

# iPhone App Dev

## Lesson 9

# Source

<https://github.com/bryanttang/iOS-Class-2015-6.git>

# Contact

[bryant.tang14mo@gmail.com](mailto:bryant.tang14mo@gmail.com)

# Summary

- **Concurrency Programming (Review)**
- **Swift**

# Swift

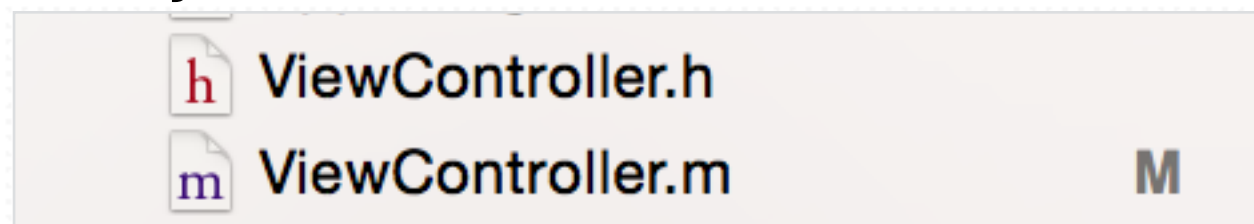


# Swift

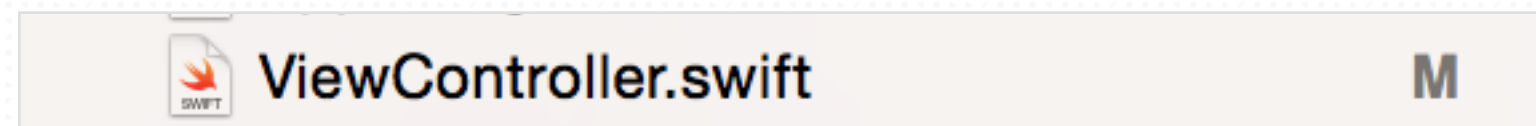
## Difference between Swift and Objective-C

- Coding syntax (JavaScript like)
- File structure

### Objective-C



### Swift



# Swift

## Mix and Match

- Import Objective-C into Swift
- Import Swift into Objective-C

# Swift

Import Swift into Objective-C

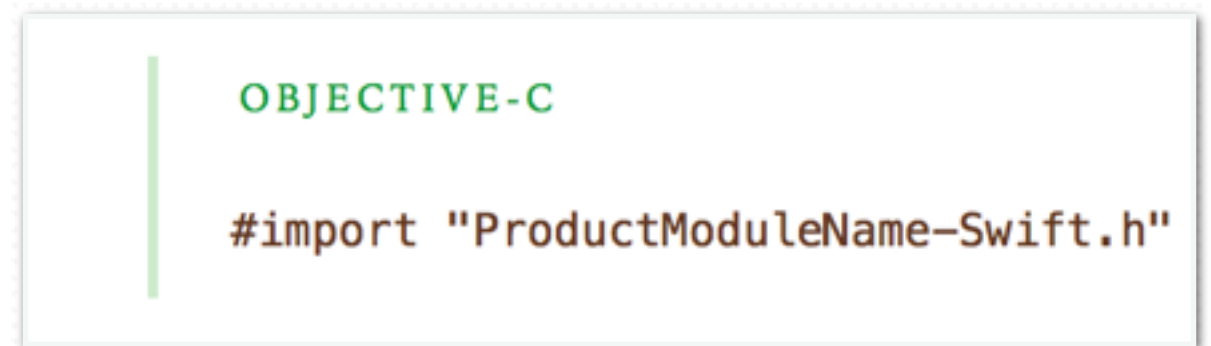
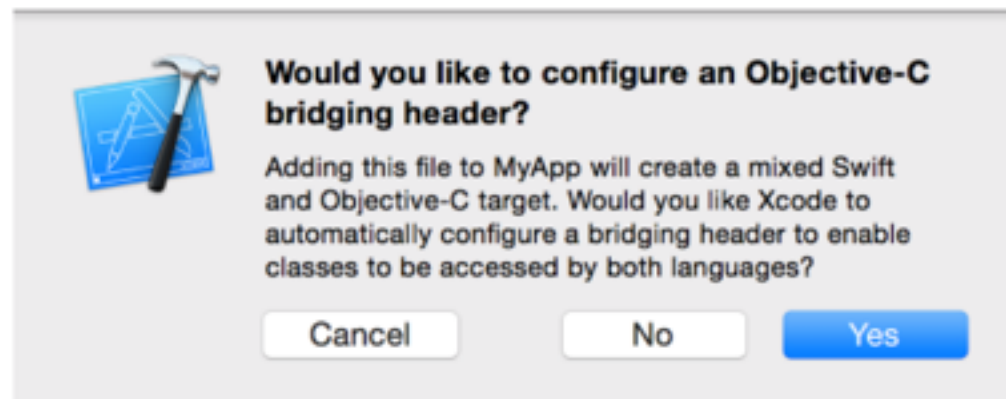
- Rely on an ***Xcode-generated header*** file
- Automatically generated file
- Name: **ProductName-Swift.h**



# Swift

## Import Swift into Objective-C

- New Swift or Import Swift file in your Target
- Allow Xcode creates the header file
- Import the header file (`ProductName-Swift.h`) in Class



# Swift

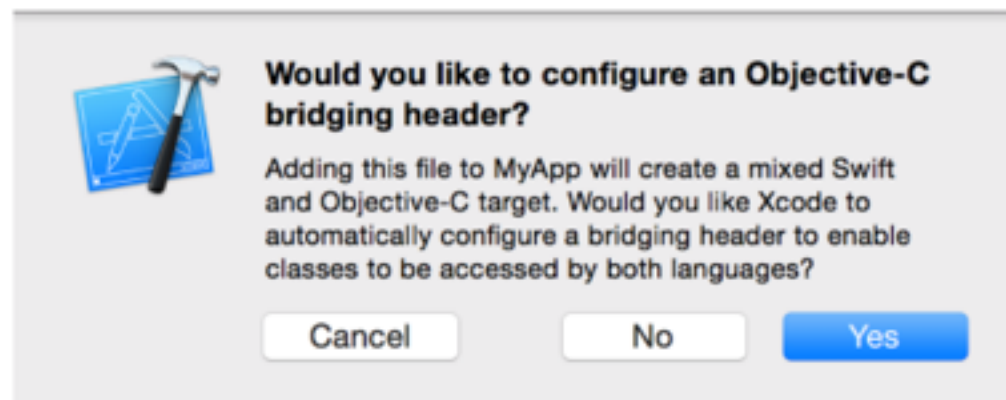
Import Objective-C into Swift

- Rely on an ***Objective-C bridging header***
- Have to create by yourself
- Name: `ProductName-Bridging-Header.h`

# Swift

## Import Objective-C into Swift

- New or Import Objective-C fileSwift file in your Target
- Allow Xcode creates the header file
- Import Class in the header file (`ProductName-Bridging-Header.h`)



```
OBJECTIVE-C
1  #import "XYZCustomCell.h"
2  #import "XYZCustomView.h"
3  #import "XYZCustomViewController.h"
```

Basic

# Basic

Import Objective-C into Swift

- Declare
- Function
- Create Class
- Condition statement
- Loop

# Declare

**var**

Declare as a variable

```
var carName : String
```

```
var carName = "BMW"
```

**Let**

Declare as a Constant

```
let carName = "BMW"
```

Optional

```
var carName : String?
```

*carName* contain some string

*carName* contain no value at all

# Function

Function without parameter

Input type

```
1 func sayGoodbye(personName: String) {  
2     println("Goodbye, \ (personName)!")  
3 }  
4 sayGoodbye("Dave")  
5 // prints "Goodbye, Dave!"
```

Function without parameter (**Objective-c**)

```
|  
- (void)sayGoodbye:(NSString*) personName{  
    NSLog(@"Goodbye, %@", personName);  
}
```

```
[self sayGoodbye:@"Dave"];
```



# Function

Function with multiple parameter

Output type

```
1 func halfOpenRangeLength(start: Int, end: Int) ->  
    Int {  
2     return end - start  
3 }  
4 println(halfOpenRangeLength(1, 10))  
5 // prints "9"
```

Function with multiple parameter (**Objective-c**)

```
- (int)halfOpenRangeLength:(int) start :(int)end{  
    return end - start;  
}
```

```
NSLog(@"%d", [self halfOpenRangeLength:1 :10]);
```



# Class

Define Properties and Methods

```
1 class Vehicle {
```

```
2     var numberOfWheels: Int
```

```
3     var maxPassengers: Int
```

```
4     func description() -> String {
```

```
5         return "\(numberOfWheels) wheels; up to \  
6         (maxPassengers) passengers"
```

```
7     }
```

```
8     init() {
```

```
9         numberOfWheels = 0
```

```
10        maxPassengers = 1
```

```
11    }
```

```
12 }
```

Properties

Methods

Initialize

# Condition statement

If

```
var temperatureInFahrenheit = 30
if temperatureInFahrenheit <= 32 {
    println("It's very cold. Consider wearing a scarf.")
}
// prints "It's very cold. Consider wearing a scarf."
```

# Condition statement

## Switch

```
let someCharacter: Character = "e"
switch someCharacter {
case "a", "e", "i", "o", "u":
    println("\(someCharacter) is a vowel")
case "b", "c", "d", "f", "g", "h", "j", "k", "l", "m",
    "n", "p", "q", "r", "s", "t", "v", "w", "x", "y", "z":
    println("\(someCharacter) is a consonant")
default:
    println("\(someCharacter) is not a vowel or a
        consonant")
}
// prints "e is a vowel"
```

# Loop

## Range

```
for index in 1...5 {  
    println("\(index) times 5 is \(index * 5)")  
}
```

## Collection

```
for item in shoppingList {  
    println(item)  
}
```

# Collection Types

- Array

```
var shoppingList = [String]() //empty array
```

```
var shoppingList = [String](count:3, repeatedValue: "abc")
```

```
var shoppingList : [String] = ["a" , "b", "c"]
```

Example:

```
for item in shoppingList {  
    println(item)  
}
```

# Collection Types

- Dictionaries

```
var namesOfIntegers = [Int: String]() //empty dictionary
```

```
var airports: [String: String] = ["YYZ": "Toronto Pearson",  
                                  "DUB": "Dublin"]
```

Example:

```
var airports = ["TYO": "Tokyo", "DUB": "Dublin"]
```

```
for (airportCode, airportName) in airports {  
    println("\(airportCode): \ (airportName)")  
}  
  
// TYO: Tokyo  
// LHR: London Heathrow
```

# Exercise: Calculator

# Exercise:

## MyTableViewController



# Reference

Swift

```
var people = ["Dave", "Brian", "Alex", "A  
let name = "Alex"  
if let index = find(people, name) {  
    println("\(name) is person \(index +  
    delegate?.didFindPersonWithName(name,  
} else {  
    println("Unable to find \(name) in +"
```



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



Please look for it at iTunes

5/27

Final Exam & Demo