindle)- In 2022 this format was abandoned. But Kindle can now also read ePub.
- PDF - is a fixed format for printing & graphics, not for eBook reading. Convert to ePub !
- > Example Software for Converting PDF to ePub, (if possible), without losing format ->
- 1. Calibre - (free) -> <https://calibre-ebook.com>
- 2. Aiseesoft -> <https://www.aiseesoft.com/pdf-to-epub-converter/> - etc...

>> To use the **Swift Programming Language Guide (SPLG)** on (page 1) above with the eBooks chapters below on pages 3, 4 and 5 ->

The index number at the end of each chapter eg. (02) in the eBooks below, refers to -> Chapter in - (Swift Programming Language Guide) page 1 above -> Chapter in - (SPLG)

Meaning when you start, eg. Chapter - 02) Control Flow (Conditionals) - in (SPLG) above, you read the corresponding chapters in the eBooks below in this order ->

1. Kodeco - Swift Apprentice (page 3) -> - [] 03) Basic Control Flow - (02)
- [] 04) Advanced Control Flow - (02)
2. BNRG - Swift Programming 7e (page 4) -> - [] 03) Conditionals. - (02)
- [] 06) Loops. - (02)
- [] 05) Switch. - (02)
3. Docs.Swift.org - The Swift Programming Language (page 5) ->
- [] 02) Basic Operators - Comparison Operators -> Exp. Parentheses - (02)

Zotero (free) for organising your research & PDF Documents -> <https://www.zotero.org>
Obsidian (free) for writing Markdown Docs, with Zotero integration -> <https://obsidian.md>

1. Kodeco - Swift Apprentice 2021 - iBooks - use the eBook reader App

I: Basics **Chapter in - (SPLG) above (page 1)**

- [] 01) Expressions, Constants & Variables - (01)
- [] 02) Types & Operations - (01)
- [] 03) Basic Control Flow - (02)
- [] 04) Advanced Control Flow - (02)
- [] 05) Functions - (03)
- [] 06) Optionals - (04)

II: Collection Types

- [] 07) Arrays, Dictionaries & Sets. - (05)
- [] 08) Closures - (08)
- [] 09) Strings - (09)
- []) Regex. - (43)

III: Creating your own Types

- [] 10) Structures - (12)
- [] 11) Properties. - (13)
- [] 12) Methods - (14)
- [] 13) Classes. - (15)
- [] 14) Advanced Classes (15)
- [] 15) Enumerations. - (10)
- [] 16) Protocols - (21)
- [] 17) Generics. - (23)

IV: Advanced Topics

- [] 18) Access Control, Code Organisation & Testing - (25)
- [] 19) Custom Operators, Subscripts & Keypaths. - (26)
- [] 20) Result Builders. - (42)
- [] 21) Pattern Matching. - (28)
- [] 22) Error Handling. - (29)
- [] 23) Encoding & Decoding Types. - (31)
- [] 24) Memory Management. - (32)
- [] 25) Value Types & Reference Types - (33)
- [] 26) Property Wrappers. - (34)
- [] 27) Protocol-Oriented Programming. - (35)
- [] 28) Advanced Protocols & Generics - (37)
- [] 29) Concurrency. - (40)
- [] 30) Conclusion

2. BNRG - The Big Nerd Ranch Guide - Swift Programming - Kindle eBook reader App

I: Getting Started **Chapter in - (SPLG) above** (page 1)

- [] 01) Getting Started
- [] 02) Types, Constants & Variables. - (01)

II: The Basics

- [] 03) Conditionals. - (02) - 1
- [] 04) Numbers. - (01)
- [] 05) Switch. - (02) - 3
- [] 06) Loops. - (02) - 2
- [] 07) Strings. - (09)

III: Collections & Functions

- [] 08) Arrays. - (05)
- [] 09) Optionals. - (04)
- [] 10) Dictionaries. - (05)
- [] 11) Sets. - (05)
- [] 12) Functions. - (07)
- [] 13) Closures. - (08)

IV: Enumerations, Structures, and Classes

- [] 14) Enumerations. - (10)
- [] 15) Structs and Classes. - (11) - (24) - building Xcode project
- [] 16) Properties - (13)
- [] 17) Initialization. - (17)
- [] 18) Value vs Reference Types. - (33)

V: Advanced Swift

- [] 19) Protocols. - (21)
- [] 20) Extensions - (27)
- [] 21) Generics. - (23)
- [] 22) Protocol Extensions. - (36)
- [] 23) Error Handling - (29)
- [] 24) Memory Management and ARC - (32)
- [] 25) Equatable, Comparable, and Hashable - (22)
- [] 26) Property Wrappers - (34)

VI: Writing Applications

- [] 27) Command line Utilities - (45)
- [] 28) iOS and macOS Apps - (46)
- [] 29) Conclusion

3. Docs Swift org - The Swift Programming Language - iBooks - use eBook reader App

Language Guide

Chapter in - (SPLG) above (page 1)

- [] 01) The Basics - Constants & Variables -> Tuples. - (01)
 - Optionals -> Implicitly Unwrapped Optionals - (04)
 - Error Handling -> Enforcing Preconditions - (29)
- [] 02) Basic Operators - Terminology -> Comp. Assignment Operators. - (01)
 - Comparison Operators -> Exp. Parentheses. - (02)
- [] 03) Strings & Characters - (09)
- [] 04) Collection Types - (05)
- [] 05) Control Flow - (06)
- [] 06) Functions - (07)
- [] 07) Closures - (08)
- [] 08) Enumerations - (10)
- [] 09) Structure & Classes - (11)
- [] 10) Properties - (13)
- [] 11) Methods - (14)
- [] 12) Subscripts - (26)
- [] 13) Inheritance - (16)
- [] 14) Initialization - (17)
- [] 15) Deinitialization - (18)
- [] 16) Optional Chaining - (30)
- [] 17) Error Handling - (29)
- [] 18) Concurrency - (40)
- [] 19) Macros - (41)
- [] 20) Type Casting - (19)
- [] 21) Nested Types - (20)
- [] 22) Extensions - (27)
- [] 23) Protocols - (21)
- [] 24) Generics - (23)
- [] 25) Opaque and Boxed Types - (38)
- [] 26) Automatic Reference Counting (ARC) - (32)
- [] 27) Memory Safety - (32)
- [] 28) Access Control - (25)
- [] 29) Advanced Operators - (39)

Language Reference - (44)

- [] 30) About Language Reference
- [] 31) Lexical Structure
- [] 32) Types
- [] 33) Expressions.
- [] 34) Statements
- [] 35) Declarations
- [] 36) Attributes
- [] 37) Patterns
- [] 38) Generic Parameters and Arguments
- [] 39) Summary of the Grammar

2. >>>. Swift self study resources & tools - for Software Developers

>> About the Swift Language & Documentations -> <https://www.swift.org>

> Swift Developers -> <https://developer.apple.com/swift/resources/>

>> Tools ->

1> M2 or M1 - Mac Mini, Mac Studio or MacBooks (Max 3yrs old) when buying.

2> External Keyboard (US International PC),

3> PC Mouse, 4> PC Monitor/s (27"- 32"),

5> 3 External Hard Disks 2TB, USB-3.0 or Thunderbolt, for 3 Backups :-

2 Backups of your (computer & work & study documents) Weekly, in case of disk failure

1 Backup stored far away (friend's house) for safety, in case of (damage or theft or fire).

> Big investment of (\$2000+) but these will Also be your working tools 7+yrs after studies. (But an old Mac at Max 8yrs (\$400) can be used for studies, while saving for a new Mac)

- Fast Internet - at least 15GB monthly, for Online courses and resources.

(Fast Internet needed for 1-3 months to download Software, eBooks & Videos).

> To install Swift on macOS -> register an Apple ID account and download Xcode from App Store or for an older macOS, download from -> <https://xcodereleases.com> and click on the latest Xcode for your macOS version.

>> About Swift on Windows -> <https://www.swift.org/blog/swift-on-windows/>

> To install Swift on Windows and Linux -> <https://www.swift.org>. click on download.

- Video instructions -> <https://www.youtube.com/watch?v=wA3UjFCYvD8>

>> Software IDE or (Code Editor)

Platform.

1. Xcode. (free)

macOS

2. Visual Studio 2022 (\$550).

Windows, macOS

3. Visual Studio Code (free).

Windows, Linux, macOS

- To Download Visual Studio Code with instructions -> <https://github.com/pwsacademy/swift-setup/blob/main/editors/vscode-windows/README.md>

- Or else Download Visual Studio Code -> <https://code.visualstudio.com/download>

- Or else Download Swift Extension for Visual Studio Code ->

<https://www.swift.org/blog/vscode-extension/>

And Click on -> Swift Extension for Visual Studio Code

> Understanding IDE's -> https://techreviewer.co/blog/best-ide-for-swift-developers?trk=organization_guest_main-feed-card_feed-article-content

>> ePUB (& PDF) - eBook Reader Apps :

Platform

1. Calibre, AppleBooks, Kavita, FBReader, Epubor, Kindle, Zotero, etc..

macOS

2. Calibre, Kavita, FBReader, Epubor, Kitaboo, Kindle, Zotero, etc..

Windows

>> Github -> <https://github.com/Sudokoders/Swift-Guide>

>> Replit -> <https://replit.com/team/SwiftKoding>

>> Kodeco Forums -> <https://forums.kodeco.com>

Top 10 Swift Help forums & discussions. -> https://blog.feedspot.com/swift_forums/

3. >>> Self-studying progression Guide to become a Software Developer

1) IT - CompTIA Certification, eBook -> A+ (\$40), Network+ (\$46), Mike Meyers (Amazon)

2) A Language Course -> Swift, Rust, Kotlin, C#, Go, Dart, Typescript, Julia, Python, R, Javascript, C/C++, Java, Assembly, Scala, Haskell, Clojure, Nim, Mojo, F#, Matlab, etc..

3a) Data Structures & Algorithms Course ->

> Data Structures - Arrays, Stacks, Queues, HashTables, Graphs, Linked Lists, Trees, etc

> Algorithms - Efficiency, Searching, Sorting, Recursion, Dynamic Programming, etc

1 - Kodeco - Data Structures & Algorithms in Swift 2021(\$60) -> (Only for Swift Language)

2 - Common-Sense Guide DataStructures & Algorithms 2e (Python) - Jay Wengrow (\$29)

3 - Grokking Algorithms - Aditya Bhargava (\$27) -> Algorithms in Motion - Beau Carnes

4 - Computer Science Intro-Lectures Online 2021 (free) - CS50x Havard University ->

- <https://www.youtube.com/watch?v=8mAITcNt710>

3b) (Advanced) Applied Math Progression course for Computer Science-> (Optional)

>> Applied Math topics for Data Structures & Algorithms ->

<https://www.enjoymathematics.com/blog/math-for-data-structures-and-algorithms>

> High School Math (A-Level) - 1-Arithmetics, 2-Algebra, 3-Geometry, 4-Trigonometry.

> College Math - 5-Discrete Math (Logic, Number Theory, Proofs, Sets, Counting, Graphs)

6-Calculus I & II, 7-Linear Algebra, 8-Calculus III, 9-Probability & Statistics.

1 Math (free) PDF -> https://nongnu.askapache.com/fhsst/Mathematics_Grade_10-12.pdf

2 Algebra & Trigonometry + (Analytic Geometry) 2e - OpenStax (free) PDF ->

<https://openstax.org/details/books/algebra-and-trigonometry-2e>

3 Discrete Mathematics w Applications - Sussana Epp (\$66) -> <https://www.cengage.uk/c/discrete-mathematics-with-applications-metric-edition-5e-epp/9780357114087/>

4 Elementary Linear Algebra w Applications EMEA 12e - Anton, Rorres 2020 (\$53)

5 Linear Algebra: Theory, Intuition, Code (Matlab - Python) - Mike Cohen 2021 (\$30)

6 Calculus 8e - James Stewart 2016 (\$82). 7 Calculus 9e - Larson, Edwards 2010 (\$75)

8 Calculus + Diff-Equations - Paul's Online Notes (free) -> <https://tutorial.math.lamar.edu>

9 Probability & Statistics for Computer Scientists 3e (R - Matlab) - M Baron 2019 (\$98)

3c) (Advanced) Data Structures & Algorithms Course -> (requires 3b Math) (Optional)

1 Algorithms 4e (Java)- Robert Sedgewick (\$72) -> <https://algs4.cs.princeton.edu/home/>

2 Algorithm Design (Pseudo Code) - Kleinberg, Tardos 2006 (\$75)

3 Introduction to Algorithms 4e (Pseudo Code) - Thomas Cormen 2022 (\$85) ->

- [https://dl.ebooksworld.ir/books/Introduction.to.Algorithms.](https://dl.ebooksworld.ir/books/Introduction.to.Algorithms.4th.Leiserson.Stein.Rivest.Cormen.MIT.Press.9780262046305.EBooksWorld.ir.pdf)

4th.Leiserson.Stein.Rivest.Cormen.MIT.Press.9780262046305.EBooksWorld.ir.pdf

4 Algorithm Design Manual 3e (C), Steven Skeina 2020 (\$56) -> <https://www.algorist.com>

5 Data Structures & Algorithms made easy 5e (Java) - Karumanchi 2016 (\$31)

3d) (Advanced) Computer Science topics -> Online (free) (Optional)

1 Teach yourself Computer Science topics - <https://teachyourselfcs.com>.

2 OpenSourceComputerScience - <https://github.com/mvillaloboz/open-source-cs-degree>

3 OSSU - <https://github.com/ossu/computer-science#introduction-to-programming>

4 Functional Computer Science - <https://functionalcs.github.io/curriculum/>

4) DevOps Courses -> Git (Gitlab, Github), Kubernetes, Gradle, Jenkins, Docker, Scrum,

1. - Kodeco - Git Apprentice 2. - Kodeco - Advanced Git

3. Video - https://youtube.com/playlist?list=PLI6nIHwoZMVfivf4PwkD1b7UT_TyEd-U

>> Then Choose Platform and/or Framework Courses

5a) Front End (Web) -> Svelte, React, Vue, ASP.Net, WebAssembly, Bootstrap, UXDesign,
1. - Swift WebAssembly. - SwiftWasm -> <https://swiftwasm.org>

5b) Client Side -> Android Studio, SwiftUI, .NET Core, Flutter, Xamarin, React, etc...

- 1 - Kodeco - Functional Programming in Swift 2 - Kodeco - Expert Swift
- 3 - Kodeco - SwiftUI by Tutorials 4 - Kodeco - Swift Data
- 5 - Kodeco - Design Patterns by Tutorials ? 6 - Kodeco - URLSession Tutorial.
- 7 - Kodeco - Modern Concurrency in Swift. 8 - Kodeco - App Design Apprentice.
- 9 - Developing Apps, SwiftUI iOS Lectures 2021 (free) - CS193p Stanford University ->
Online - <https://cs193p.sites.stanford.edu>
- 10 - Kodeco - SwiftUI Animation by Tutorials 11 - Kodeco - Adv. iOS App Architecture ?

6a) Back End (IT) -> Django, Echo, Rails, Spring, NodeJS, ExpressJS, Rocket, Vapor, etc
> Courses - Servers+, CCNA, Databases, NewSQL, API, Parallel/Distributed Systems, etc
> Servers - Linux+, Windows Server, Mac Server, SystemAdministrator, Package Manager
1 - Kodeco - SwiftNIO 2 Tutorial 2 - Kodeco.- Server-Side Swift with Vapor.
3 - Vapor 4 - <https://www.youtube.com/watch?v=RhYZh3rTMKU> + Vapor 4 Lesson 6
4 - Swift Package Manager -> <https://www.swift.org/package-manager/>

6b) Cloud Platforms -> Amazon AWS, Microsoft Azure, Google Cloud, IBM Bluemix, etc.
> Courses - (IaaS, PaaS, SaaS), Cloud Servers, Blockchain, Monolithic & Microservices,
Virtualization & Containerization. + AWS Amplify, Azure App., Google Firebase, etc..
1 - Swift Apps on AWS -> <https://aws.amazon.com/developer/language/swift/>

7) App Dev. -> App Store, Google Play, MS Play, Front End, Back End, IoT, Embedded,
1 - Kodeco - An App from Scratch 2 - Kodeco - Polishing the App
3 - Kodeco - iOS Test-Driven Dev. Tutorials 4 - Kodeco - Real-World iOS by Tutorials
5 - Videos - <https://youtube.com/playlist?list=PLI6nIHwoZMVGybuZBNoylkD30W7jYDjAZ>

8) Cyber Security -> Malware, APT, Phishing, Ransomware, Password, SQL Injection, etc
> Penetration Tests - White Box, Grey Box, Black Box - (Network, Application, Workflow)
> Attacks - Web Apps, Client Side, Networks, Wireless, Social Engineering, Physical, etc..
> Certifications - (1. A+ 2.CCNA 3.Linux+ 4.Security+ 5.eJPT 6.SSCP 7.OSCP 8.CISSP,)
> CTF Training - (1 Picot CTF, 2 Root-me, 3 TryHackMe, 4 HackTheBox, 5 BattleHack etc)
> Books - <https://www.intelligent.com/best-cyber-security-books/>

9) Developer Jobs @ -> Companies, Freelancer, StartUps, Indie Developer, etc...

- 1 - Cracking the Coding Interview 6e - Gayle L McDowell ->
- <https://www.crackingthecodinginterview.com>
- 2 System Design Interview Vol.1+2 - Alex Xu -> <https://www.byte-by-byte.com/resources/>
- 3 - Kodeco - Hack Your Job Search.
- > Code Challenge websites -> (Requires - 2 Language, (3a,3b,3c) Adv. Data & Algorithms)
- 1 - LeetCode - <https://leetcode.com> 2 - HackerRank - <https://www.hackerrank.com>.
- 3 - Codewars - <https://www.codewars.com>. 4 - Coderbyte - <https://coderbyte.com> etc.

This Software Developer guide is provided by the SudoKoders, for a Cheaper and a more Practical (less theory) self-study alternative to say, programming Bootcamps, etc..
It's an ongoing project and If errors are found please do inform and add suggestions, so that together we can improve on it. kodesudo@gmail.com - Ug, Africa.

kodesudo@gmail.com

Note: Lots of information has been provided to help make for a better personal decision, but it can also be confusing. Here are some rough examples on how to choose and use:
> These 3 Chapters:

2) A Language + **3a)** Data Structures & Algorithms + **4)** Git (Gitlab, Github, etc...)

Are all compulsory for everyone to study. - Thereafter one should choose a Platform and/or Frameworks of choice. - Languages chosen will depend on the Frameworks of choice.

Example 1: Study for a Web site/app Developer

Level - (Intermediate). Est. Time - 1-2+ years (2+ hours daily)

2) Languages - Javascript + CSS + HTML - and later afterwards also + Typescript.

- Check the Web for books & courses (theOdinProject) for extensive language tutorials.

3a) Data Structures & Algorithms - Study only the **(3a)** Data (skip the Kodeco Swift book).

No need to study **(3b, 3c, 3d)** - (Advanced) Data Structures & Algorithms. <- (Optional).

4) DevOps Course -> - Git (Gitlab, Github, etc..). - Check the Web for tutorials

> Then Choose Platform and/or Framework Courses

5a) Front End (Web) -> Svelte or React or Vue, etc. Choose one. + UX Design <-(Optional)
Check the Web for books & tutorials of the Framework of choice.

7) App Dev. -> App Store, Google Play, MS Play, Front End, Back End, IoT, Embedded,

Check the web for books & tutorials on how to develop Web sites and/or Web Apps.

If possible, Publish at least 3 Web sites and/or Web Apps, for your **Portfolio**.

9) Developer Jobs @ -> Companies, Freelancer, StartUps, Indie Developer, etc...

Time for Job interviews !

Note: Except for **1)** IT, **3b,3c)** Data, **4)** DevOps, **8)** Cyber Security **9)** Developer Jobs, The guide Info provided is for Swift Language, as a rough example.

Example 2: Study for 3D CAD Models Developer and/or 3D Games Developer

Level - (Advanced). Est. Time - 2-4+ years (2+ hours daily)

2) Choose A Language - Rust or Swift or Kotlin or C# or C++ (depending on Frameworks)
Except for Swift, check the Web for books & courses for the other languages.

3b)+(3a+3c) CS Math + Adv. Data Structures & Algorithms courses. **3d)** This is (Optional),
(For **(3a)** - skip the Kodeco Swift book, - unless studying the Swift language),

Courses - 3D Graphics, Animations, Game Design and Development, Physics I, AI, etc..

4) DevOps Course -> Git (Gitlab, Github, etc..) - Check the Web for books & tutorials

> Then Choose Platform and/or Framework Courses

5b) Client Side -> 2D + 3D - Unreal, Unity, Godot, Blender, Quake, Maya, SceneKit, etc...
+ APIs - DirectX, Vulkan, OpenGL, Metal, etc... Choose.

- Check the Web for the tutorials of the 3D Engine + API of choice - Or how to create one.

7) App Dev. -> App Store, Google Play, MS Play, Front End, Back End, IoT, Embedded,

Check the web for books & tutorials on how to develop 3D CAD and/or Game Apps.

If possible, Publish at least 3 of own Apps to the platform of choice, for your **Portfolio**.

9) Developer Jobs @ -> Companies, Freelancer, StartUps, Indie Developer, etc...

Time for Job interviews !

> How to build a Software Developer Portfolio ->

1. -> <https://www.actitime.com/productivity/how-to-build-a-great-developer-portfolio>

2. -> <https://www.codecademy.com/resources/blog/software-developer-portfolio-tips/>

3. -> <https://arc.dev/developer-blog/software-engineer-portfolio/>

4. >>>. More useful Online Swift Programming resources

> A) Swift Quick References & Cheat Sheets :-

1. Swift Developer -> <https://developer.apple.com/swift/resources/>
2. Swiftly Cheat Sheet -> <https://swiftly.dev>
3. QuickRef Cheat Sheet -> <https://quickref.me/swift.html>
4. Kodeco -> <https://koenig-media.raywenderlich.com/uploads/2020/12/RW-Swift-5.1-Cheatsheet-1.0.pdf>
5. Swift in 1 Video -> <https://www.newthinktank.com/2020/01/learn-swift-5-one-video/>

> B) Swift Online tutorials :-

1. - 41 Best Resources to learn Swift (YouTube, Books, Courses, & Tutorials) -> <https://www.mltut.com/best-resources-to-learn-swift/>
2. - 10 Best Swift Courses to Take in 2023 -> <https://www.classcentral.com/report/best-swift-courses/>

> C) Some of the Swift YouTube channels :-

1. Code with Chris - iOS Dev. - YouTube - <https://www.youtube.com/@CodeWithChris>
2. Lets Build That App - <https://www.youtube.com/@LetsBuildThatApp/featured>
3. Sean Allen - iOS Dev. YouTube - <https://www.youtube.com/c/SeanAllen>
4. Paul Hudson - Hacking with Swift - <https://www.youtube.com/@twostraws>
5. iOS Academy - iOS Dev. YouTube - <https://www.youtube.com/@iOSAcademy>
6. Kilo Loco - YouTube - https://www.youtube.com/@Kilo_Loco
7. Kavsoft - YouTube - <https://www.youtube.com/@Kavsoft>
8. Swiftful Thinking - YouTube - <https://www.youtube.com/@SwiftfulThinking/featured>
9. Steward Lynch - iOS Dev. YouTube - <https://www.youtube.com/@StewartLynch>
10. Vincent Pradeilles - https://www.youtube.com/@v_pradeilles/videos

> D) Swift Online Blogs :-

1. Swift by Sundell - <https://www.swiftbysundell.com>
2. Appcoda - <https://www.appcoda.com>
3. SwiftLee Blog - <https://www.avanderlee.com>

> E) Other general Online Courses (MOOCs) ->

- 1 Coursera 2 Pluralsight Skills 3 EdX 4 Udacity 5 LinkedIn 6 Udemy 7 Codecademy
- 8 Future Learn 9 CBT Nugget 10 DataCamp 11 Khan Academy 12 freeCodeCamp
- 13 theOdinProject 14 GeeksforGeeks 15 W3School 16 Neso Academy etc...

> MIT ocw ComputerScience - <http://zafar.cc/2016/12/9/getting-bs-in-cs-2-dot-0/#fnref:1>

Note: Estimated from Hardest to Easiest languages to learn:- C++, Rust, Swift, Haskell, Scala, C#, Java, Kotlin, Go, PHP, Ruby, Julia, C, R, Python, Javascript, CSS, HTML ...

> Est.Time to learn this guide (Swift + Data Structures and Algorithms + build Apps) is -> 2+ years (2+ hours daily). Prerequisite:- Primary (Middle) School Math. (P7)

> + Est.Time for the Optional (Math for CS + Advance Data Structures & Algorithms) is -> 1-2+ years (2+ hours daily). Prerequisite:- Secondary (High) School Math (O-Level)

> However depending on the Level of Self Discipline, Motivation & Learning skills, and the Level of Prior self knowledge, like in Programming or Mathematics or IT or Computer Science, this will shorten or extend the time it takes to complete, significantly.

5 >>> **Applied Math** for **Computer/Data Science, Engineers, AI, Machine Learning, etc.** - more useful resources (Optional)

>> The **Math Learning Progression** for a self-study Math course for CS, AI, Engineers,->
1 Arithmetics, 2 Algebra I,II & III, 3 Geometry & (Analytic Geometry), 4 Trigonometry,
5 Discrete Math (Logic, Number Theory, Proofs, Sequences, Sets, Counting, Graphs, etc),
6 Calculus I & II, 7 Linear Algebra, 8 Calculus III, 9 Probability & Statistics.
> Videos - https://youtube.com/playlist?list=PLI6nIHwoZMVEAXnfl_ipmzSo_ovw13SZu

> **1,2,3,4) - Arithmetic, Algebra, Geometry & Trigonometry** (Equations, Functions, etc.)
1 A graphical approach to Algebra & Trigonometry 7e - Hornsby, Lial 2018 (\$85)
2 College Algebra (Precalculus) 5e - Robert Blitzer 2014 (\$85)

> **5) - Discrete Math** (Logic, Number Theory, Proofs, Sequences, Sets, Counting, Graphs)
1 Discrete Mathematics, Open Introduction 3e - Levin 2021 (free) PDF -> <https://discrete.openmathbooks.org/dmoi3.html>
2 Book of Proof 3e - Hammack 2018 (free) PDF -> <https://www.people.vcu.edu/~rhammack/BookOfProof/Main.pdf>
3 Discrete Mathematics for Computer Scientists - Stein, Clifford 2011 (\$75)
4 Discrete Mathematical Structures 6e - Kolman, Busby 2014 (\$44) (needs LinearAlgebra)

> **6) - Calculus I,II** (Limits, Single-Variable (Differentials (Derivatives) & Integrals), ODE,)
1 Calculus with Analytic Geometry 2e - George Simmons 1996
2 Calculus and Analytic Geometry 9e -Thomas, Finney 1995
3 Calculus I, II, & III 2e - Weinstein 1985 (free) PDF -> <https://www.cds.caltech.edu/~marsden/volume/Calculus/> 4 Essential Calculus Skills - Chris McMullan 2018 (\$12)

> **7) - Linear Algebra** (Linear Eq, Vectors, Matrices, Determinants, LinearTransformations)
1 Linear Algebra - Cherney, Denton 2013 (free) PDF -> <https://www.math.ucdavis.edu/~linear/linear-guest.pdf>
2 Elementary Linear Algebra 6e - Larson, Falvo 2009
3 Linear Algebra and its Applications 5e - David Lay 2016 (\$59)
4A MIT-18.06SC - Linear Algebra 2011 (free) -> <https://ocw.mit.edu/courses/18-06sc-linear-algebra-fall-2011/>
4B Introduction to Linear Algebra 5e - Gilbert Strang 2016 (\$15)
5 Schaum's Outline of Linear Algebra 6e - Lipschutz, Lipson 2017 (\$13)
6 Coding the Matrix (Python) - Philip Klein 2013 (\$35) -> <https://codingthematrix.com>
7 Matrix Computation 4e (Matlab) - Golub, Loan 2012 (\$40)

> **8) - Calculus III** (Vectors, Multi-Variable (Partial Derivatives & Multi Integrals), 2ndODE)
1 Thomas' Calculus : early Transcendentals SI Units 15e - Hass, Weir 2022 (\$85)
2 Calculus with Multi Variables : Essential Skills - Chris McMullan 2021 (\$14)
3 Differential Equations with Boundary-Value Problems 7e - Zill 2009 (\$69)-> (ODE & PDE)

> **9) - Probability & Statistics** (Random Variables, Distributions, Inference, Regression,)
1 Elementary Statistics: International 13e - Mario Triola 2017 (\$75) -> (basic concepts)
2 Introduction to Probability & Statistics 13e - Mendenhall 2009 (\$82) -> (basic concepts)
3 Introduction to Probability & Statistics for Engineers and Scientists 6e - Ross 2021 (\$80)
4 Schaum's Outline of Probability & Statistics 4e - Spiegel, Schiller 2012 (\$24)
5 Applied Statistics & Probability for Engineers 5e - Montgomery, Runger 2011 (\$96)
6 Introduction to Statistical Learning: with Applications (R, Python) 2e - James 2023 (free).

6. >>>. Other useful Online Programming self-study resources

- > **1) IT - CompTIA Certifications** -> <https://www.comptia.org/certifications>
- > Video - https://youtube.com/playlist?list=PLI6nIHwoZMVHR8lUKm66MXmflGWFX_EqC
- > **2) Software Programming languages** -> Videos.
https://youtube.com/playlist?list=PLI6nIHwoZMVFYNOeWPh_YnquNoT34DsF
- > **3) Data Structures & Algorithms** ->. Videos
<https://youtube.com/playlist?list=PLI6nIHwoZMVHlyJlc7k2mMtM1h39-CPLi>
- > Dynamic Programming - <https://www.youtube.com/watch?v=oBt53YbR9Kk&t=7018s>
- > **4) DevOps, Software Engineering, System Design, App Architecture, etc..** ->
https://youtube.com/playlist?list=PLI6nIHwoZMVEIf6p_UKC8E_GilygJ6M6v
- > **5a) Front End** -> Web
<https://youtube.com/playlist?list=PLI6nIHwoZMVH2NjuEtI0Y26yLzEV6uFVN>
- > **5b) Client Side** -> Mobile, Desktop PC, Tablets, TV's, Watches, Embedded, IoT.
<https://youtube.com/playlist?list=PLI6nIHwoZMVEmqucLpIWZ7obkeiYcrSPk>
- > **6a) Back End** ->
https://youtube.com/playlist?list=PLI6nIHwoZMVFuFeodeAin0wacEI_I8IIF
- > **6b) Cloud Platforms** ->
<https://youtube.com/playlist?list=PLI6nIHwoZMVH4vxgN-FHJ2xS7V-ecqVIO>
- > **7) App Dev** ->
<https://youtube.com/playlist?list=PLI6nIHwoZMVEXUImrKwk5jAn4tE--1BpK>
- > **8) Cyber Security** ->
https://youtube.com/playlist?list=PLI6nIHwoZMVGcRbENdwPziVY7_sZEHE8m
- > **9) Developer Jobs** ->
<https://youtube.com/playlist?list=PLI6nIHwoZMVGTfxjzeEIUuXCBhZPfuD7R>
- >> More Interesting Resources.
- > **11) Data Science & Analytics, AI, DL, ML, etc..** ->
<https://youtube.com/playlist?list=PLI6nIHwoZMVEXtMrdbYEa2fanAWpsTT35>
- > **12) IoT, Embedded, Robotics, etc... System Developer** ->
<https://youtube.com/playlist?list=PLI6nIHwoZMVFCPfkRok5qljivotfB-aUZ>
- > **13) Computer Science + other topics** ->
<https://youtube.com/playlist?list=PLI6nIHwoZMVHRYuAjGtKfNwMdYtYYHsdG>

Happy Coding !