Mobile Development on the iOS Platform

Master's Degree in Computer Science
Mobile Programming and Multimedia
University of Padua

Alberto De Bortoli - 03/05/2022

Who I am

- University of Padua alumni (Master’s degree in 2011)
- Worked in the mobile space since then
- Currently working in London at Just Eat Takeaway.com as Principal Software Engineer
- More info at albertodebortoli.com

Today’s agenda

- History of native iOS development
- Languages
- Frameworks
- Tools and Package Managers

~45 min

- Documentation and Resources
- Demo
- Q&A

~45 min

History of native iOS development

History of native iOS development

Operating System and devices

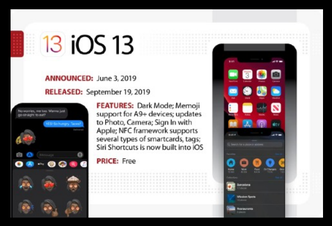
- Drastically evolved over the years
- Started in 2008 with iPhone OS 2
- Very limited set of features
- iPhone OS rebranded as iOS in 2010

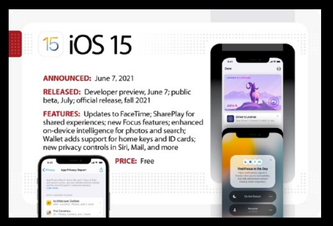
Source: https://www.computerworld.com/article/2975868/the-evolution-of-ios.html

History of native iOS development

Operating System and devices

- New iPhone models and new iOS major releases every year
- Ever increasing set of features



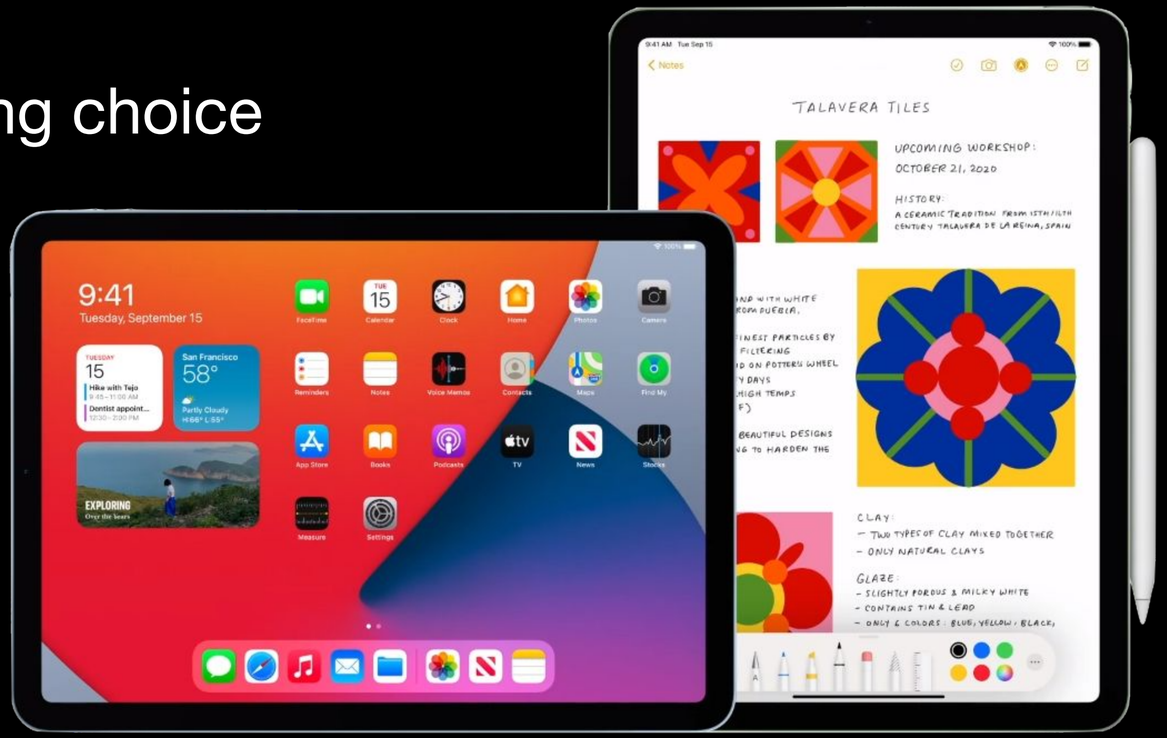


Source: https://www.computerworld.com/article/2975868/the-evolution-of-ios.html

History of native iOS development

Operating System and devices

- iPadOS introduced in 2019
- Same as iOS, just a marketing branding choice

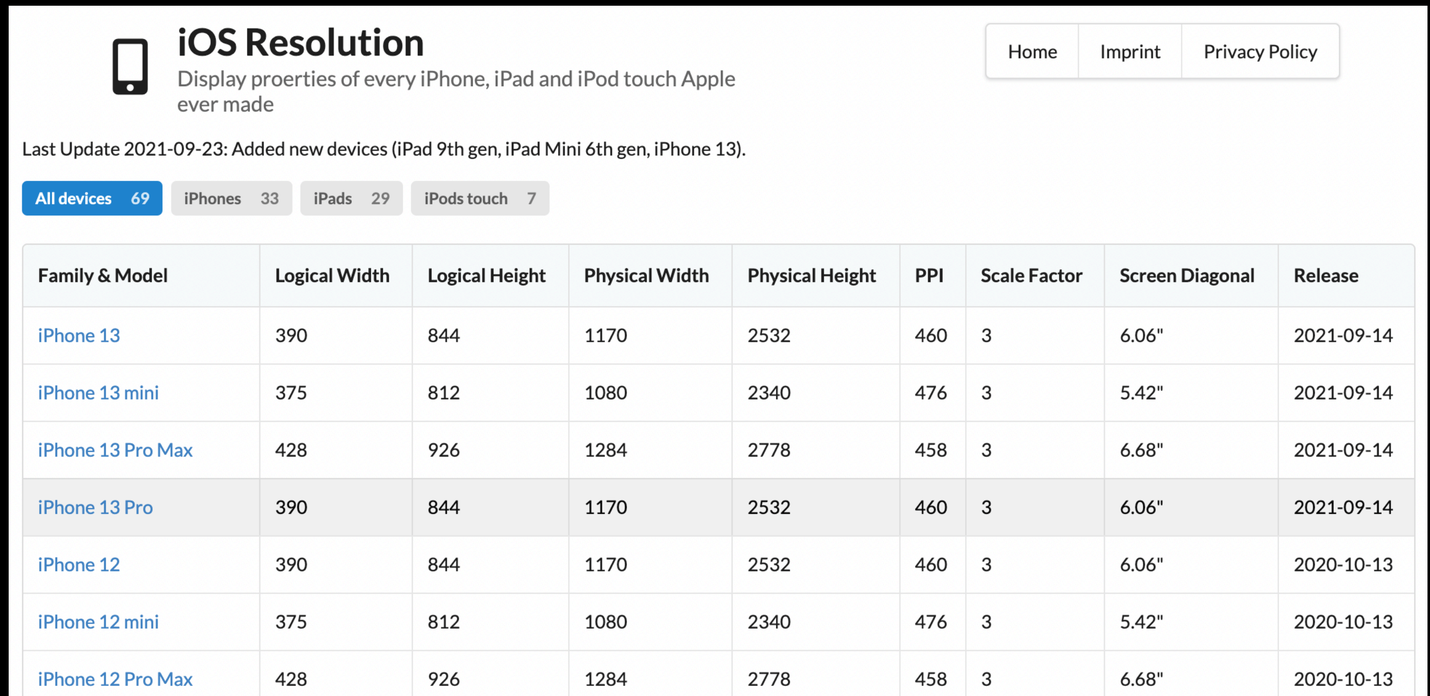


Source: https://www.computerworld.com/article/2975868/the-evolution-of-ios.html

History of native iOS development

Operating System and devices

- Device fragmentation not particularly relevant anymore
- Developers don’t usually care much about specific resolutions, more about layouting rules



iOS Resolution

Display proerties of every iPhone, iPad and iPod touch Apple ever made

Home | Imprint | Privacy Policy

Last Update 2021-09-23: Added new devices (iPad 9th gen, iPad Mini 6th gen, iPhone 13).

All devices 69 | iPhones 33 | iPads 29 | iPods touch 7

Family & Model	Logical Width	Logical Height	Physical Width	Physical Height	PPI	Scale Factor	Screen Diagonal	Release
iPhone 13	390	844	1170	2532	460	3	6.06"	2021-09-14
iPhone 13 mini	375	812	1080	2340	476	3	5.42"	2021-09-14
iPhone 13 Pro Max	428	926	1284	2778	458	3	6.68"	2021-09-14
iPhone 13 Pro	390	844	1170	2532	460	3	6.06"	2021-09-14
iPhone 12	390	844	1170	2532	460	3	6.06"	2020-10-13
iPhone 12 mini	375	812	1080	2340	476	3	5.42"	2020-10-13
iPhone 12 Pro Max	428	926	1284	2778	458	3	6.68"	2020-10-13

Source: https://www.ios-resolution.com/

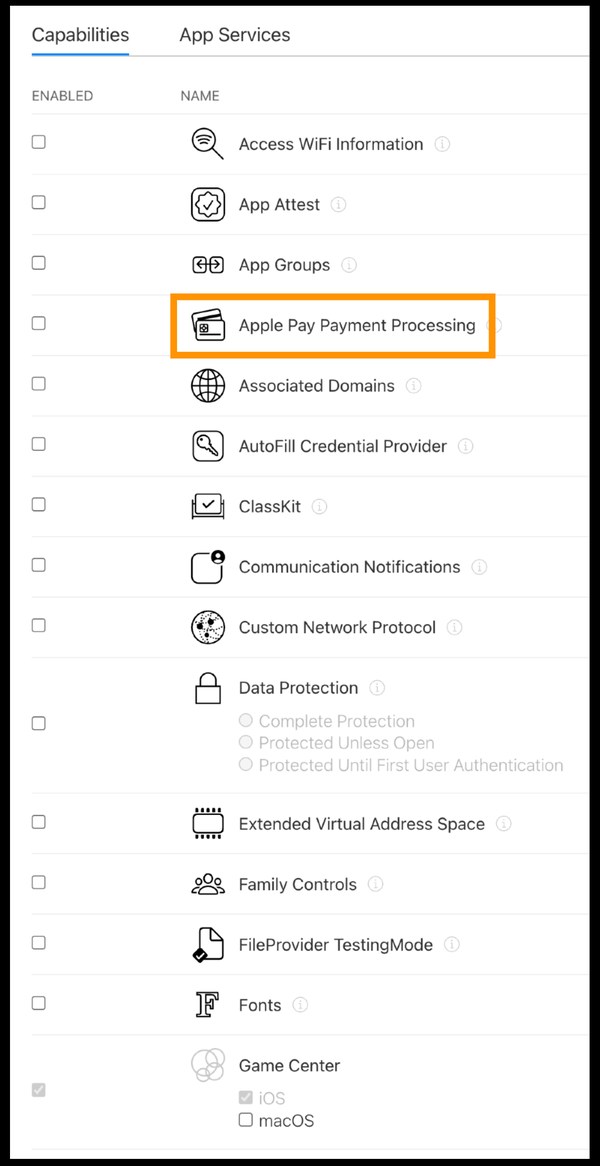
History of native iOS development

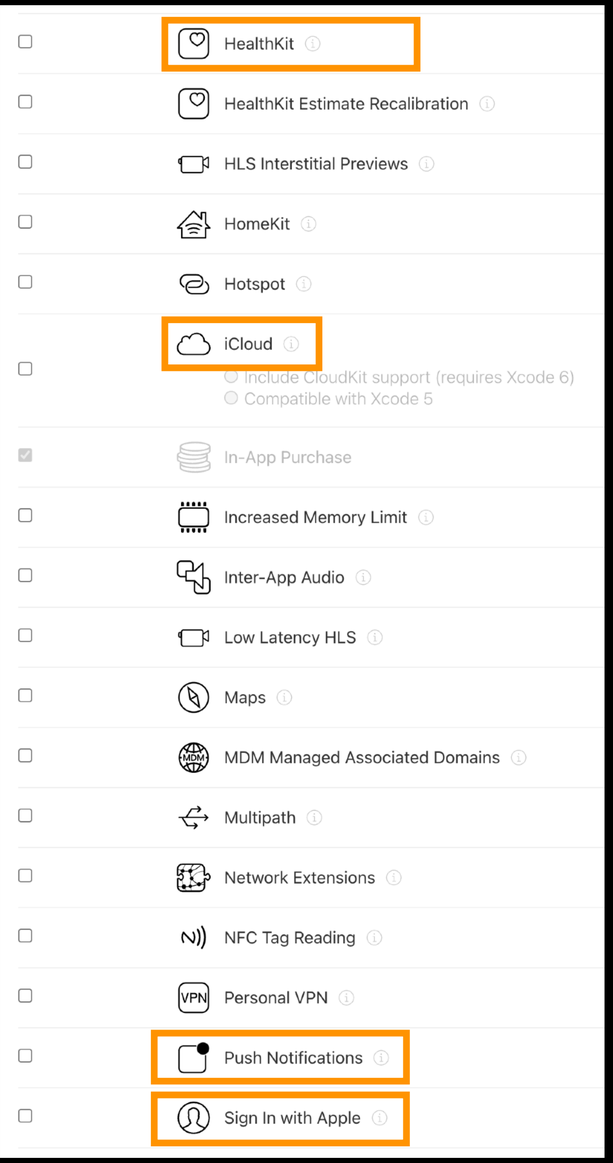
Operating System and devices

Capabilities:

- Ever growing list
- Enabled per app in App Store Connect
- Defined in an Entitlements file (XML) in the app project

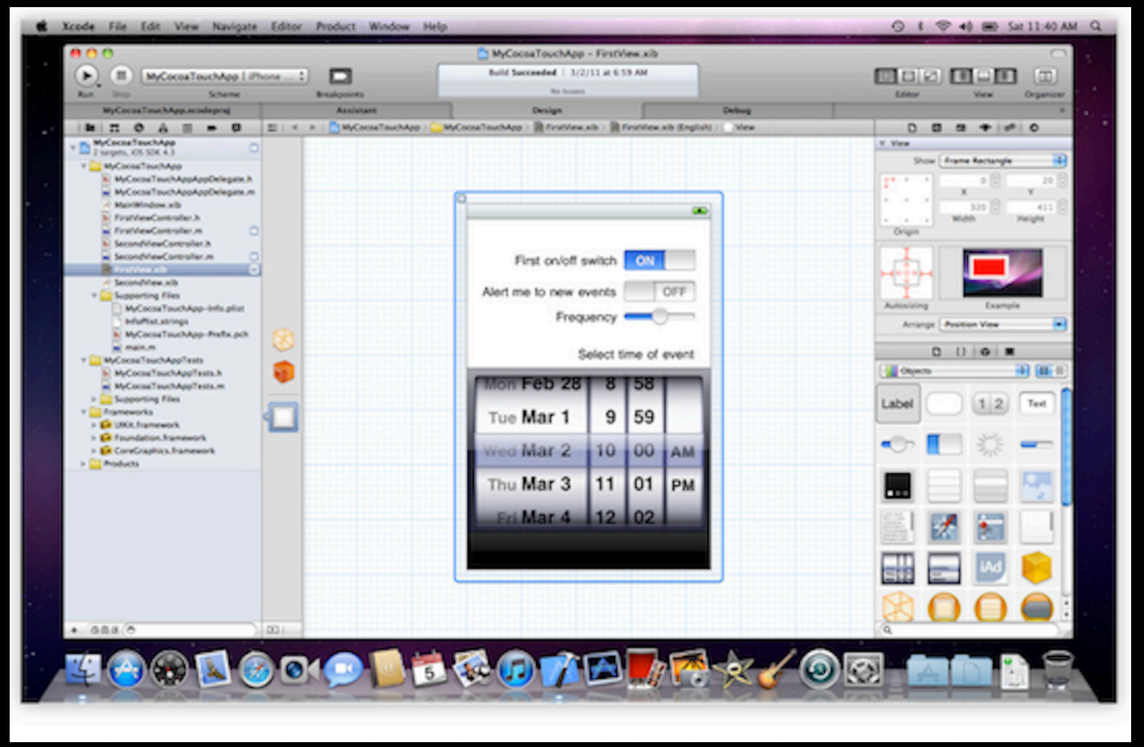
Source: https://appstoreconnect.apple.com/





History of native iOS development
IDE

- Xcode as the official IDE
- New major releases almost every year
- Requires Apple’s hardware



2010: Xcode 4 was a major turning point in Xcode’s history, and the successive improvements were all built on the changes made during this era.

Source: https://martiancraft.com/blog/2022/01/xcode-through-the-years/

History of native iOS development

IDE

- Xcode as the official IDE
- New major releases almost every year
- Requires Apple’s hardware

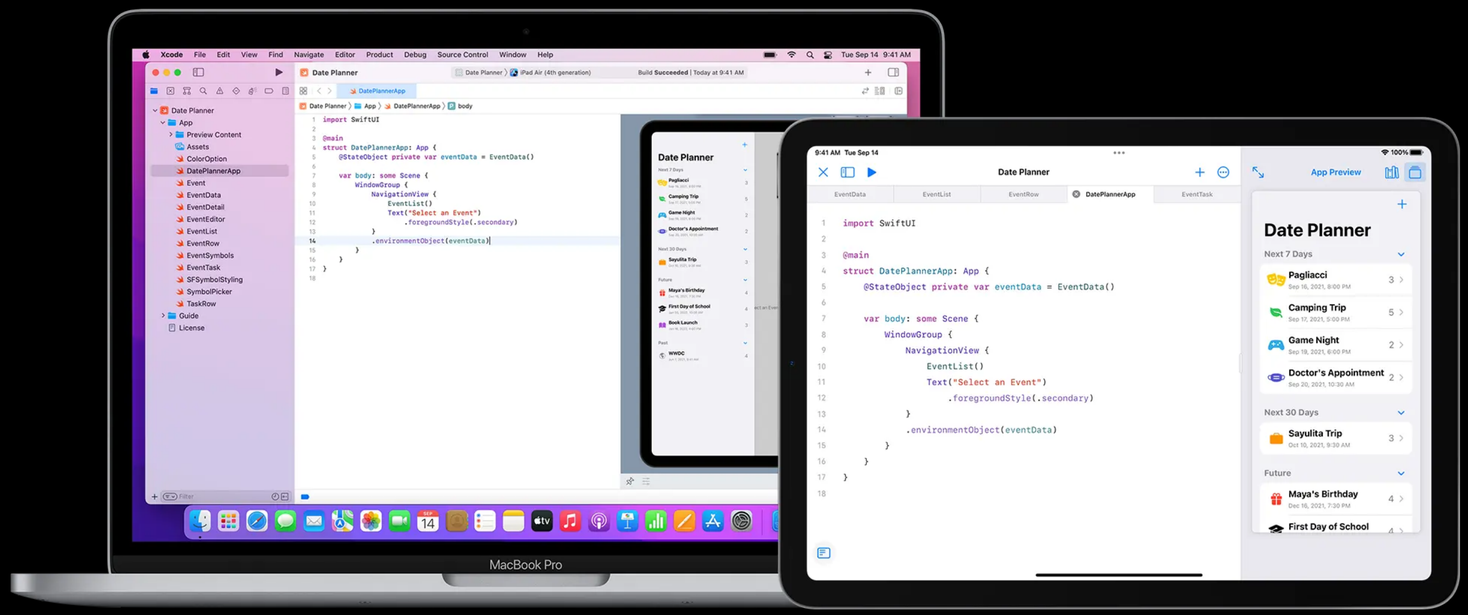
2019-2021: Xcode 11/12/13 running SwiftUI code with the visual inspector opened for debugging.

Source: https://martiancraft.com/blog/2022/01/xcode-through-the-years/

History of native iOS development

IDE

- Xcode as the official IDE
- New major releases almost every year
- Requires Apple’s hardware



2021 onwards: Swift Playgrounds 4 running on macOS and iPadOS, sharing the same codebase for the same app. Apps built with Swift Playgrounds 4 on iPadOS can be run and delivered to the App Store for publication.

Source: https://martiancraft.com/blog/2022/01/xcode-through-the-years/

Languages

Languages
Objective-C

- Objective-C as the historical language used at Apple/NeXT
- Written by Brad Cox in the early ’80 (been around for ~40 years)
- Very dynamic and allows introspection
- Takes inspiration from Smalltalk (message sending instead of function calls)
- No support for namespaces: classes and symbols prefixed with NS (e.g. NSString)
- Manual memory management (reference counting) in first incarnations

```
NSString *str = [[NSString alloc] initWithString: @“Hello”];
NSString *combinedStr = [str stringByAppendingFormat:@" World"];
NSLog(@"Combined string: %@", combinedStr);
```

Languages
Swift

- Swift was announced in 2014
- Originally written by Chris Lattner at Apple
- Latest version 5.6 (5.7 in development)
- Strongly typed
- Cutting-edge set of features
- Open-sourced by Apple on GitHub (https://github.com/apple/swift)
- Evolution in the open (https://github.com/apple/swift-evolution)

```
var str = "Hello"
str.append(" World")
print(str)
```

Reference: https://swift.org/

Frameworks

Frameworks

Native Apple frameworks

- Huge list of frameworks from Apple
- Fundamentals:
  - Foundation
  - UIKit
  - SwiftUI

iOS SDK Frameworks

The following lists the frameworks that are part of the iOS SDK as of version 13.0.

- ARKit
- AVFoundation.AVFAudio
- AVFoundation
- AVKit
- Accelerate
- Accelerate.vImage
- Accelerate.vecLib
- Accounts
- AdSupport
- AddressBook
- AddressBookUI
- AssetsLibrary
- AudioToolbox
- AudioUnit
- AuthenticationServices
- BackgroundTasks
- BusinessChat
- CFNetwork
- CallKit
- CarPlay
- ClassKit
- CloudKit
- Contacts
- ContactsUI
- CoreAudio
- CoreAudioKit
- CoreAudioTypes
- CoreBluetooth
- CoreData
- CoreFoundation — lower-level C System APIs
- CoreGraphics
- CoreHaptics
- CoreImage
- CoreLocation

Frameworks

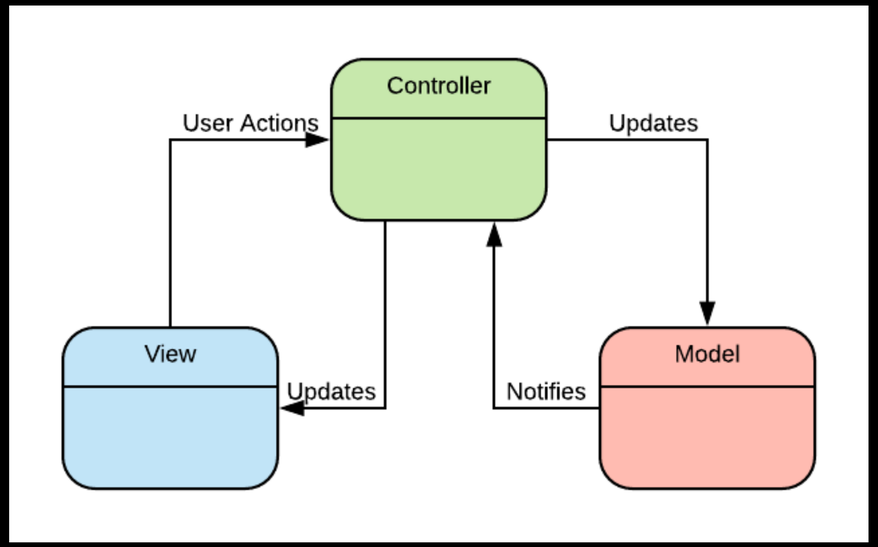
Foundation and the Swift standard library

- Essential data types, collections, dates, time calculation, text processing, sorting, networking, and operating-system services
- Foundation could be considered the “standard library” of Objective-C (NSArray, NSDictionary, NSString, etc.)
- The Swift standard library contains the fundamental data types
- File system access, calendar and time-zone-aware date handling, localization functionality etc. are not included in the Swift standard library and remain in Foundation.

Reference: https://developer.apple.com/documentation/foundation

Frameworks
UIKit

- Construct and manage a graphical, event-driven user interface
- Manages life-cycle events and scenes
- Provides classes for UI components as well as utilities to manage interactions/gestures and resources
- MVC and delegation as some of the main patterns (implicitly) promoted by Apple
- Imperative UI framework



Reference: https://developer.apple.com/documentation/uikit

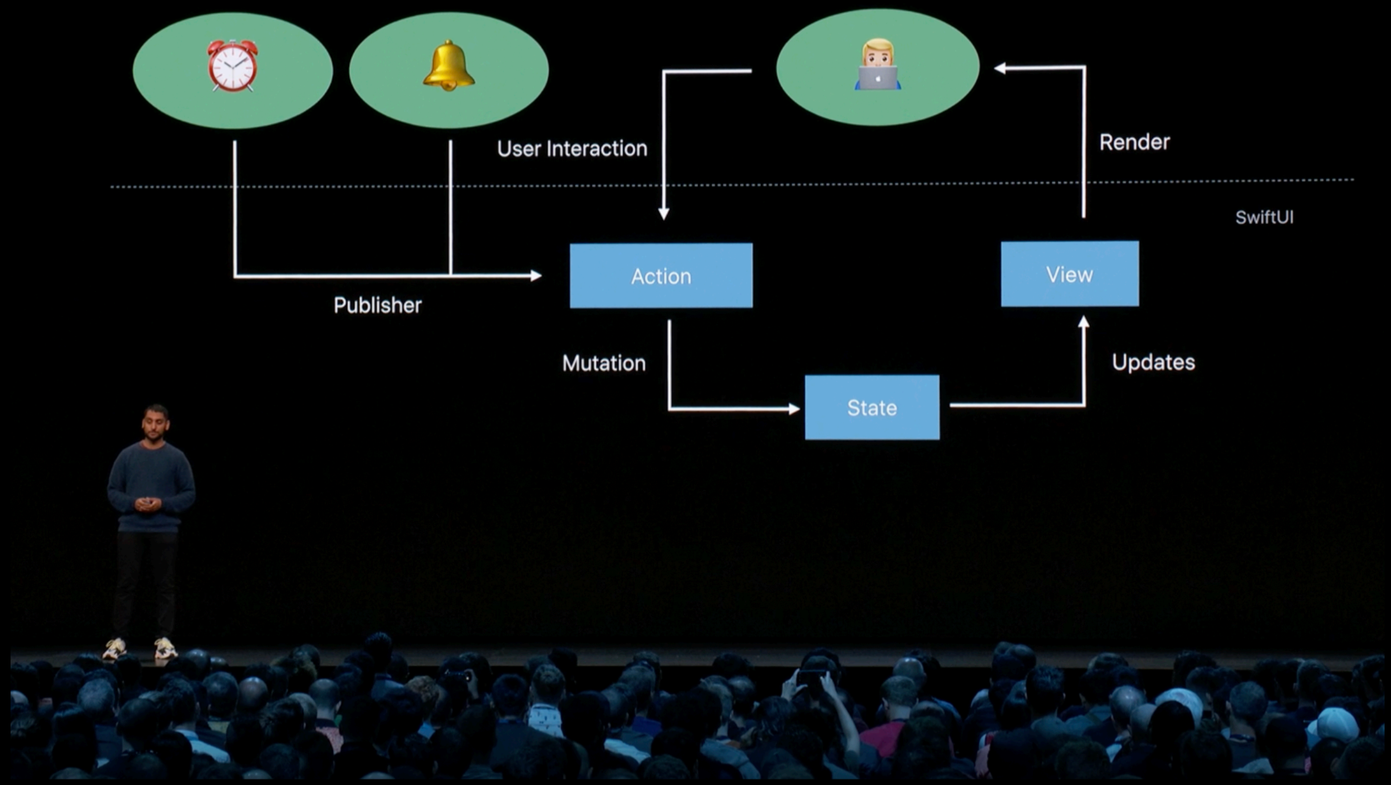
Frameworks

SwiftUI

- Massive leap forward
- Swift evolved to enable SwiftUI
- The way forward to write interfaces
- Cross OSes (tvOS, watchOS, macOS, iOS/iPadOS)
- Localisations, dynamic type and accessibility almost for free

Frameworks
SwiftUI

- Data Flow Through SwiftUI
- Declarative UI framework



Unnamed pattern (on purpose) but in reality it’s a reincarnation of Flux/Redux

Source: https://developer.apple.com/videos/play/wwdc2019/226/

Frameworks

SwiftUI

Layouting rules

1. Parent Proposes Size for Child

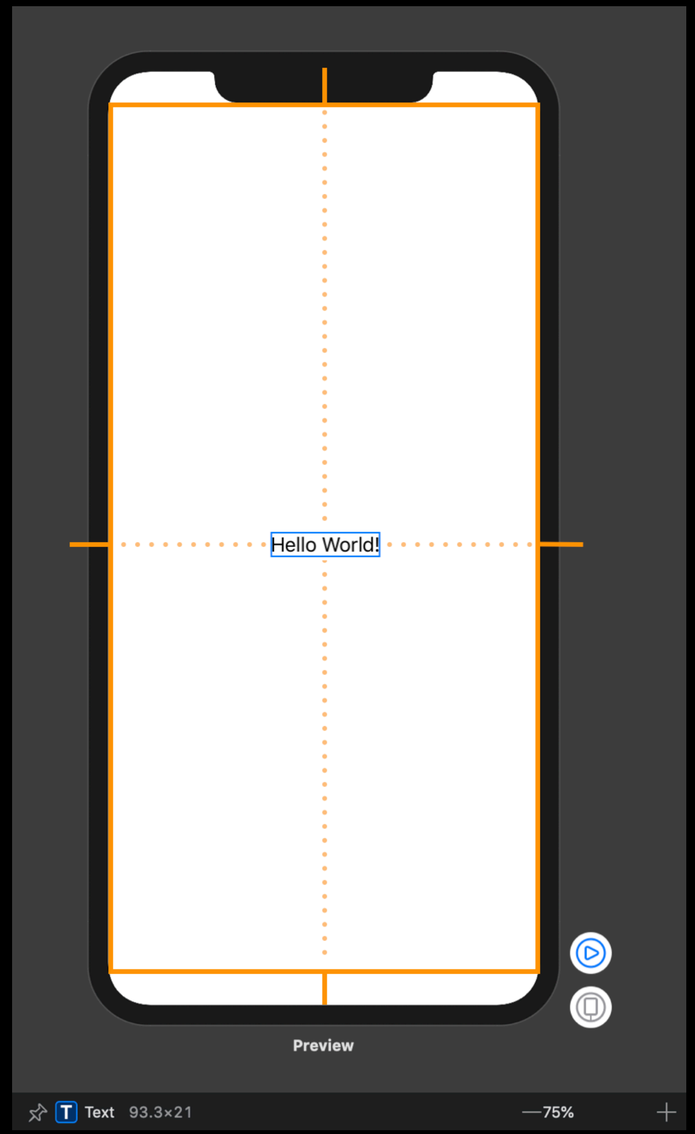
First, the root view offers the text a proposed size – in this case, the entire safe area of the screen, represented by an orange rectangle.

2. Child Chooses its Size

Text only requires that much size to draw its content. The parent has to respect the child's choice. It doesn't stretch or compress the child.

3. Parent Places Child in Parent’s Coordinate Space

And now the root view has to put the child somewhere, so it puts in right in the middle.

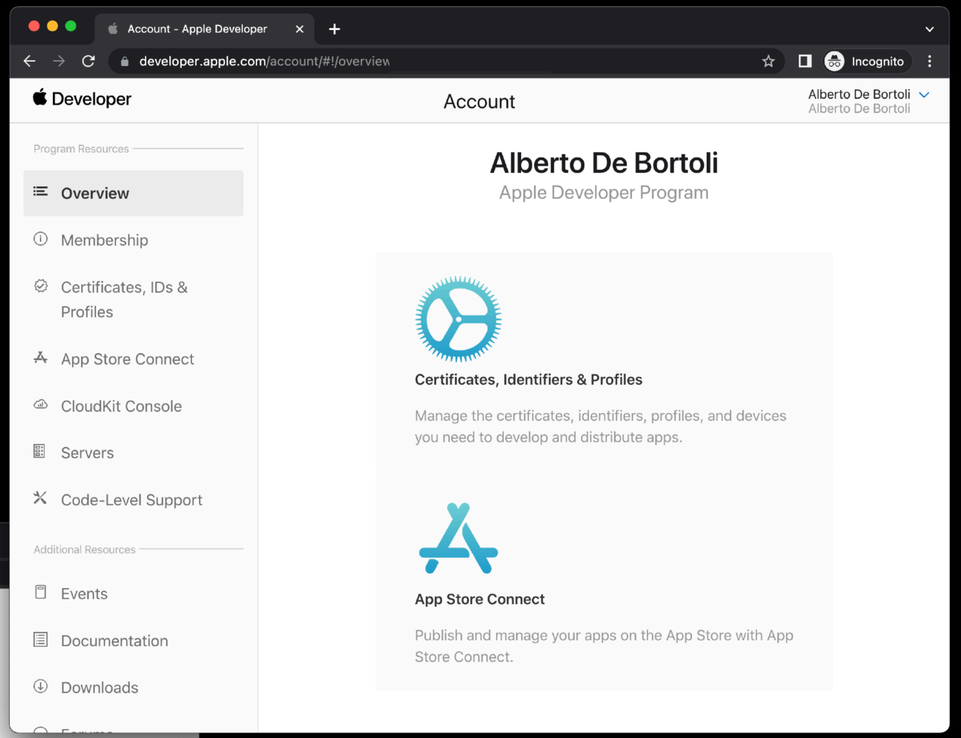


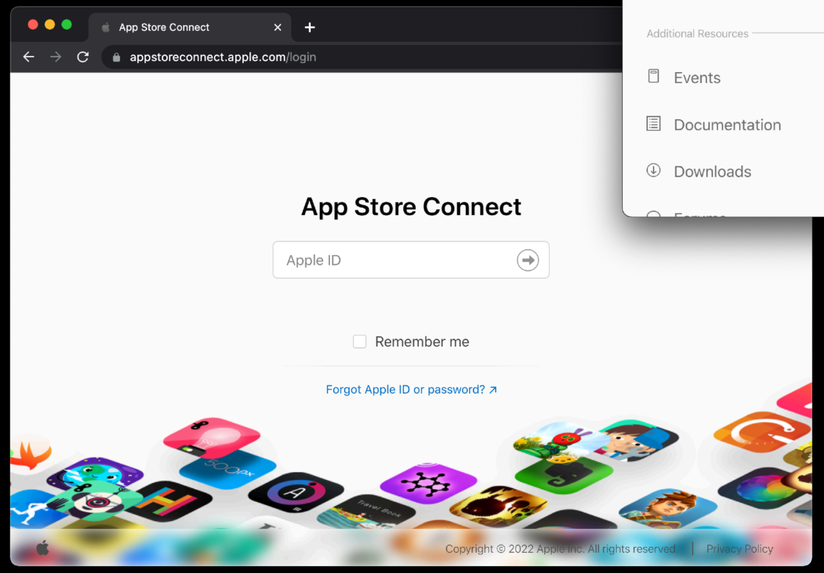
Tools and Package Managers

Tools

App Store Connect and Developer Portal

- The portals to setup and configure all the things for your apps
- Developer License $99



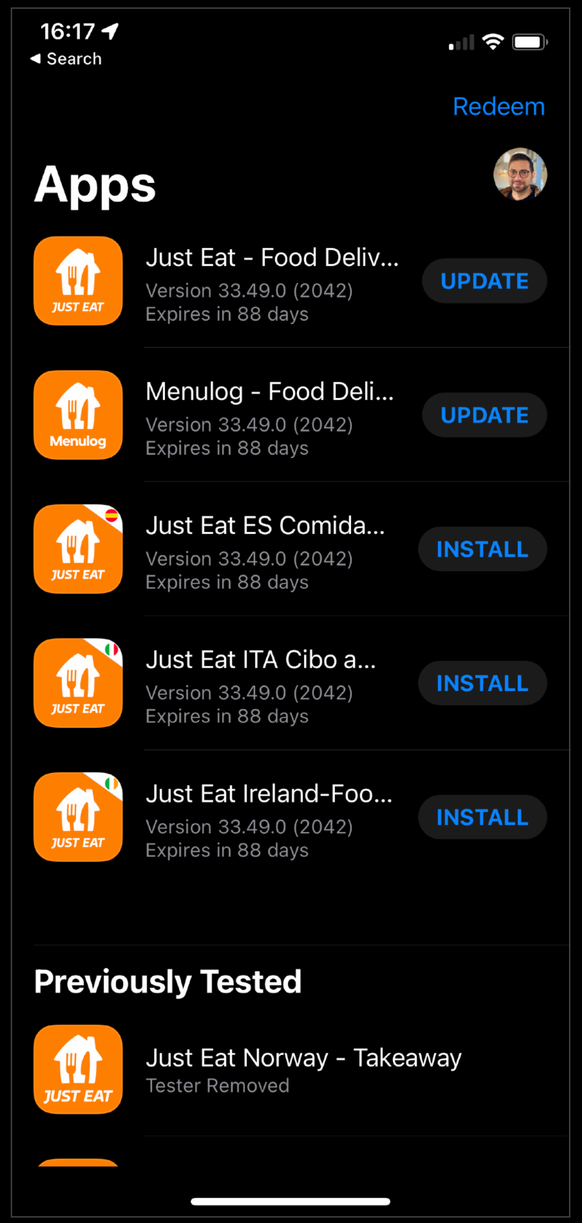


References:
https://appstoreconnect.apple.com/
https://developer.apple.com/

Tools

TestFlight

- The way to distribute apps to testers using the Apple Developer license
- Dashboard on App Store Connect (per app)
- Dedicated app on the App Store



Reference: https://developer.apple.com/testflight/

Tools
Fastlane

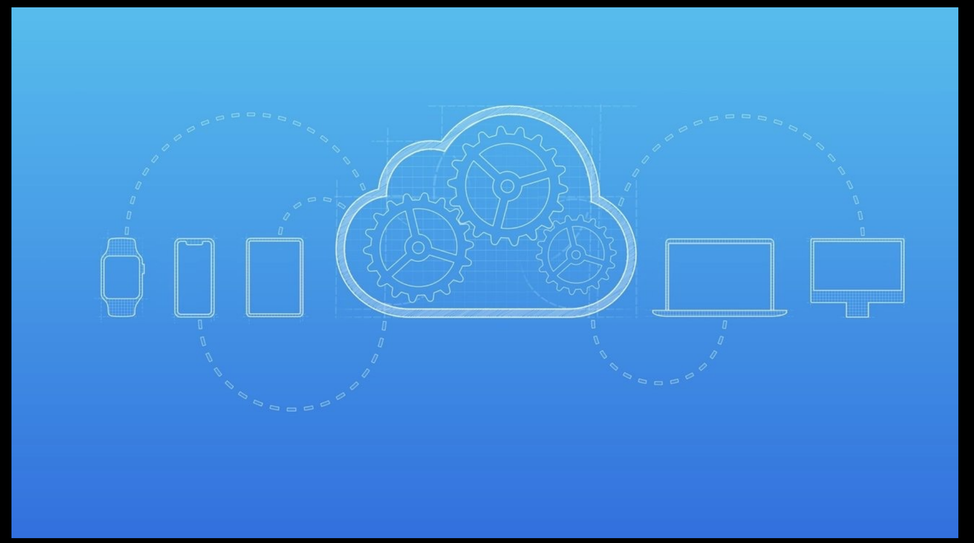
- Fastlane simplifies the tools and processes provided by Apple
  - Automate screenshots and metadata
  - Beta/App Store deployment
  - Code signing, provisioning profiles and private keys management
  - `xcodebuild` commands
- Fastlane is the only third-party tool that became a standard de facto

Reference: https://fastlane.tools/

Tools

Xcode Cloud

- Continuous integration and delivery service
- New solution from Apple
- Integrates with the Apple ecosystem
- Meant to be easy
- Announced in 2021, still in beta



Reference: https://developer.apple.com/xcode-cloud/

Package Managers

CocoaPods

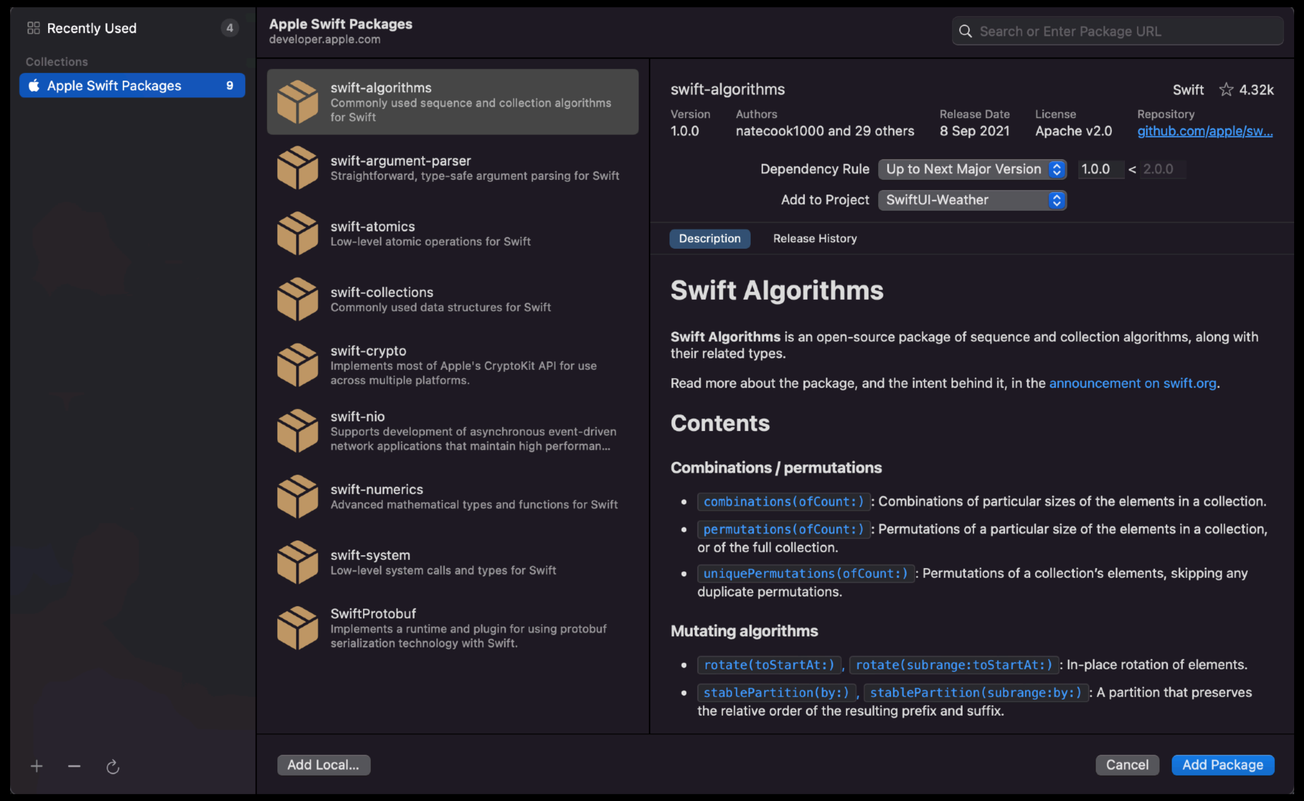
- Third-party package manager to overcome the lack from Apple (not the case anymore)
- The historical de facto standard for managing dependencies
- Automated solution
- The frameworks (called pods) are defined in a manifest called “podspec”
- Ruby-based

Reference: https://cocoapods.org/

Package Managers

Swift Package Manager

- Long-awaited native dependency manager
- Gradually taking over CocoaPods
- Integrated with Xcode
- The frameworks (called packages) are defined in a manifest named `Package.swift`
- Swift-based

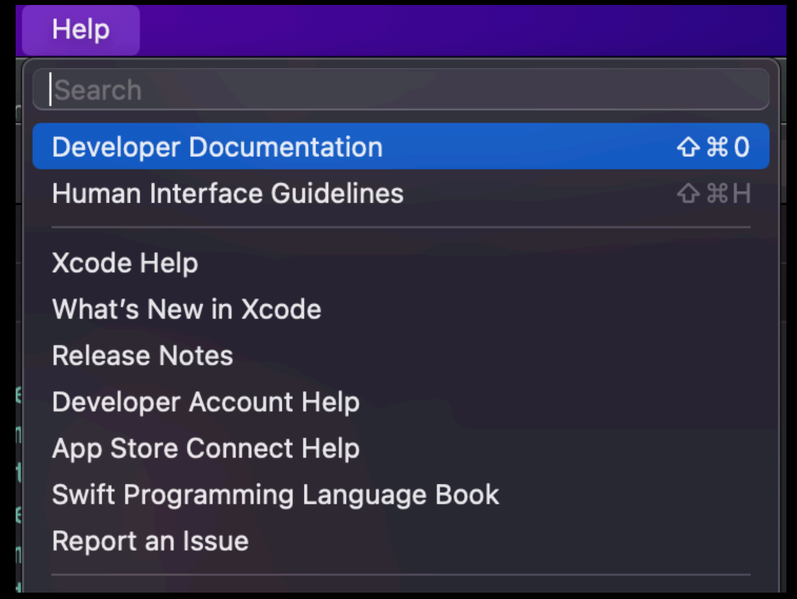


Documentation and Resources

Documentation and Resources

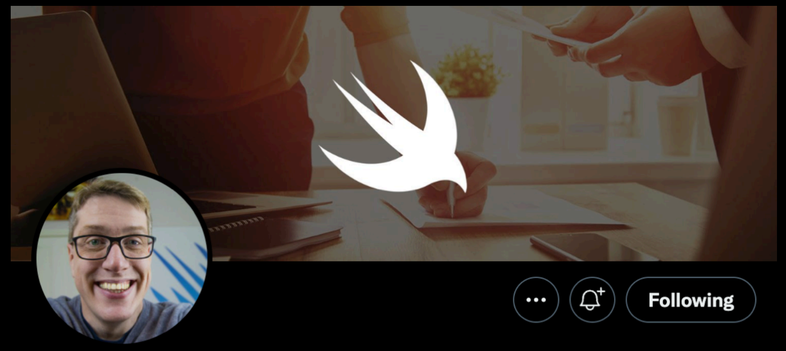
Official documentation

- Apple has become extremely good in providing documentation, examples and tutorials for all level.
- Main places to go are swift.org and developer.apple.com
- Easily accessible from within Xcode



Documentation and Resources

Authors and speakers

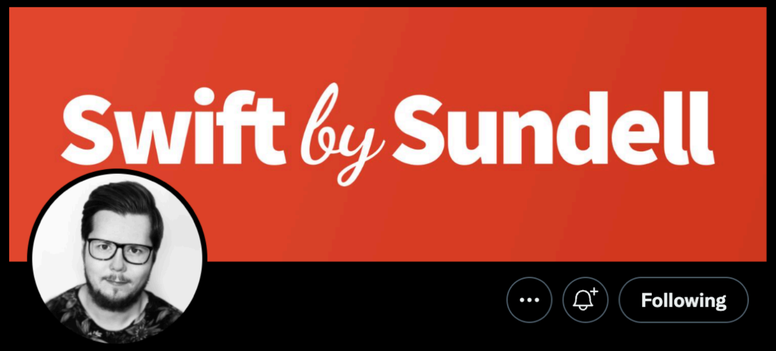


Paul Hudson
@twostraws

Learning and sharing at Hacking with Swift, author of Swift, SwiftUI, and iOS books, public speaker, Rubik's cube enthusiast, and herder of my kids. (He/him)

Bath, UK · hackingwithswift.com · Joined July 2009

257 Following **68.3K** Followers

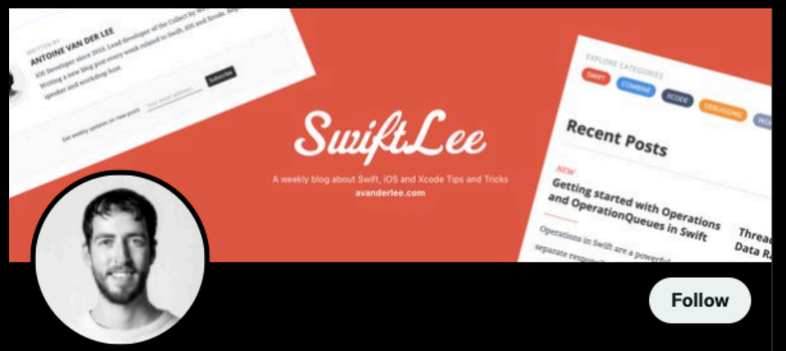


John Sundell
@johnsundell

Creator of @swiftbysundell, co-host of @stacktracepod, Swift developer, avid prototyper and amateur chef 👨‍💻👩‍💻👨‍🍳

Gdańsk, Poland · swiftbysundell.com · Joined February 2011

315 Following **46.7K** Followers



Antoine v.d. SwiftLee
@twannl

#iosdev #iOS @WeTransfer

#swift blog: avanderlee.com
Newsletter & discounts: avanderlee.com/swiftlee-weekl...
Simulator enhanced: rocketsim.app

Amsterdam, The Netherlands · avanderlee.com · Joined December 2009

711 Following **28.6K** Followers

Documentation and Resources

Conferences

- WWDC (Worldwide Developer Conference) - Cupertino, CA
  https://developer.apple.com/wwdc22/
- #Pragma Conference - Bologna, IT
  https://pragmaconference.com/
- SwiftHeroes - Turin, IT
  https://swiftheroes.com/

Demo

Demo

Xcode / SwiftUI / Documentation

- Make a start to build a weather app UI
- Time is extremely limited (~40 min)
- Basic interactions with Xcode
- Navigating the documentation

Source: https://www.youtube.com/watch?v=HXoVSbwWUIk

Q&A