[**Assignment 2**](https://bblearn.griffith.edu.au/webapps/blackboard/content/listContent.jsp?course_id=_48251_1&content_id=_2071246_1)

The purpose of this assignment is to write more advanced iOS and Swift applications. The assignment consists of a programming task and a number of exercises that will be made available incrementally as milestones (so check the milestones inside this folder regularly to make sure you won't miss out on your next milestone!) Milestones are due every week and need to be submitted in or before the lab they are due in.

**Some Hints**

* Use WKWebView to display web content (you can load a web page through the loadRequest() method)!
* In order to use WKWebView, you need to import WebKit in your project and link it against the WebKit framework!
* To create a URL request from a string, you can use the NSURLRequest(URL:) convenience initialiser, e.g.
  + NSURLRequest(URL: NSURL(string: "http://griffith.edu.au/")!)
* Saving data in property list format is easy, e.g. if you have an array of strings, you can save it using
  + array.writeToFile(fileName, atomically: true)
* Loading such data back is equally easy, e.g. if you want to create a mutable array from the saved array above, use
  + optionalArray = NSMutableArray(contentsOfFile: fileName);
* iOS uses a sandbox, so you can only load and save files from certain locations.  E.g. if you want to save to a file called data.plist, you can use

        let documentsDirectory = NSSearchPathForDirectoriesInDomains(.DocumentDirectory, .UserDomainMask, true)[0] as NSString

        let fileName = documentsDirectory.stringByAppendingPathComponent("data.plist")

* If the above does not seem to work for you for writing files, check that you have used .DocumentDirectory and **not** .DocumentationDirectory (which is read-only — this is a common mistake)!
* Core Data documents can be managed using the [managedObjectContext](http://developer.apple.com/library/ios/documentation/uikit/reference/UIManagedDocument_Class/Reference/Reference.html#//apple_ref/occ/instp/UIManagedDocument/managedObjectContext) property of [UIManagedDocument](http://developer.apple.com/library/ios/#documentation/uikit/reference/UIManagedDocument_Class/Reference/Reference.html)
* The [NSFetchedResultsController](http://developer.apple.com/library/ios/#documentation/CoreData/Reference/NSFetchedResultsController_Class/Reference/Reference.html) is a controller class that provides all the necessary methods you can use to update a table from a [UITableViewController](http://developer.apple.com/library/ios/#documentation/uikit/reference/UITableViewController_Class/Reference/Reference.html) subclass!
* more hints to come -- check back regularly!

### [Week 6 Reading Task (Milestone 1 of Assignment 2)](https://bblearn.griffith.edu.au/webapps/blackboard/content/launchLink.jsp?course_id=_48251_1&content_id=_2071247_1&mode=view)

#### Objective

Now it is time to get an understanding of the more advanced concepts of [the Swift programming language](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/).

You may not be very familiar with some of these concepts as many of them are unique to modern programming languages (such as Swift), but you won't find them in older languages (such as Java or C++).  While you won't need to memorise every single detail about the programming language syntax (this is what the reference manual is for), it is important that you read the material carefully to understand the concepts discussed.  You will need them when writing your own programs!

#### Sections to Read

All the reading material for this week is contained in both [the Swift Programming Language book (available for free online](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/) or as an [eBook](https://itunes.apple.com/au/book/swift-programming-language/id881256329?mt=11)) as well as the [Using Swift with Cocoa and Objective-C](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/BuildingCocoaApps/index.html) book (free [online](https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/BuildingCocoaApps/index.html#//apple_ref/doc/uid/TP40014216-CH2-XID_0) or as an [eBook](https://itunes.apple.com/au/book/using-swift-cocoa-objective/id888894773?mt=11)).  The labs are open this week and it is highly recommended that you use Xcode playgrounds to experiment with the concepts you are reading about (rather than just reading the book and trying to memorise its content).  Make sure you read the following sections this week (most subsections will be dark red to indicate more advanced concepts that you might not be so familiar with - read them very carefully to make sure you fully understand these concepts!):

##### Language Guide

If you haven't already done so, read the remainder of the Swift Language Guide, in particular, you may want to (re-)read the following sections to ensure you have a thorough understanding of the concepts discussed:

* [Functions](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Functions.html)
* [Closures](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Closures.html)
* [Enumerations](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Enumerations.html)
* [Optional Chaining](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/OptionalChaining.html)
* [Extensions](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Extensions.html)
* [Protocols](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Protocols.html)
* [Error Handling](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/ErrorHandling.html)
* [Generics](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Generics.html)
* [Access Control](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/AccessControl.html)
* [Advanced Operators](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/AdvancedOperators.html)

##### Using Swift with Cocoa and Objective-C

This book shows some of the more intricate concepts on how Swift is integrated in iOS.  Read the following sections:

* [Getting Started](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/BuildingCocoaApps/)
* [Interoperability](https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/BuildingCocoaApps/InteractingWithObjective-CAPIs.html#//apple_ref/doc/uid/TP40014216-CH4-XID_25)
  + [Interacting with Objective-C APIs](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/BuildingCocoaApps/InteractingWithObjective-CAPIs.html)
  + [Working with Cocoa Data Types](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/BuildingCocoaApps/WorkingWithCocoaDataTypes.html)
  + [Adopting Cocoa Design Patterns](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/BuildingCocoaApps/AdoptingCocoaDesignPatterns.html)
  + [Interacting with C APIs](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/BuildingCocoaApps/InteractingWithCAPIs.html)