



iOS Application Development

Lecture 2: Seminar Topics and Unit 1

Simon Völker & Philipp Wacker
Media Computing Group
RWTH Aachen University

hci.rwth-aachen.de/ios



RWTHAACHEN
UNIVERSITY

Seminar

- 2 Presentations per Session
 - 20 min presentation, ~10 min discussion
 - 3 people per group
 - Dates:
 - 18.11, 19.11, 25.11, 26.11, 2.12, 3.12, 9.12
 - Order is not fixed yet!
- Finished version 1 week before the presentation
 - 15 min slide and content discussions 1 week before the presentation

Seminar

- Framework overview
- Conceptual structure
- Show how to use them

- Structure:
 - Brief introduction
 - Basis steps how to use the framework
 - Demo (live coding)
 - Explain one or two advanced features and show how to use them.
- Deliverables: Slides and Demo code

Topic: Sprite Kit

- 2D Gaming Engine
- Visuals are Sprites, Shapes
- Includes:
 - Animations
 - Physics
 - Lighting
 - Particle effects

We will show a very short demo presentation



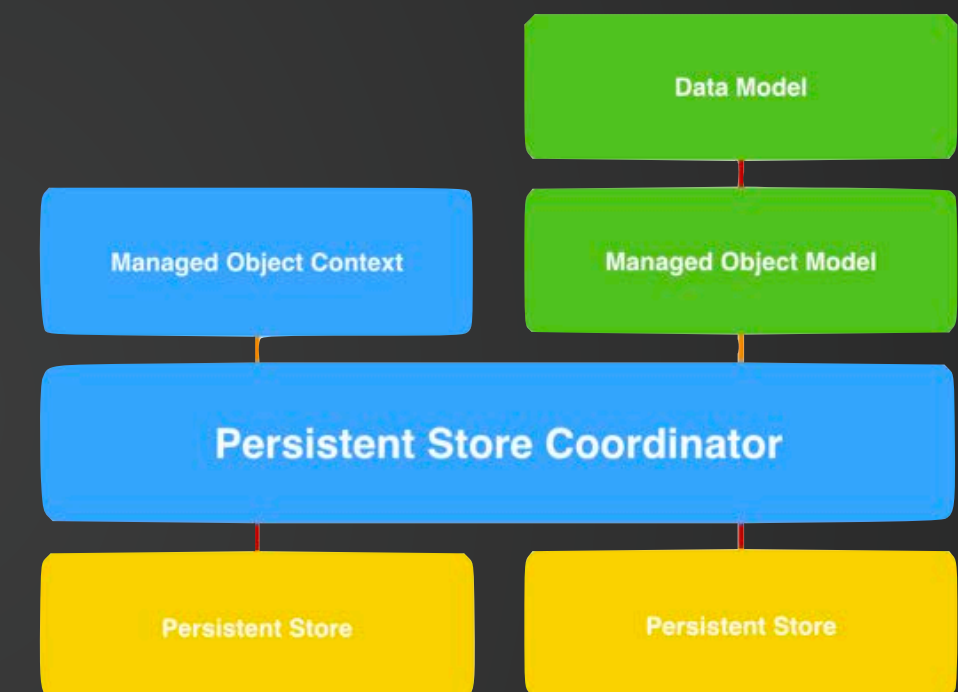
1. Core Data

- Manages the model in MVC
- Persistent data storage
- Managed objects
- Alternative to SQL

<https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/CoreData/index.html>

<https://developer.apple.com/videos/play/wwdc2017/210/>

<https://developer.apple.com/videos/play/wwdc2019/230/>



2. Scene Kit

- High-level 3D game engine
- Scene graphs
- Animations
- Physics

<https://developer.apple.com/scenekit/>

<https://developer.apple.com/videos/play/wwdc2017/605/>

<https://developer.apple.com/videos/play/wwdc2017/604/>



3. Metal

- Low level renderer
- Modern alternative to OpenGL
- Precompiled shaders
- Multithreading
- Speed increase by reducing CPU load

<https://developer.apple.com/metal/>

<https://developer.apple.com/videos/play/wwdc2019/601/>

<https://developer.apple.com/videos/play/wwdc2019/613/>



4. CloudKit & iCloud Drive

- Cloud storage
- Cross devices files sharing
- Storage optimization

<https://developer.apple.com/icloud/>

<https://developer.apple.com/videos/play/wwdc2017/226/>



5. Core ML + Create ML

- Machine learning
- Computer vision (Vision Framework)
- Natural language processing

<https://developer.apple.com/machine-learning/>

<https://developer.apple.com/videos/play/wwdc2019/209/>

<https://developer.apple.com/videos/play/wwdc2019/704/>

<https://developer.apple.com/videos/play/wwdc2019/430/>

<https://developer.apple.com/videos/play/wwdc2018/708/>

<https://developer.apple.com/videos/play/wwdc2018/709/>



6. SiriKit

- Add Siri to your app
- Voice commands

<https://developer.apple.com/sirikit/>

<https://developer.apple.com/videos/play/wwdc2017/214/>



7. WatchOS

- Design apps for the Apple Watch
- Native WatchOS apps

<https://developer.apple.com/videos/play/wwdc2019/208/>

<https://developer.apple.com/videos/play/wwdc2018/239/>

<https://developer.apple.com/videos/play/wwdc2018/206/>



8. MapKit, CoreLocation

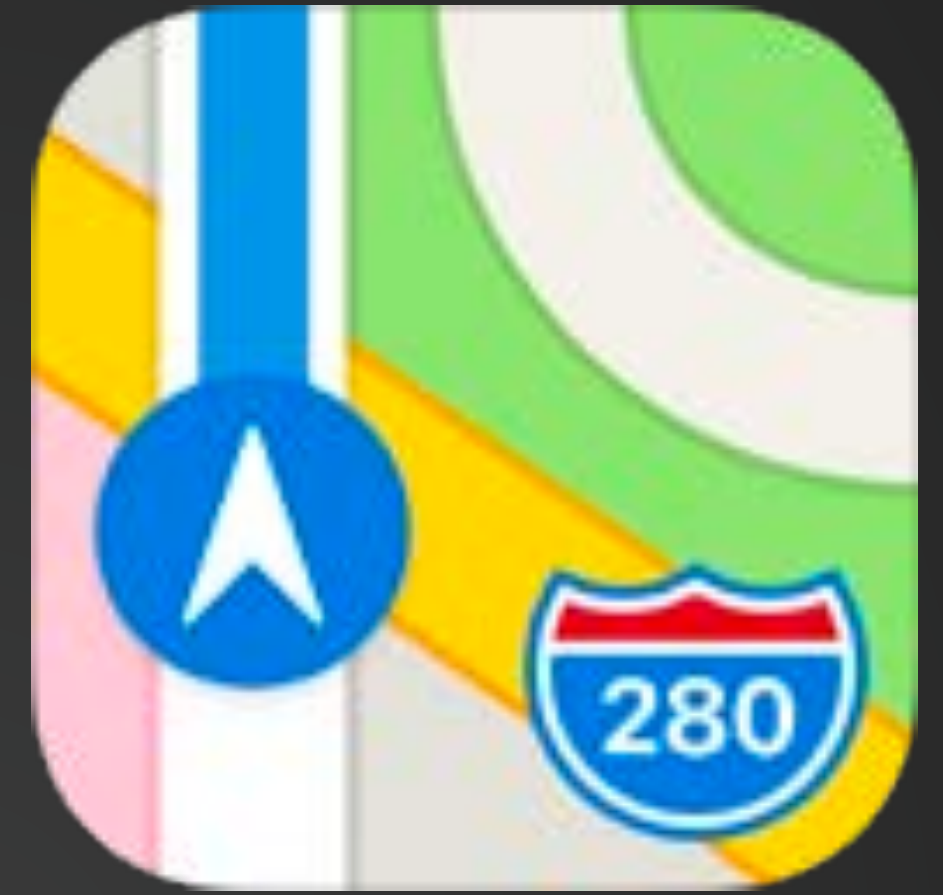
- Map navigation
- Location detection
- Geofencing

<https://developer.apple.com/maps/>

<https://developer.apple.com/videos/play/wwdc2019/705/>

<https://developer.apple.com/videos/play/wwdc2017/713/>

<https://developer.apple.com/videos/play/wwdc2019/236/>



9. Networking on iOS

- CoreBluetooth
- Sockets
- Bonjour

<https://developer.apple.com/videos/play/wwdc2019/901/>

<https://developer.apple.com/videos/play/wwdc2019/713/>

<https://developer.apple.com/videos/play/wwdc2019/712/>

<https://developer.apple.com/videos/play/wwdc2017/712/>

<https://developer.apple.com/videos/play/wwdc2017/707/>

<https://developer.apple.com/videos/play/wwdc2017/709/>



10. Extensions & Inter-App communication

- Deep Linking
- App Extensions (Services)
- Files app

<https://developer.apple.com/app-extensions/>

<https://developer.apple.com/videos/play/wwdc2017/250/>



11. Debugging in Xcode and Instruments

- Instruments
- Energie consumptions

<https://developer.apple.com/videos/play/wwdc2019/411/>

<https://developer.apple.com/videos/play/wwdc2019/412/>

<https://developer.apple.com/videos/play/wwdc2018/412/>

<https://developer.apple.com/videos/play/wwdc2018/410/>

<https://developer.apple.com/videos/play/wwdc2018/416/>



12. ClassKit

- Create class exercises
- Schoolwork apps
- Student and teach classroom curriculum support

<https://developer.apple.com/videos/play/wwdc2019/247/>

<https://developer.apple.com/videos/play/wwdc2018/215/>



13. RealityKit & Reality Composer

- Simulate and render 3D content in AR
- Prototype AR scenes and apps

<https://developer.apple.com/videos/play/wwdc2019/603/>

<https://developer.apple.com/videos/play/wwdc2019/605/>

<https://developer.apple.com/videos/play/wwdc2019/609/>

<https://developer.apple.com/videos/play/wwdc2019/415/>



14. Bringing People into AR

- Detect people and object in a AR scene
- Body tracking

<https://developer.apple.com/videos/play/wwdc2019/610/>

<https://developer.apple.com/videos/play/wwdc2019/607/>



15. GameplayKit

- Gameplay logics
- Agents
- Pathfinding
- Rule systems

<https://developer.apple.com/documentation/gameplaykit>

<https://developer.apple.com/videos/play/wwdc2016/608/>

https://developer.apple.com/library/archive/documentation/General/Conceptual/GameplayKit_Guide/index.html



16. SwiftUI

- Design Apps using simple SwiftUI framework
- Declarative syntax
- Design tools

<https://developer.apple.com/videos/play/wwdc2019/216/>

<https://developer.apple.com/videos/play/wwdc2019/204/>

<https://developer.apple.com/videos/play/wwdc2019/231/>

<https://developer.apple.com/videos/play/wwdc2019/231/>

<https://developer.apple.com/videos/play/wwdc2019/226/>

<https://developer.apple.com/videos/play/wwdc2019/237/>



17. AVKit

- Low-Latency HLS
- Media playback
- Multi-Camera capture for iOS

<https://developer.apple.com/videos/play/wwdc2019/249/>

<https://developer.apple.com/videos/play/wwdc2019/502/>

<https://developer.apple.com/videos/play/wwdc2019/503/>



Forming Groups & Vote for the Topic

Give each topic a different rating.

Please rate your interest in each topic on a scale from 1 (highly interested) to 17 (not interested at all).

Topic

1. Core Data
2. Scene Kit
3. Metal
4. CloudKit & iCloud Drive
5. Core ML + Create ML
6. SiriKit
7. WatchOS
8. MapKit, CoreLocation
9. Networking on iOS
10. Extensions & Inter-App communication
11. Debugging in Xcode and Instruments
12. ClassKit
13. RealityKit & Reality Composer
14. Bringing People into AR
15. GameplayKit
16. Swift UI
17. AVKit

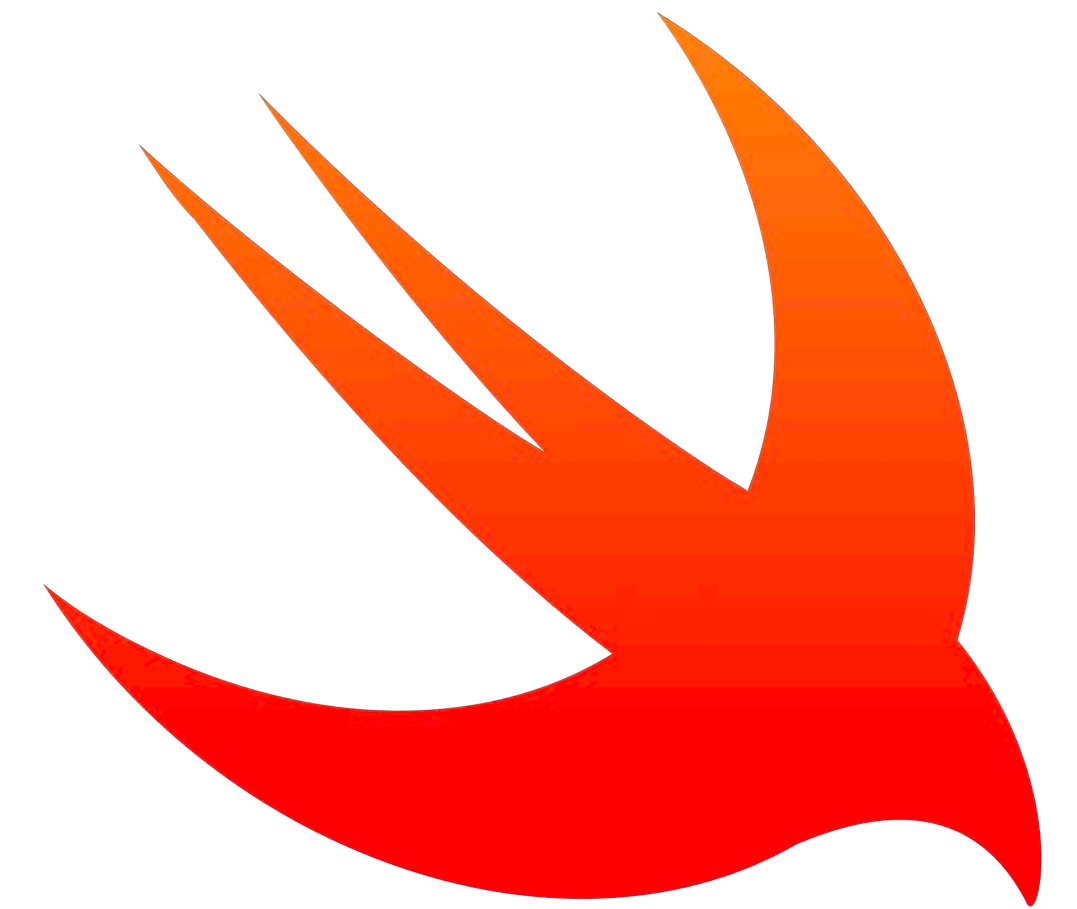
CHAPTER 1

Swift



History

- Introduced at WWDC 2014
- Influenced by C and Objective-C
- But designed to be simpler to learn and not dependent on older programming languages
- “Safe, fast, and expressive”
- Open source since 2015



Characteristics

- Clean syntax
- Optionals
- Type inference
- Type safety
- Automatic Reference Counting (ARC)

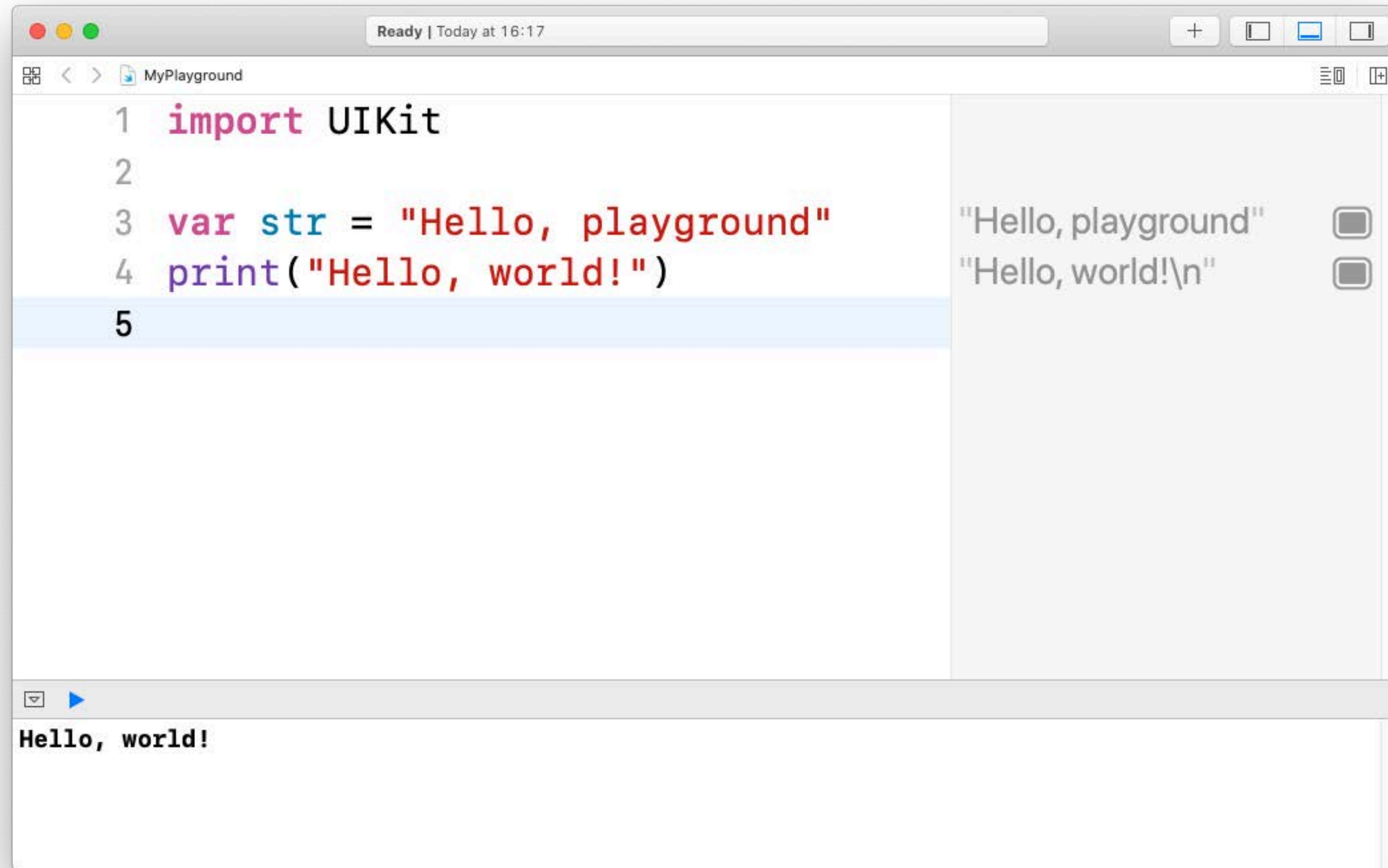


Characteristics

- Tuples and multiple return values
- Generics
- Fast and concise iteration over collections
- Structs that support methods, extensions, and protocols
- Map, filter, reduce, and other functional programming patterns
- Powerful error handling



Playgrounds



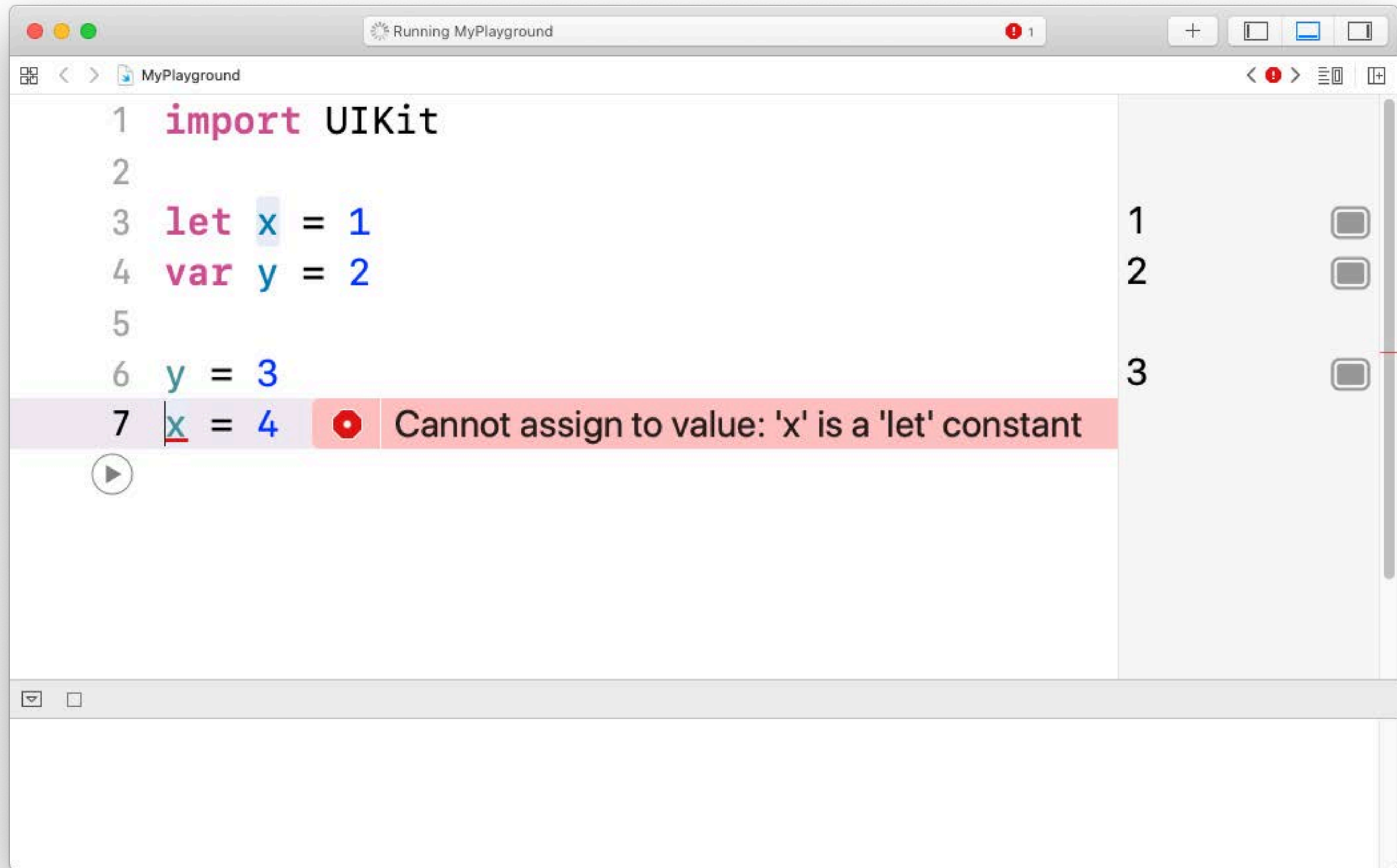
Variables and Constants

- Variables are declared with `var`

```
var x = 100
```

- Constants are declared with `let`

```
let pi = 3.14  
pi = 3.1415926 // error! a constant can't be changed afterwards
```

Type Inference

- Swift automatically chooses the adequate data type for a variable/constant

```
var x = 100
    x = 99.5 // error! x is of type Integer

var x = 99.5
    x = 100 // correct (x = 100.0, type: Double)
```

- You can also explicitly specify the type

```
var aString : String
```

Data Types & Type Inference

```
var x1 = 100 // Int
var x2 = 0.5 // Double
var x3 = x1 + x2 // error! can't add Int and Double
var x3 = Double(x1) + x2 // works! explicit type casting to Double
var x4 = 0.5 + 100 // works! compiler adds before setting the data type
print("x4 = \(x4)") // x4 = 100.5
var 😂 = "LOL" // String
```

Optionals

- By default, variables and constants cannot be nil
- Optionals: variables that can also be nil

```
var opt:Int? = 3  
  
opt = nil
```

- Normal variables and Optionals are incompatible to each other

```
var number = Int("abc")    // nil. Type of Number: Int?  
  
print(number + 3)          // error! Int? != Int  
  
var i:Int = number         // error!
```

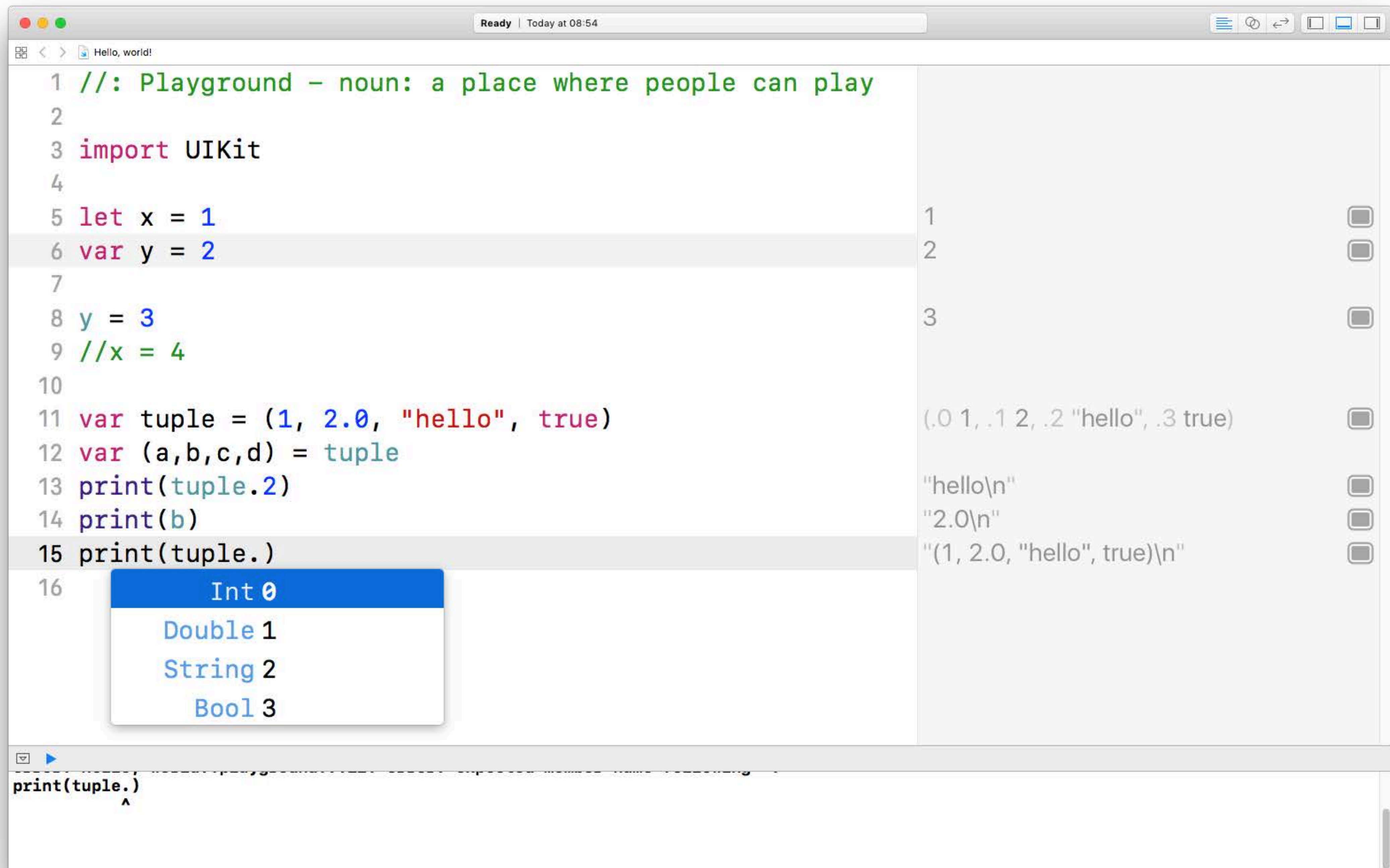

Tuples

- Tuples can have multiple elements of different types

```
var tuple = (1, 2.0, "hello", true)

var (a,b,c,d) = tuple // a = 1, ...

print(tuple.2) // "hello" (starts at 0)
```



Control Flow

- If/else

```
var x=3

if x<0 {
    print("x is negative")
} else if x==0 {
    print ("x is zero")
} else {
    print("x is positive")
}
```

- Ternary Operator

```
var largest: Int
let a = 15
let b = 4

if a > b {
    largest = a
}
else {
    largest = b
}

largest = a > b ? a : b
```

Control Flow

- Switch

```
let pt = (0.0, 0.0)

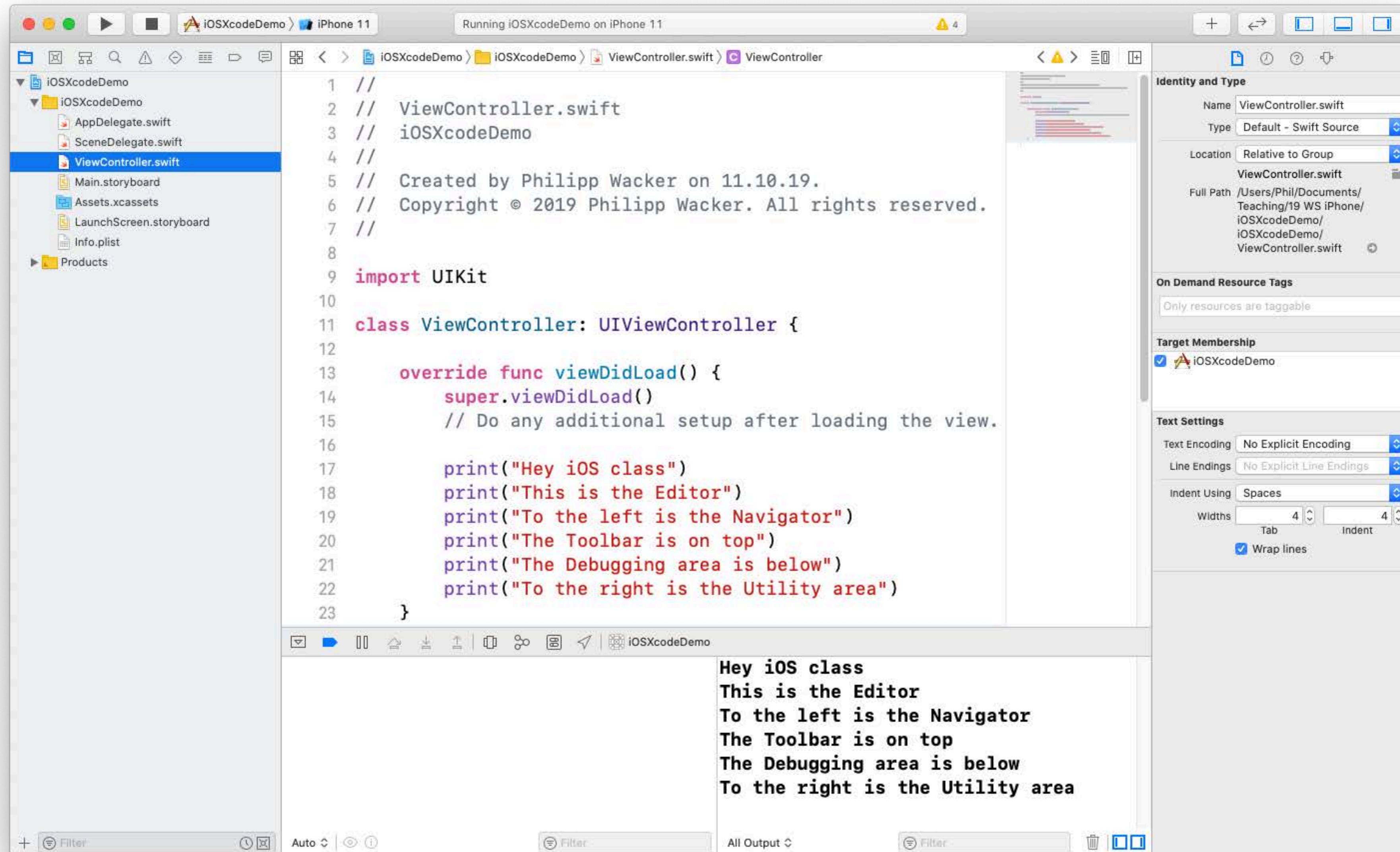
switch pt {
case (0,0):
    print("Origin.")
case (_,0):
    print ("On x-axis.")
case (0,_):
    print ("On y-axis.")
default:
    print ("Elsewhere.")
}
```

```
switch distance {
case 0...9:
    print("You are close")
case 10...500:
    print("Take a car")
default:
    print("Too far away.")
}
```

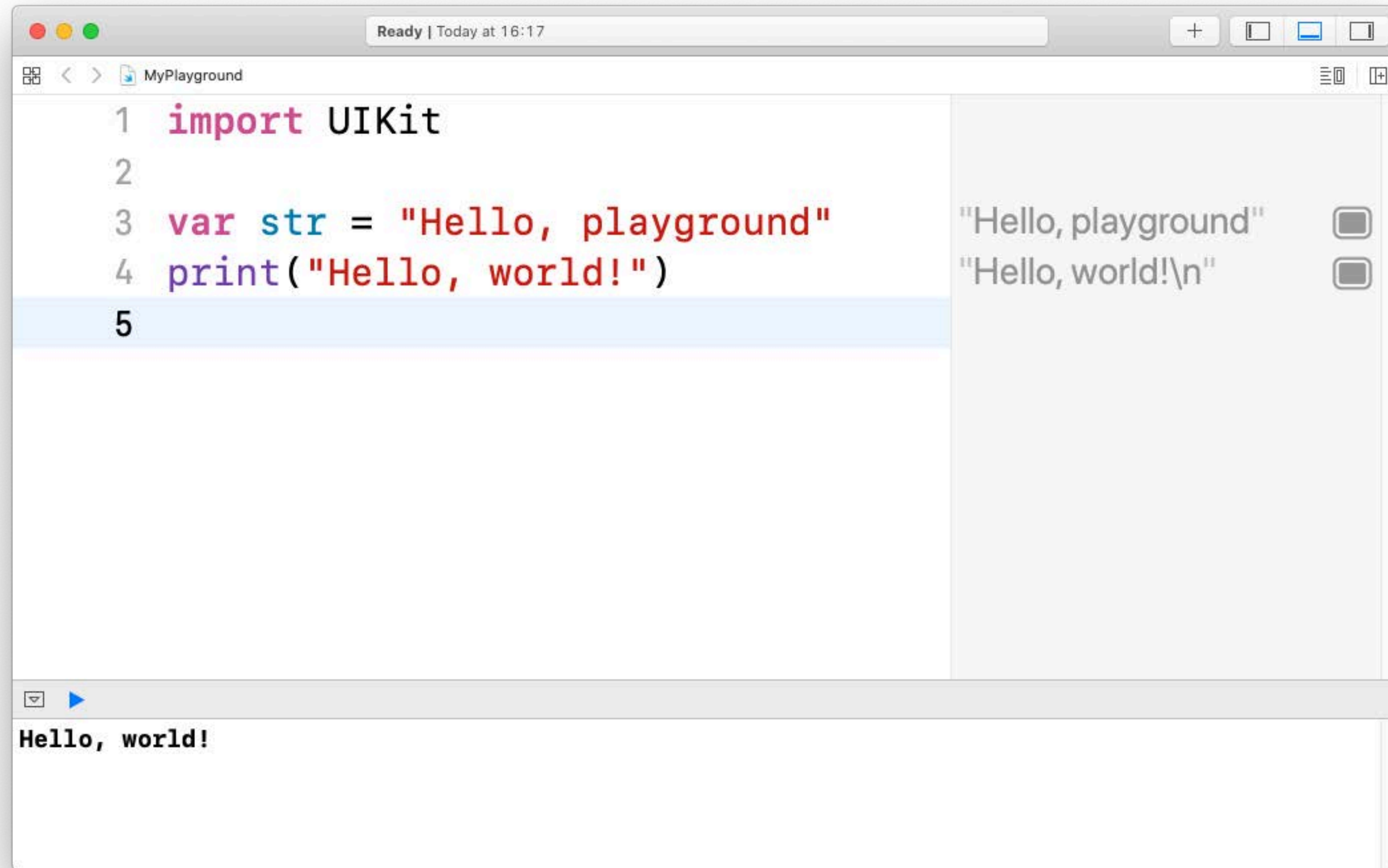

CHAPTER 4

Development Environment

Xcode



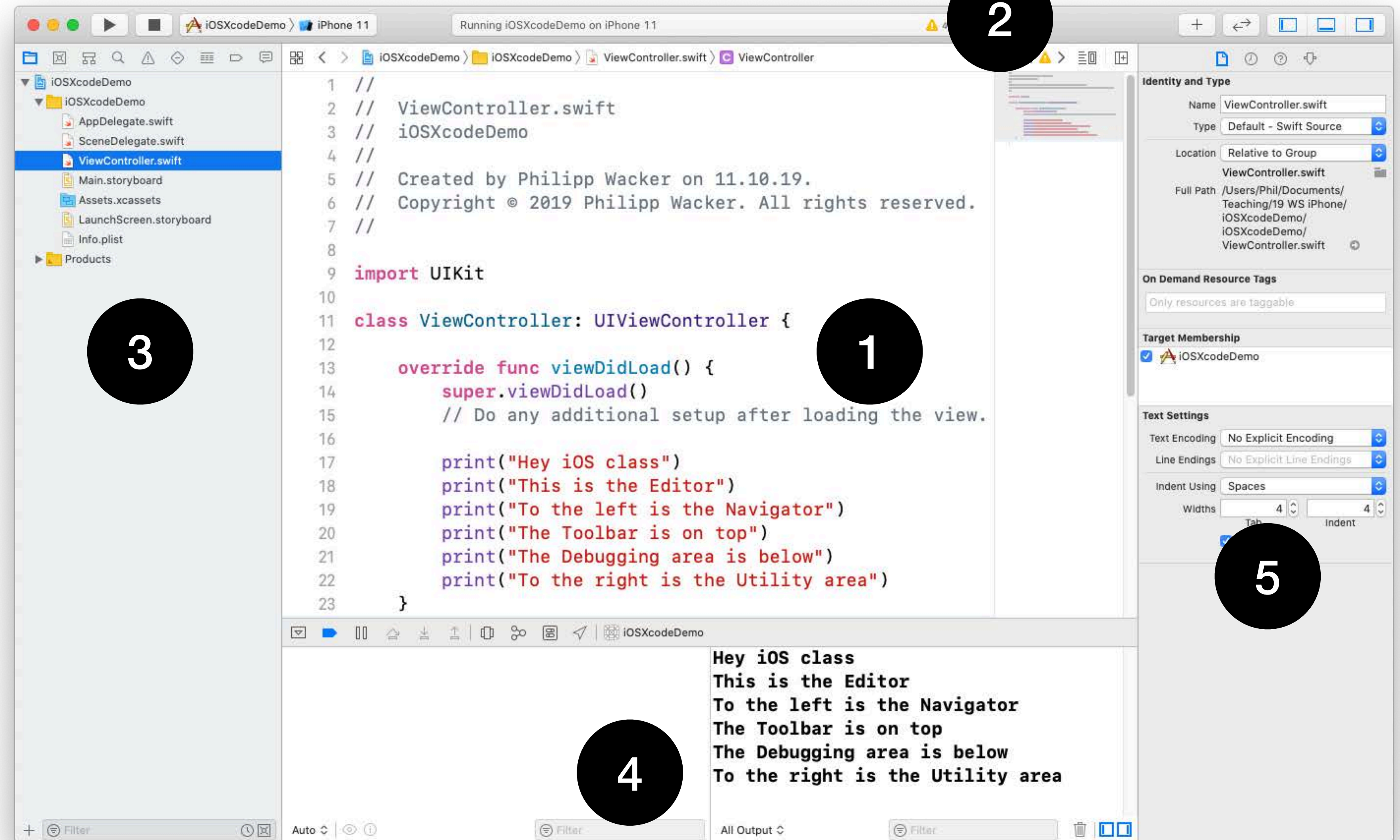
Playground Demo



Xcode

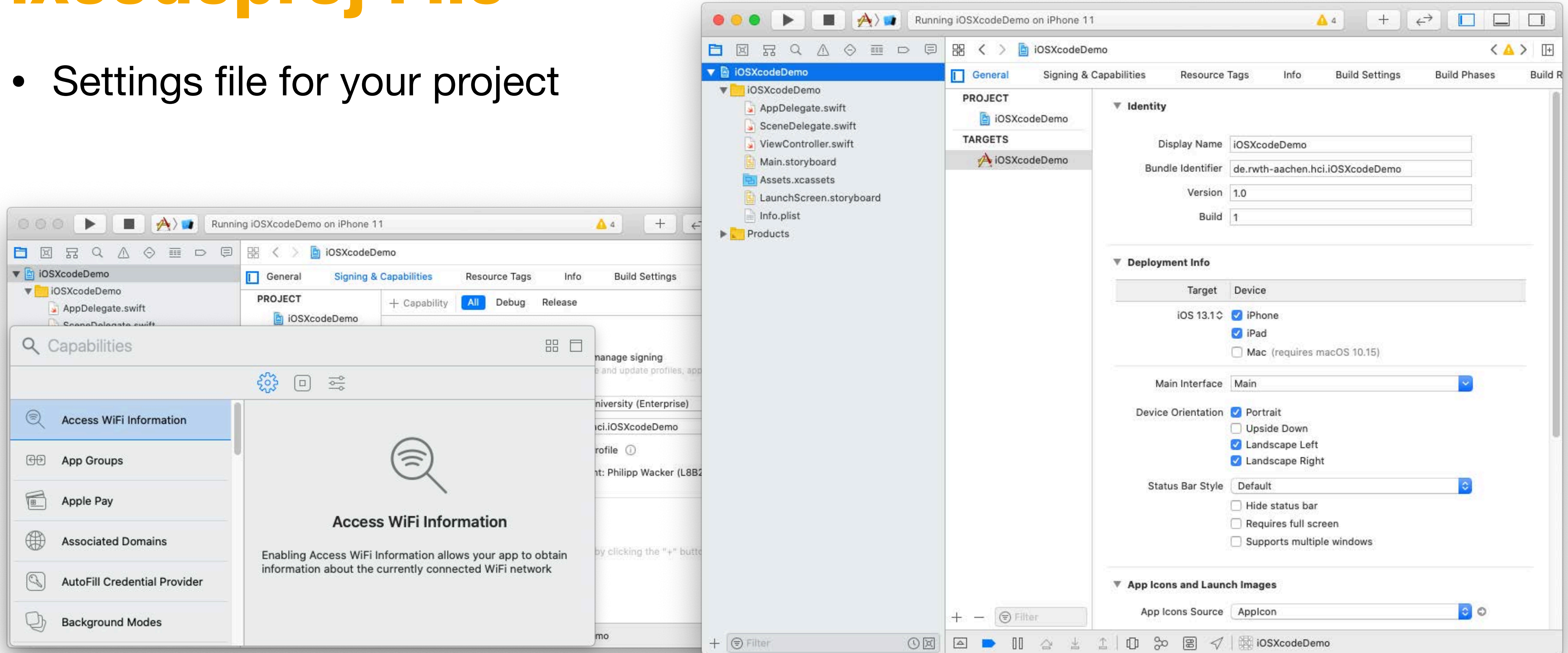
- 5 areas

1. Editor
2. Toolbar
3. Navigator
4. Debugging
5. Utility

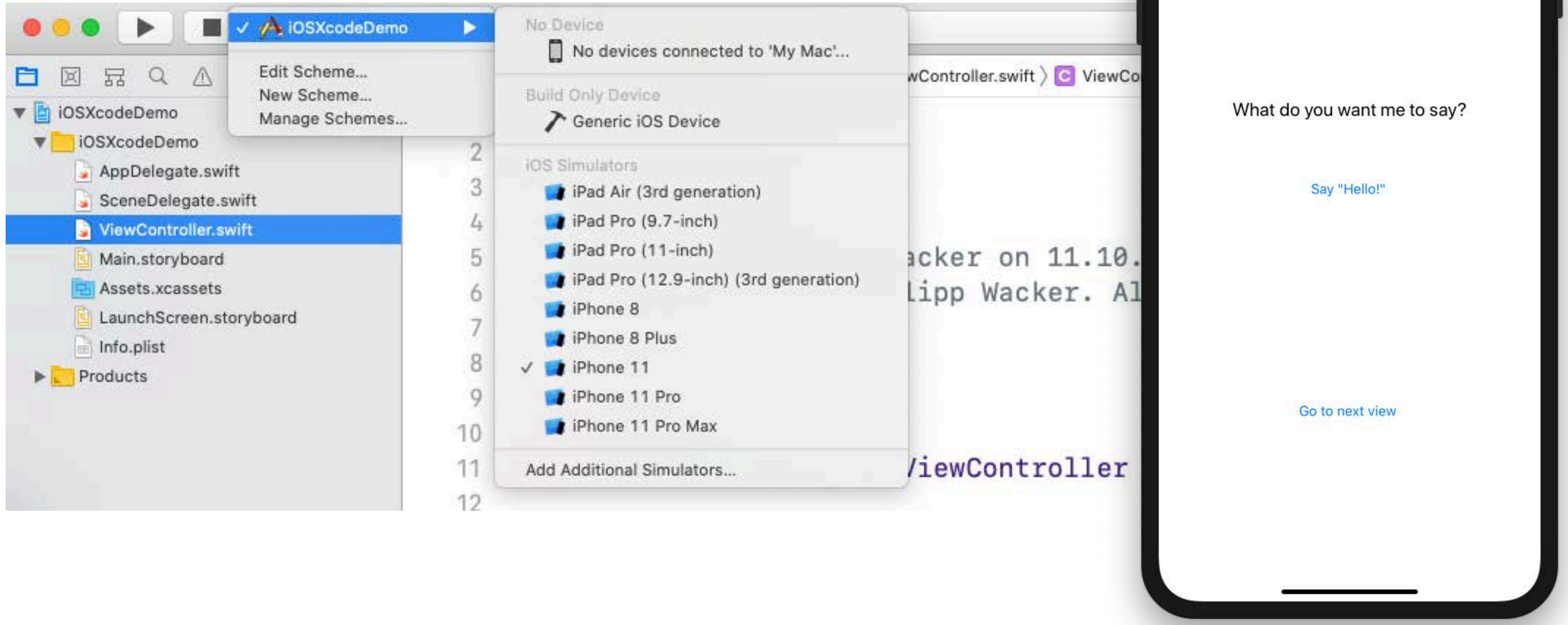


.xcodeproj File

- Settings file for your project



Building/Running



Warnings & Errors

- Warnings don't prevent your app from compiling & running

- Code that never gets executed
- Variable that does not change
- Deprecated code



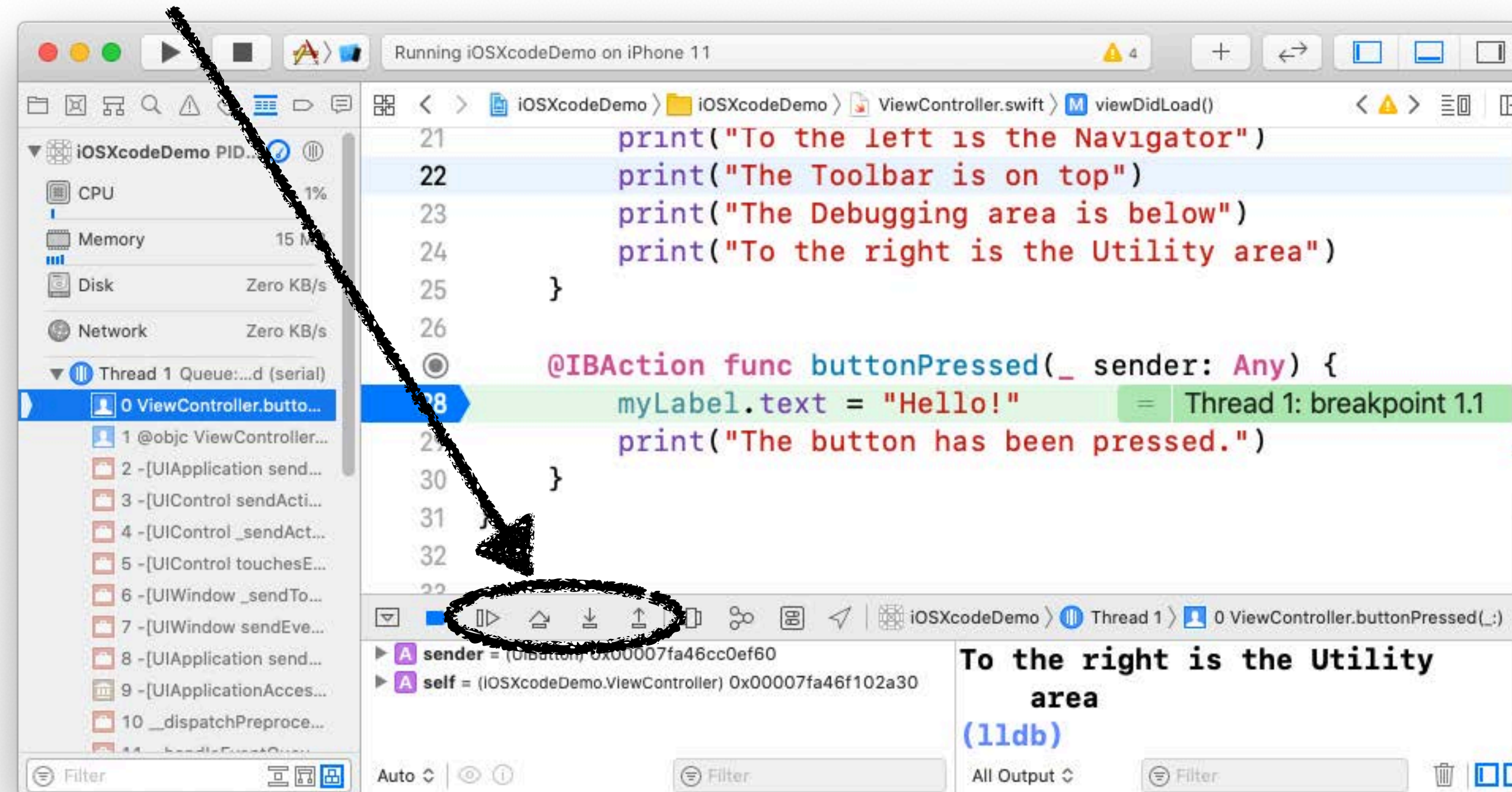
- Errors prevents the app from building

- Invalid code (typo, variable declaration, function calling)
- Xcode often provides suggestions & fixes



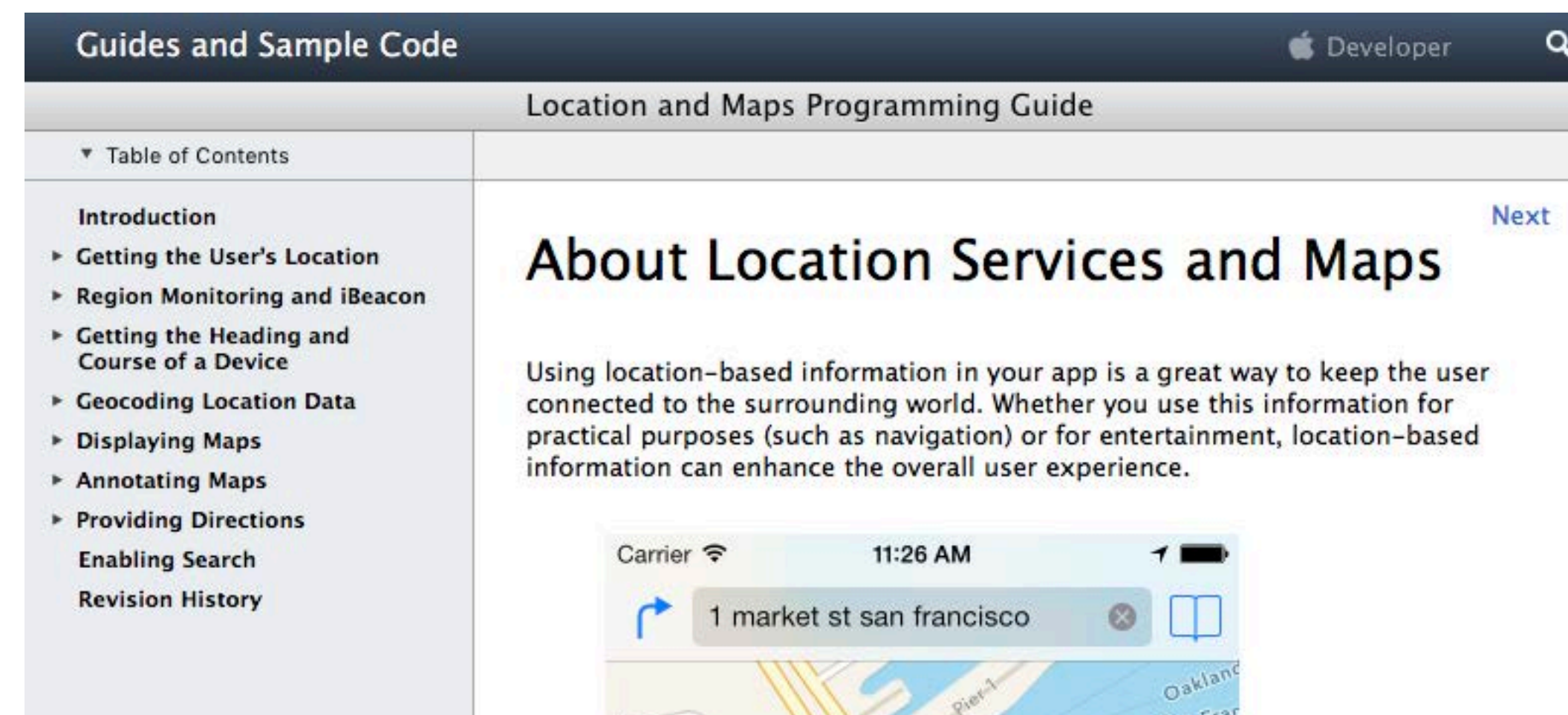
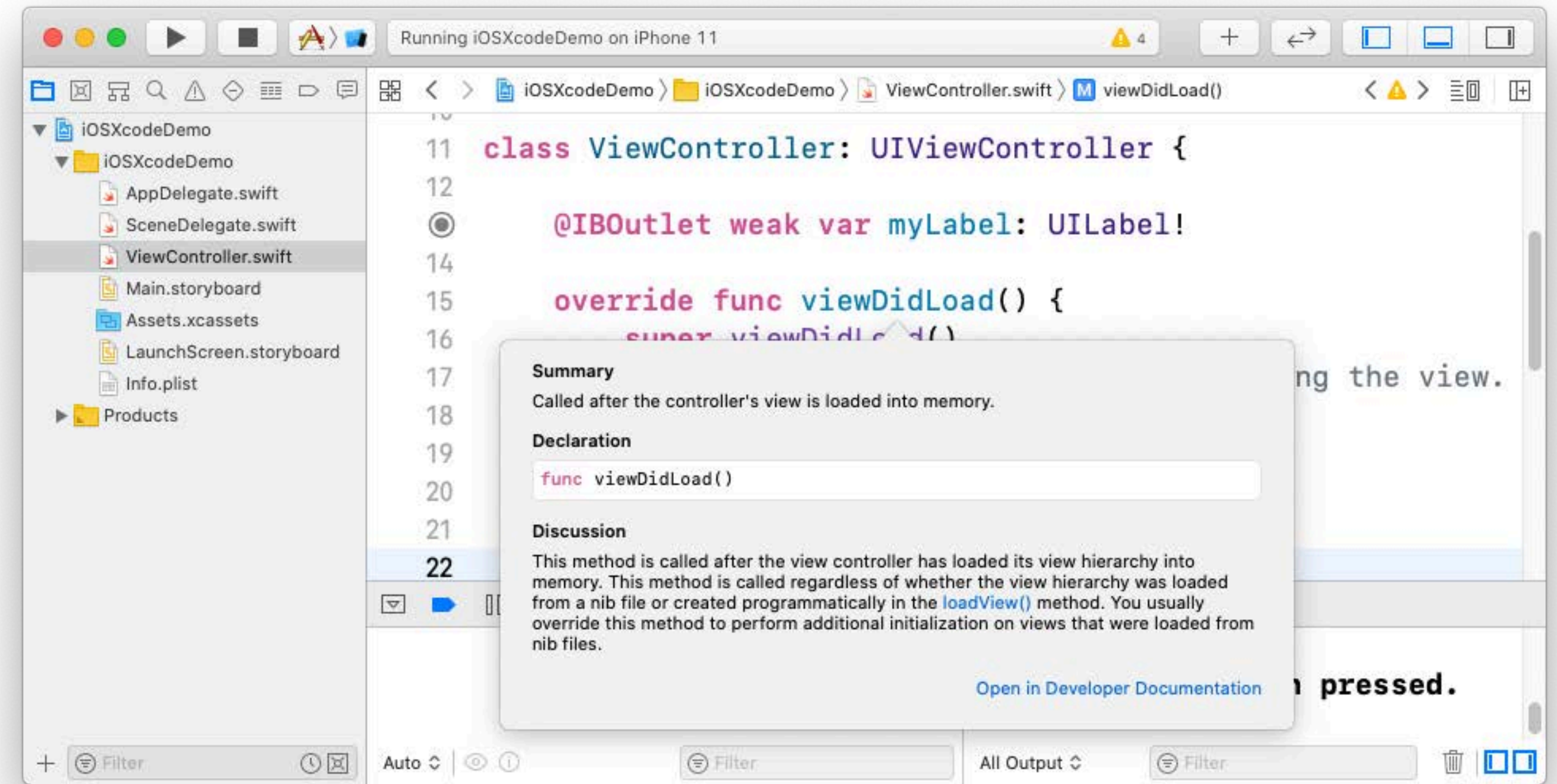
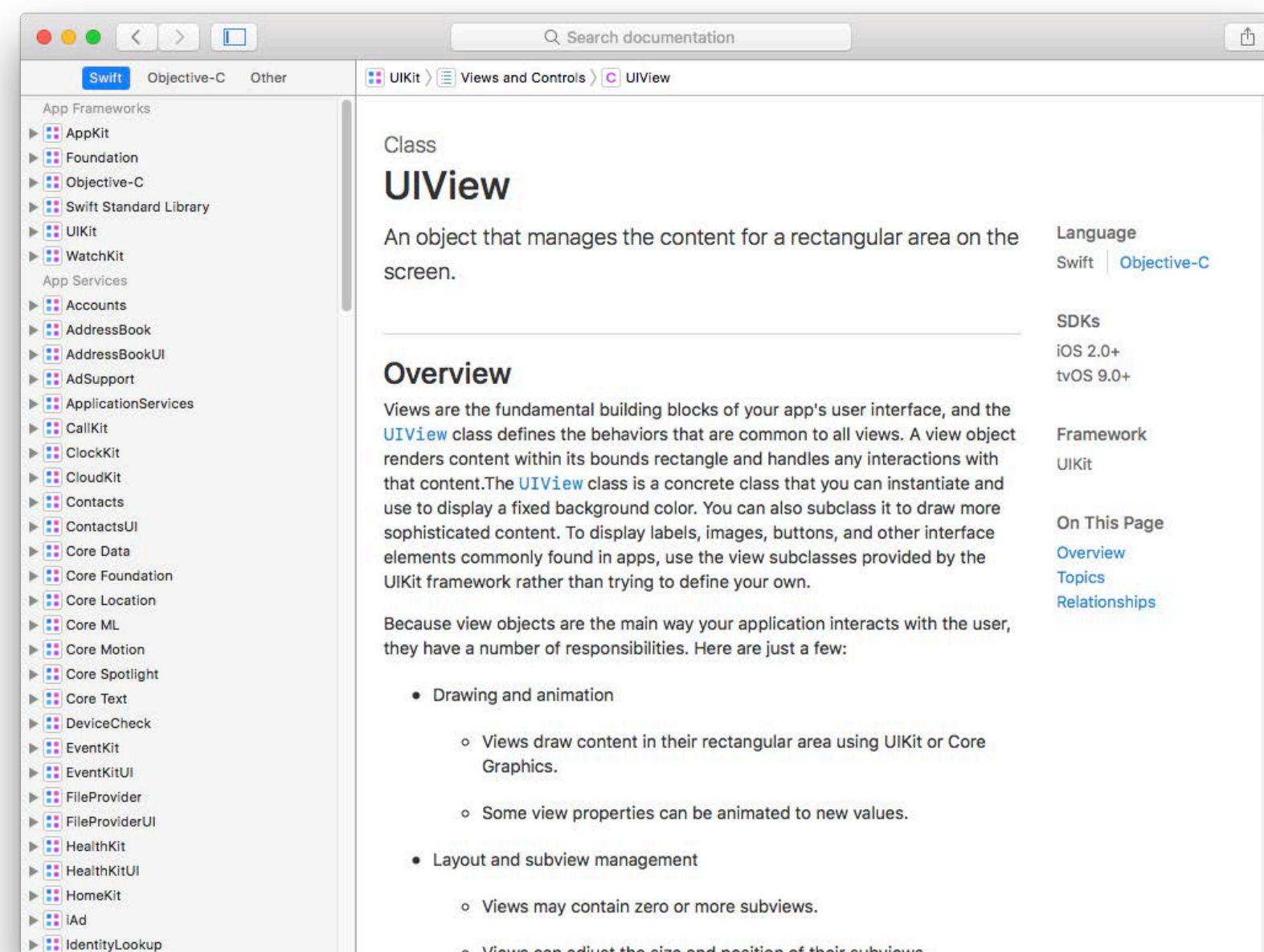
Debugging

- Set breakpoints for execution on simulator and device
- Continue, Step over, Step into, Step out



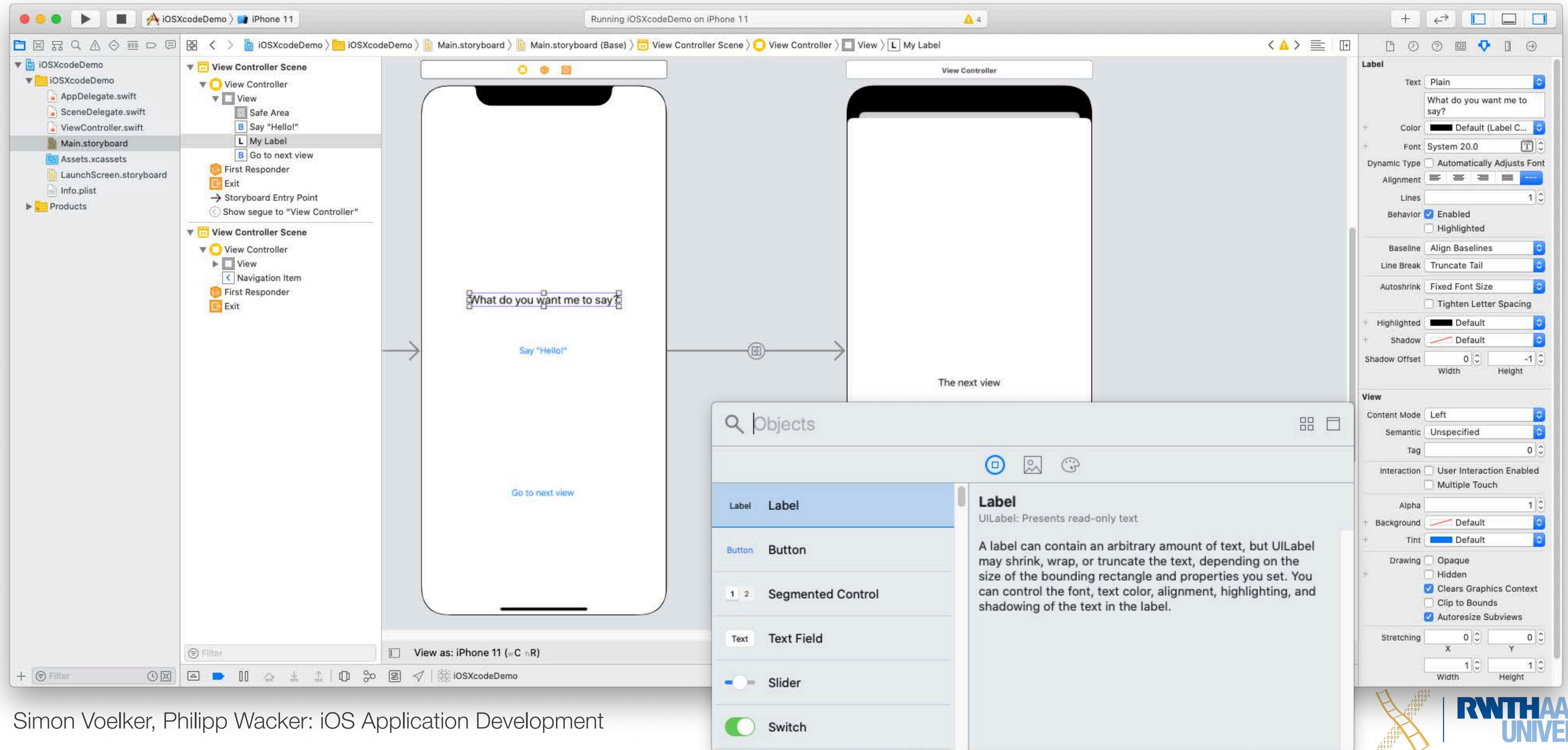
Documentation

- Quick Help (Option+Click)
- Documentation Browser
- Programming Guides



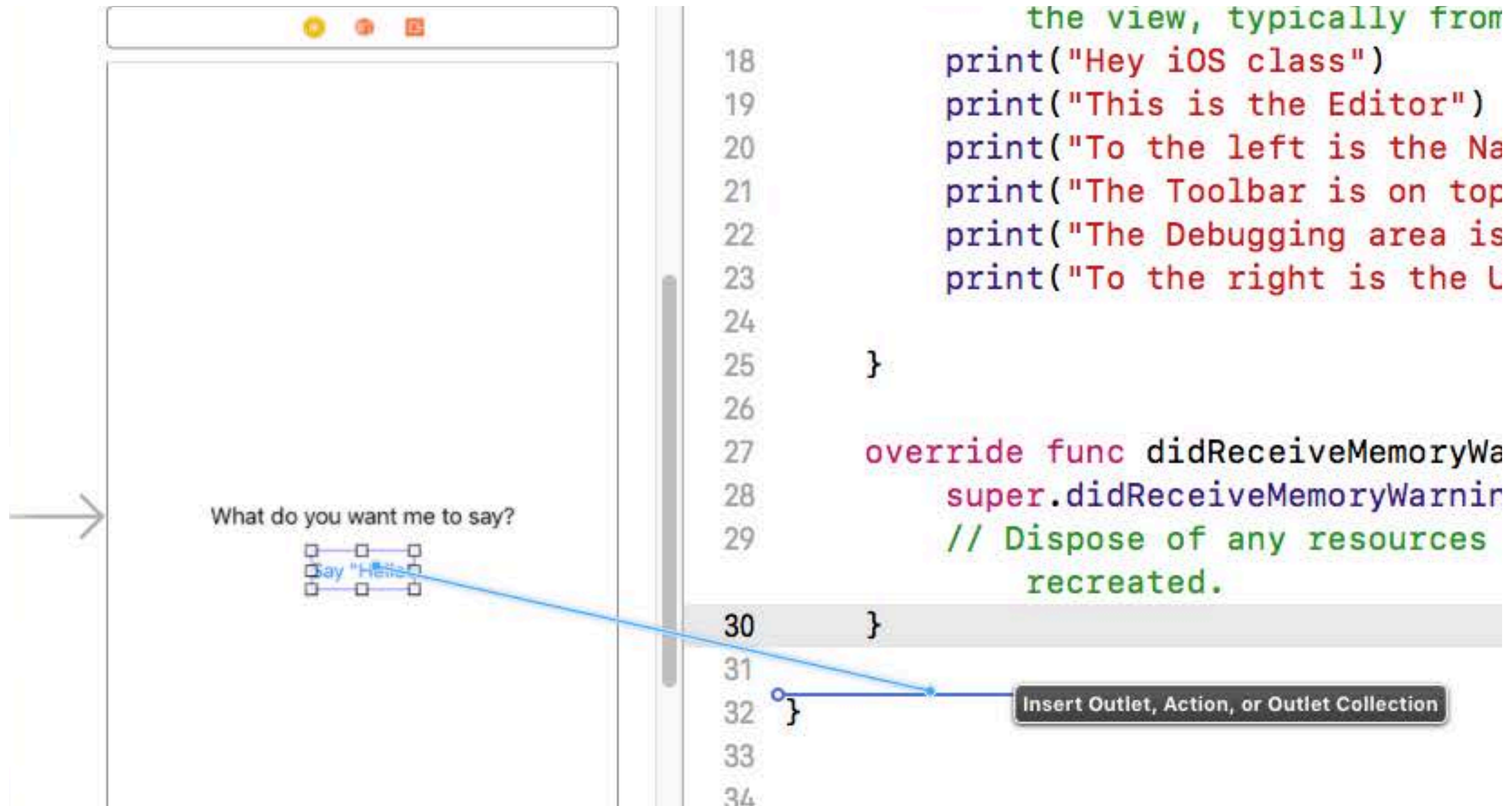
Interface Builder

- Visually define your UI



Outlets & Actions

- Connect your UI elements with your code: Right-click + Drag



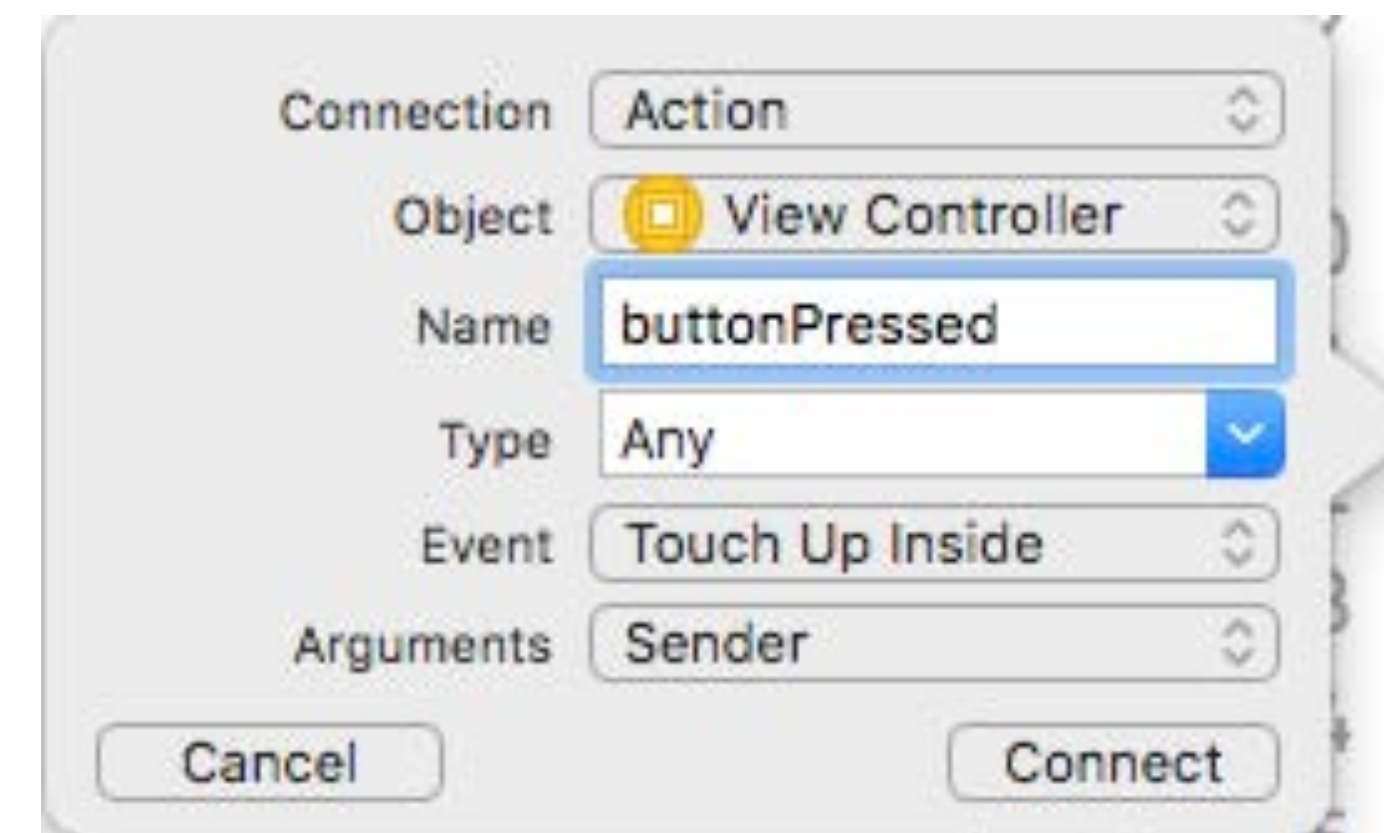
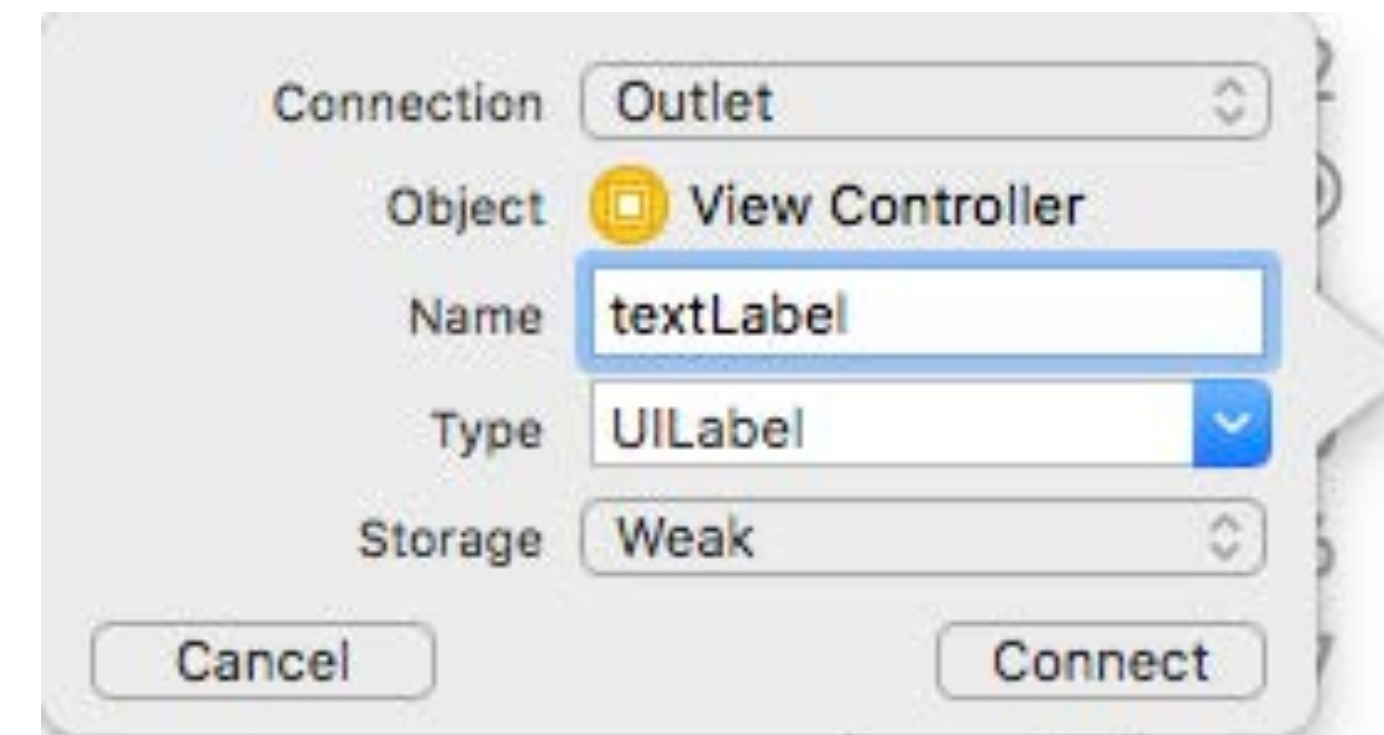
Outlets & Actions

- IBOutlet
 - Access the UI element from code

```
@IBOutlet weak var textLabel: UILabel!
```

- IBAction
 - Receive UI events

```
@IBAction func buttonPressed(_ sender: Any) {}
```



Summary

- Basic concept of Swift
- Data Types, Control Flow, Tuples
- Development Environment
 - Xcode
 - Playgrounds
- Tomorrow: Unit 2: Strings, Classes and Structs...

