

Introduction to Swift 3.1 and Xcode

Presented at Learn Swift NOVA on July 15, 2017

Background

Present

- Lead Developer, *The Washington Post*
- Freelance, *The iOS Consultant*
- Instructor, *Betamore & General Assembly*

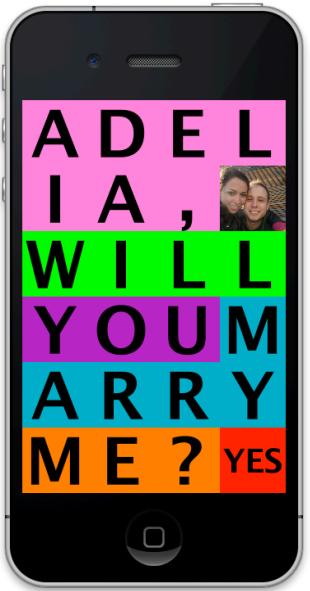
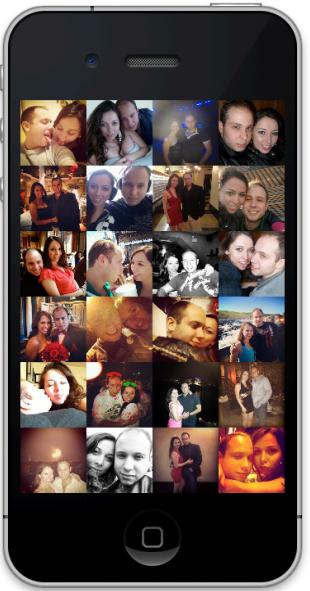
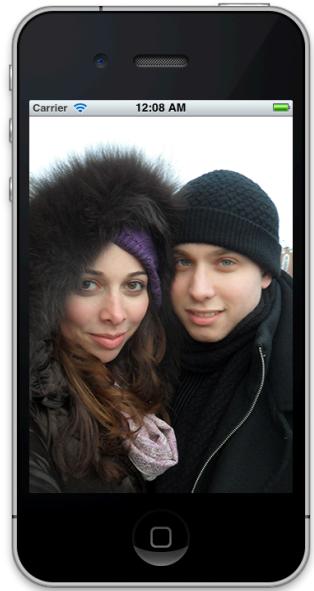
Past

- Experimental Nuclear Physics, MS
- Biological Physics, BS
- Firefox Add-Ons Developer



App Proposal

I proposed to my wife with an app in 2012. We were married in 2013.
The [New York Times](#) published a couple¹ pieces² on it.



¹ [Proposals \(NYT\)](#)

² [Proposals that aren't Prosaic \(NYT\)](#)

The Essentials

Hardware

Initial investment is expensive!

- **Computer**
 - Macbook Air, Macbook Pro, iMac: \$999+
- **Testing Devices**
 - iPhone: \$399+ (without subsidies)
 - iPad (WiFi-only): \$329+
 - Apple Watch: \$269+
 - Apple TV: \$149+

Software - Apple Developer Account³

- **Free account**
 - Use your existing iCloud account.
- **Ship app to the App Store**
 - \$99 personal or corporate account.
- **Ship app outside the App Store**
 - \$299 enterprise account.

³ [Apple Developer Center](#)

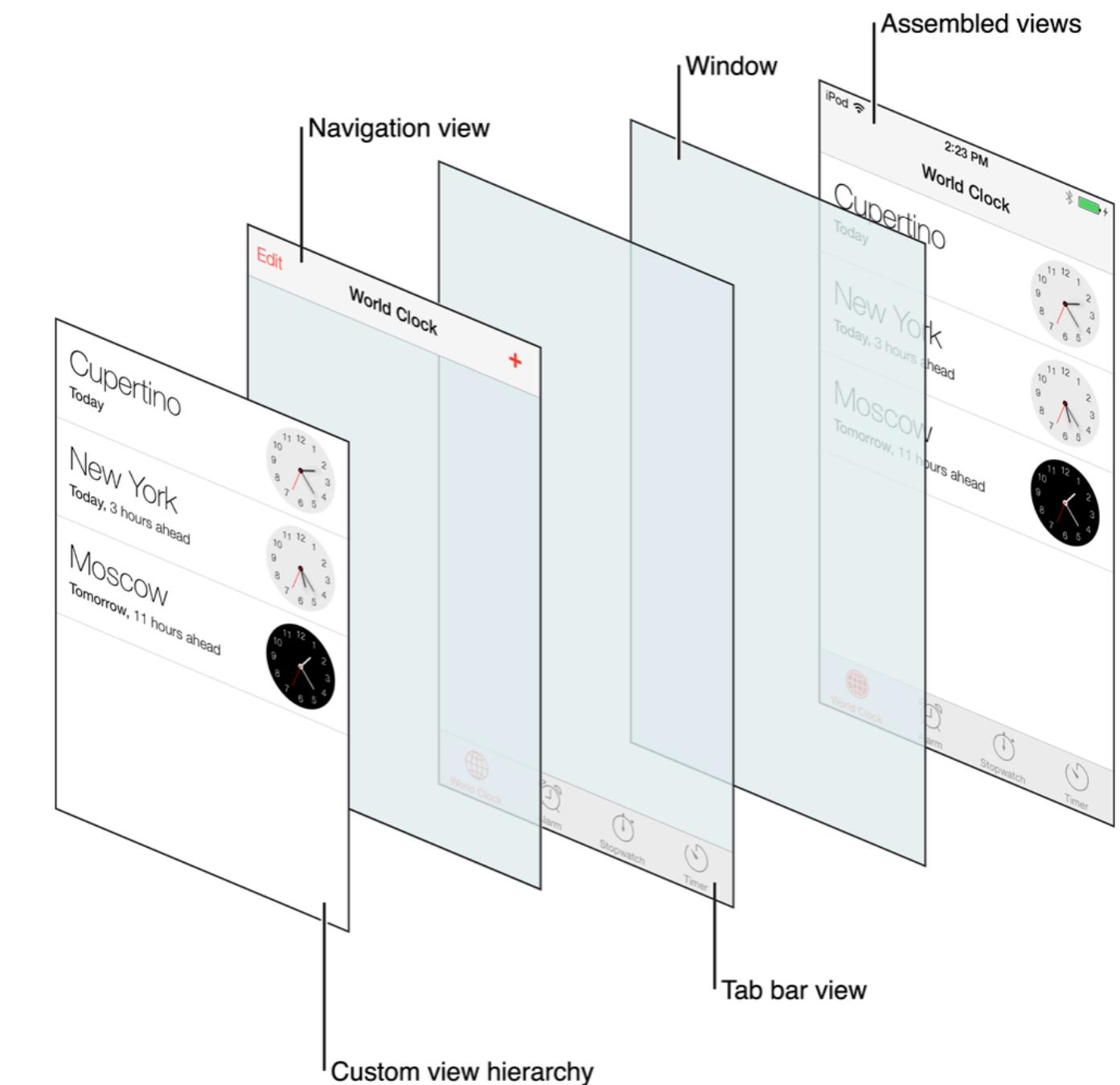
Software - Xcode⁴

- **Integrated Development Environment (IDE)**
 - Code Editor, Compiler, Debugger, etc.
- Includes Simulator and Testing tools

⁴ [Xcode Homepage](#)

User Interface Catalog⁵

- User interface elements are an app's building blocks.
- **Views** (Viewable)
 - Labels, Tables, Alerts, Images, etc.
- **Controls** (Interactable)
 - Buttons, Sliders, Switches, etc.



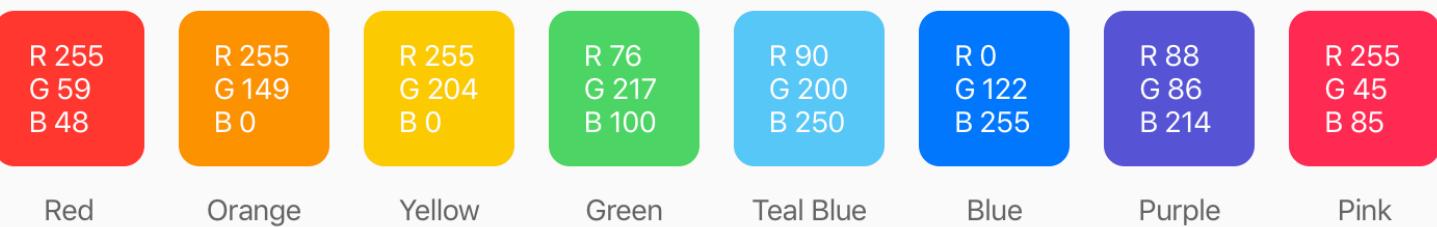
⁵ [UIKit User Interface Catalog Homepage](#)

Human Interface Guidelines (HIG)⁶

- Design Recommendations
- Design Resources¹³

Color

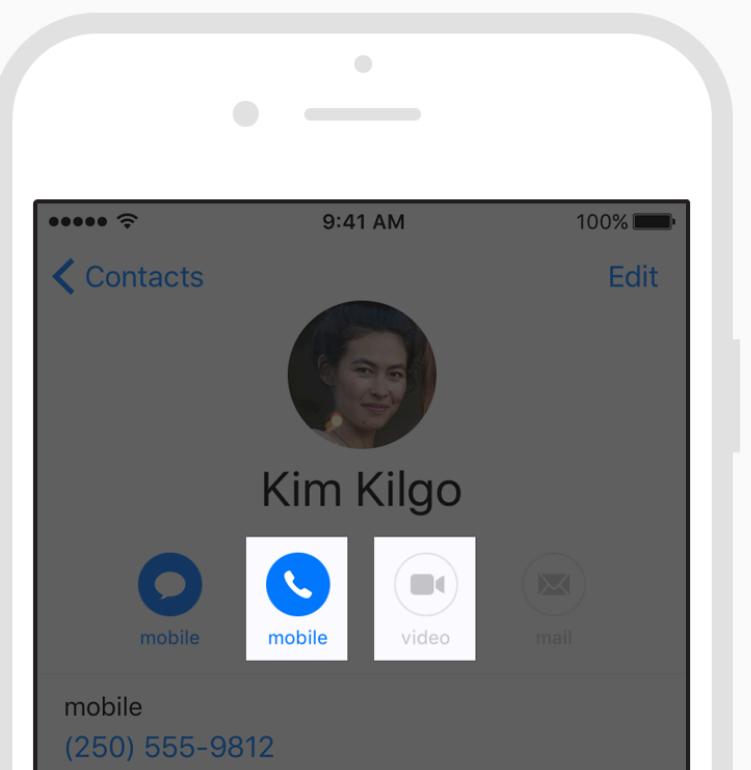
In iOS, color can indicate interactivity, impart vitality, and provide visual continuity. Look to the system's color scheme for guidance when picking app tint colors that look great individually and in combination, on both light and dark backgrounds.



Use complementary colors throughout your app. The colors in your app should work well together, not conflict or distract. If pastels are essential to your app's style, for example, use a coordinating set of pastels.

In general, choose a limited color palette that coordinates with your app logo. Subtle use of color is a great way to communicate your brand.

Consider choosing a key color to indicate interactivity throughout your app. In Notes, interactive elements are yellow. In Calendar, interactive elements are red. If you define a key color that denotes interactivity, make sure other colors don't compete with it.



Avoid using the same color for interactive and noninteractive elements. If interactive and noninteractive elements have the same color, it's hard for people to know where to tap.

⁶ [iOS Human Interface Guidelines Homepage](#)

¹³ [Swift Playgrounds for iPad and Xcode](#)

Summary

- Initial investment is \$1,000+ for hardware.
- Create a free Apple Developer Account.
- Read the UIKit User Interface Catalog.
- Read The Human Interface Guidelines (HIG).

Learn to Code

Programming Languages (Native)

- **Two** languages to choose from:
 - Swift
 - Objective-C
- Which one should I choose?

“Swift is a successor to the C and
Objective-C languages.”

~ Apple

Learning Swift

- Everyone Can Code Series by Apple⁸
 - Intro to App Development w/Swift⁹
 - App Development with Swift
- Stanford's CS193P on iTunesU¹⁰



⁸ [Everyone Can Code Series on iBooks](#)

⁹ [Intro to App Development with Swift on iBooks](#)

¹⁰ [CS193P on iTunes U](#)

Programming Languages (Hybrid)

Apps can also be built using:

- **Web Languages (JavaScript + HTML + CSS)**
- React Native, PhoneGap
- **C Sharp (C#)**
- Xamarin
- **Ruby**
- Ruby Motion

I cannot comment on the hybrid approach as it's not my forté.

Stuck Programming?

- Stack Overflow¹¹
 - Have a problem? Ask here! Provide some sample code.
 - 99% chance that a similar problem has already been solved.
- Apple Developer Forums¹²

¹¹ [Stack Overflow](#)

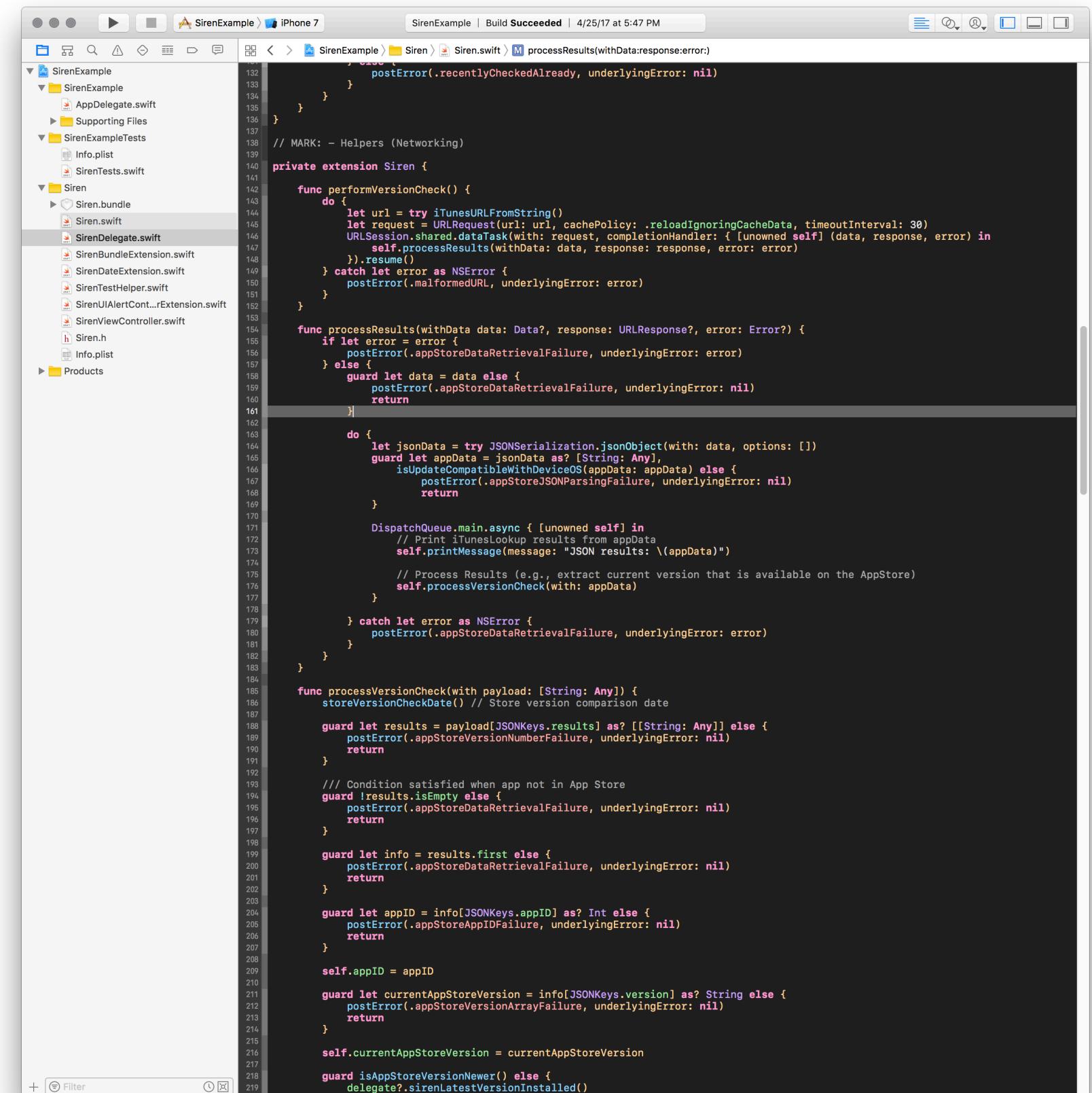
¹² [Apple Developer Forums](#)

Your First App

Remember Xcode? 😊

- Write Code
- Test Code
- Debug Code
- Submit Apps to App Store
- Also, Playgrounds!¹³

¹³ [Swift Playgrounds for iPad and Xcode](#)



The screenshot shows the Xcode interface with the project 'SirenExample' open. The left sidebar displays the file structure, including files like AppDelegate.swift, Supporting Files, SirenExampleTests, Info.plist, Siren, SirenDelegate.swift, Siren.bundle, Siren.swift, SirenBundleExtension.swift, SirenDateExtension.swift, SirenTestHelper.swift, SirenUIAlertControllerExtension.swift, SirenViewController.swift, and Siren.h. The main editor window shows the code for the Siren.swift file, specifically the processResults(withData:response:error:) function. The code handles various error cases, such as malformed URLs and JSON parsing failures, and performs version checks against the App Store.

```
private extension Siren {
    func performVersionCheck() {
        do {
            let url = try iTunesURLFromString()
            let request = URLRequest(url: url, cachePolicy: .reloadIgnoringCacheData, timeoutInterval: 30)
            URLSession.shared.dataTask(with: request, completionHandler: { [unowned self] (data, response, error) in
                self.processResults(data: data, response: response, error: error)
            }).resume()
        } catch let error as NSError {
            postError(.malformedURL, underlyingError: error)
        }
    }

    func processResults(withData data: Data?, response: URLResponse?, error: Error?) {
        if let error = error {
            postError(.appStoreDataRetrievalFailure, underlyingError: error)
        } else {
            guard let data = data else {
                postError(.appStoreDataRetrievalFailure, underlyingError: nil)
                return
            }

            do {
                let jsonData = try JSONSerialization.jsonObject(with: data, options: [])
                guard let appData = jsonData as? [String: Any],
                      isUpdateCompatibleWithDeviceOS(appData: appData) else {
                    postError(.appStoreJSONParsingFailure, underlyingError: nil)
                    return
                }

                DispatchQueue.main.async { [unowned self] in
                    // Print iTunesLookup results from appData
                    self.printMessage(message: "JSON results: \(appData)")

                    // Process Results (e.g., extract current version that is available on the AppStore)
                    self.processVersionCheck(with: appData)
                }
            } catch let error as NSError {
                postError(.appStoreDataRetrievalFailure, underlyingError: error)
            }
        }
    }

    func processVersionCheck(with payload: [String: Any]) {
        storeVersionCheckDate() // Store version comparison date

        guard let results = payload[JSONKeys.results] as? [[String: Any]] else {
            postError(.appStoreVersionNumberFailure, underlyingError: nil)
            return
        }

        /// Condition satisfied when app not in App Store
        guard !results.isEmpty else {
            postError(.appStoreDataRetrievalFailure, underlyingError: nil)
            return
        }

        guard let info = results.first else {
            postError(.appStoreDataRetrievalFailure, underlyingError: nil)
            return
        }

        guard let appID = info[JSONKeys.appID] as? Int else {
            postError(.appStoreAppIDFailure, underlyingError: nil)
            return
        }

        self.appID = appID

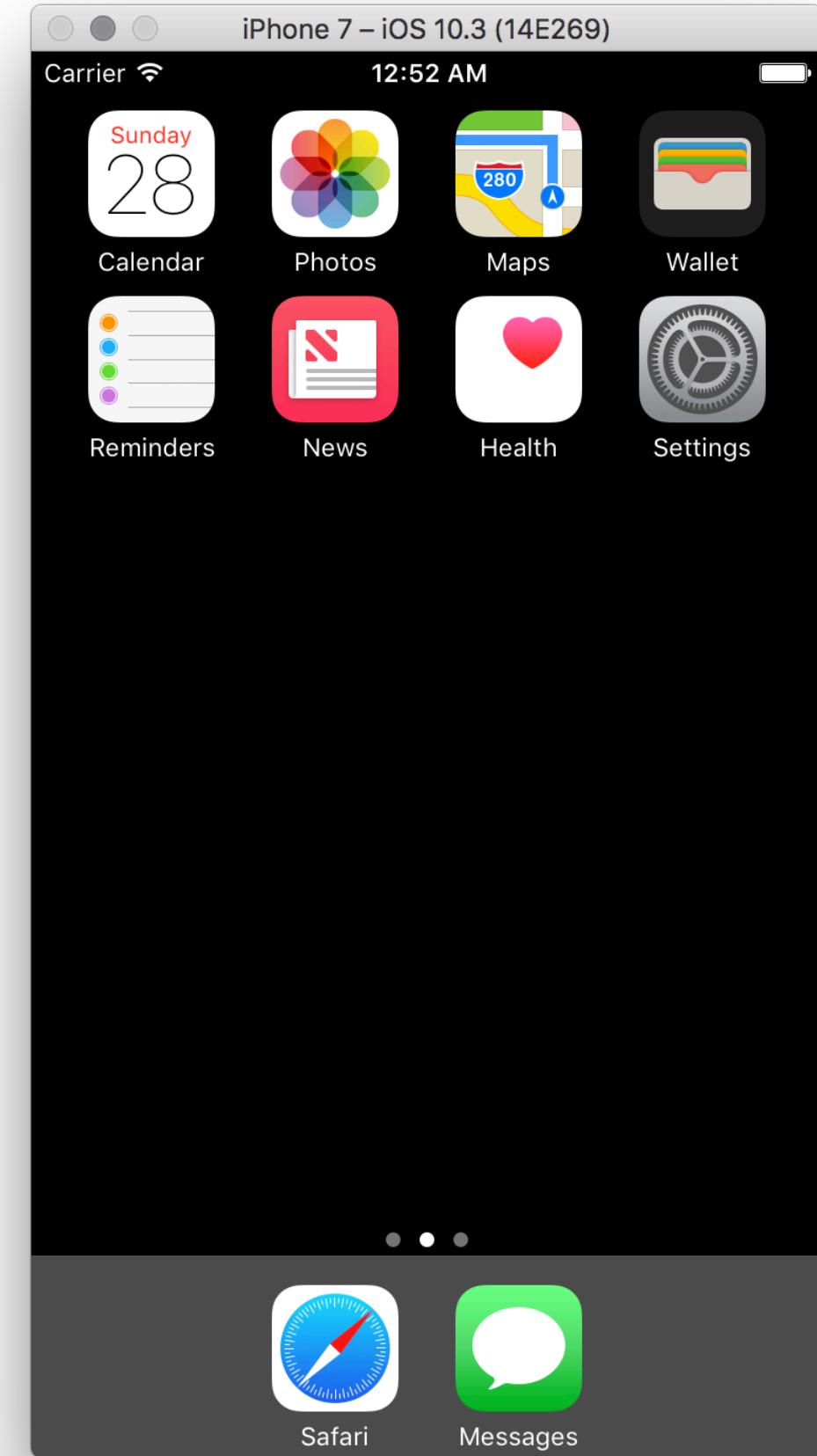
        guard let currentAppStoreVersion = info[JSONKeys.version] as? String else {
            postError(.appStoreVersionArrayFailure, underlyingError: nil)
            return
        }

        self.currentAppStoreVersion = currentAppStoreVersion

        guard isAppStoreVersionNewer() else {
            delegate?.sirenLatestVersionInstalled()
        }
    }
}
```

The iOS Simulator¹⁴

- Simulate all devices on your computer:
 - iPhone
 - iPad
 - Apple Watch
 - Apple TV
- Simulate Location, Notifications, Connectivity Issues, etc.



¹⁴ Bundled with Xcode

Summary

- Learn Swift for free from Apple or Stanford.
- Use Swift to build your apps.
- If you get stuck, ask Stack Overflow for help.
- Use Xcode to write, test, and deploy your apps.

The iOS Community

Worldwide Developers Conference (WWDC)

Part 1 of 2

- **Every June, Apple holds a conference in California.**
 - 5000+ Developers invited via lottery
 - 1000 Apple Engineers
- **Announcements**
 - New versions of iOS, macOS, watchOS, and tvOS.
 - Occasionally, hardware refreshes.

Worldwide Developers Conference (WWDC)

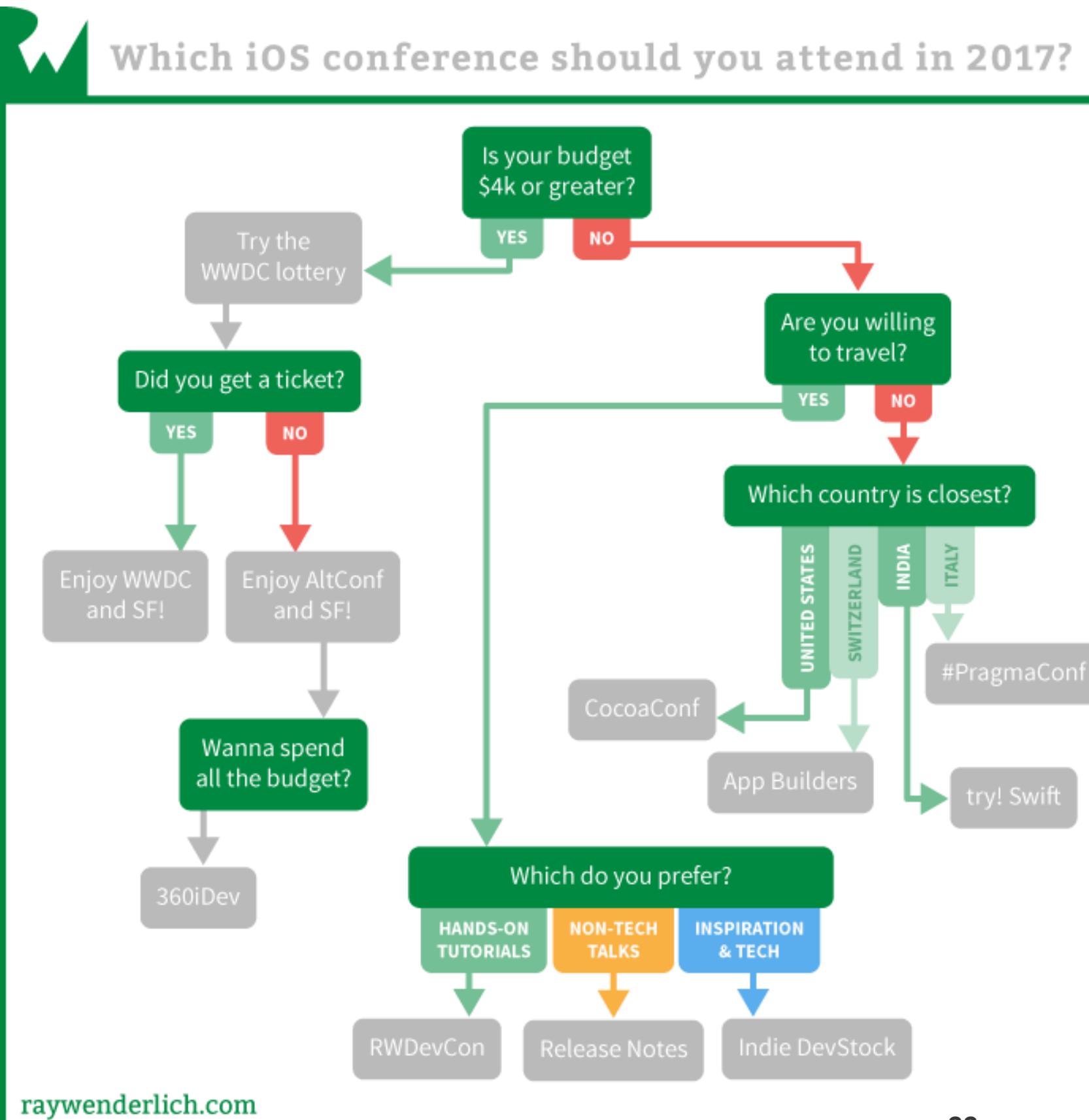
Part 2 of 2

- **200+ Lectures/Talks**
 - Weeklong event unveiling forthcoming improvements for developers in iOS, macOS, etc.
- **Livestreams**
 - Keynote is livestreamed.
 - Some talks are livestreamed.
 - Video and PDF transcripts made available after each talk.
- **Assistance from Apple Engineers**
 - 1-on-1 Design Meetings.
 - Code assistance from the engineers building the APIs.

Other Conferences¹⁵

- [360|iDev](#) - Indy Development
- [AltConf](#) - Free
- [App Builders](#) - Switzerland
- [CocoaConf](#) - Traveling Conference w/Tutorials
- [FrenchKit](#) - France
- [iOSDevCampDC](#) - Technical Conference at Capital One in Virginia
- [iOSDevUK](#) - Held at a University in the UK
- [Pragma Conference](#) - Italy
- [Release Notes](#) - Business of App Development in Chicago
- [RWDevCon](#) - Hands on Tutorials
- [SwiftAlps](#) - Switzerland
- [SwiftSummit](#) - San Francisco
- [try! Swift](#) - Traveling Conference
- [UIKonf](#) - Berlin

¹⁵ [Ray Wenderlich: Top 10 iOS Conferences](#)



Local Meetups (DC, MD, VA)

- [Baltimore iOS Development Meetup](#)
- [CocoaHeads DC](#)
- [Columbia iOS Development Meetup](#)
- [ModevDC](#)

Podcasts

- [Core Intuition](#) - Indy Development
- [Release Notes](#) - Indy Development
- [Runtime](#) - General iOS & Swift Development
- [SwiftCoders](#) - Interviews with Swift Developers
- [Swift Unwrapped](#) - Deep-dive into Swift language development
- [Under the Radar](#) - Indy Development

Demo