XCode and Swift Development Environment

Cindy Royal, Ph.D
Mobile Media Development
School of Journalism and Mass Communication
Texas State University

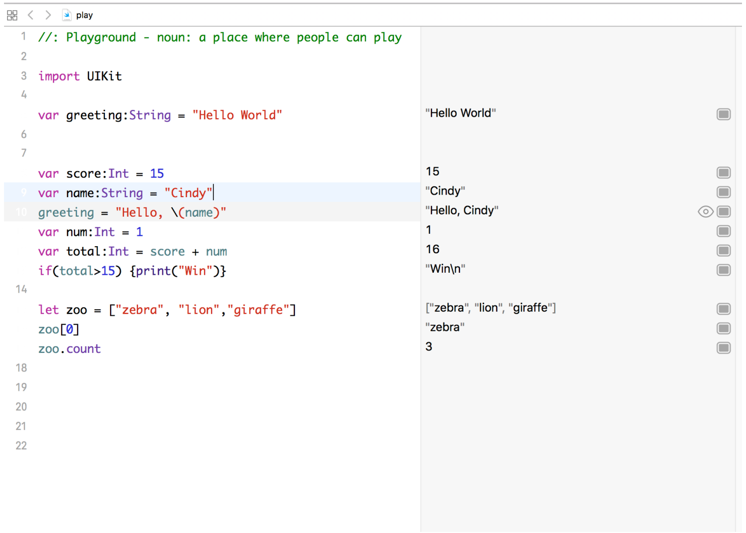
Versions

- iOS 13 – we’ll be developing for the current version
- XCode - if you have a Mac, download it from the App Store. Our computers already have it. It will then be found in your Applications folder. We are using XCode 11.3 at the time of setting up this lesson.
- Swift 5 is the current version of Swift.

XCode Playground

- When you open XCode, you have the choice to open a playground or a project. A playground lets you experiment with concepts.
- In a blank playground, play with the following code:

```
var greeting:String = "Hello World”
var score:Int = 15
var name:String = "Cindy"
greeting = "Hello, \(name)"
var num:Int = 1
var total:Int = score + num
```



```
//: Playground - noun: a place where people can play

import UIKit

var greeting:String = "Hello World"                "Hello World"


var score:Int = 15                                 15
var name:String = "Cindy"                          "Cindy"
greeting = "Hello, \(name)"                        "Hello, Cindy"
var num:Int = 1                                    1
var total:Int = score + num                        16
if(total>15) {print("Win")}                        "Win\n"

let zoo = ["zebra", "lion","giraffe"]              ["zebra", "lion", "giraffe"]
zoo[0]                                             "zebra"
zoo.count                                          3
```

Variables

Notes for naming variables:

- Names can't contain mathematical symbols.
- Names can't contain spaces.
- Names can't begin with a number

Camelcase – you will often see this in some styles of programming, where the first letter of a name of something is lowercase, but in other first letters are capitalized. You can decide what style is best for you, but there are some accepted standards.

Ex., var welcomeMessage

Programming Concepts

If Statements

```
if(total>15) {print("Win")}
```

Lists and Loops

```
let zoo = ["zebra", "lion","giraffe"]
zoo[0]
zoo.count
```

Comments

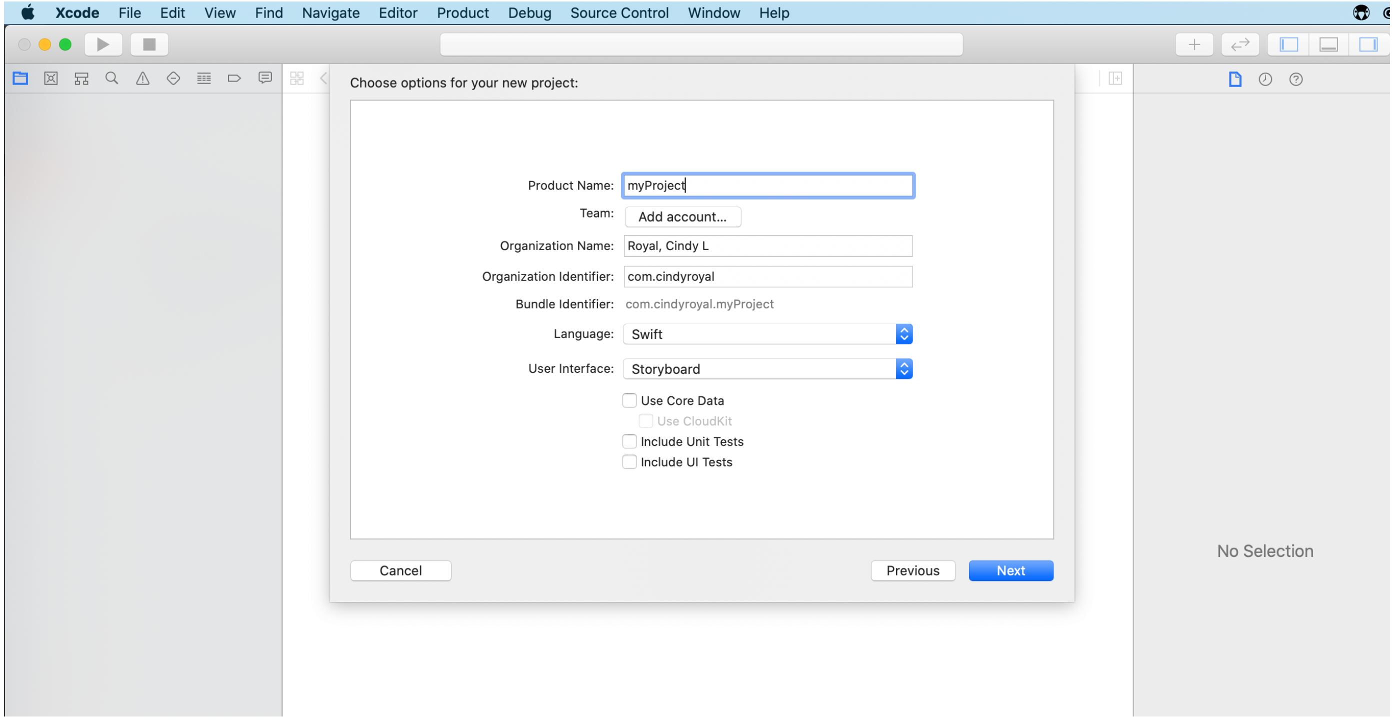
Use comments to ignore lines of code //

Types

String, Int, Double, Bool

New Project

Choose Single View App



Choose options for your new project:

Product Name: myProject

Team: Add account...

Organization Name: Royal, Cindy L

Organization Identifier: com.cindyroyal

Bundle Identifier: com.cindyroyal.myProject

Language: Swift

User Interface: Storyboard

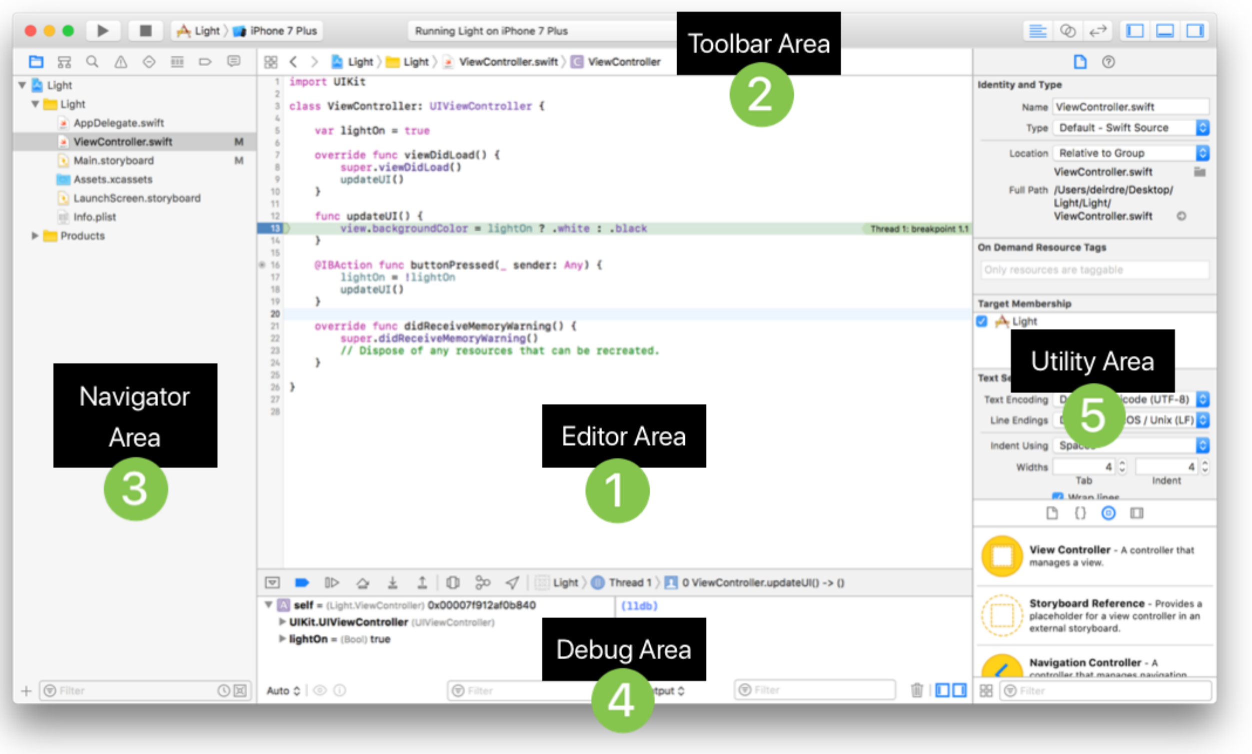
☐ Use Core Data

☐ Use CloudKit

☐ Include Unit Tests

☐ Include UI Tests

XCode Interface



```
import UIKit

class ViewController: UIViewController {

    var lightOn = true

    override func viewDidLoad() {
        super.viewDidLoad()
        updateUI()
    }

    func updateUI() {
        view.backgroundColor = lightOn ? .white : .black
    }

    @IBAction func buttonPressed(_ sender: Any) {
        lightOn = !lightOn
        updateUI()
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

}
```

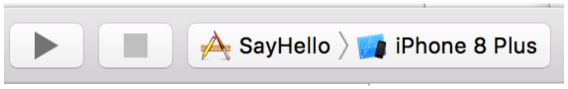
```
self = (Light.ViewController) 0x00007f912af0b840
  UIKit.UIViewController (UIViewController)
  lightOn = (Bool) true
```

```
(lldb)
```

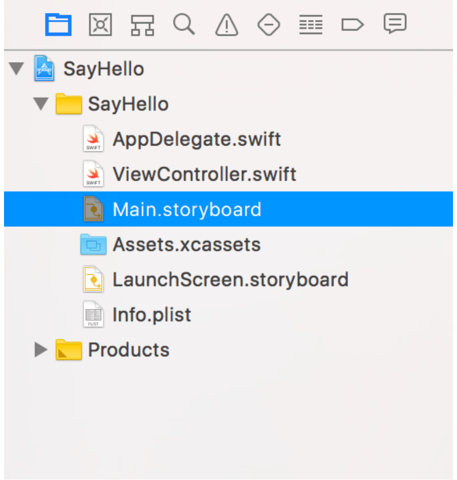
Name: ViewController.swift
Type: Default - Swift Source
Location: Relative to Group
ViewController.swift
Full Path: /Users/deirdre/Desktop/Light/Light/ViewController.swift

Running Light on iPhone 7 Plus
Light 〉 iPhone 7 Plus
Light 〉 Light 〉 ViewController.swift 〉 ViewController
Thread 1: breakpoint 1.1
Light 〉 Thread 1 〉 0 ViewController.updateUI() -> ()
Light
Light
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
Identity and Type
On Demand Resource Tags
Target Membership
View Controller - A controller that manages a view.
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.
Navigation Controller - A controller that manages navigation
Toolbar Area
2
Navigator Area
3
Editor Area
1
Debug Area
4
Utility Area
5

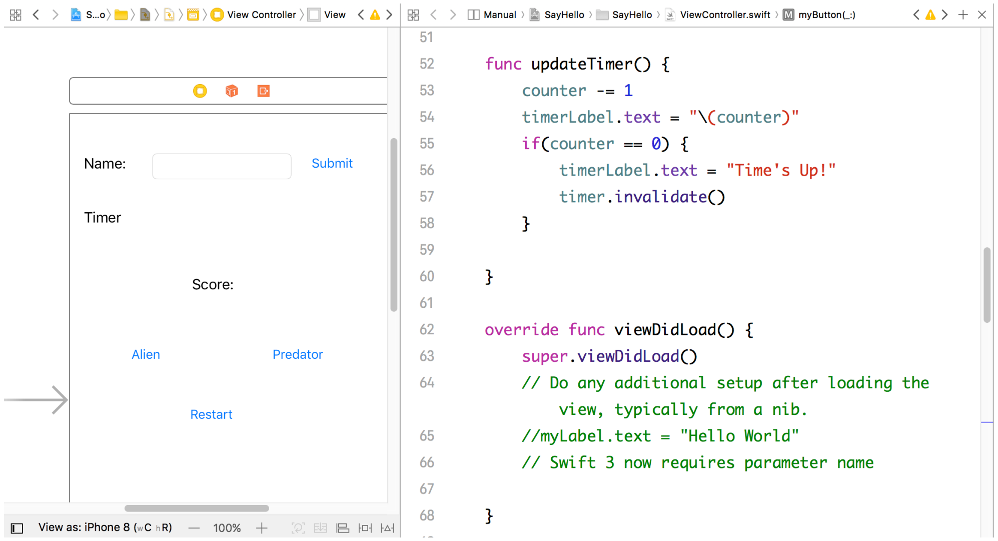
Library in Swift 5 is now a floating panel accessed by the Library button (+)



Toolbar

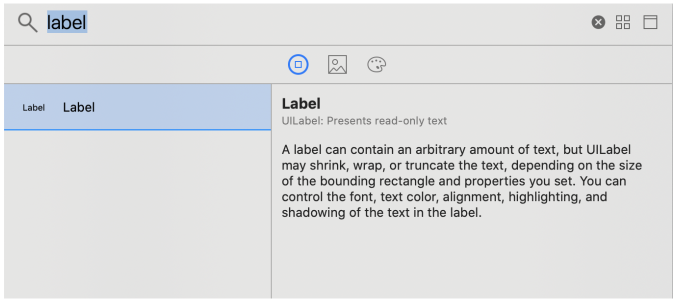


Navigator



```swift

    func updateTimer() {
        counter -= 1
        timerLabel.text = "\(counter)"
        if(counter == 0) {
            timerLabel.text = "Time's Up!"
            timer.invalidate()
        }

    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
        //myLabel.text = "Hello World"
        // Swift 3 now requires parameter name

    }
```

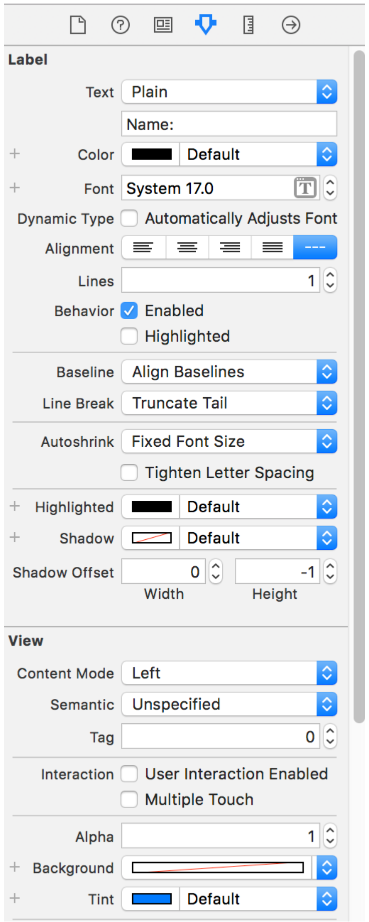
Editor – Storyboard and Code
Use option and then select the ViewController in the File Inspector to see both open in Editor



Label
UILabel: Presents read-only text

A label can contain an arbitrary amount of text, but UILabel may shrink, wrap, or truncate the text, depending on the size of the bounding rectangle and properties you set. You can control the font, text color, alignment, highlighting, and shadowing of the text in the label.

Library



Label

Text: Plain

Name:

Color: Default

Font: System 17.0

Dynamic Type: ☐ Automatically Adjusts Font

Alignment

Lines: 1

Behavior: ☑ Enabled
☐ Highlighted

Baseline: Align Baselines

Line Break: Truncate Tail

Autoshrink: Fixed Font Size
☐ Tighten Letter Spacing

Highlighted: Default

Shadow: Default

Shadow Offset: 0 (Width) -1 (Height)

View

Content Mode: Left

Semantic: Unspecified

Tag: 0

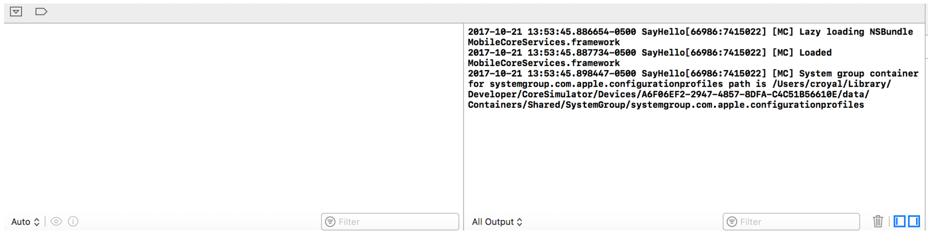
Interaction: ☐ User Interaction Enabled
☐ Multiple Touch

Alpha: 1

Background

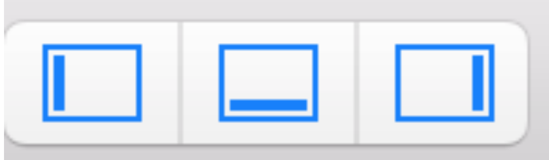
Tint: Default

Inspectors and Views



```
2017-10-21 13:53:45.886654-0500 SayHello[66986:7415022] [MC] Lazy loading NSBundle
MobileCoreServices.framework
2017-10-21 13:53:45.887734-0500 SayHello[66986:7415022] [MC] Loaded
MobileCoreServices.framework
2017-10-21 13:53:45.898447-0500 SayHello[66986:7415022] [MC] System group container
for systemgroup.com.apple.configurationprofiles path is /Users/croyal/Library/
Developer/CoreSimulator/Devices/A6F06EF2-2947-4857-8DFA-C4C51B56610E/data/
Containers/Shared/SystemGroup/systemgroup.com.apple.configurationprofiles
```

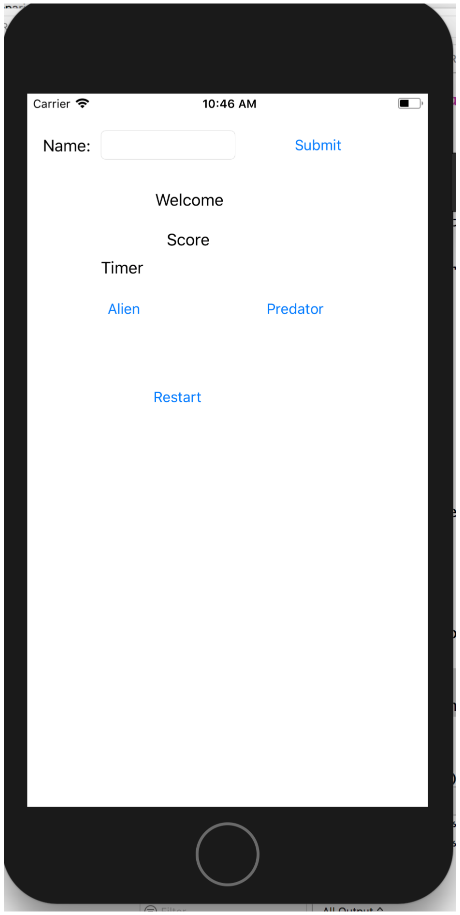
Debug



Toggle panels

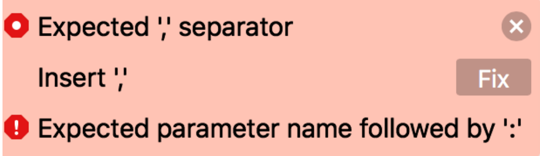
Simulator

- When you press the run button (triangle in top left of interface), the Simulator is launched to show you what your app looks like and how it works.
- Change Simulator to the type of phone you want to make your app using the main toolbar. Then choose the Run button. You will see the Simulator and then the app launches.
- If you need to change the size of the device in the Simulator, just click and drag on the edges.
- Make sure you click the stop button (square) after each run session.



Errors

- As you are working on your app, you may (you will!) see icons with exclamation marks in them. They are either warnings (yellow) or errors (red). In some cases, you may have to debug them. We will address as we go. These error messages will be an important part of debugging and troubleshooting.



Expected ',' separator

Insert ','

Expected parameter name followed by ':'

- As you are coding, if you make a mistake and need to start a section over, make sure you delete any connections in the Connections Inspector on the right panel.

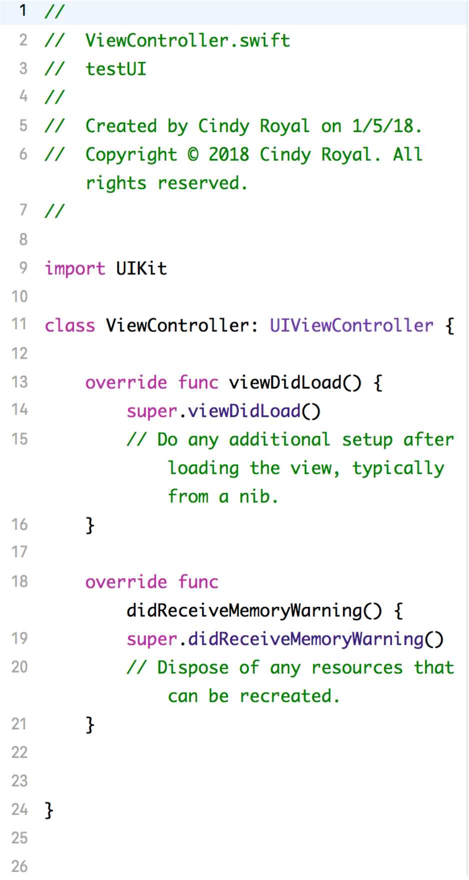
Labels

- From Library, select and drag Label
- Click ctrl and drag from label into the ViewController section.

  @IBOutlet weak var myLabel: UILabel!

- This creates a "connection" from the label to the code. You can see this connection defined in the Connections Inspector on the right.
- Can then give the label a value

  myLabel.text = "Hello World" – put this code in the viewDidLoad method



```
//
//  ViewController.swift
//  testUI
//
//  Created by Cindy Royal on 1/5/18.
//  Copyright © 2018 Cindy Royal. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


}


```

Text Fields

- Allows user input
- Search the Library for a Text Field and drag to Storyboard
- Ctrl and drag to make a connection in the code

```
IBOutlet weak var myName: UITextField!
```

Buttons

- Find a button in the Library and drag to the Storyboard
- Ctrl and drag to code

```
@IBAction func myButton(_ sender: Any) {

 }
```

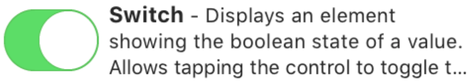
- Creates a function for the button in which you can insert commands to happen when the user engages with the button. This also creates a “connection” that can be viewed in the Connections Inspector.

UI Elements

- Images – drag image to Assets.xcassets (find in file navigator)

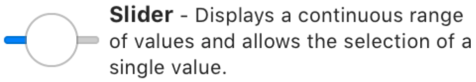
  Find an Image View in the Library and drag that to the screen. You can then pick the image from the Utilities area with the Attributes Inspector. Size that image appropriately for the screen.

- Switch



Switch - Displays an element showing the boolean state of a value. Allows tapping the control to toggle t...

- Slider



Slider - Displays a continuous range of values and allows the selection of a single value.

Onward

- That’s a basic overview of how XCode and Swift work together
- We’ll be using these concepts and more for the next several classes to go through some exercises to practice.
- Note: for OSX Catalina, there is a new method of developing Swift interfaces called SwiftUI. We will be using the Storyboard method in this class this semester, but be ready for this update to change the way iOS apps are developed in the future.
- Stay focused, pay attention to where you are putting code, recognize error messages, try to solve problems!