

Swift Essential Fundamentals

Swift is a general-purpose, multi-paradigm, compiled programming language developed by Apple Inc. for iOS, macOS, watchOS, tvOS, and Linux. Swift is a modern-language, optimised for modern hardware.

Student must know fundamentals of Swift. To use it right way and

Prerequisites

- **SKILLS AND FINISHED COURSES**
 - No programming skill is required
 - Better to finish Swift Playgrounds (free iPad application in App Store)
- **TOOLS**
 - macOS 10.13 or later (free with Mac computer)
 - Xcode 9.3 or later (free in App Store)

Lessons

1 - 2 Days

GETTING STARTED

- Getting Xcode
- Creating playgrounds
- Accessing Documents

KNOWING VARIABLES AND CONSTANTS

- Simple Value
- Basic Operations
- Strings and characters

BASIS

- Control Flow
- Functions and Closures
- Object and Classes
- Enumerations and Structures
- Methods

ADVANCED BASIS

- Protocols and Extensions
- Error Handling
- Generics
- Automatic Reference Counting

ADVANCED

- Memory Safety
- Advanced Operators
- Access Control

Enablement

- iOS 9++ Essential Fundamentals

iOS 9++ Essential Fundamentals

To know how to build iOS application developer must know how to build AutoLayout. For more advance, developer must able to customise any view and control for their own business wills. And real developer must know how to handle dependencies for code, connect to api, and automate CI/CD.

Prerequisites

- **SWIFT ESSENTIAL FUNDAMENTALS**
- **TOOLS**
 - macOS 10.13 or later (free with Mac computer)
 - Xcode 9.3 or later (free in App Store)

Lessons

Day 1

INTRODUCTION

- Introduce to Xcode
- AppDelegate and Application Life Cycle

BUILDING UI

- Building basic UI via storyboard - UIView, UILabel, UITextField, UIImage, UIButton, UIStackView, and many more.
- View Debugging.

WORKING WITH UI AND CODE

- Connect UI to Code
- View Controller and View Controller Life Cycle
- Use delegate pattern
- Custom Control

* After first day student will know how to build UI with basic of auto layout.

Day 2

MORE CODE

- Unit testing

WORKING WITH TABLE VIEW

- UI Table View (second scene in storyboard)
- UI Navigation View
- Add, Edit, Delete, Persist Data

* After finish day 2 student will able to build simple multi-page application.

iOS 9++ In Real Life

To be faster in development iOS application, you should know about dependencies. good dependency will help you craft iOS application faster. And to use framework will more easier to maintain application.

And not to ruin your OS system: what you need to know before use CocoaPods is rbenv or rvm but we prefer rbenv. So let's understand and use it.

For now on mostly iOS application is not stand-alone. But it were client-sever application, so what you need to know is how to implement api client.

Sometimes we not know what goes wrong in application, Then use Xcode tools for debugging it may help you to find out.

If you release your application you will never know when application is crash, how application going to crash, or how many time app is close itself on which iDevice and what iOS it run. This time Crashlytics crash report will help you. By the way, they are many other tools but we prefer Crashlytics crash report.

Prerequisites

- **SKILLS AND FINISHED COURSES**
 - Swift Essential Fundamentals
- **ENVIRONMENTS**
 - Broadband internet
 - Disk space and Disk speed, Performance CPU

Lessons

1 - 3 Days

First part

- Working with dependencies
 - CocoaPods
 - Use sudo as your own risk, but we have a solution (rbenv)
 - Why do we need rbenv or rvm
 - How to install and use rbenv
 - Bundler (Gem)
 - Carthage
 - Better to know Homebrew (The missing package manager for macOS)
 - Swift Package Manager
 - Just acknowledge, not to use it right now.

Second part

- Connect to API (Alamofire or Moya)
- Brief on NSURLSession, AFNetworking
- Debugging with Xcode
- Crashlytics crash report
- Crashlytics beta
- Firebase Analytics

Git version control

Git is a most popular software configuration management tool (SCM).

SCM is use to track what is going on in project.

Prerequisites

- BROADBAND INTERNET

Lessons

Day 1

- Understand GIT
- Command
 - Start a working area
 - clone Clone a repository into a new directory
 - init Create an empty Git repository or reinitialize an existing one
 - Work on the current change
 - add Add file contents to the index
 - mv Move or rename a file, a directory, or a symlink
 - reset Reset current HEAD to the specified state
 - rm Remove files from the working tree and from the index
 - Examine the history and state
 - log Show commit logs
 - show Show various types of objects
 - status Show the working tree status
 - Grow, mark and tweak your common history
 - branch List, create, or delete branches
 - checkout Switch branches or restore working tree files
 - commit Record changes to the repository
 - diff Show changes between commits, commit and working tree, etc
 - merge Join two or more development histories together
 - rebase Reapply commits on top of another base tip
 - tag Create, list, delete or verify a tag object signed with GPG
 - collaborate
 - remote
 - fetch Download objects and refs from another repository
 - pull Fetch from and integrate with another repository or a local branch
 - push Update remote refs along with associated objects
 - other
 - reset
 - reflog
 - stash
 - etc.

iOS Automate - build, ship, run

What ever you do. Computer will help you with no doubts. No sleep. No mistake. But we need to tell him what exactly he need to do. You happy, your boss happy.

Prerequisites

- **SKILLS AND FINISHED COURSES**
 - iOS 9++ In real life (First part)
 - GIT version control
- ***BROADBAND INTERNET**
- ****MACHINE(S) FOR GITLAB AND JENKINS**

Lessons

1 - 2 days

- Build Configuration, Scheme, Target Memberships

First part

- App Distribution
 - Understanding
 - distribute certificate*.
 - developer certificate*.
 - provisioning profile*.

Second part

- Nice to meet you fastlane
 - Build
 - Test
 - Distribute
 - Etc.

Third part

- Know Jenkins, and other ci tools.
 - Jenkins**
 - TravisCI*
 - CircleCI*
 - Bitrise*
- Know CI friends
 - CodeCov*
 - Gitlab CI**
 - Github*
 - Facebook Mention-Bot*

iOS Unit Testing and UI Testing

To make sure application still running as it were, unit test is the fastest tool.
To make sure you integrate it right, ui testing is the best tool.

Prerequisites

- **SWIFT ESSENTIAL FUNDAMENTALS AND SAME TOOLS**
- **BETTER IF YOU FINISH IOS 9++ ESSENTIAL FUNDAMENTALS**

Lessons

1 - 2 days

- Introduction to testing

First part

- Unit testing
 - Creating a Unit Test target
 - How to run unit test
 - Create testing function and using XCTAssert
 - Debugging a test
 - Using XCTestExpectation
 - Faking objects and interactions
 - Fake input from stub
 - Fake update to Mock Object

Second part

- UI Testing with Xcode (Build simple calculator with UIBuilder)
 - Creating a UITest target
 - Create test
 - Record test

Third part

- Performance testing
- Code Coverage

More about iOS Application Development

Prerequisites

- **SKILLS AND FINISHED COURSES**
 - Swift Essential Fundamentals and same tools
 - Better if you finish iOS 9++ Essential Fundamentals
 - iOS Automate - build, ship, run (First part)
- **TOOLS (FOR PUSH NOTIFICATION)**
 - Real iDevice (iPhone, iPod, iPad) that able to access to internet
 - Apple developer account (Any type)

Lessons

2 Days

- Working with push notifications (Firebase to APNs)
 - Getting Start
 - Enabling Push Notification Service
 - Registering for Push Notification
 - Create an SSL Certificate and PEM file
 - Sending a Push Notification
 - Create APNs key
 - Handling Push Notifications
 - It happens when receive a Push Notification
 - Actionable Notifications
 - Handling Notifications Actions
 - Silent Push Notifications
 - Use Firebase to send push for you
- Working with designer
 - Sketch
 - Zeplin
 - Marvel
 - InVision
- BITCODE
 - Understanding
 - What prevent to using BTICODE
 - Pros and Cons
- HTTP/HTTPS debugging with Charles and netfox

Reactive Programming in iOS with RxSwift

RxSwift is a reactive programming library written in and for Swift language. For composing asynchronous and event-based code by using observable sequences and functional style operators, allowing for parameterized execution via schedulers.

Prerequisites

- **SWIFT ESSENTIAL FUNDAMENTALS AND SAME TOOLS**
- **BETTER IF YOU FINISH IOS 9++ ESSENTIAL FUNDAMENTALS**

Lessons

2 - 4 days

- Introduction to Reactive Programming with RxSwift
 - Hello RxSwift
 - Observables
 - Subjects
 - Filter Operators
 - Transform Operators
 - Combine Operators
 - Time Based Operators
 - Scheduler
 - Hot & Cold
- RxCocoa
- Error handling
- Testing

About Instructor

Wasith Theerapattrathamrong (ไบไฟ)

Has been in software development and related fields for several years, who love his work and can do it with quality and passion.

Experience

- iOS application development.
- Microservices architecture.
- Agile software development.
- Network engineer, System Engineer.

Certifications

- **Microsoft Technology Associate** - Microsoft Certification ID: 11571303
 - **Software Development Fundamentals** - License ID: F138-0648
 - **Database Administration Fundamentals** - License ID: F138-0649
 - **Networking Fundamentals** - License ID: F138-0650
 - **Windows Server Administration Fundamentals** - License ID: F139-5240
 - **Security Fundamentals** - License ID: F139-5241
- **Information Technology Professionals Examination**
 - **Fundamental Information Technology Engineer (FE)** - License ID: THFE14S001
 - **Information Technology Passport (IP)** - License ID: THIP12S0351

Contact

Email: phai@live.com

Mobile: +66-85-566-6513



Me (on a chair) and student in iOS with Swift class. (One of many classes)