iPhone App Dev

Lesson 9

Source

https://github.com/bryanttang/iOS-Class-2015-9.git

Contact

bryant.tang14mo@gmail.com

Summary

- Concurrency Programming (Review)
- Play Audio
- Swift

Play audio

- Two types of sound
  - Playing UI Sound Effects Using System Sound Services
  - Playing Sounds Easily with the AVAudioPlayer Class

Play audio

- Playing UI Sound Effects Using System Sound Services
  - No longer than 30 seconds in duration
  - In linear PCM or IMA4 (IMA/ADPCM) format
  - Packaged in a .caf, .aif, or .wav file

Play audio

- How to ?

1. Creating a sound ID object
2. Playing a system sound

Sample code

```
@property CFURLRef
soundFileURLRef;

@property   SystemSoundID
soundFileObject;
```

```
// Get the main bundle for the app
    CFBundleRef mainBundle = CFBundleGetMainBundle ();

// Get the URL to the sound file to play. The file in this case
// is "tap.aif"
    _soundFileURLRef  = CFBundleCopyResourceURL (
                                                mainBundle,
                                                CFSTR ("bullet_3"),
                                                CFSTR ("wav"),
                                                NULL
                                                );

    // Create a system sound object representing the sound file
    AudioServicesCreateSystemSoundID (
                                     _soundFileURLRef,
                                     &_soundFileObject
                                     );
```

Play audio

- Playing UI Sound Effects or Invoking Vibration Using System Sound Services
  - Play sounds of any duration
  - Play sounds from files or memory buffers
  - Loop sounds
  - Play multiple sounds simultaneously (although not with precise synchronization)
  - Control relative playback level for each sound you are playing
  - Seek to a particular point in a sound file, which supports application features such as fast forward and rewind
  - Obtain audio power data that you can use for audio level metering

Play audio

- The AVAudioPlayer class lets you play sound in any audio format available in iOS, as described in Table 1-1.

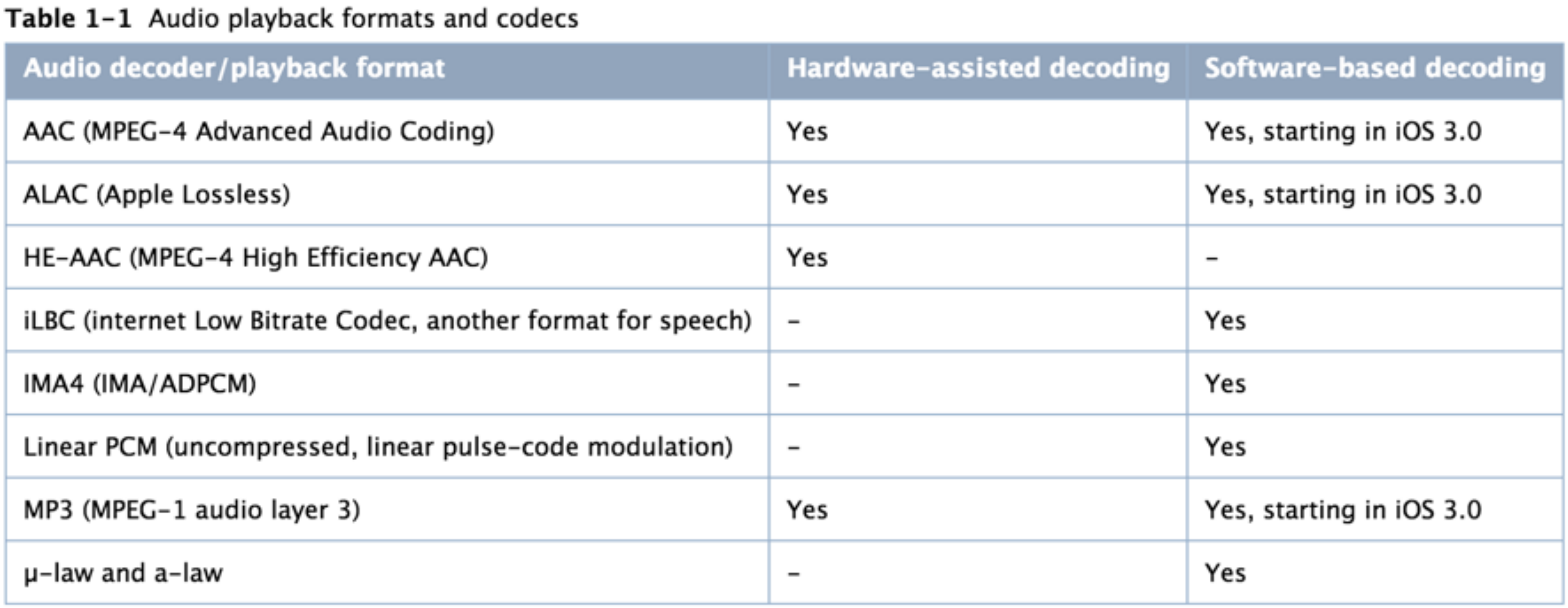


Table 1-1 Audio playback formats and codecs

Audio decoder/playback format	Hardware-assisted decoding	Software-based decoding
AAC (MPEG-4 Advanced Audio Coding)	Yes	Yes, starting in iOS 3.0
ALAC (Apple Lossless)	Yes	Yes, starting in iOS 3.0
HE-AAC (MPEG-4 High Efficiency AAC)	Yes	-
iLBC (internet Low Bitrate Codec, another format for speech)	-	Yes
IMA4 (IMA/ADPCM)	-	Yes
Linear PCM (uncompressed, linear pulse-code modulation)	-	Yes
MP3 (MPEG-1 audio layer 3)	Yes	Yes, starting in iOS 3.0
µ-law and a-law	-	Yes

Play audio

- How to ?

1. Define the sound path
2. Create an AVAudioPlayer object
3. Configure AVAudioPlayer object

Sample code

```
//Get Sound Path
NSString *soundFilePath = [[NSBundle mainBundle] pathForResource: @"bg_sound"
                                  ofType: @"wav"];
NSURL *fileURL = [[NSURL alloc] initFileURLWithPath: soundFilePath];
//Create an AVAudioPlayer
newPlayer = [[AVAudioPlayer alloc] initWithContentsOfURL: fileURL
                                        error: nil];
//Configuration
    newPlayer.delegate = self;
    newPlayer.volume = 0.5;
    [newPlayer setNumberOfLoops:-1];
    [newPlayer prepareToPlay];
```

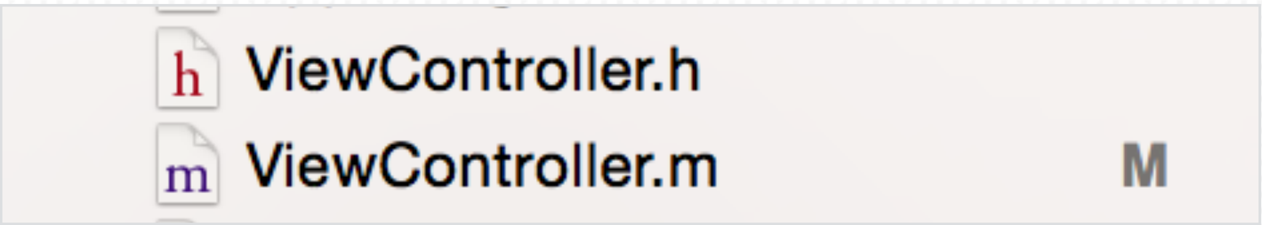
Swift

Swift

Difference between Swift and Objective-C

- Coding syntax (javaScript like)
- File structure

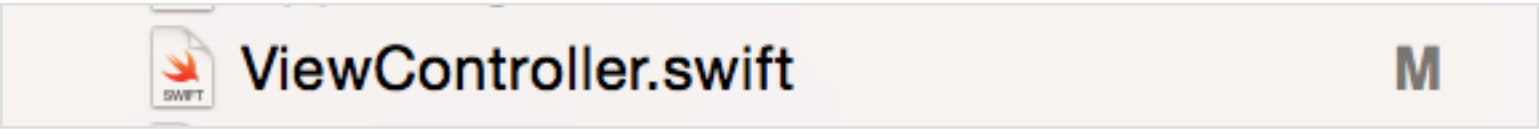
Objective-C



ViewController.h

ViewController.m M

Swift



ViewController.swift M

Swift

Mix and Match

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

Swift

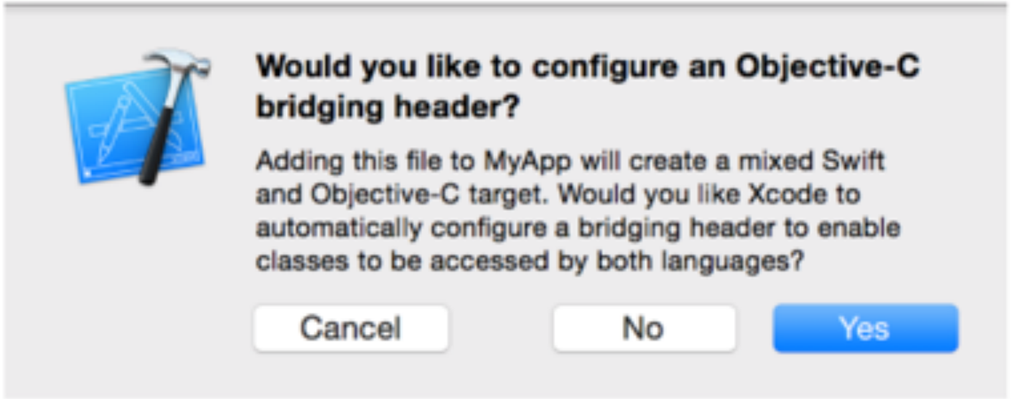
Import Swift into Objective-C

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

Swift

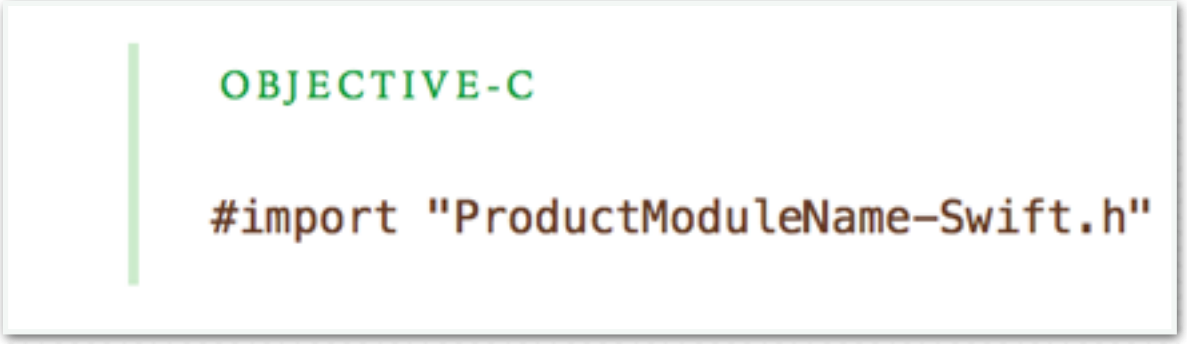
Import Swift into Objective-C

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



Would you like to configure an Objective-C bridging header?

Adding this file to MyApp will create a mixed Swift and Objective-C target. Would you like Xcode to automatically configure a bridging header to enable classes to be accessed by both languages?



OBJECTIVE-C

```
#import "ProductModuleName-Swift.h"
```

Swift

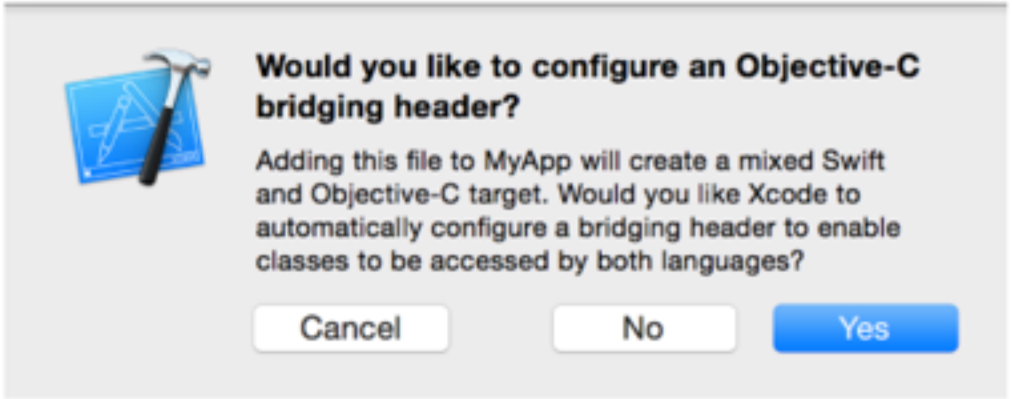
Import Objective-C into Swift

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

Swift

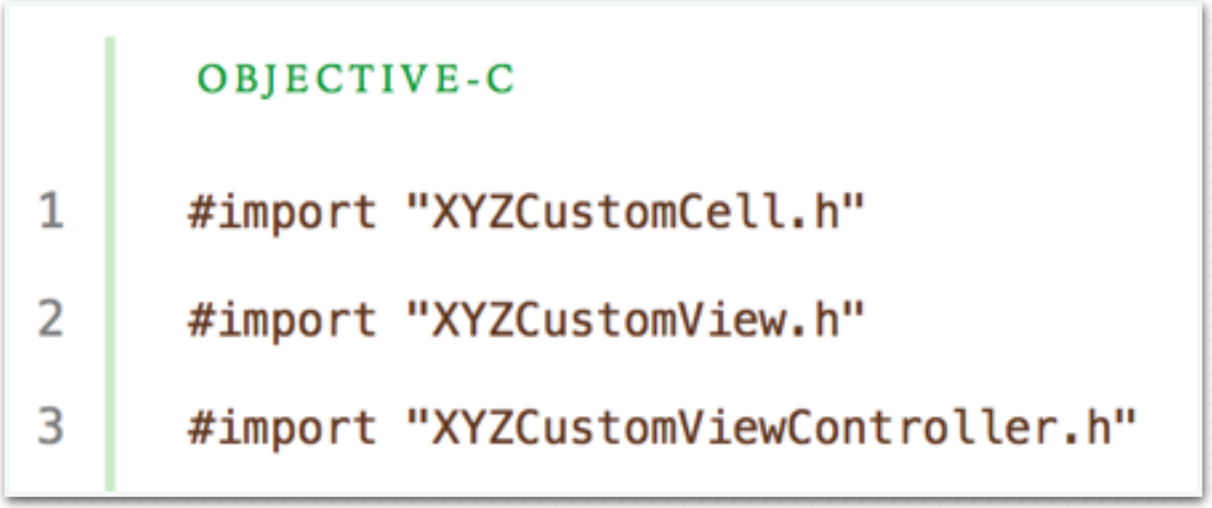
Import Objective-C into Swift

- 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)



Would you like to configure an Objective-C bridging header?

Adding this file to MyApp will create a mixed Swift and Objective-C target. Would you like Xcode to automatically configure a bridging header to enable classes to be accessed by both languages?



OBJECTIVE-C

```
#import "XYZCustomCell.h"
#import "XYZCustomView.h"
#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
```

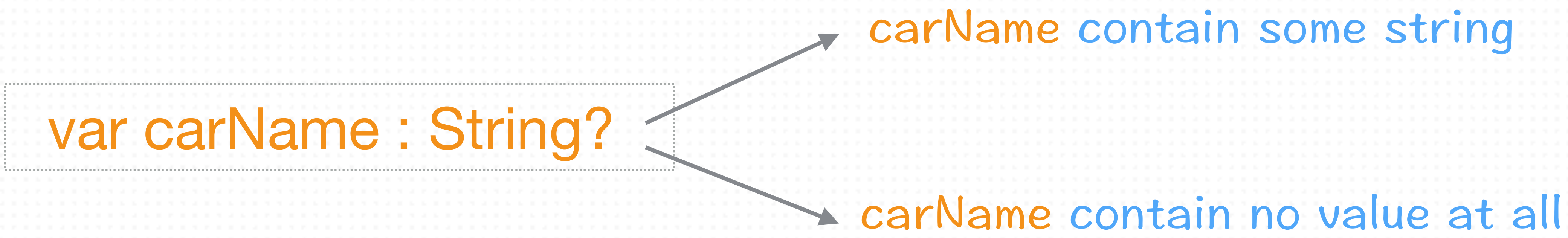
```
var carName = “BMW”
```

Let

Declare as a Constant

```
let carName = “BMW”
```

Optional



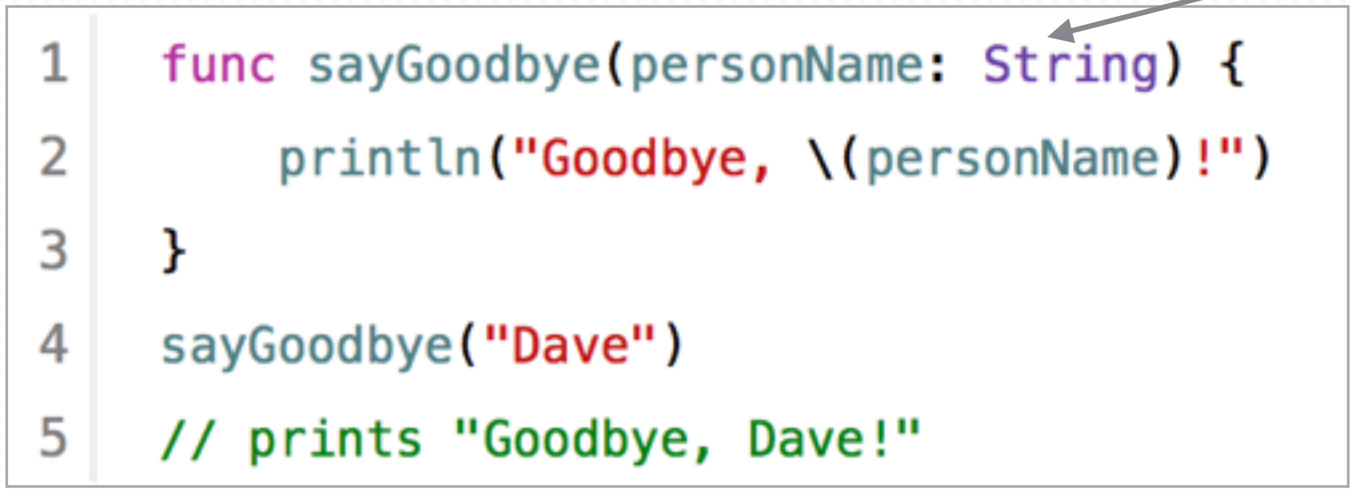
```
var carName : String?
```

carName contain some string

carName contain no value at all

Function

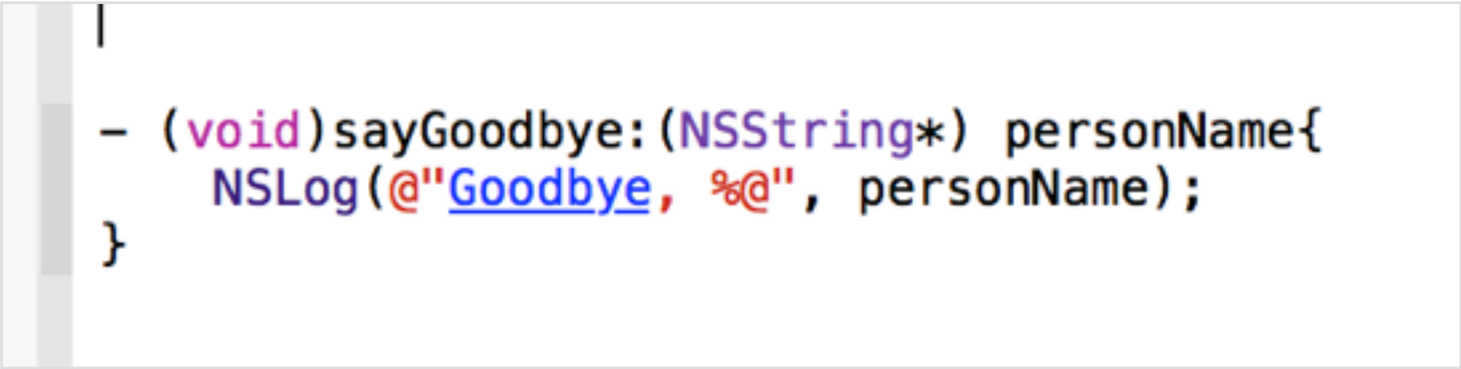
Function without parameter



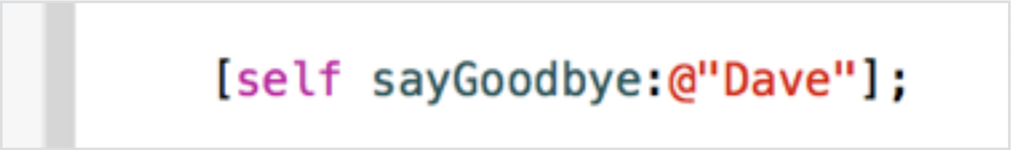
```swift
func sayGoodbye(personName: String) {
    println("Goodbye, \(personName)!")
}
sayGoodbye("Dave")
// prints "Goodbye, Dave!"
```

Input type

Function without parameter (Objective-c)



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

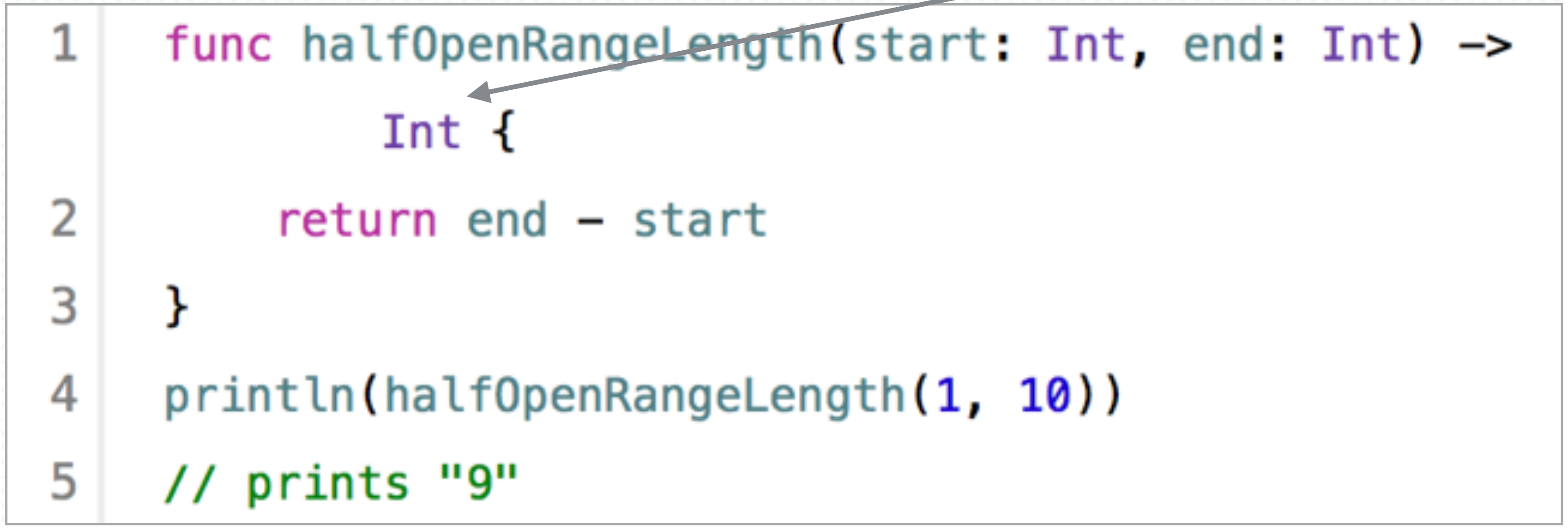


```objc
[self sayGoodbye:@"Dave"];
```

Function

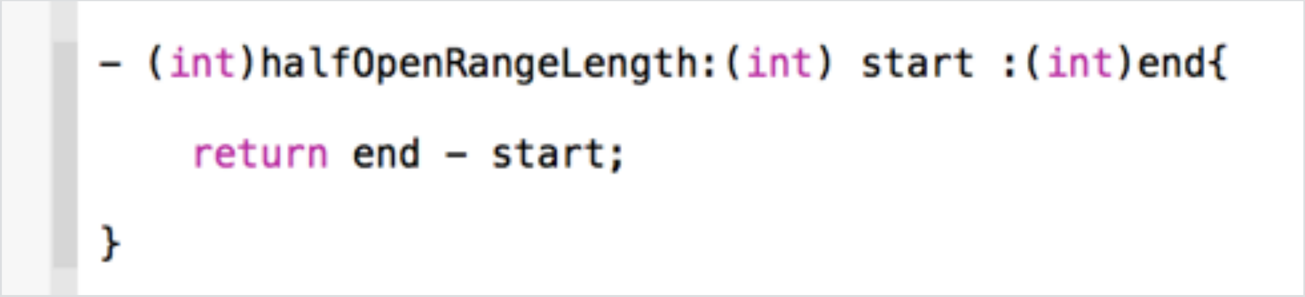
Function with multiple parameter

Output type

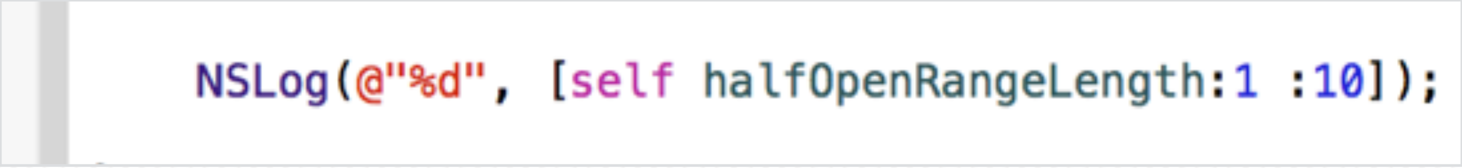


```swift
func halfOpenRangeLength(start: Int, end: Int) ->
        Int {
    return end - start
}
println(halfOpenRangeLength(1, 10))
// prints "9"
```

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



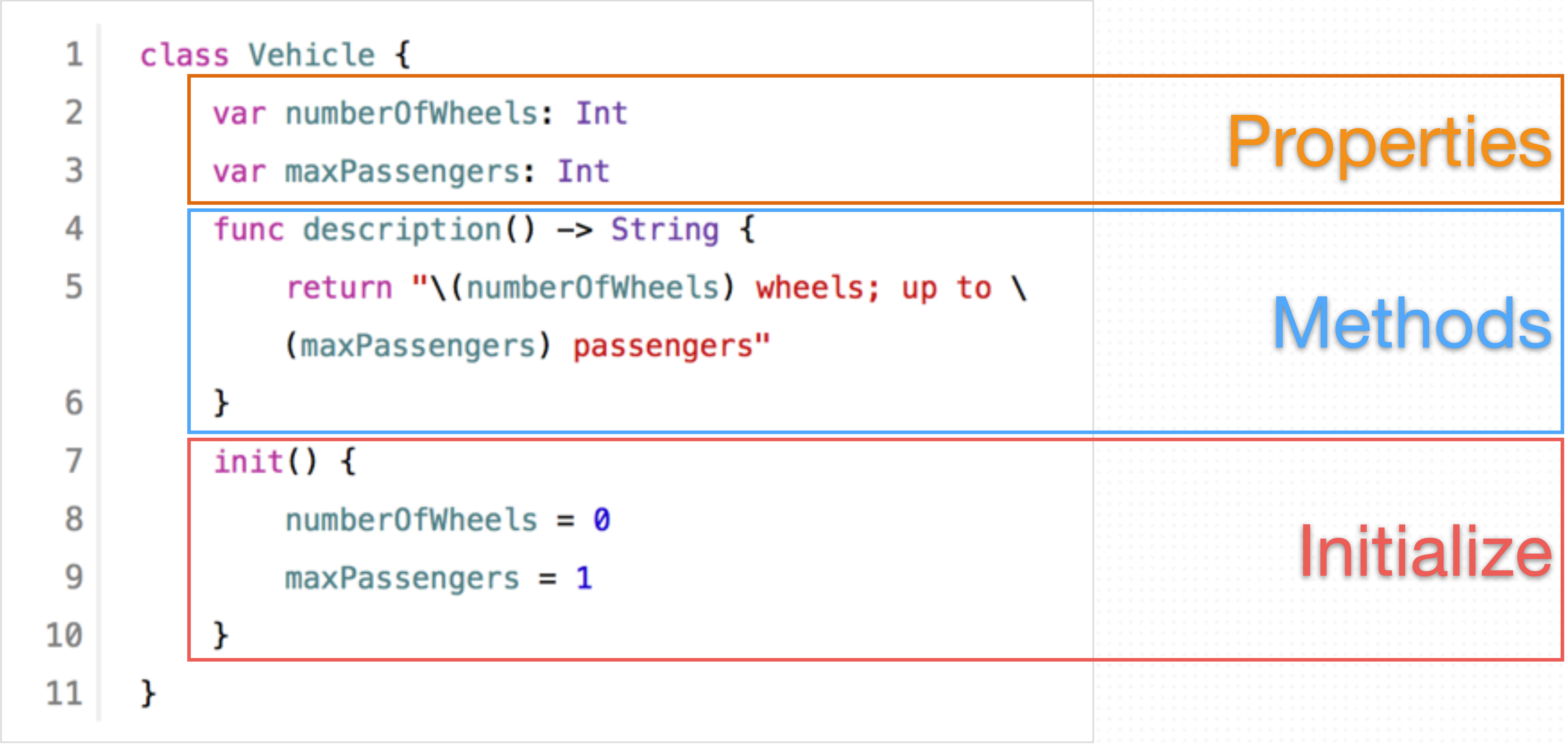
```objc
- (int)halfOpenRangeLength:(int) start :(int)end{
    return end - start;
}
```



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

Class

Define Properties and Methods



```
class Vehicle {
    var numberOfWheels: Int
    var maxPassengers: Int
    func description() -> String {
        return "\(numberOfWheels) wheels; up to \
        (maxPassengers) passengers"
    }
    init() {
        numberOfWheels = 0
        maxPassengers = 1
    }
}
```

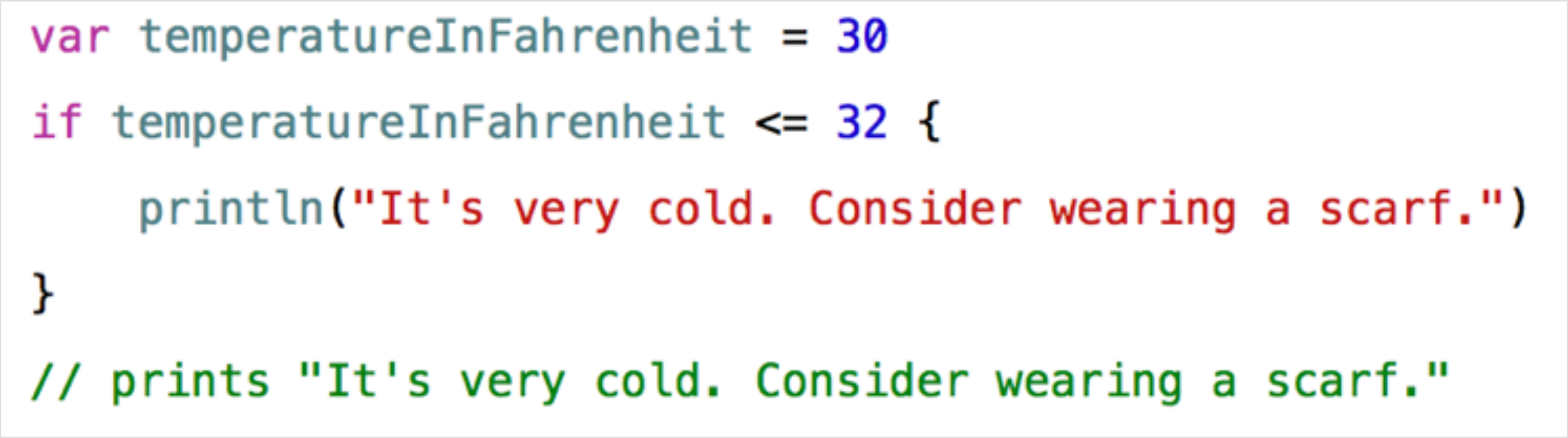
Properties

Methods

Initialize

Condition statement

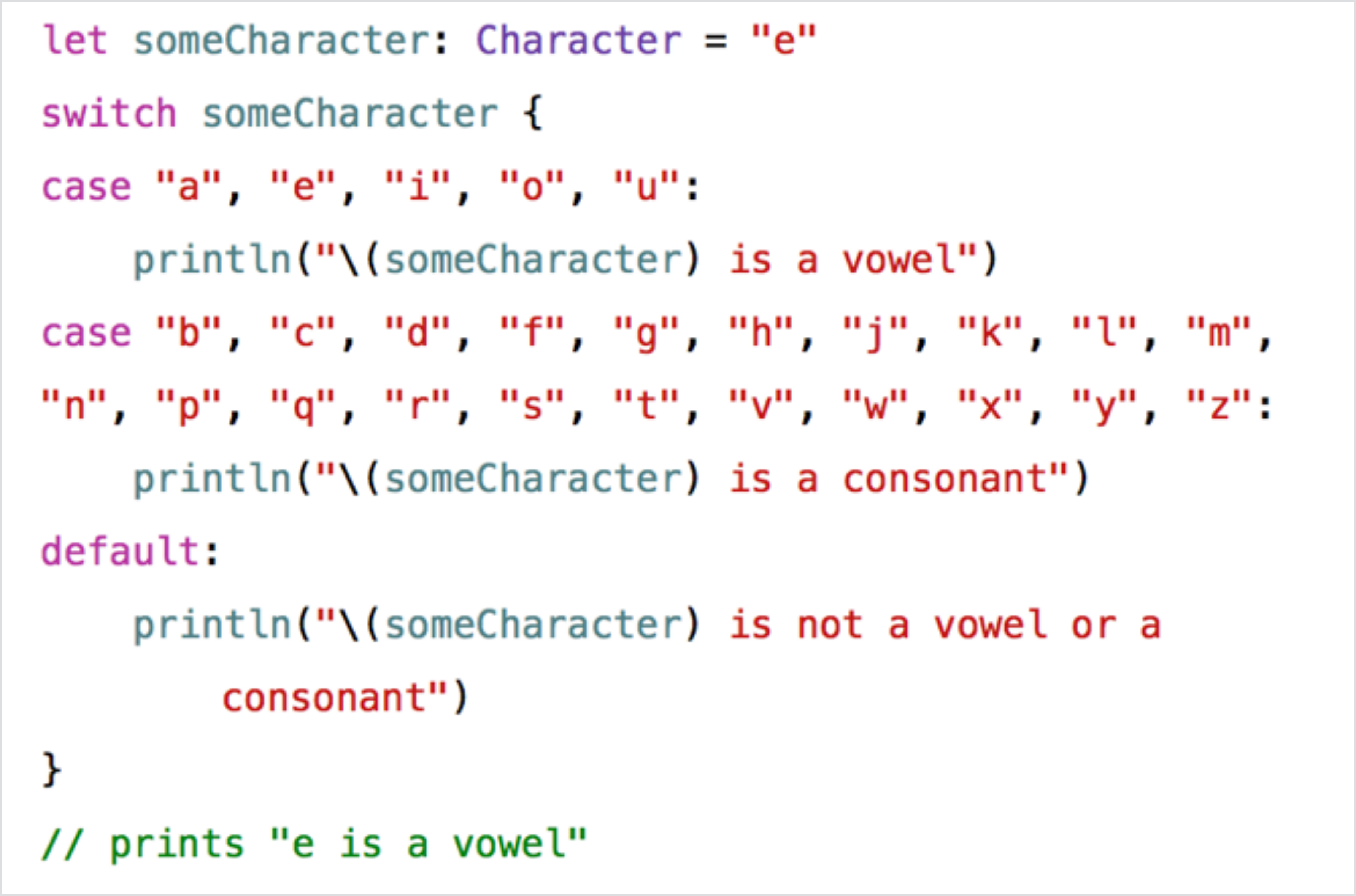
If



```
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

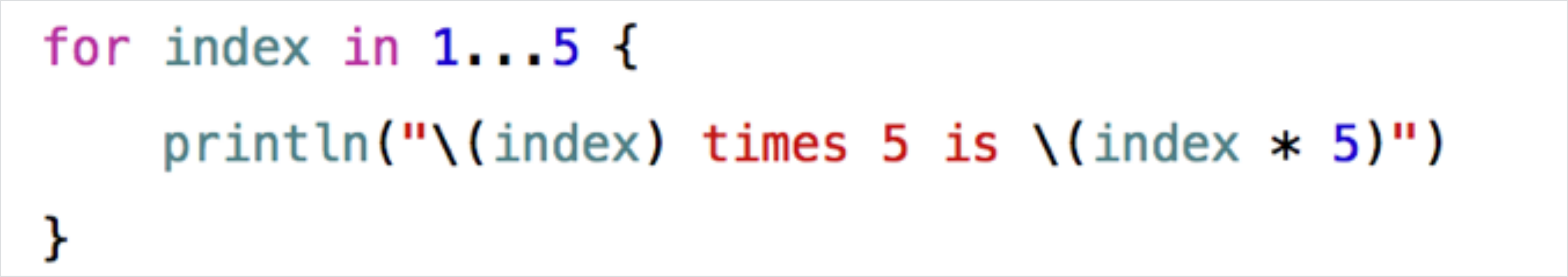
Switch



```
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"
```

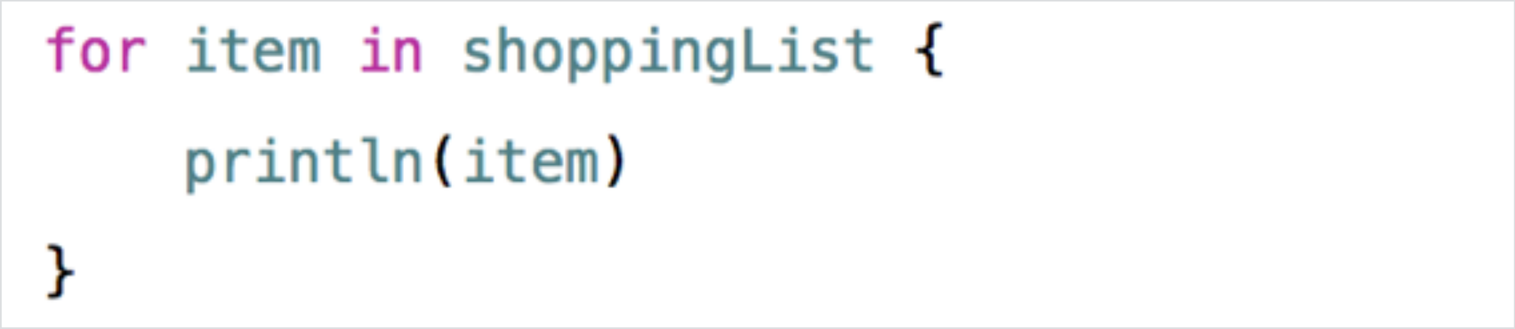
Loop

Range



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

Collection



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

Collection Types

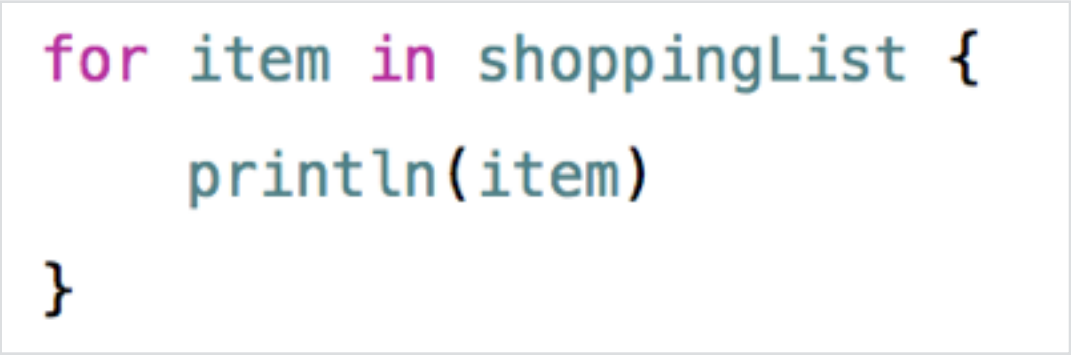
- Array

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)
}
```

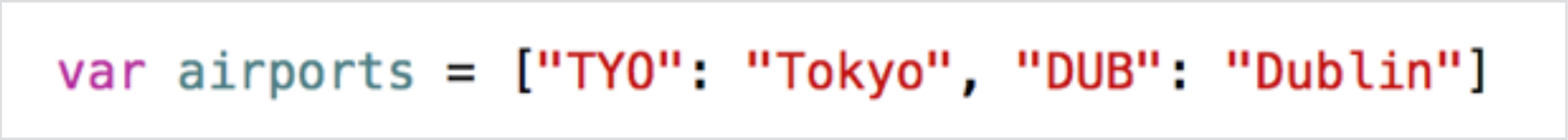
Collection Types

- Dictionaries

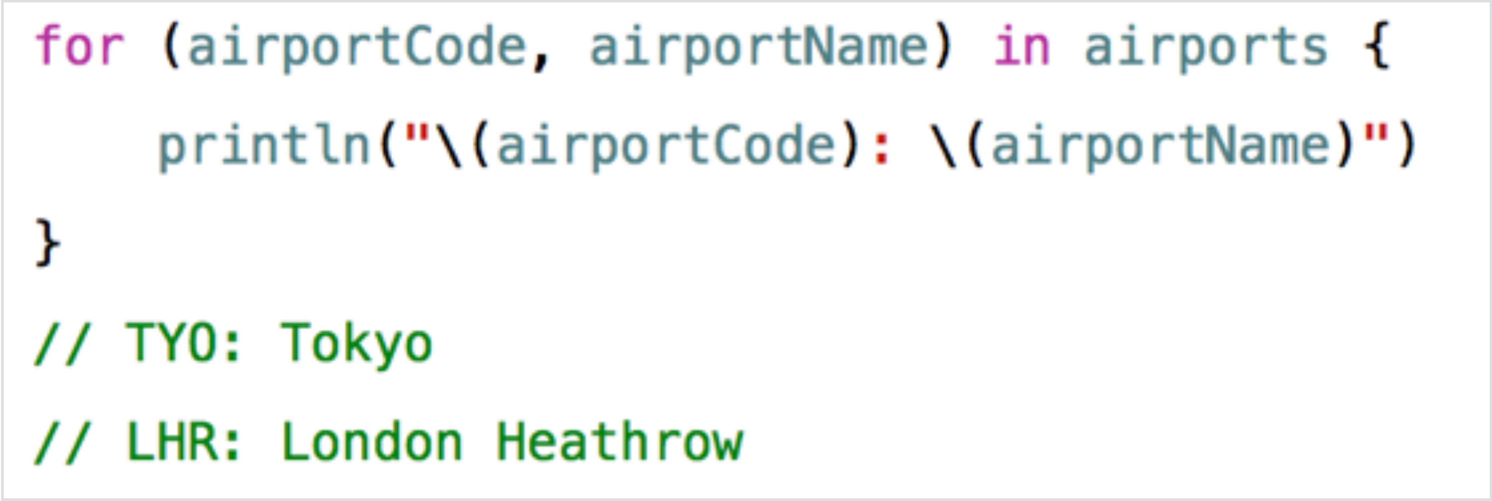
```
var namesOfIntegers = [Int: String]()   //empty dictionary

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

Example:



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

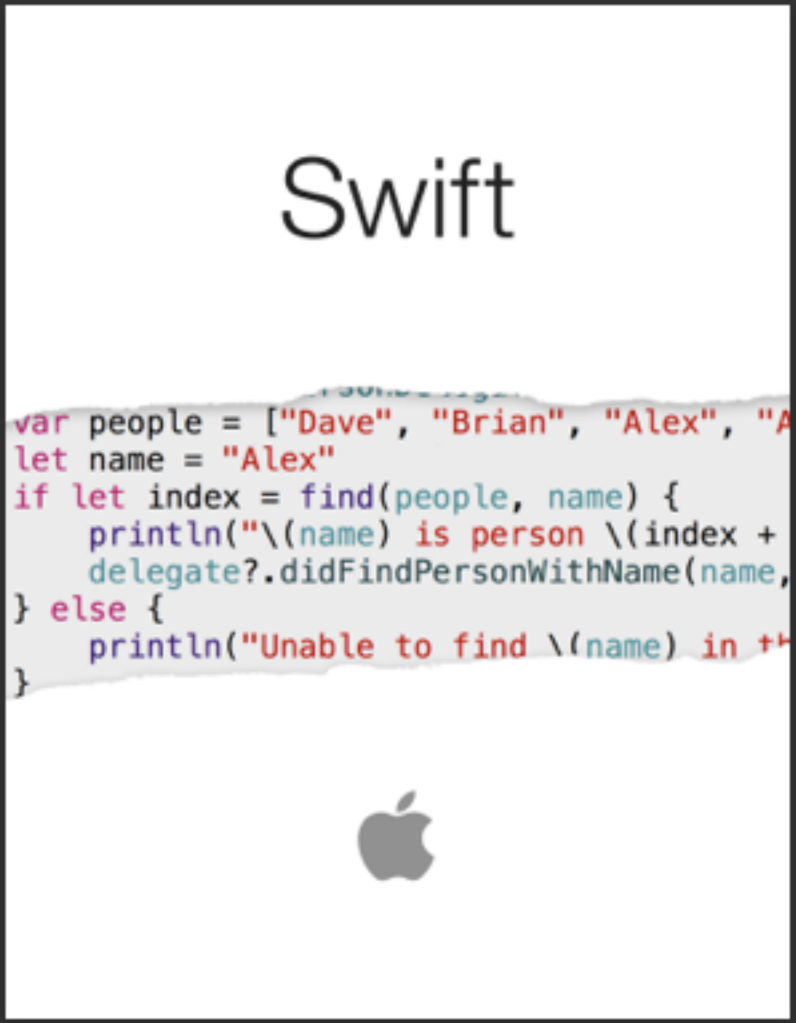


```
for (airportCode, airportName) in airports {
    println("\(airportCode): \(airportName)")
}
// TYO: Tokyo
// LHR: London Heathrow
```

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 th
}
```

Using Swift with Cocoa and Objective-C

Please look for it at iTunes

10/22

Final Exam & Demo

Good Luck!