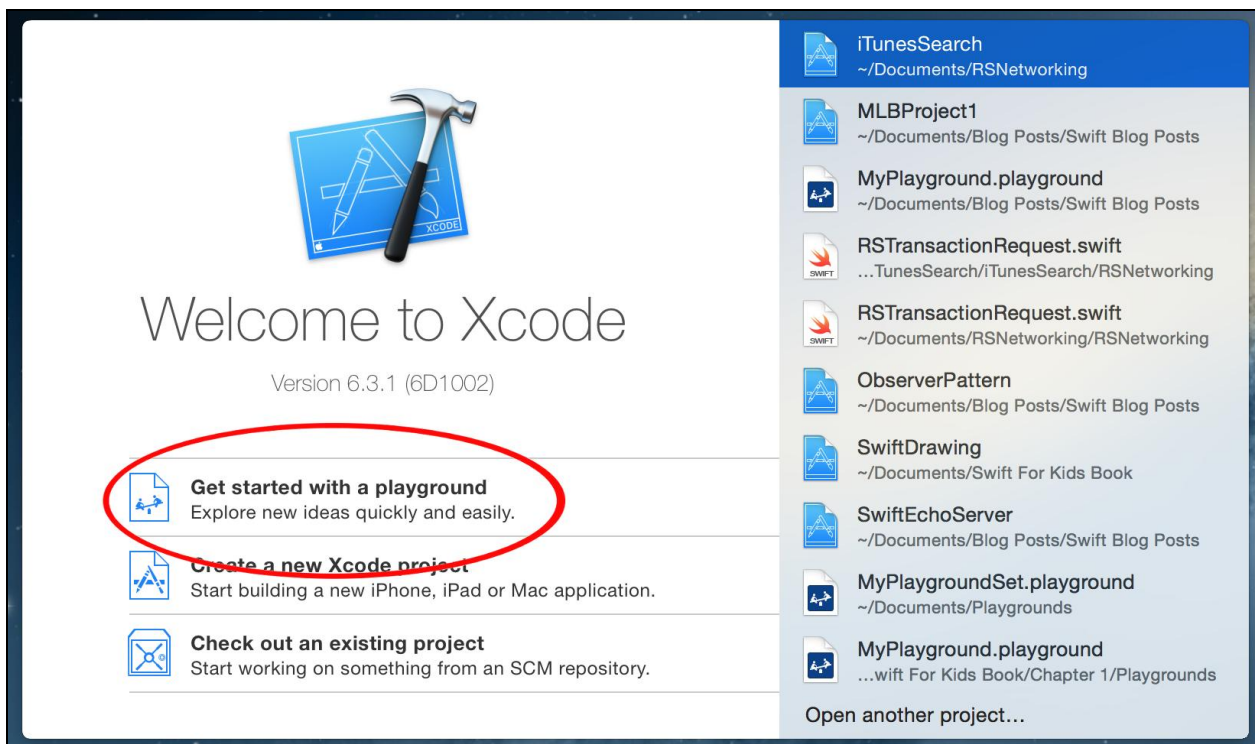
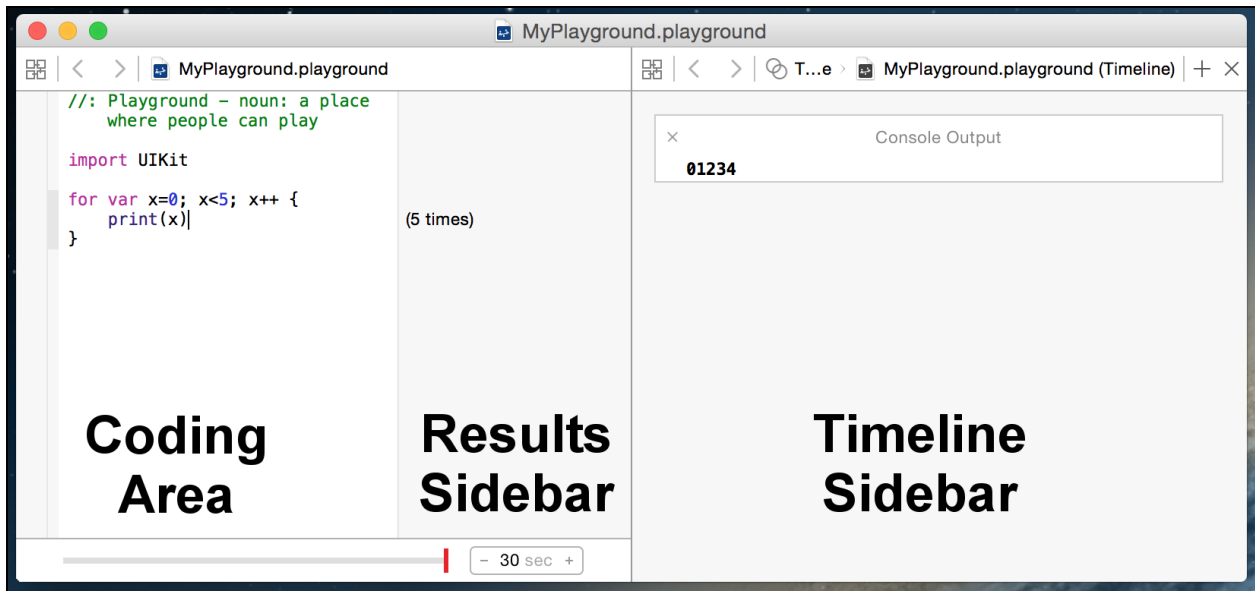
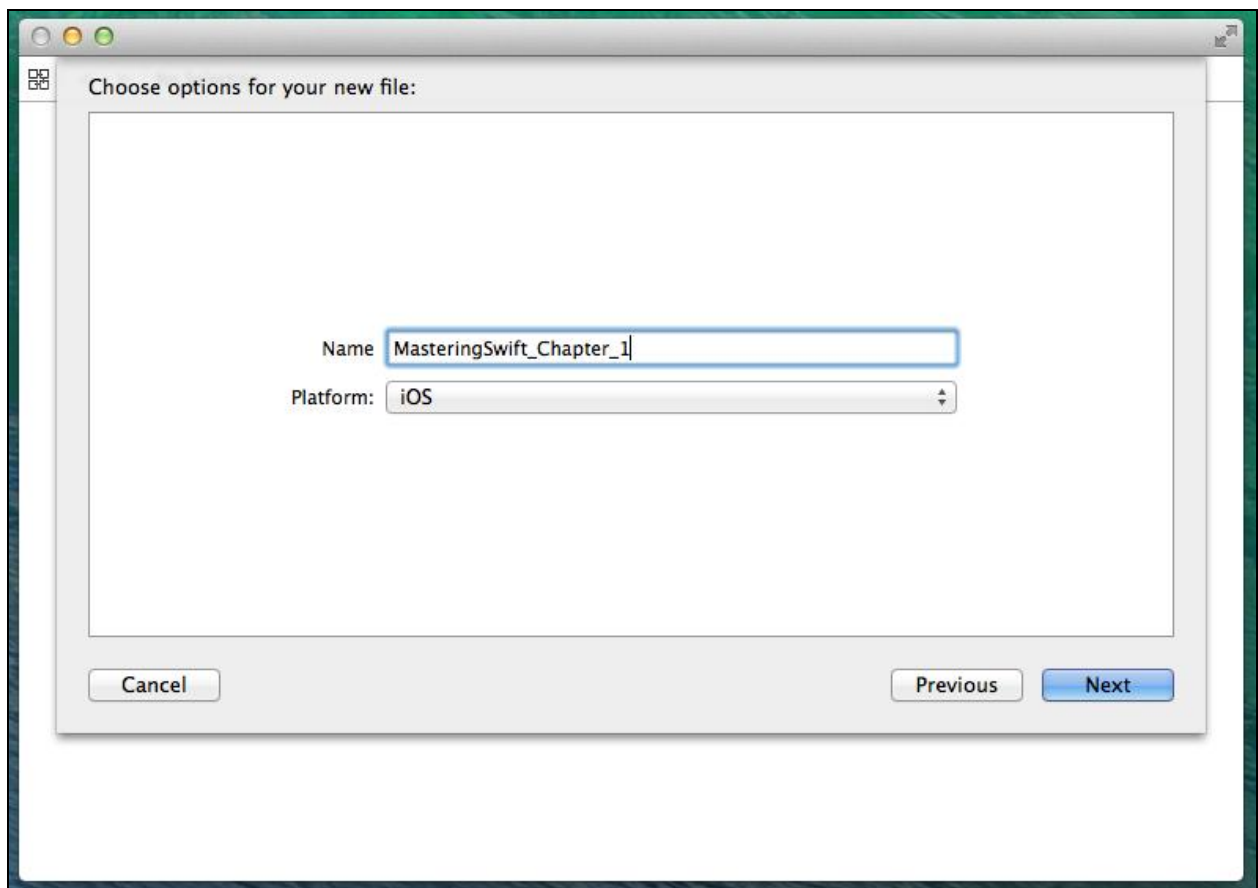
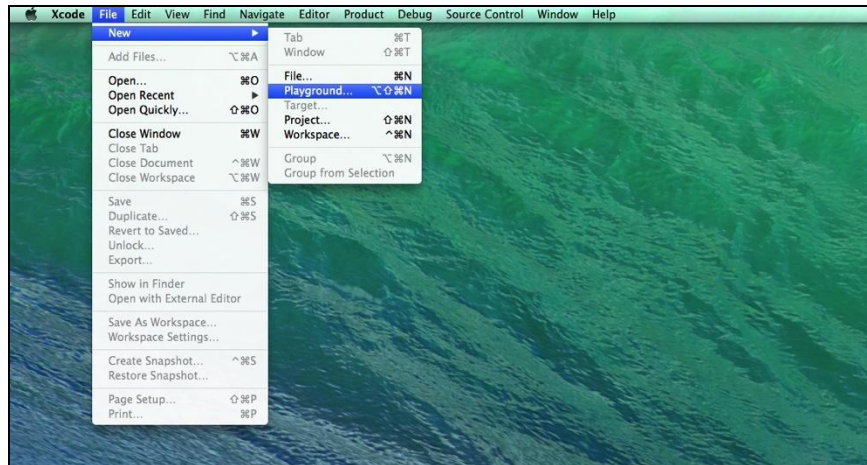
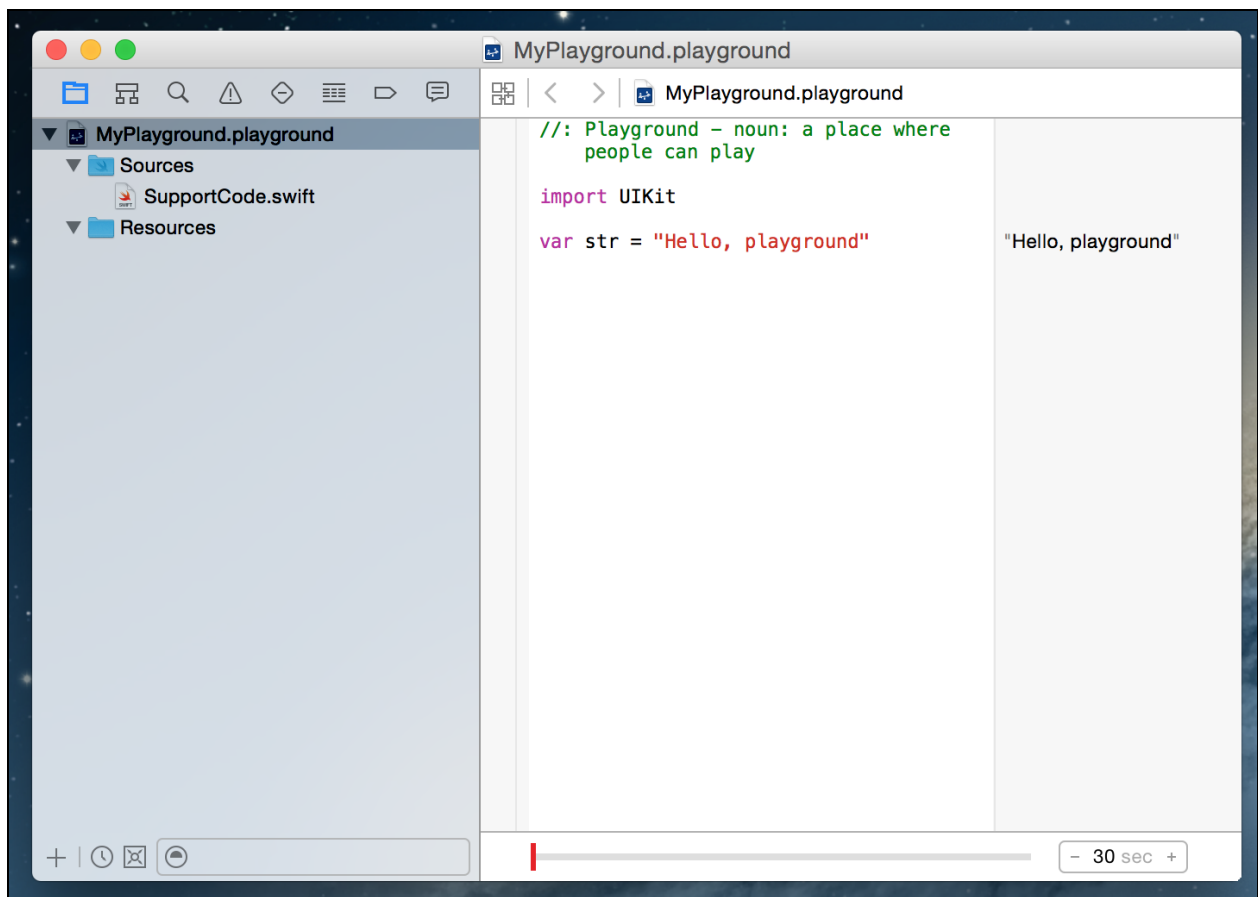
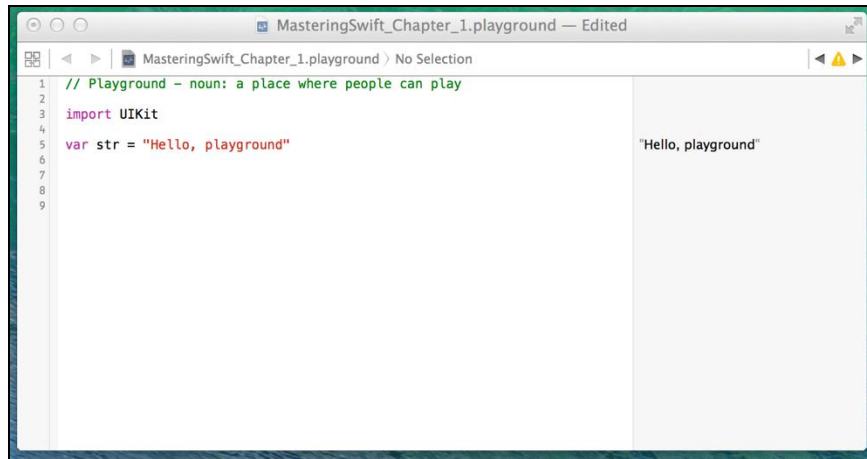
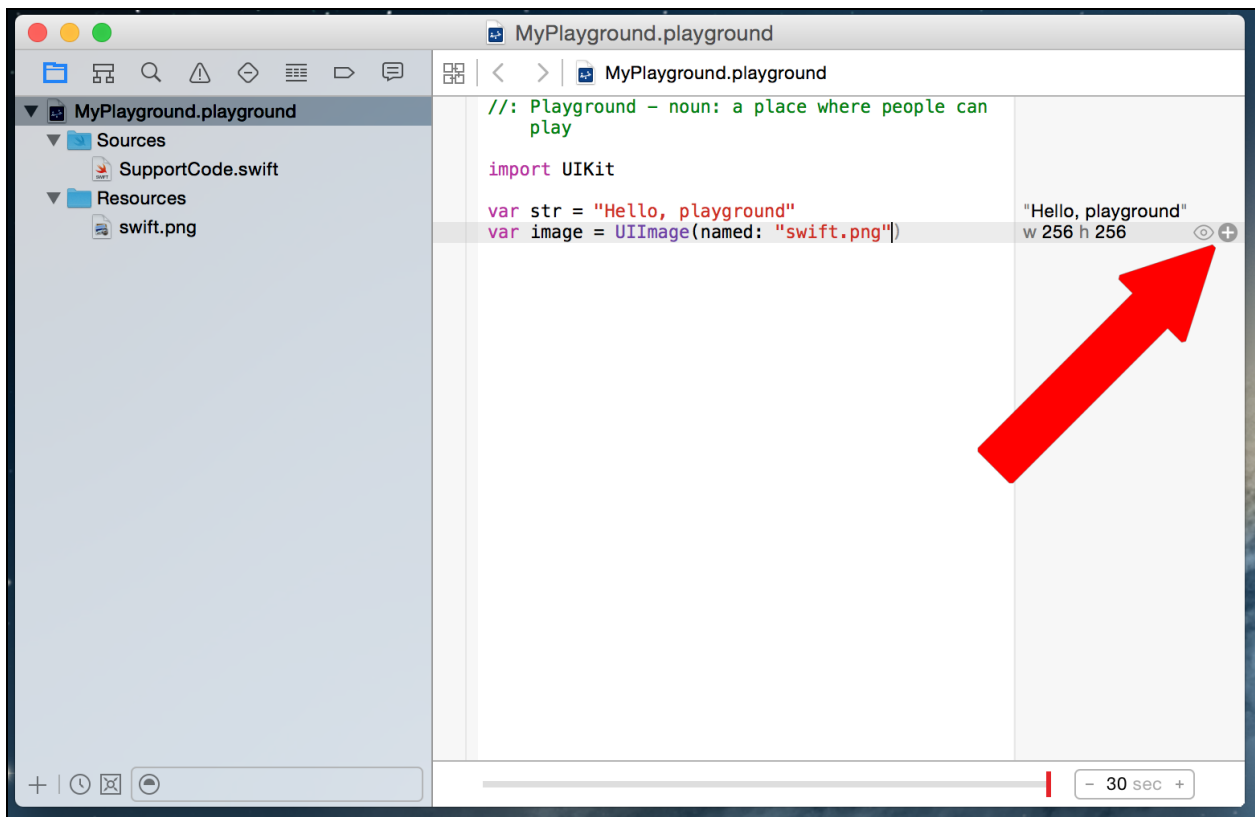
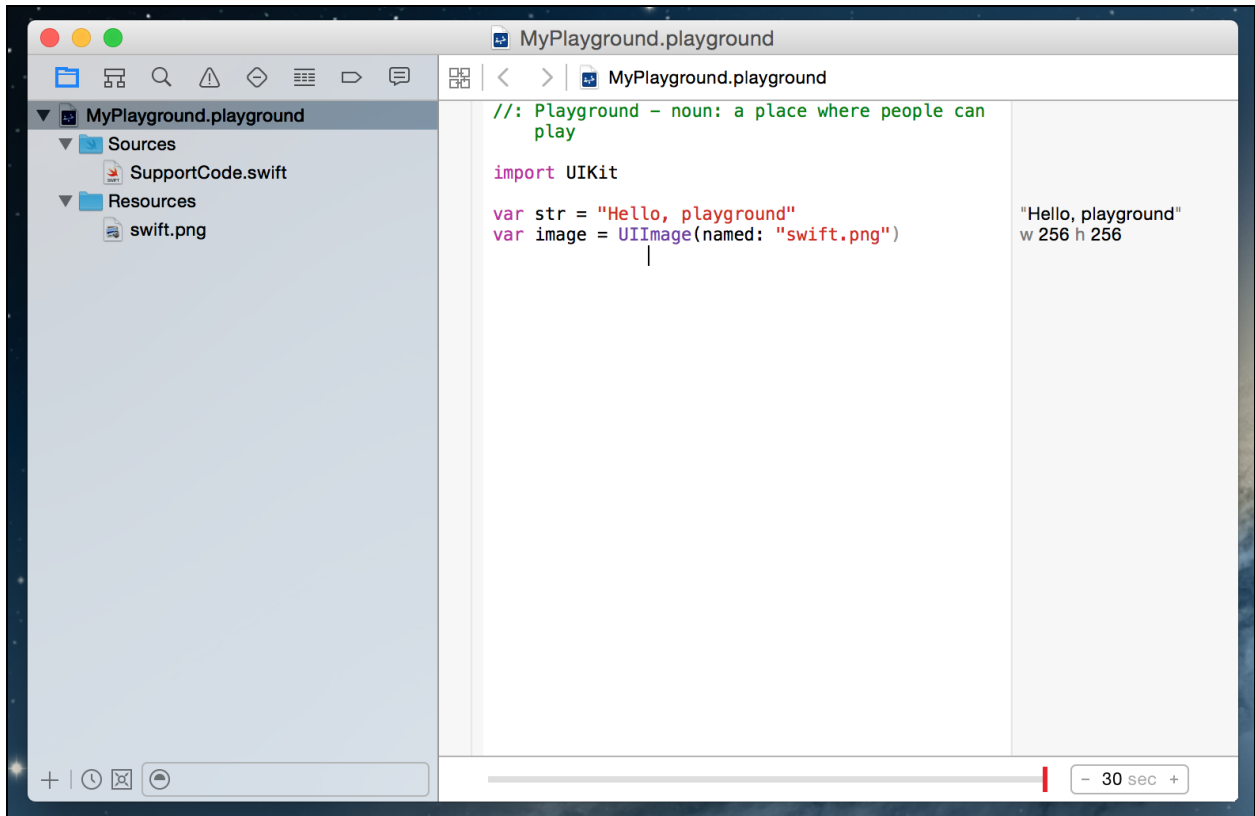


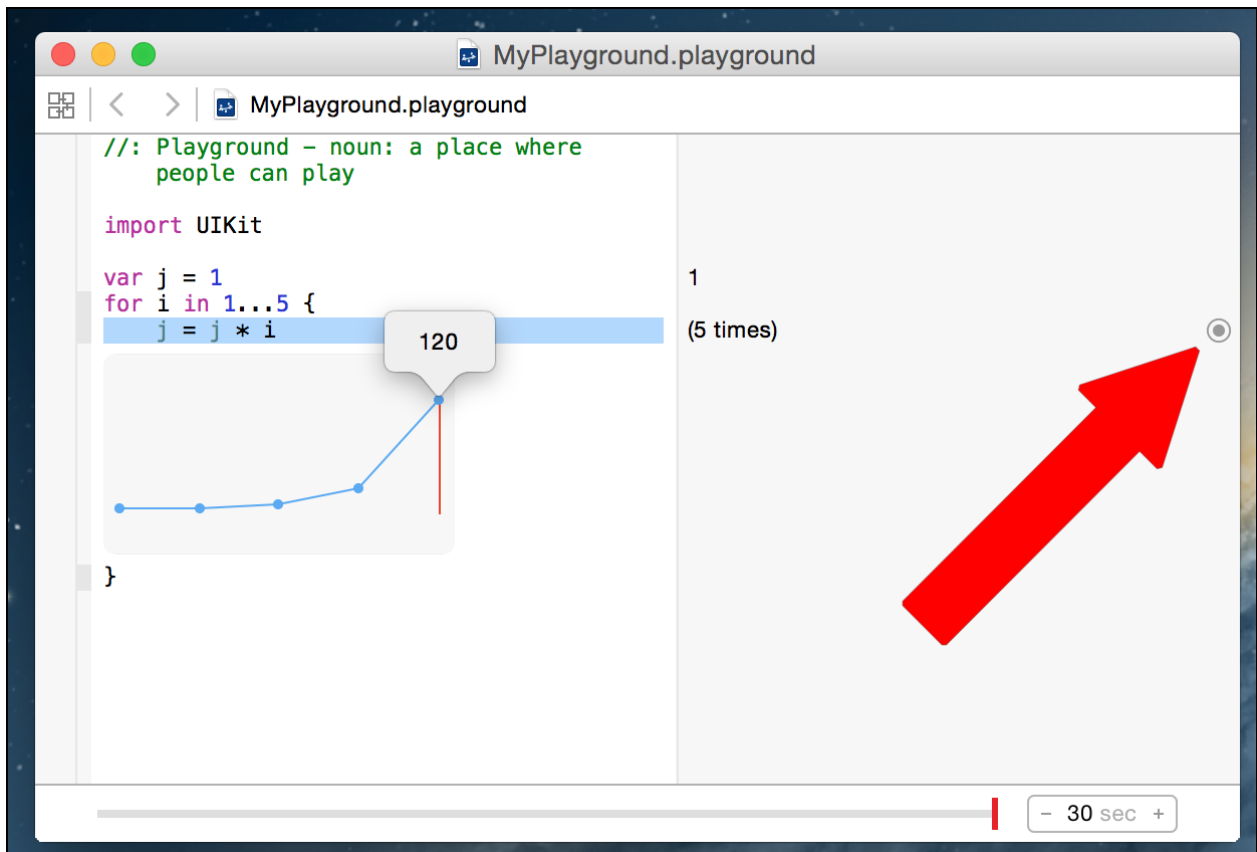
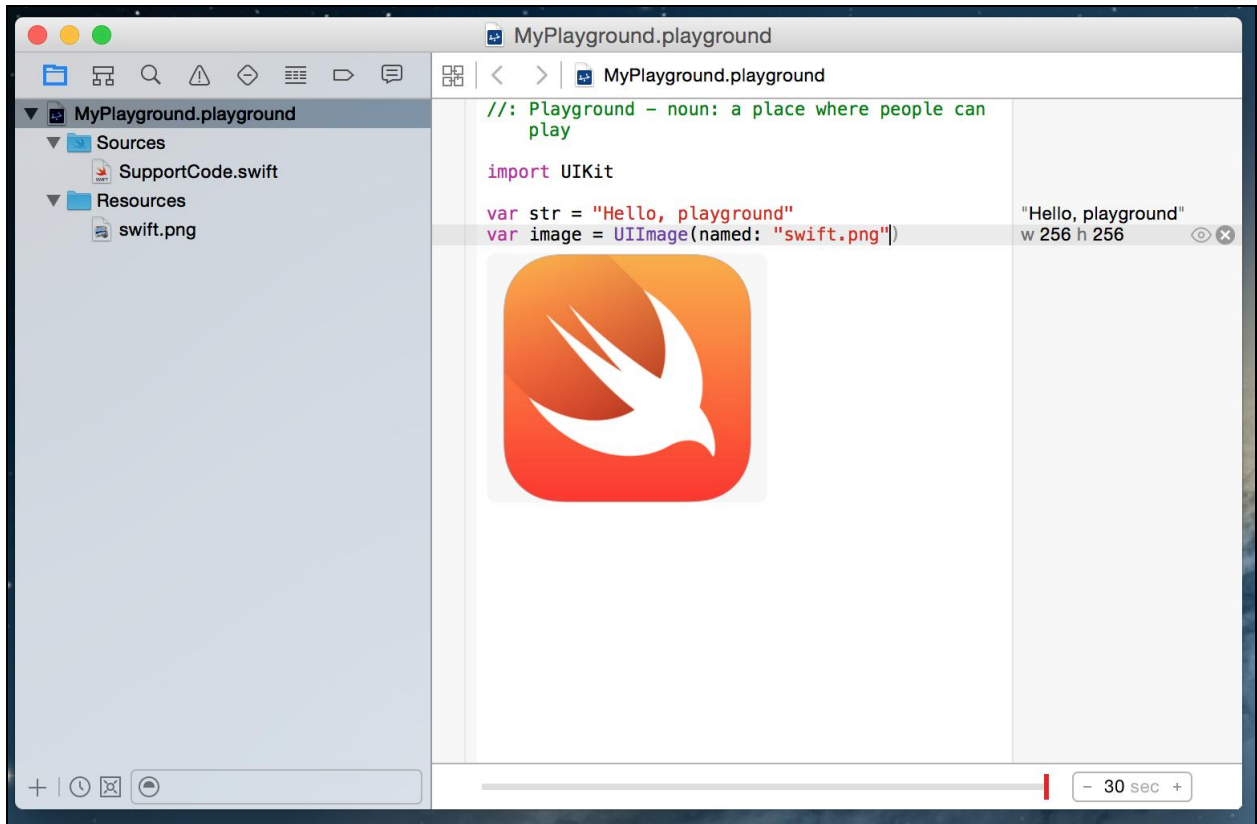
Chapter 1

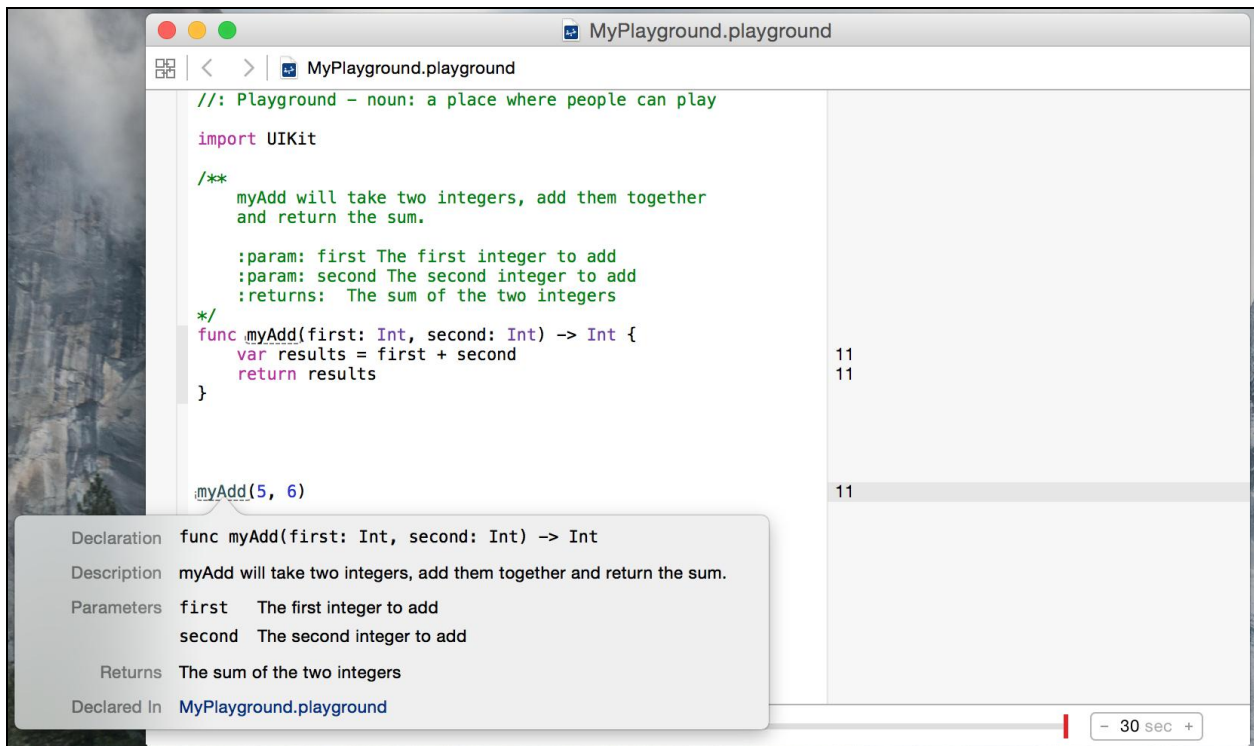
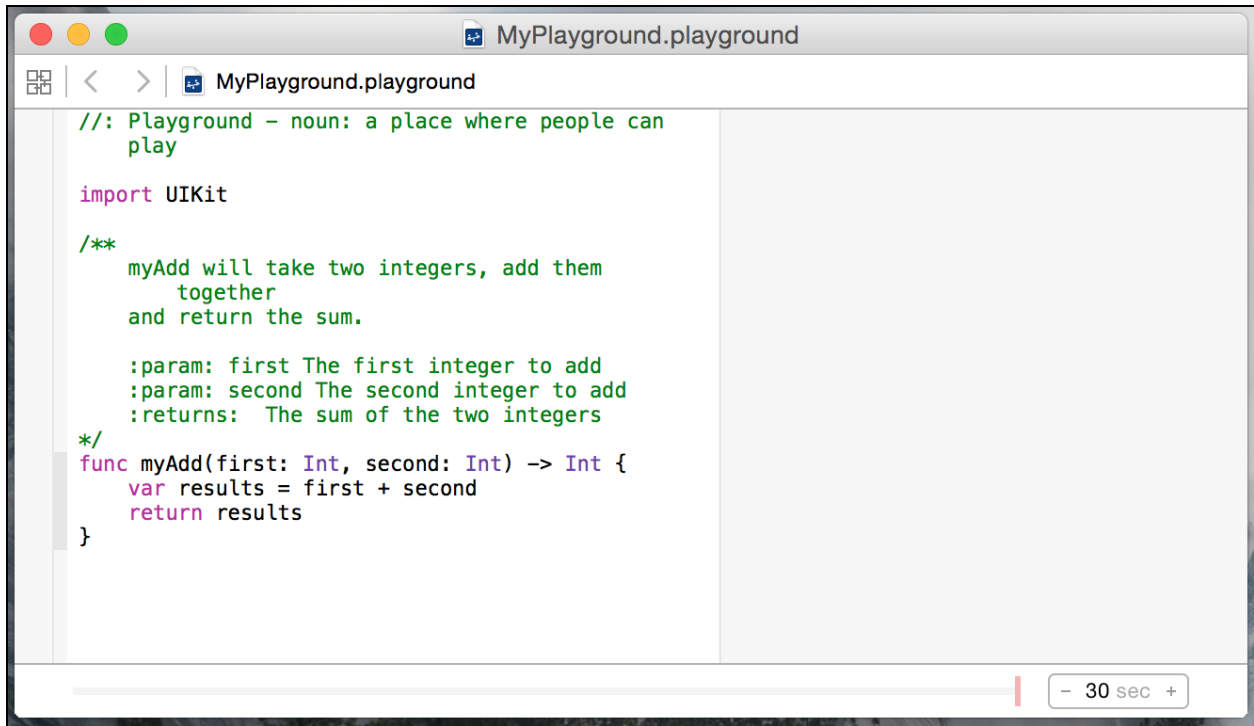












The screenshot shows a Swift playground window titled "MasteringSwift_Chapter_1.playground — Edited". The code editor contains three lines of code:

```
1  
2 println("Hello from Swift")  
3 println("Hello from Swift");
```

The output console on the right displays the results of the execution:

```
"Hello from Swift"  
"Hello from Swift"
```

The screenshot shows a Swift playground window titled "MasteringSwift_Chapter_1.playground — Edited". The code editor contains ten lines of code:

```
1  
2 var x = 1  
3  
4 if x == 1 {  
5     println("x == 1")  
6 }  
7  
8 if (x==1) {  
9     println("X == 1")  
10 }
```

The output console on the right displays the results of the execution:

```
1  
"x == 1"  
"X == 1"
```

```
MasteringSwift_Chapter_1.playground — Edited
MasteringSwift_Chapter_1.playground > No Selection

1 import UIKit
2
3 var x = 1
4 var y = 1
5 var z = 1
6
7 //Single conditional statement, no parentheses
8 if x == 1 {
9     println("X == 1")
10 }
11
12 //Multiple conditional statements, parentheses
13 if (x == 1) && (y == 1) && (z == 1) {
14     println("All vars == 1")
15 }
16
17 //Multiple conditional statements, no Parentheses
18 if x == 1 && y == 1 && z == 1 {
19     println("All vars == 1")
20 }
21 |
```

1
1
1
"X == 1"
"All vars == 1"
"All vars == 1"

```
MasteringSwift_Chapter_1.playground — Edited
MasteringSwift_Chapter_1.playground > No Selection

1
2 var x = 1
3
4 if x == 1 {
5     println("x == 1")
6 }
7
8 if x == 1
9     println("x == 1")
```

1
"x == 1"
Expected ')' after 'if' condition


```
MasteringSwift_Chapter_1.playground — Edited
MasteringSwift_Chapter_1.playground > No Selection

1 import UIKit
2
3 var i = 1;
4
5 if i = 1 {
6     println("HI")
7 }
8
9 while i = 1 {
10     println("HI")
11 }
```

1

Type '0' does not conform to protocol 'BooleanType' "i==1"

Type '0' does not conform to protocol 'BooleanType'

```
MasteringSwift_Chapter_1.playground — Edited
MasteringSwift_Chapter_1.playground > No Selection

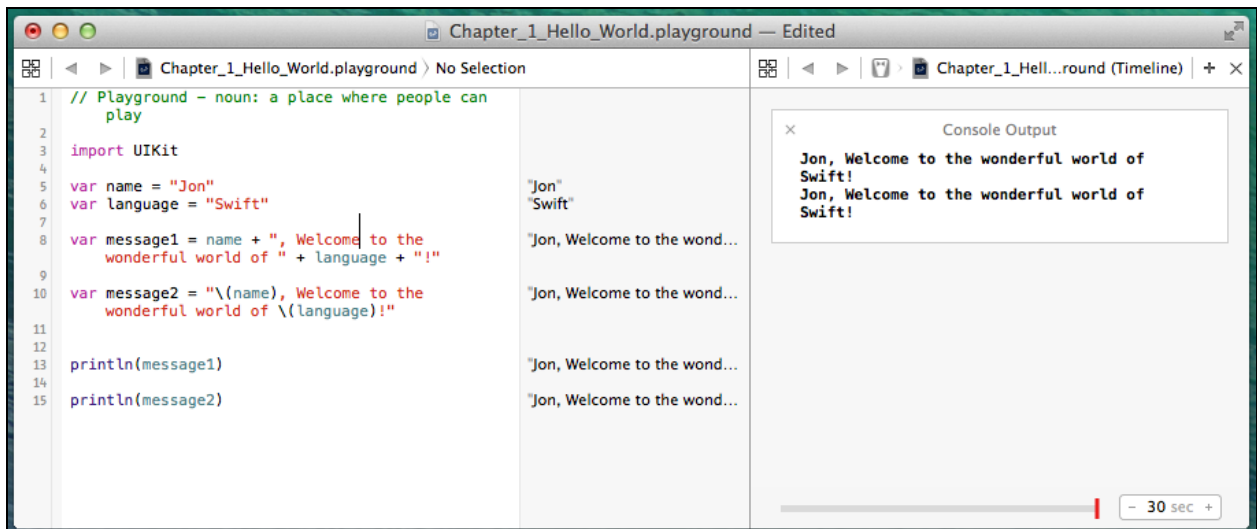
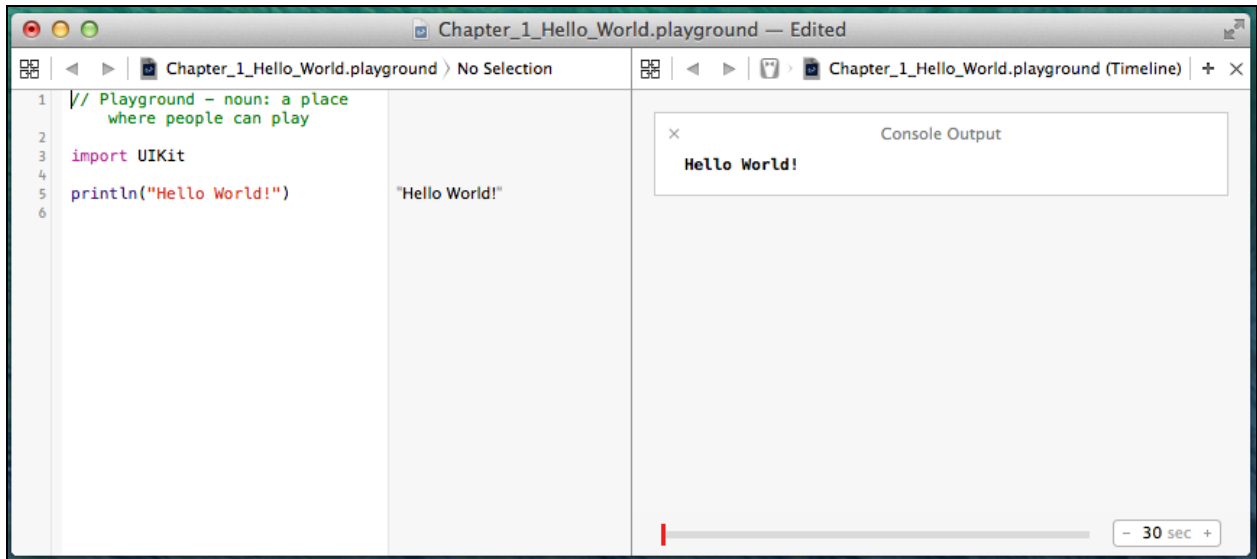
1 import UIKit
2
3 //The i block
4 var i=1
5 if i==1 {
6     println("HI")
7 }
8
9 //The j block|
10 var j = 1
11 if (j == 1) {
12     println("HI")
13 }
14
```

1

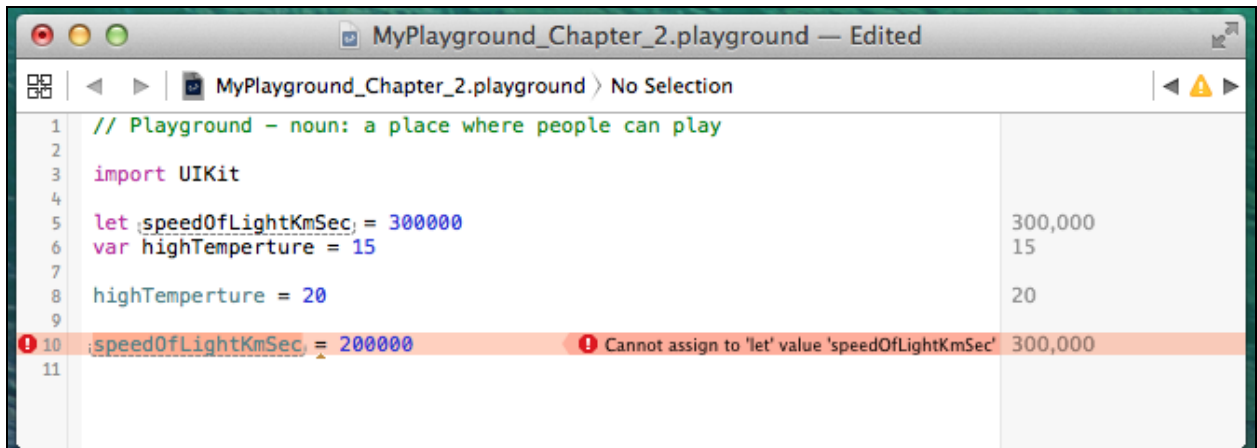
"HI"

1

"HI"



Chapter 2



The image shows a screenshot of a Swift playground editor window titled "MyPlayground_Chapter_2.playground — Edited". The editor displays the following code:

```
1 // Playground - noun: a place where people can play
2
3 import UIKit
4
5 var integerVar = 10
6
7 integerVar = "My String"
8
```

A red error message is displayed on line 7: "Type 'Int' does not conform to protocol 'StringLiteralConvertible'". The value "10" is visible in the right-hand pane of the playground.

```
MyPlayground_Chapter_2.playground — Edited
MyPlayground_Chapter_2.playground > No Selection

1 // Playground – noun: a place where
  // people can play
2
3 import UIKit
4
5 var a = UInt8.max           255
6 var b = UInt8.min          0
7
8 var c = UInt16.max          65,535
9 var d = UInt16.min          0
10
11 var e = UInt32.max          4,294,967,295
12 var f = UInt32.min          0
13
14 var g = UInt64.max          18446744073709551615
15 var h = UInt64.min          0
16
17 var j = UInt.max            18446744073709551615
18 var k = UInt.min            0
19
20 var l = Int8.max             127
21 var m = Int8.min            -128
22
23 var n = Int16.max            32,767
24 var o = Int16.min           -32,768
25
26 var p = Int32.max            2,147,483,647
27 var q = Int32.min           -2,147,483,648
28
29 var r = Int64.max            9,223,372,036,854,775,807
30 var s = Int64.min           -9,223,372,036,854,775,808
31
32 var t = Int.max              9,223,372,036,854,775,807
33 var u = Int.min              -9,223,372,036,854,775,808
34
35
36
```

The screenshot shows a Swift playground window titled "MyPlayground_Chapter_2.playground — Edited". The code in the editor is as follows:

```
1 // Playground - noun: a place where  
2   people can play  
3 import UIKit  
4  
5 var z = 95  
6 var b = 0b1011111  
7 var c = 0o137  
8 var d = 0x5f  
9  
10
```

The output pane on the right displays the memory addresses for each variable:

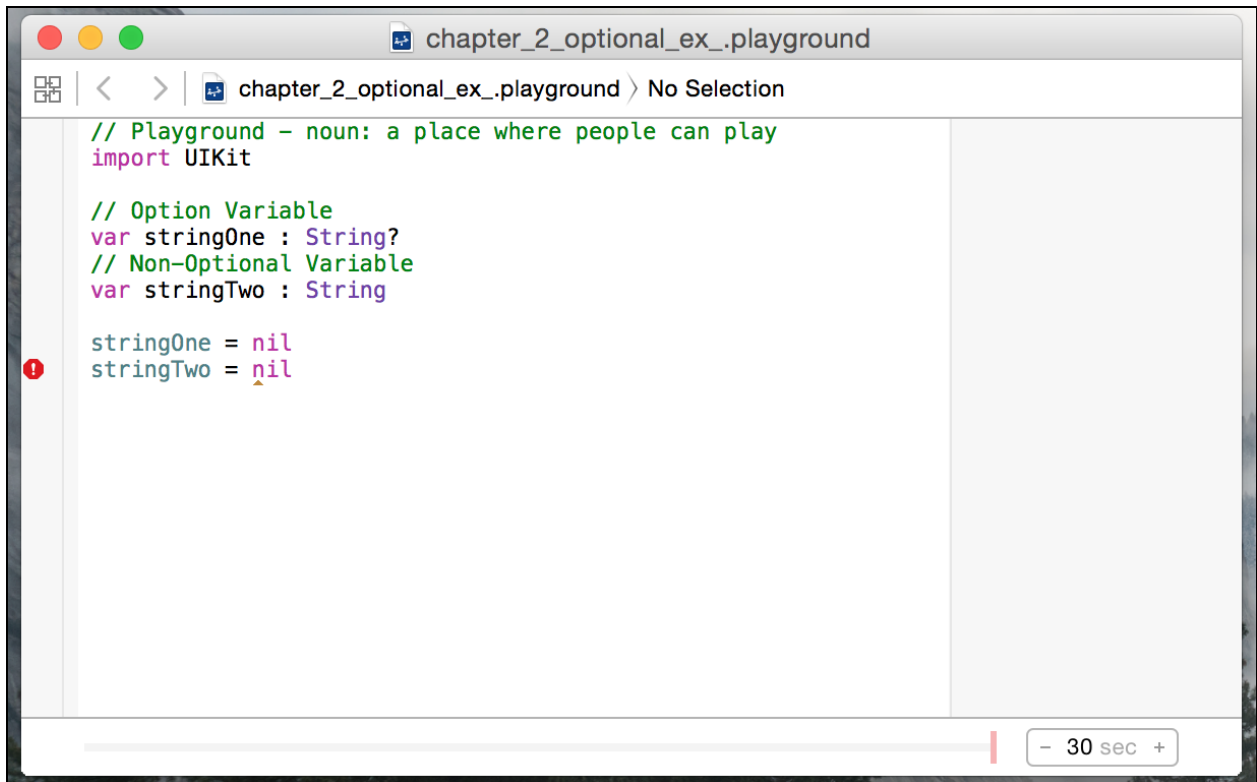
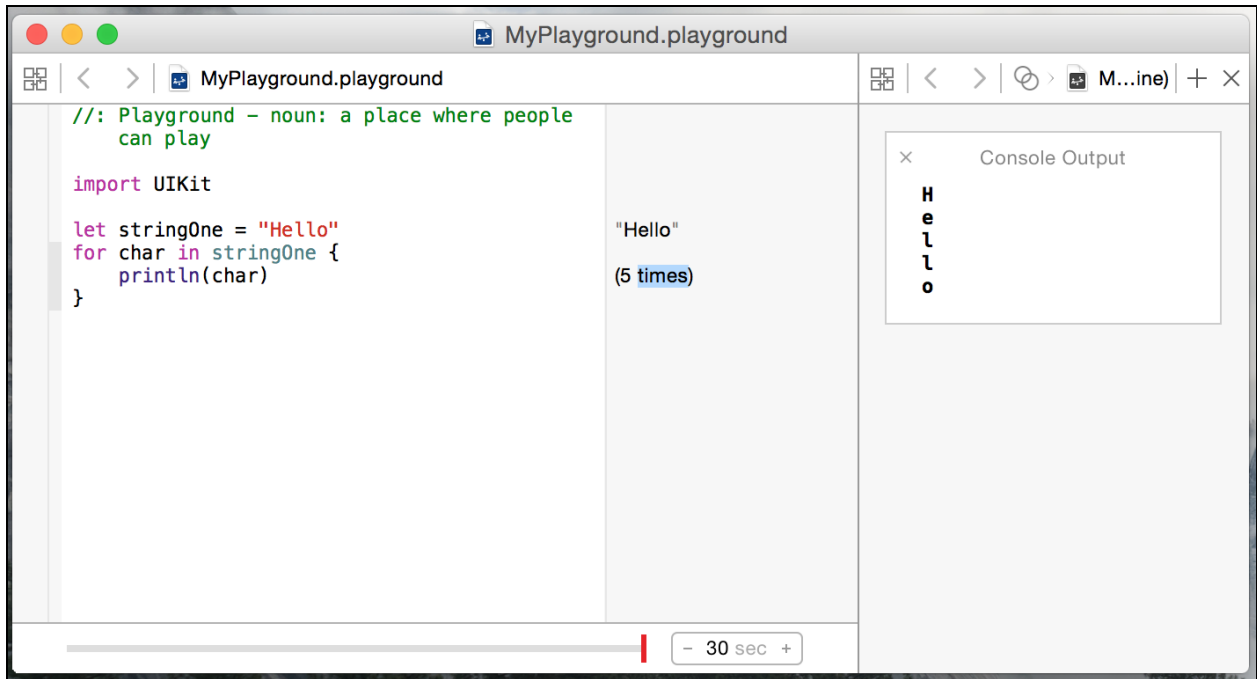
```
95  
95  
95  
95
```

The screenshot shows a Swift playground window titled "MyPlayground_Chapter_2.playground — Edited". The code in the editor is as follows:

```
1 // Playground - noun: a place where people can play  
2  
3 import UIKit  
4  
5 var x : Float = 3.14  
6  
7 var y = 3.14  
8
```

The output pane on the right displays the memory addresses for each variable:

```
3.14000010490417  
3.14
```



```
MyPlayground_Chapter_2.playground — Edited
MyPlayground_Chapter_2.playground > No Selection

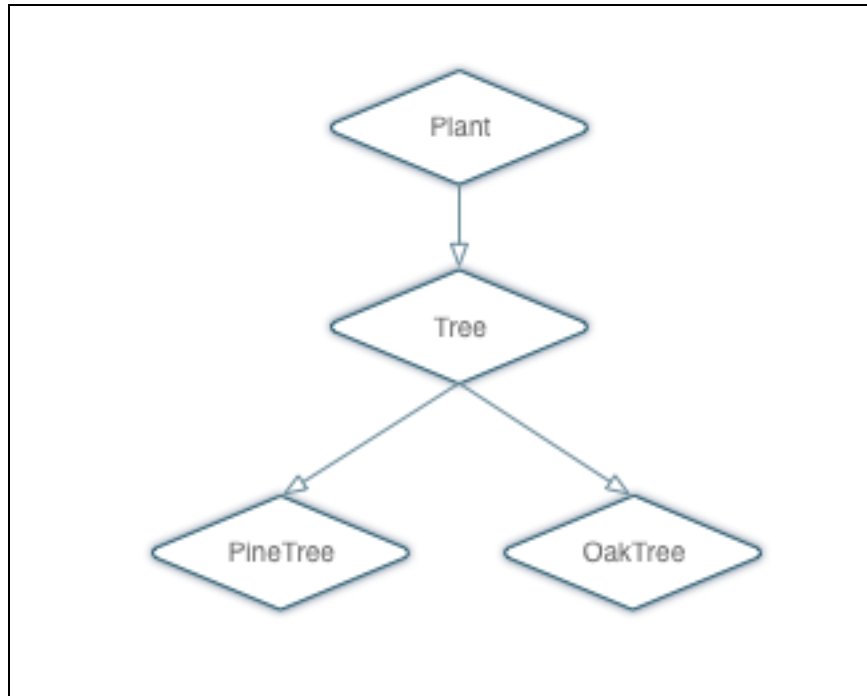
1 // Playground - noun: a place where people can play
2
3 import UIKit
4
5 //Non-Optional Variable
6 var stringOne : String
7
8 println(stringOne)
9
10
```

! Variable 'stringOne' used before being initialized

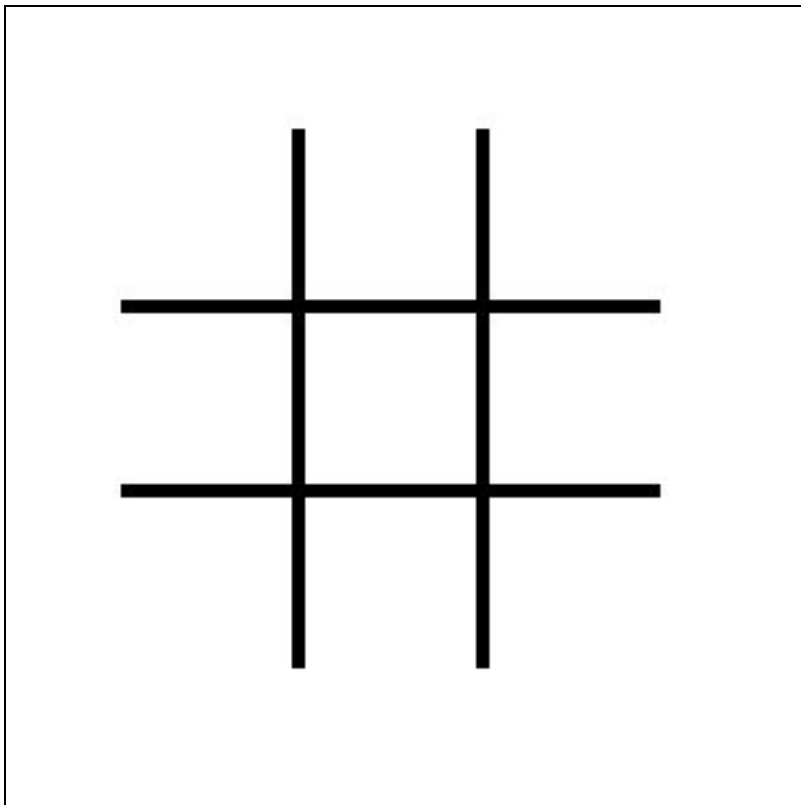
```
MyPlayground_Chapter_2.playground
MyPlayground_Chapter_2.playground > No Selection

1 // Playground - noun: a place where people can play
2
3 import UIKit
4
5 //Optional Variable
6 var stringOne : String? nil
7
8 //-----stringOne is nil-----//
9 //Explicitly check for nil
10 if stringOne != nil {
11     println(stringOne)
12 } else {
13     println("Explicit Check: stringOne is nil") "Explicit Check: stringOne is nil"
14 }
15
16 //Optional binding
17 if let tmp = stringOne {
18     println(tmp)
19 } else {
20     println("Optional Binding: stringOne is nil") "Optional Binding: stringOne is nil"
21 }
22
23 //Optional chaining
24 var newString = stringOne?.lastPathComponent nil
25
26 //-----Add value to stringOne-----//
27 stringOne = "http://www.packtpub.com/all" {Some "http://www.packtpub.com/all"}
28
29 //Explicitly check for nil
30 if stringOne != nil {
31     println(stringOne)
32 } else {
33     println("Explicit Check: stringOne is nil") "Optional("http://www.packtpub.com/all")"
34 }
35
36 //Optional binding
37 if let tmp = stringOne {
38     println(tmp)
39 } else {
40     println("Optional Binding: stringOne is nil") "http://www.packtpub.com/all"
41 }
42
43 //Optional chaining
44 var pathString = stringOne?.lastPathComponent {Some "all"}
```

Chapter 5



Chapter 7



Choose a template for your new project:

iOS

Application

Framework & Library

Other

cocos2d

OS X

Application

Framework & Library

System Plug-in

Other

cocos2d



Master-Detail
Application



Page-Based
Application



Single View
Application



Tabbed
Application



Game

Single View Application

This template provides a starting point for an application that uses a single view. It provides a view controller to manage the view, and a storyboard or nib file that contains the view.

Cancel

Previous

Next

Choose options for your new project:

Product Name: ObjectiveCProject

Organization Name: Jon Hoffman

Organization Identifier: hoffman.jon

Bundle Identifier: hoffman.jon.ObjectiveCProject

Language: Objective-C

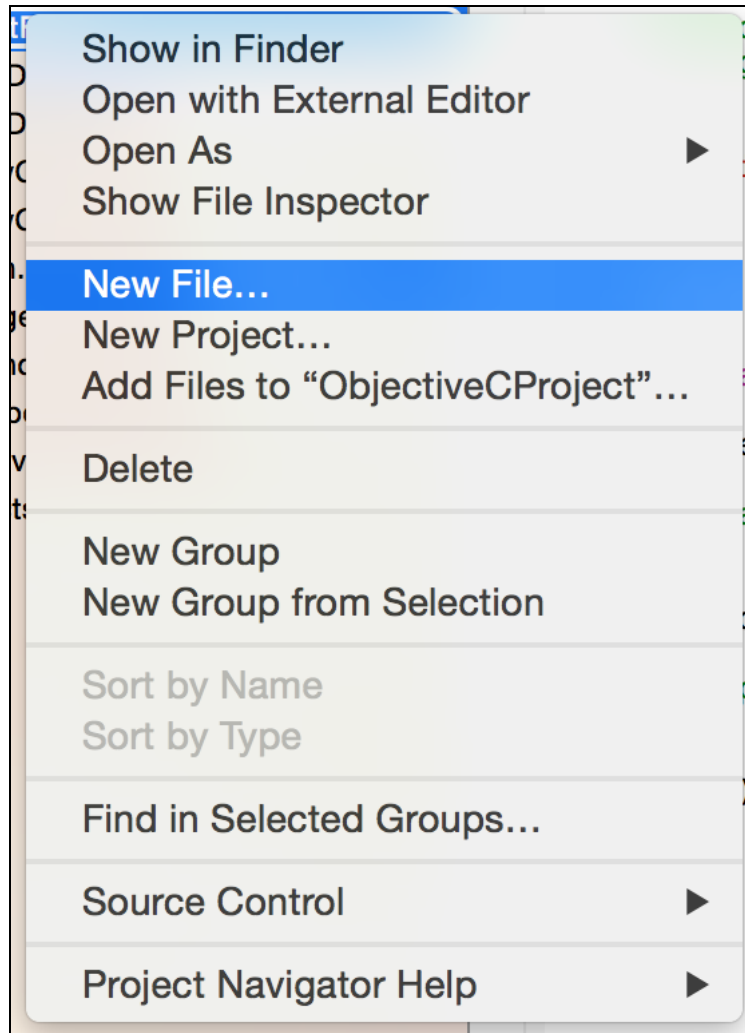
Devices: Universal

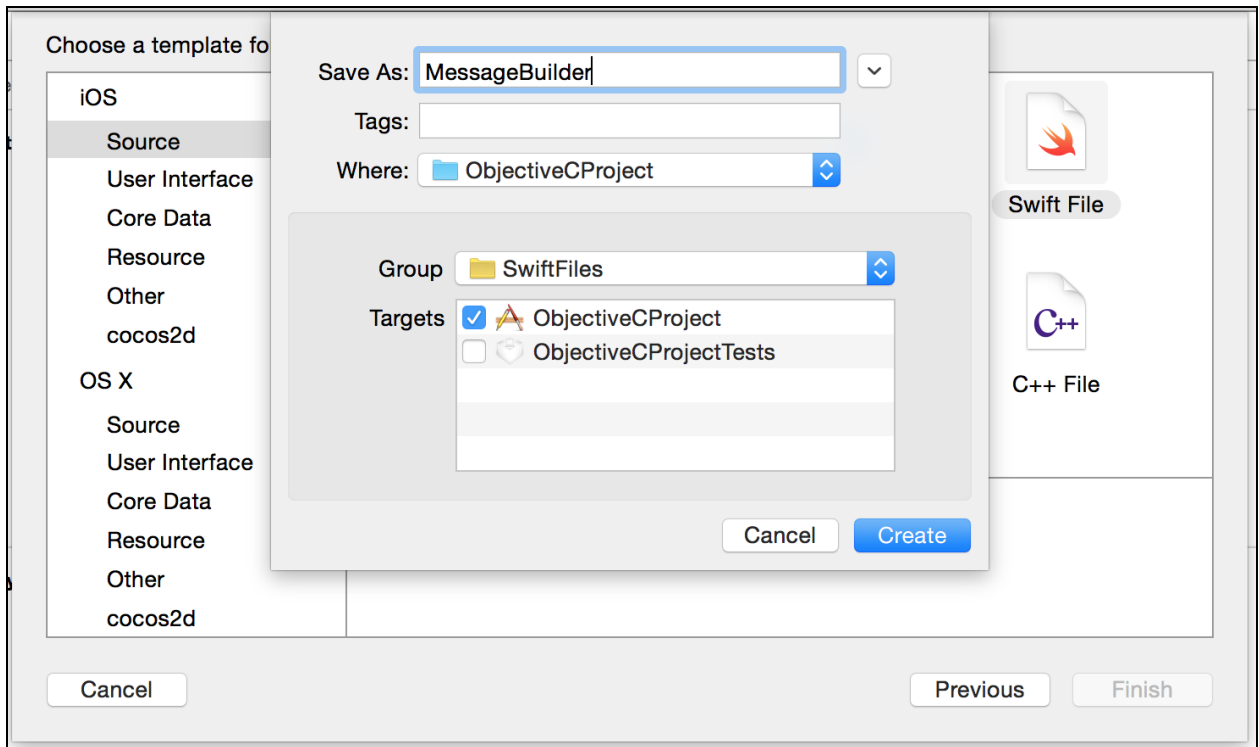
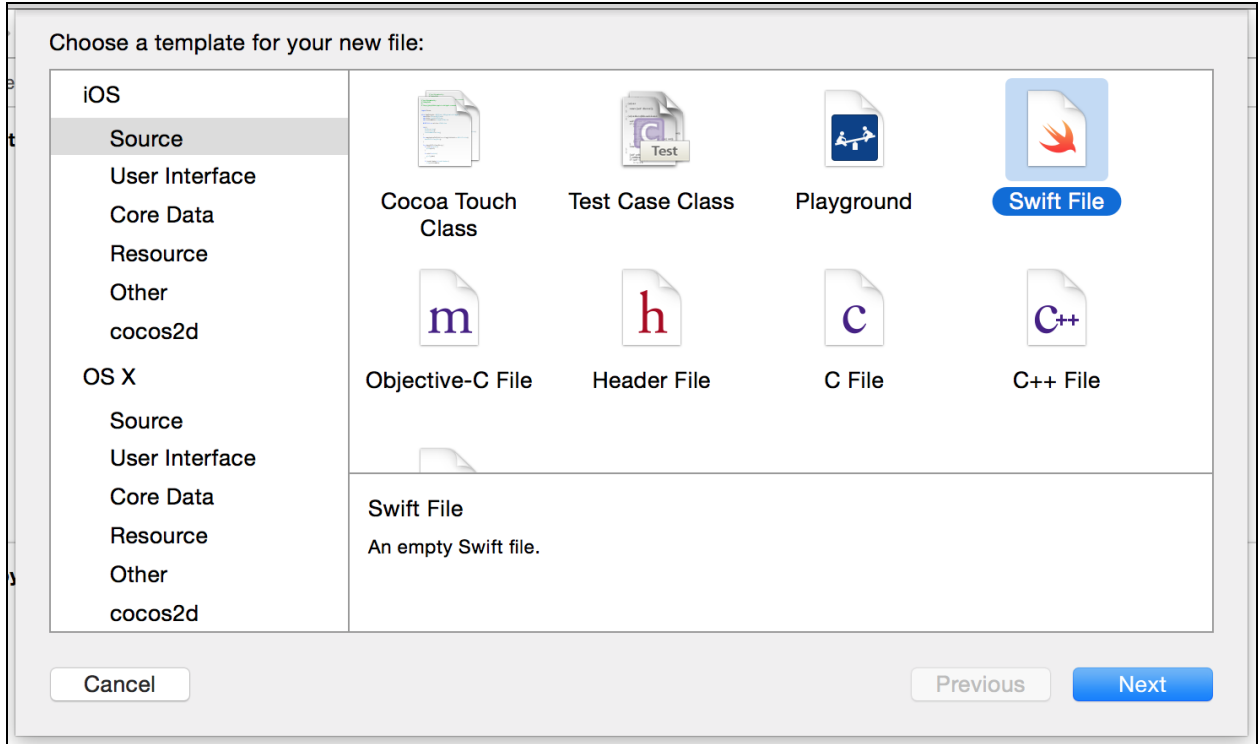
Use Core Data

Cancel

Previous


Next









Choose a template for


- iOS
 - Source
 - User Interface
 - Core Data
 - Resource
 - Other
 - cocos2d
- OS X
 - Source
 - User Interface
 - Core Data
 - Resource
 - Other
 - cocos2d



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

Adding this file to ObjectiveCProject 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 File	Header File	C File	C++ File



Swift File

Swift File

An empty Swift file.

ObjectiveCProject: Ready | Today at 5:06 PM

ObjectiveCProject

ObjectiveCProject 2 targets, iOS SDK 8.1

- ObjectiveCProject
 - SwiftFiles
 - MessageBuilder.swift A
 - Objectiv...-Header.h A
 - AppDelegate.h
 - AppDelegate.m
 - ViewController.h M
 - ViewController.m M
 - Main.storyboard
 - Images.xcassets
 - LaunchScreen.xib
 - Supporting Files
 - ObjectiveCProjectTests
 - Products

ObjectiveCProject

Basic All Combined Levels +

▼ Packaging

Setting	ObjectiveCProject
Info.plist File	ObjectiveCProject/Info.plist
Product Name	ObjectiveCProject

▼ Apple LLVM 6.0 - Language - Modules

Setting	ObjectiveCProject
▶ Enable Modules (C and Objec... Yes ▾	

▼ Asset Catalog Compiler - Options

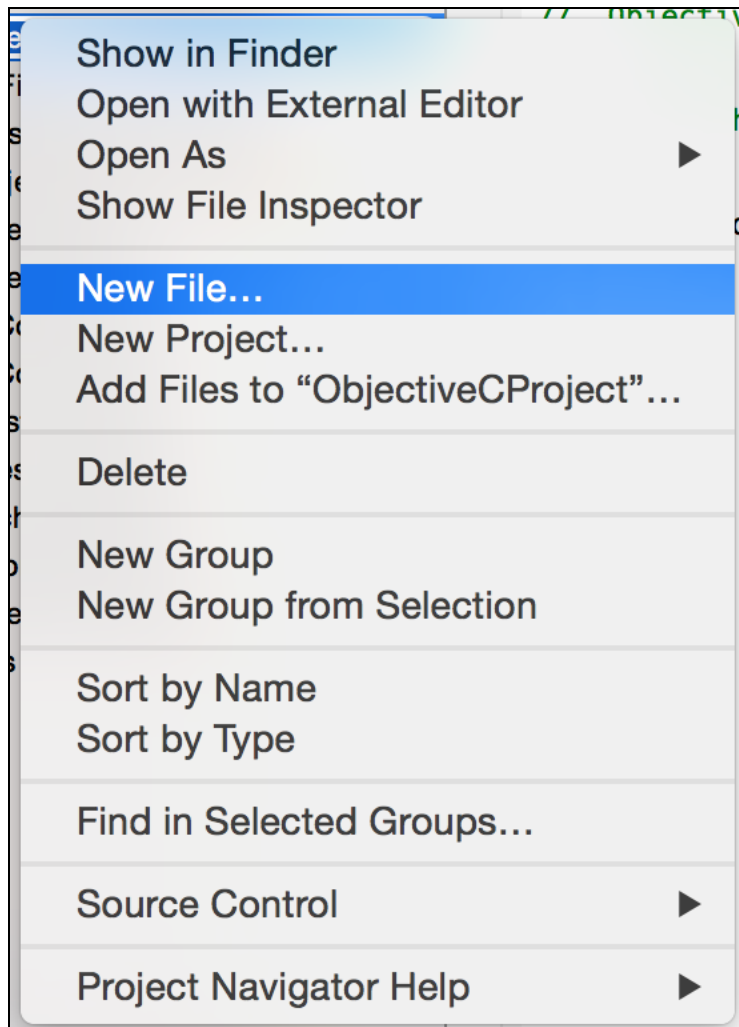
Setting	ObjectiveCProject
Asset Catalog App Icon Set N...	AppIcon

▼ Swift Compiler - Code Generation









Setting	ObjectiveCProject
Objective-C Bridging Header	ObjectiveCProject/ObjectiveCProject-Bridging-Header.h
▼ Optimization Level	<Multiple values> ▾
Debug	None [-Onone] ▾
Release	Fastest [-O] ▾

▼ User-Defined

Setting	ObjectiveCProject
---------	-------------------



Choose a template for your new file:

iOS				
Source	Cocoa Touch Class	Test Case Class	Playground	Swift File
User Interface				
Core Data	Objective-C File	Header File	C File	C++ File
Resource				
Other				
cocos2d				
OS X				
Source				
User Interface				
Core Data				
Resource				
Other				
cocos2d				

Cancel Previous Next

Choose options for your new file:

Class:

Subclass of: ▾

Also create XIB file

▾

Language: ▾

Cancel Previous Next

Chapter 13

