***Swift vs Objective-C:***

* **About Objective C:**

<https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>

* “It’s a superset of the C programming language and provides object-oriented capabilities and a dynamic runtime. Objective-C inherits the syntax, primitive types, and flow control statements of C and adds syntax for defining classes and methods”
* Basically C with object oriented functionality
* Some classes provided by Cocoa and Cocoa Touch (others you define yourself)
* “Although Objective-C includes syntax for exception handling, Cocoa and Cocoa Touch use exceptions only for programming errors (such as out of bounds array access), which should be fixed before an app is shipped.

All other errors—including runtime problems such as running out of disk space or not being able to access a web service—are represented by instances of the NSError class. Your app should plan for errors and decide how best to handle them in order to present the best possible user experience when something goes wrong.”

* **Why Swift is better:**

[**https://www.wired.com/2014/07/apple-swift/**](https://www.wired.com/2014/07/apple-swift/)

* ‘faster and more effective means of building software apps for iPhones, iPads, and Macs”
* “Part of Swift’s edge is that it’s built for the average programmer. It’s designed for coding even the simplest of mobile apps, and with a rather clever tool Apple calls “Playgrounds,” it offers an unusually effective way of teaching yourself to code”
* “Swift isn’t just a language. It’s a language that’s tightly woven with everything developers need to build their software. This includes not only an integrated development environment, or IDE—an interface where coders can actually write their software—but also various other tools, such as a debugger that can help weed errors from their code.”
* “But it’s a big advance over Objective-C, a language that dates back to the mid-80s and, frankly, isn’t as easy to use as more modern languages”
* Swift is not a compromise between compiler-languages (objective C) and interpreted languages (Python) but rather the best of both worlds. No compile time, and runs quickly.

[**http://www.infoworld.com/article/2920333/mobile-development/swift-vs-objective-c-10-reasons-the-future-favors-swift.html**](http://www.infoworld.com/article/2920333/mobile-development/swift-vs-objective-c-10-reasons-the-future-favors-swift.html)

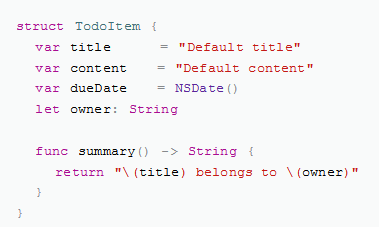
* Cleaner syntax makes code easier to read and debug
* Combines .h and .c into a single .swift file; able to instantly sort out dependencies
* Swift will detect bugs relating to using null variables; allowing the user to fix it right away (where Objective C will not detect the error and may lead to unpredictable behavior)
* Swift handles memory management; no risk of large memory loss
* Less code: can concatenate strings with “+”, compiler can figure out types, String interpolation to minimize errors
* “namespaces are based on the target that a code file belongs to” therefore you can incorporate other open source projects into your code without fear of same-name conflicts.
* Dynamic libraries so Swift can evolve faster
* Playgrounds to test out short code without creating a whole app; real-time feedback to facilitate efficiency

***Swift Programming:***

<https://www.makeschool.com/online-courses/tutorials/learn-swift-by-example-part-1-structs/structs-in-swift>

* Standard library mostly consists of structs rather than classes
* Structs = value type; Classes = reference type ( I think references are basically pointers)
* *Defining a STRUCT:*
  + struct TodoItem {
    - var title: String
    - var content: String
    - var dueDate: NSDate
    - let owner: String
    - }

*(“let” keyword = variable cannot be changed after creation)*

* *Initializing a STRUCT:*
  + var todoItem = TodoItem(title: “Get Milk”, content: “really urgent!”, dueDate: NSDate(), owner: “User1”)
* Swift structs can contain methods (difference between C and Swift)
  + **
* *‘Mutating’ keyword for methods to change member variables*

***iOS Limitations:***

The development of iOS applications requires the XCode IDE, which is only available for Mac OS. Because of this, iOS development requires an Apple computer. The other option is to install OS X and access it via a VM. HOWEVER, downloading Mac OS on non-Apple licensed devices is against Apple’s SLA (<http://www.apple.com/legal/sla/>). Also, uploading applications via iTunes may also require Mac OS.

***Reading Data from sqlite into XCode***

<https://xcode4all.wordpress.com/2011/05/26/reading-data-from-a-sqlite-database/>