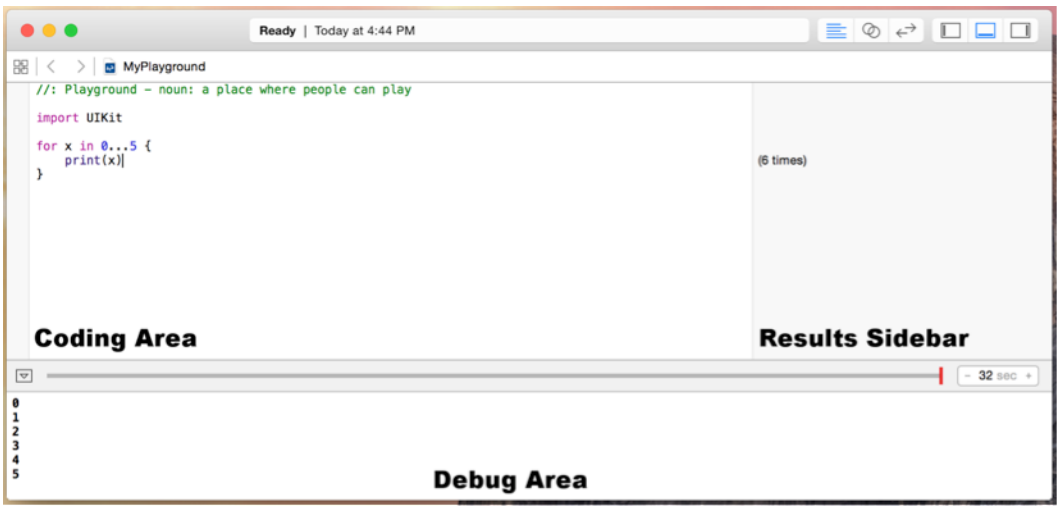
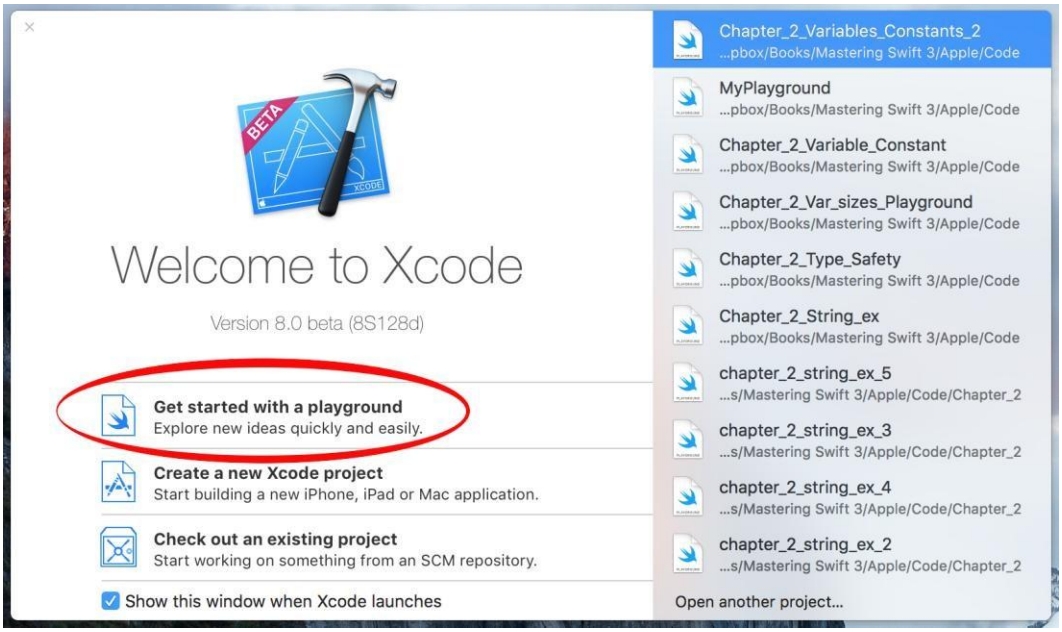
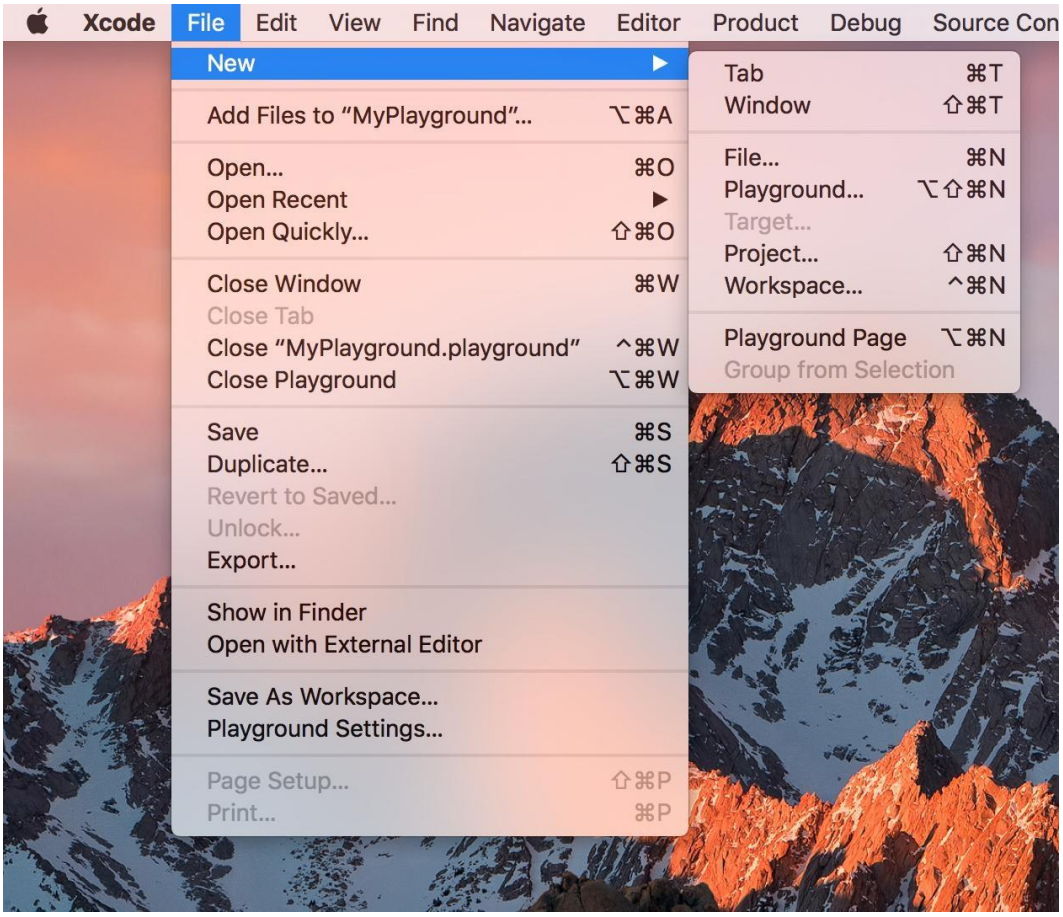
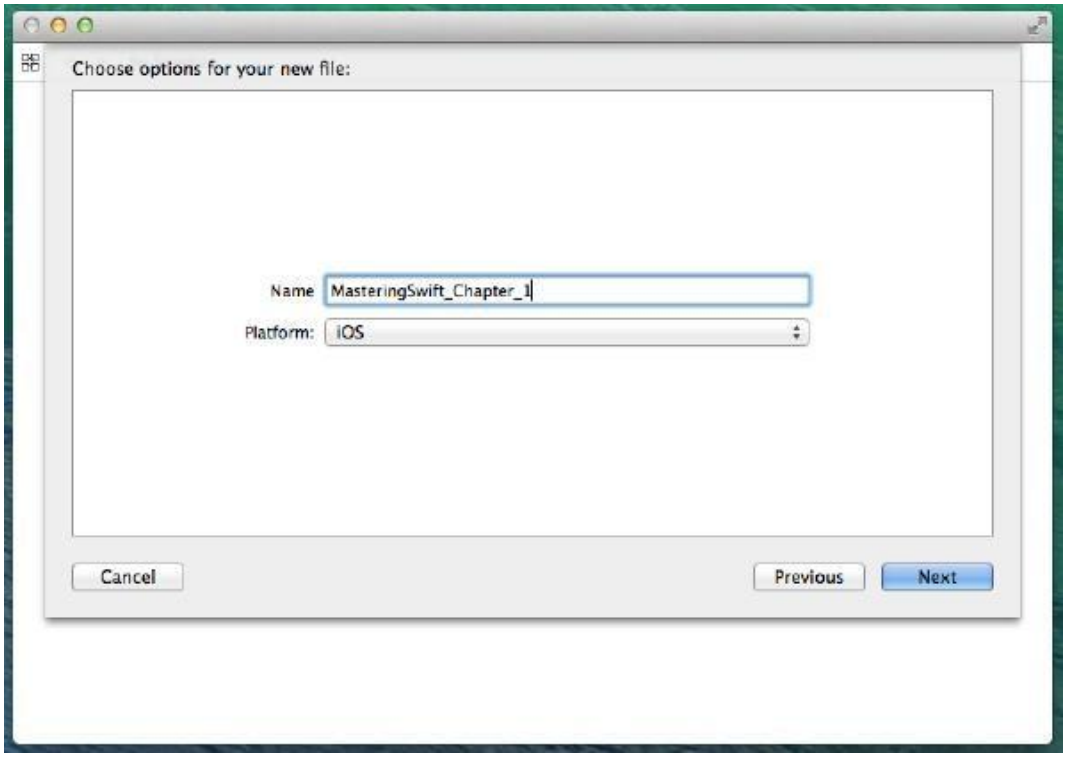
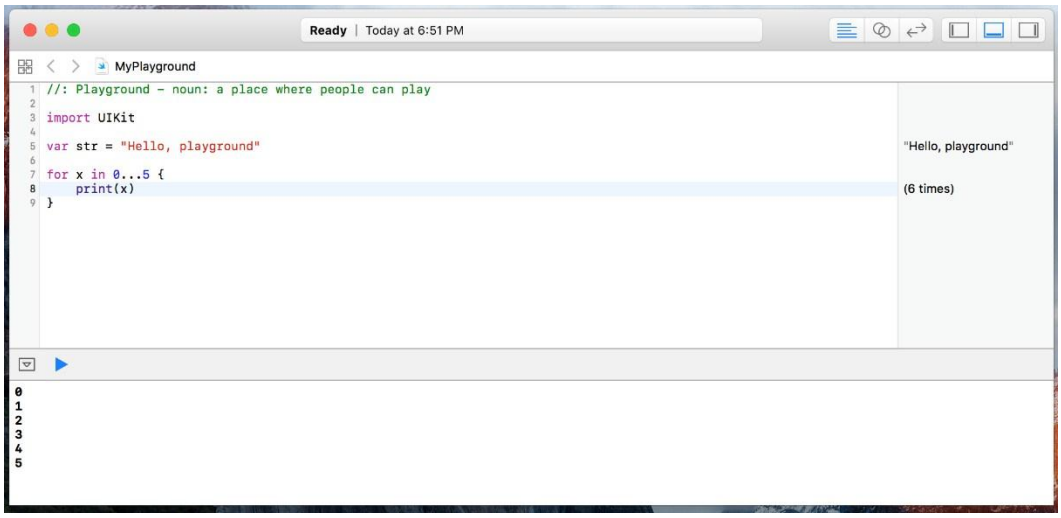
Chapter 1: Taking the First Steps with Swift

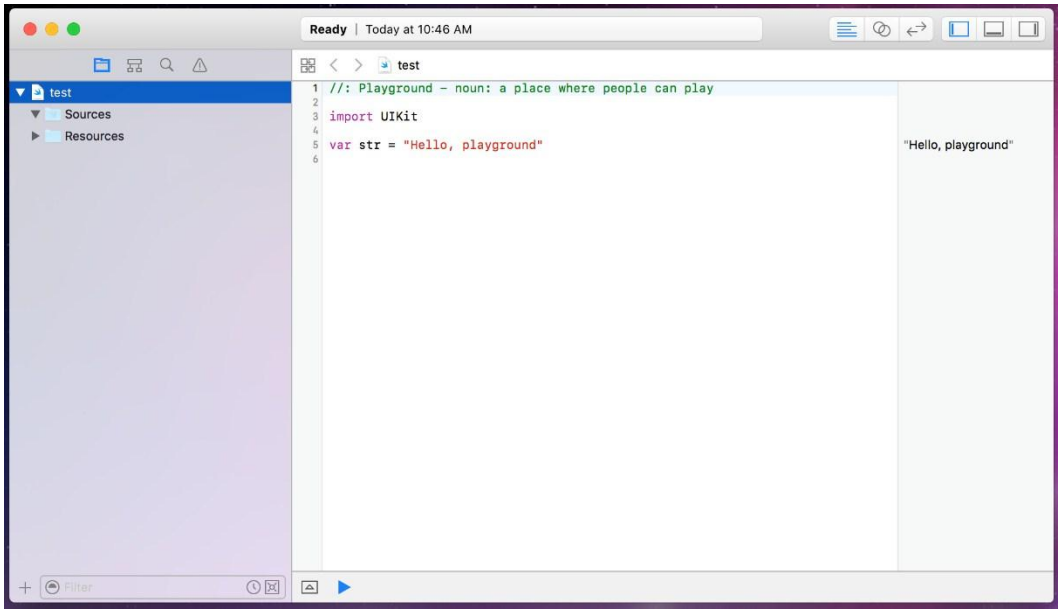


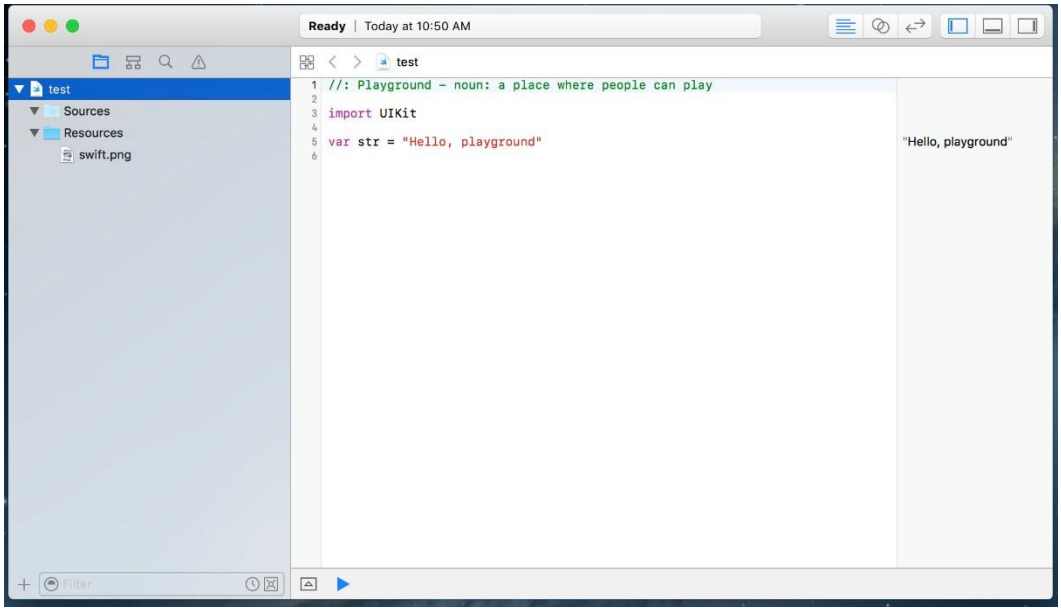


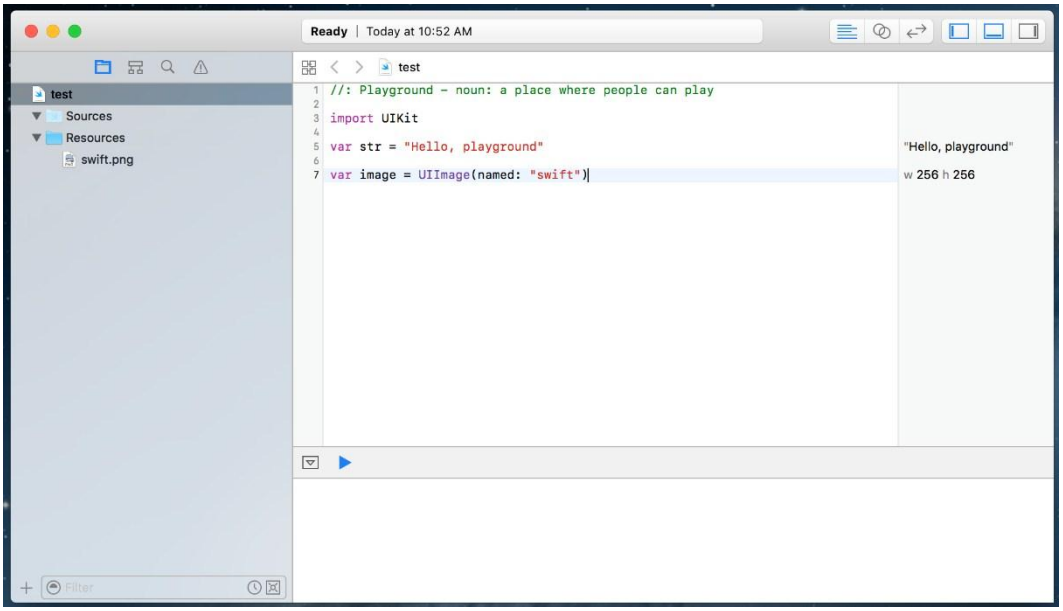

Xcode
File
Edit
View
Find
Navigate
Editor
Product
Debug
Source Con
New
Add Files to “MyPlayground”... ⌥⌘A
Open... ⌘O
Open Recent
Open Quickly... ⇧⌘O
Close Window ⌘W
Close Tab
Close “MyPlayground.playground” ^⌘W
Close Playground ⌥⌘W
Save ⌘S
Duplicate... ⇧⌘S
Revert to Saved...
Unlock...
Export...
Show in Finder
Open with External Editor
Save As Workspace...
Playground Settings...
Page Setup... ⇧⌘P
Print... ⌘P
Tab ⌘T
Window ⇧⌘T
File... ⌘N
Playground... ⌥⇧⌘N
Target...
Project... ⇧⌘N
Workspace... ^⌘N
Playground Page ⌥⌘N
Group from Selection

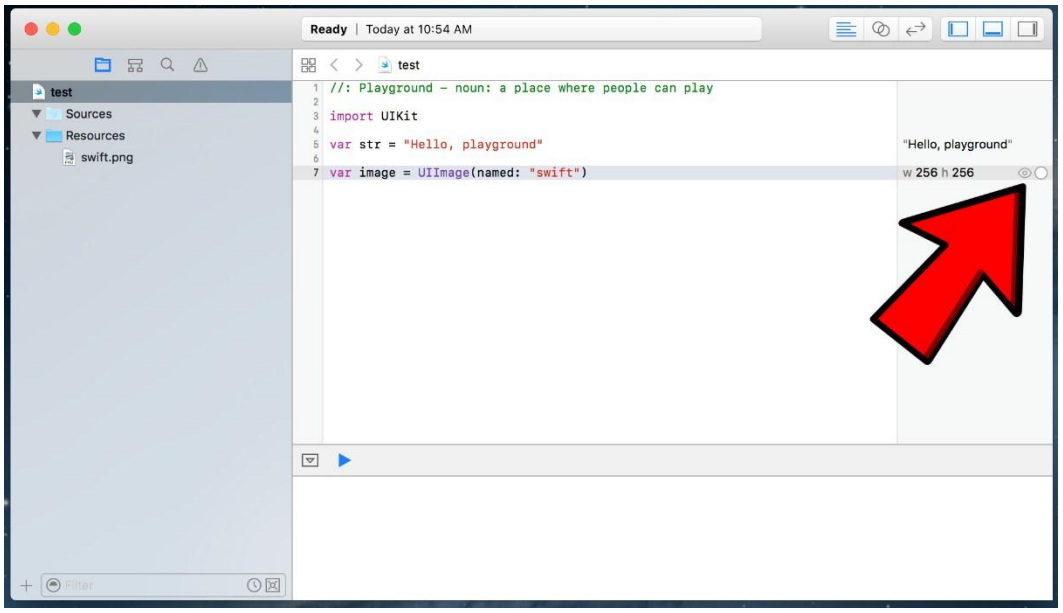

Choose options for your new file:
Name
MasteringSwift_Chapter_1
Platform:
iOS
Cancel
Previous
Next

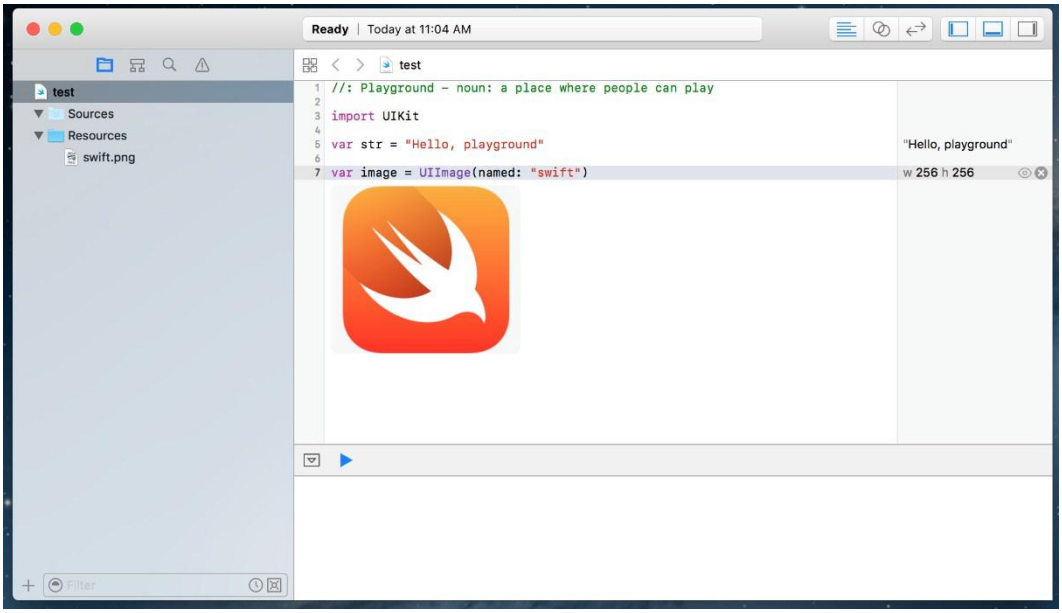

Ready | Today at 6:51 PM
MyPlayground
//: Playground - noun: a place where people can play
import UIKit
var str = "Hello, playground"
for x in 0...5 {
print(x)
}
"Hello, playground"
(6 times)
0
1
2
3
4
5

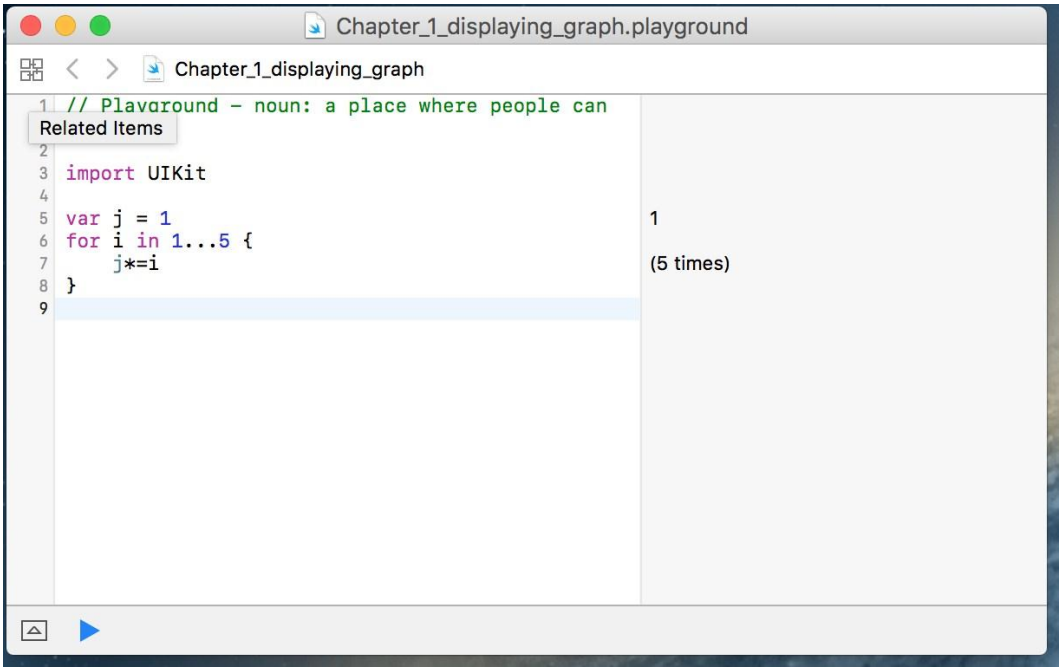

Ready | Today at 10:46 AM
test
test
Sources
Resources
1 //: Playground - noun: a place where people can play
2
3 import UIKit
4
5 var str = "Hello, playground"
6
"Hello, playground"
Filter

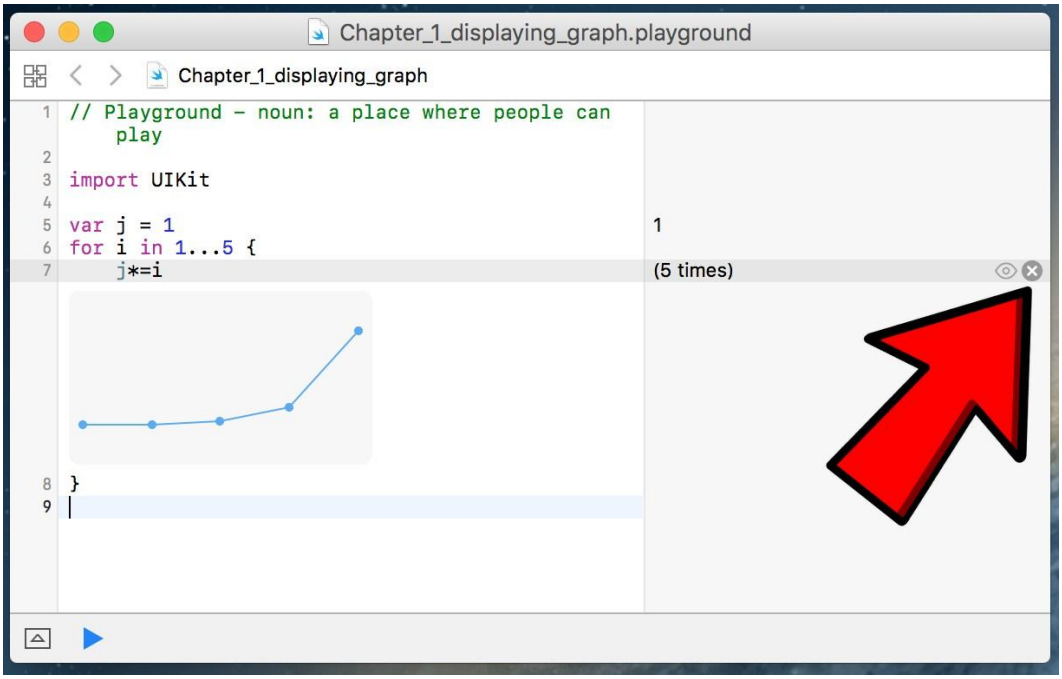

Ready | Today at 10:50 AM
test
test
Sources
Resources
swift.png
1 //: Playground - noun: a place where people can play
2
3 import UIKit
4
5 var str = "Hello, playground"
6
"Hello, playground"
Filter

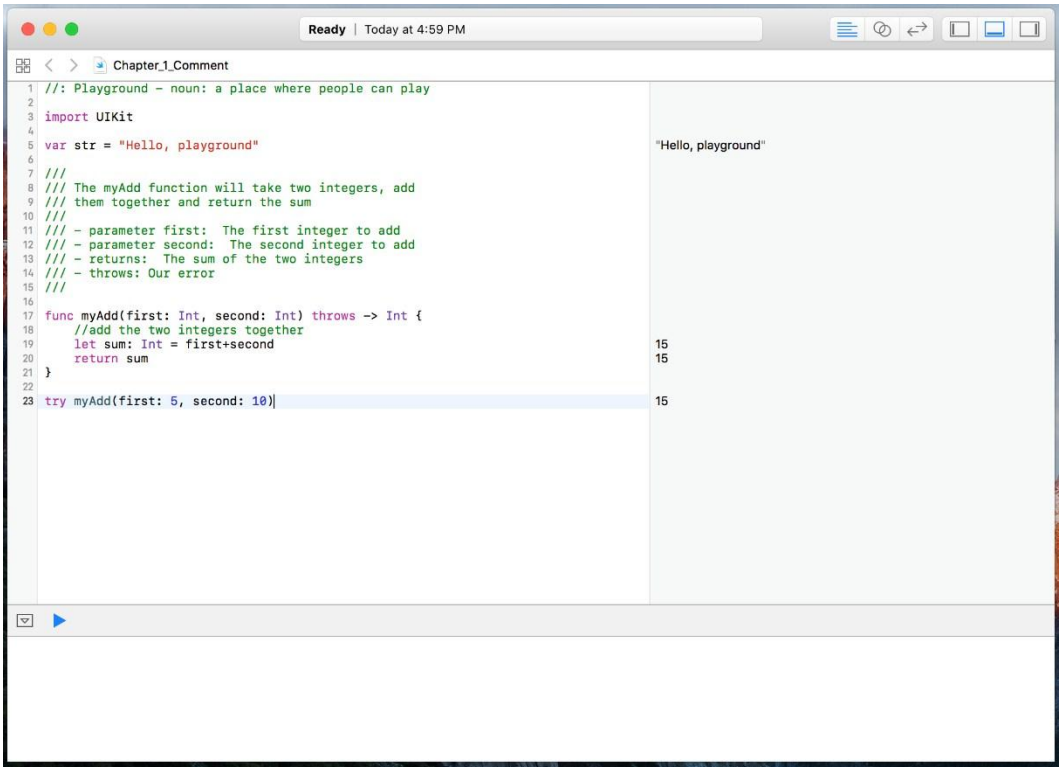

Ready | Today at 10:52 AM
test
test
Sources
Resources
swift.png
1 //: Playground - noun: a place where people can play
2
3 import UIKit
4
5 var str = "Hello, playground"
6
7 var image = UIImage(named: "swift")
"Hello, playground"
w 256 h 256

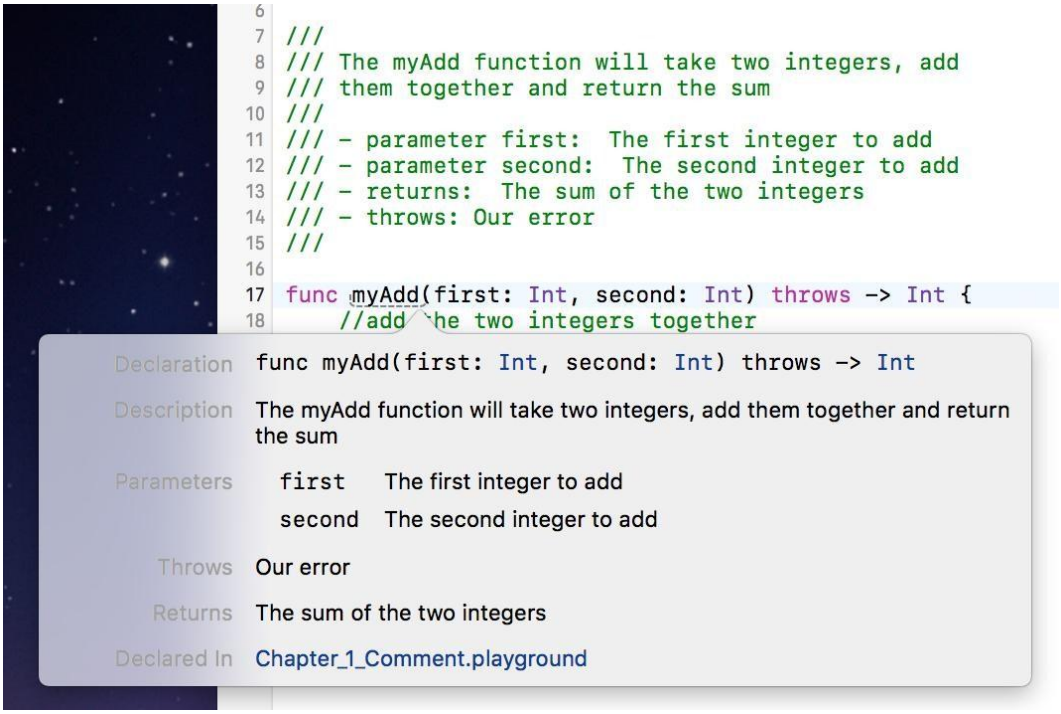

Ready | Today at 10:54 AM
test
test
Sources
Resources
swift.png
1 //: Playground - noun: a place where people can play
2
3 import UIKit
4
5 var str = "Hello, playground"
6
7 var image = UIImage(named: "swift")
"Hello, playground"
w 256 h 256

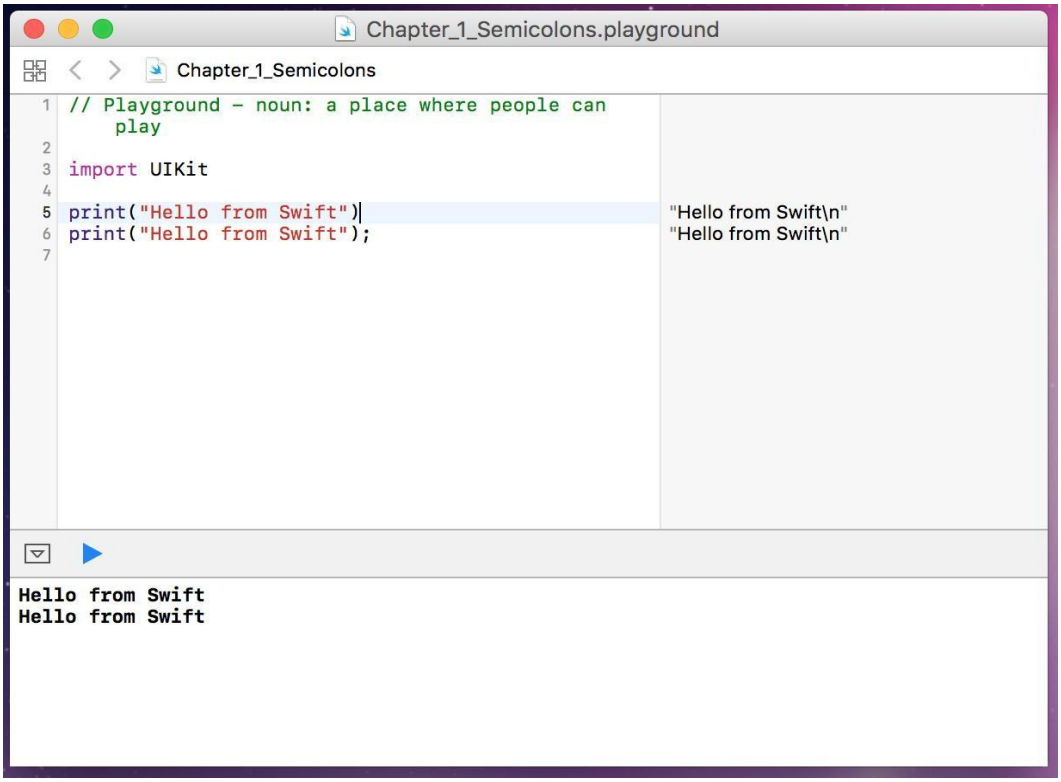

Ready | Today at 11:04 AM
test
Sources
Resources
swift.png
test
//: Playground - noun: a place where people can play
import UIKit
var str = "Hello, playground"
"Hello, playground"
var image = UIImage(named: "swift")
w 256 h 256

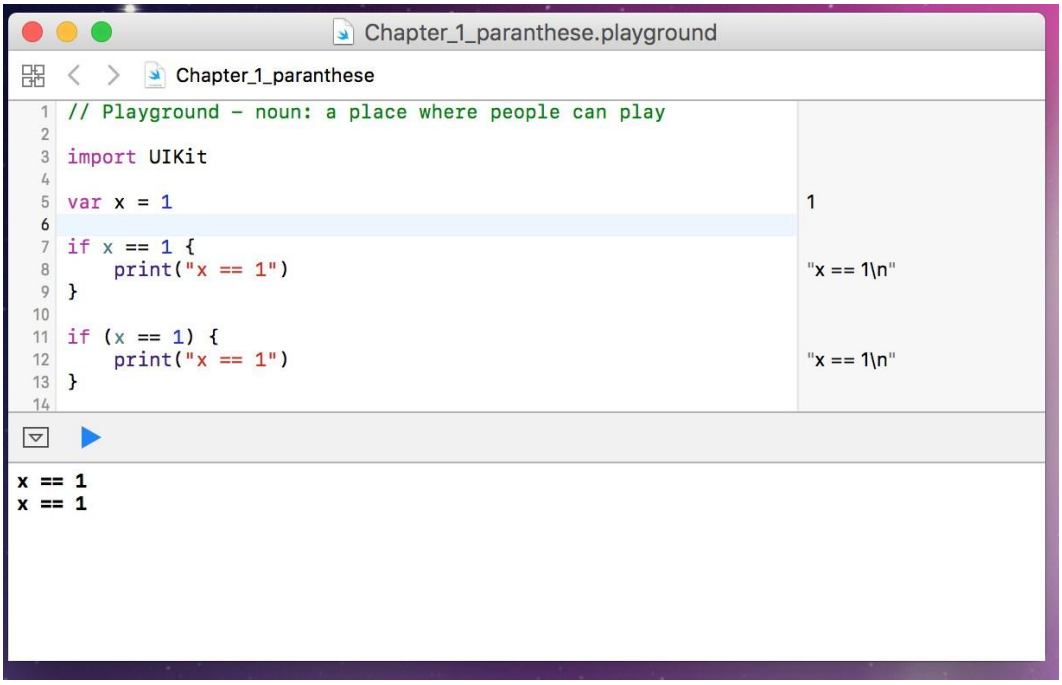

Chapter_1_displaying_graph.playground
Chapter_1_displaying_graph
// Playground - noun: a place where people can
Related Items
import UIKit
var j = 1
for i in 1...5 {
j*=i
}
1
(5 times)

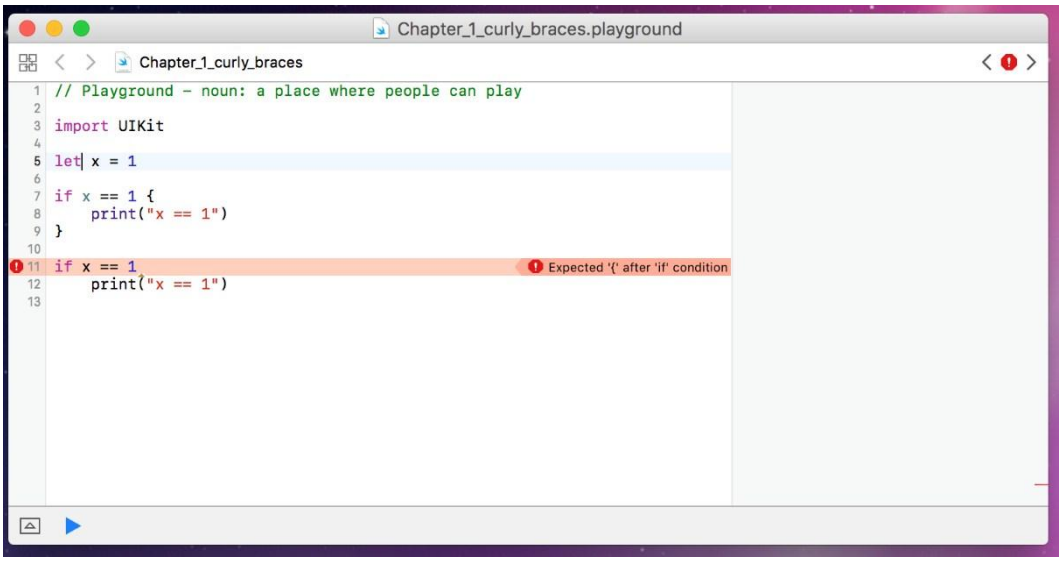

Chapter_1_displaying_graph.playground
Chapter_1_displaying_graph
1 // Playground - noun: a place where people can play
2
3 import UIKit
4
5 var j = 1
6 for i in 1...5 {
7 j*=i
8 }
9
1
(5 times)

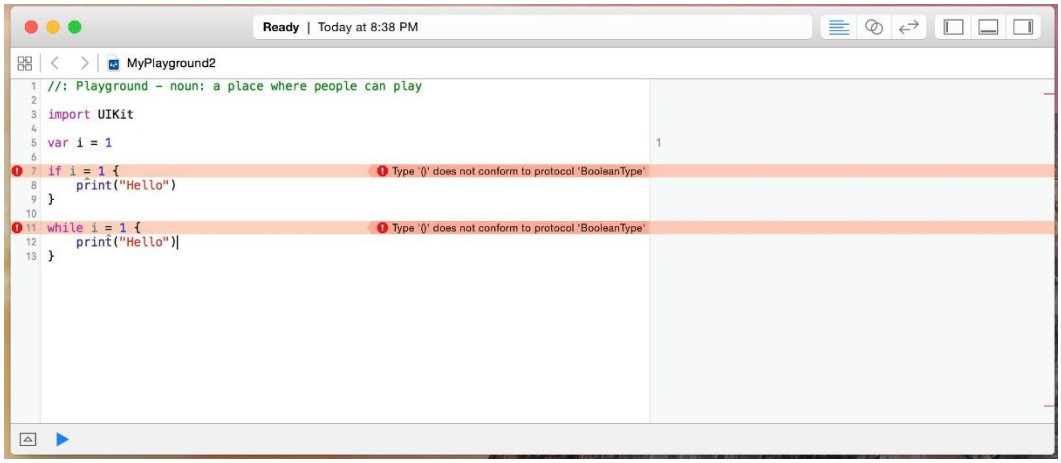

Ready | Today at 4:59 PM
Chapter_1_Comment
//: Playground - noun: a place where people can play
import UIKit
var str = "Hello, playground"
"Hello, playground"
/// The myAdd function will take two integers, add
/// them together and return the sum
/// - parameter first: The first integer to add
/// - parameter second: The second integer to add
/// - returns: The sum of the two integers
/// - throws: Our error
func myAdd(first: Int, second: Int) throws -> Int {
//add the two integers together
let sum: Int = first+second
15
return sum
15
try myAdd(first: 5, second: 10)
15

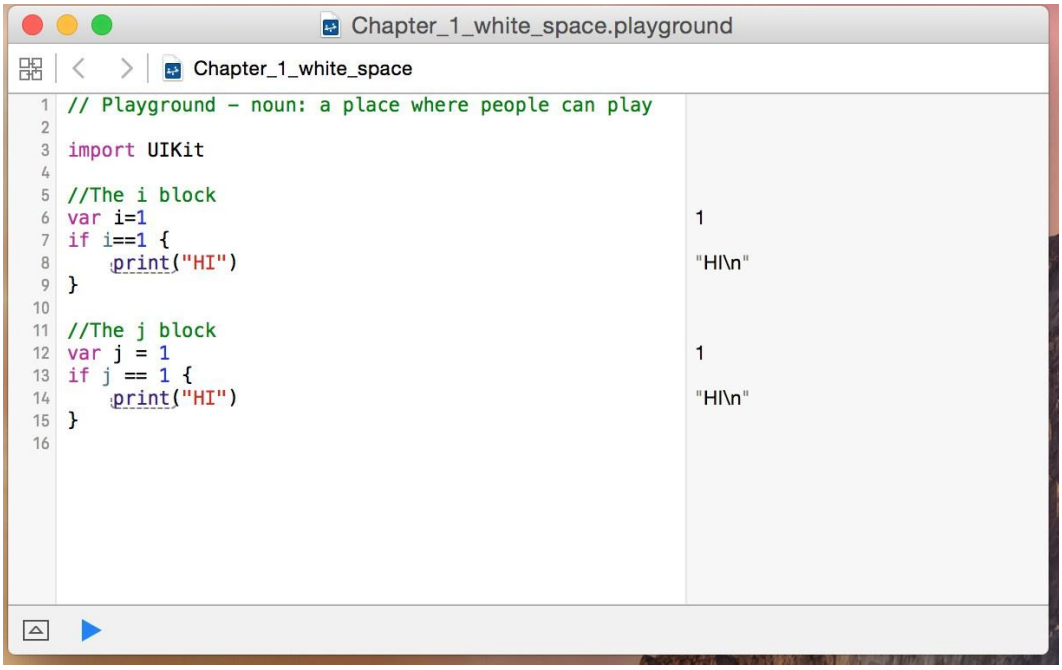

6
7 ///
8 /// The myAdd function will take two integers, add
9 /// them together and return the sum
10 ///
11 /// - parameter first: The first integer to add
12 /// - parameter second: The second integer to add
13 /// - returns: The sum of the two integers
14 /// - throws: Our error
15 ///
16
17 func myAdd(first: Int, second: Int) throws -> Int {
18 //add the two integers together
Declaration
func myAdd(first: Int, second: Int) throws -> Int
Description
The myAdd function will take two integers, add them together and return the sum
Parameters
first
The first integer to add
second
The second integer to add
Throws
Our error
Returns
The sum of the two integers
Declared In
Chapter_1_Comment.playground


Chapter_1_Semicolons.playground
Chapter_1_Semicolons
1 // Playground - noun: a place where people can
play
2
3 import UIKit
4
5 print("Hello from Swift")
6 print("Hello from Swift");
7
"Hello from Swift\n"
"Hello from Swift\n"
Hello from Swift
Hello from Swift

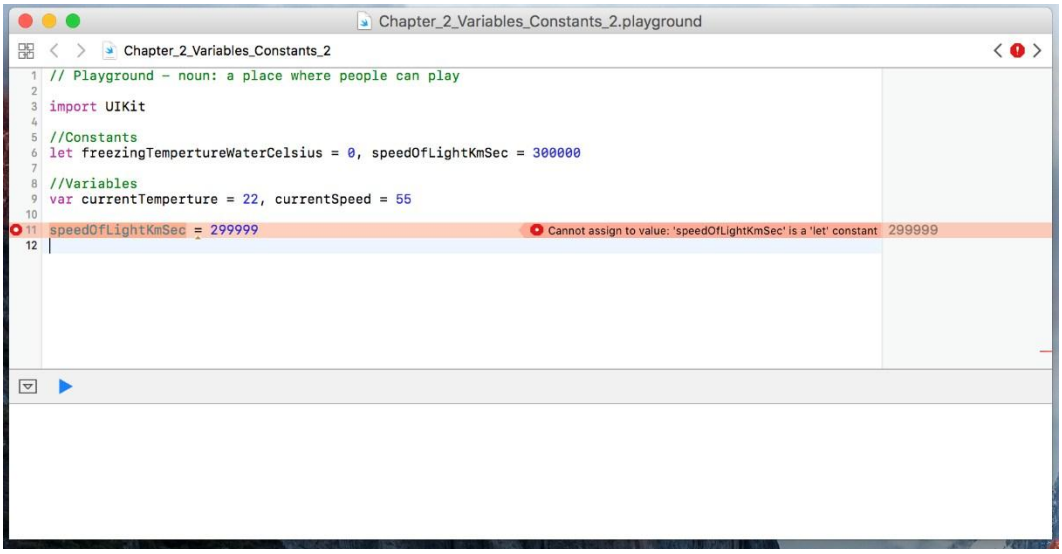

Chapter_1_paranthese.playground
Chapter_1_paranthese
// Playground - noun: a place where people can play
import UIKit
var x = 1
1
if x == 1 {
print("x == 1")
"x == 1\n"
}
if (x == 1) {
print("x == 1")
"x == 1\n"
}
x == 1
x == 1

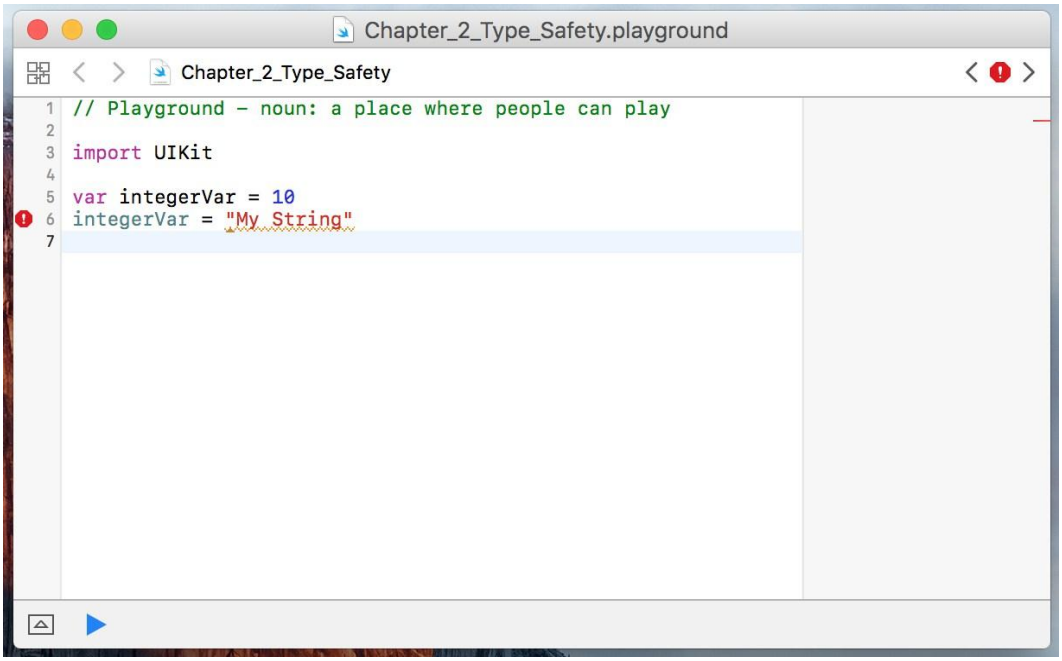

Chapter_1_curly_braces.playground
Chapter_1_curly_braces
// Playground - noun: a place where people can play
import UIKit
let x = 1
if x == 1 {
print("x == 1")
}
if x == 1
Expected '{' after 'if' condition
print("x == 1")

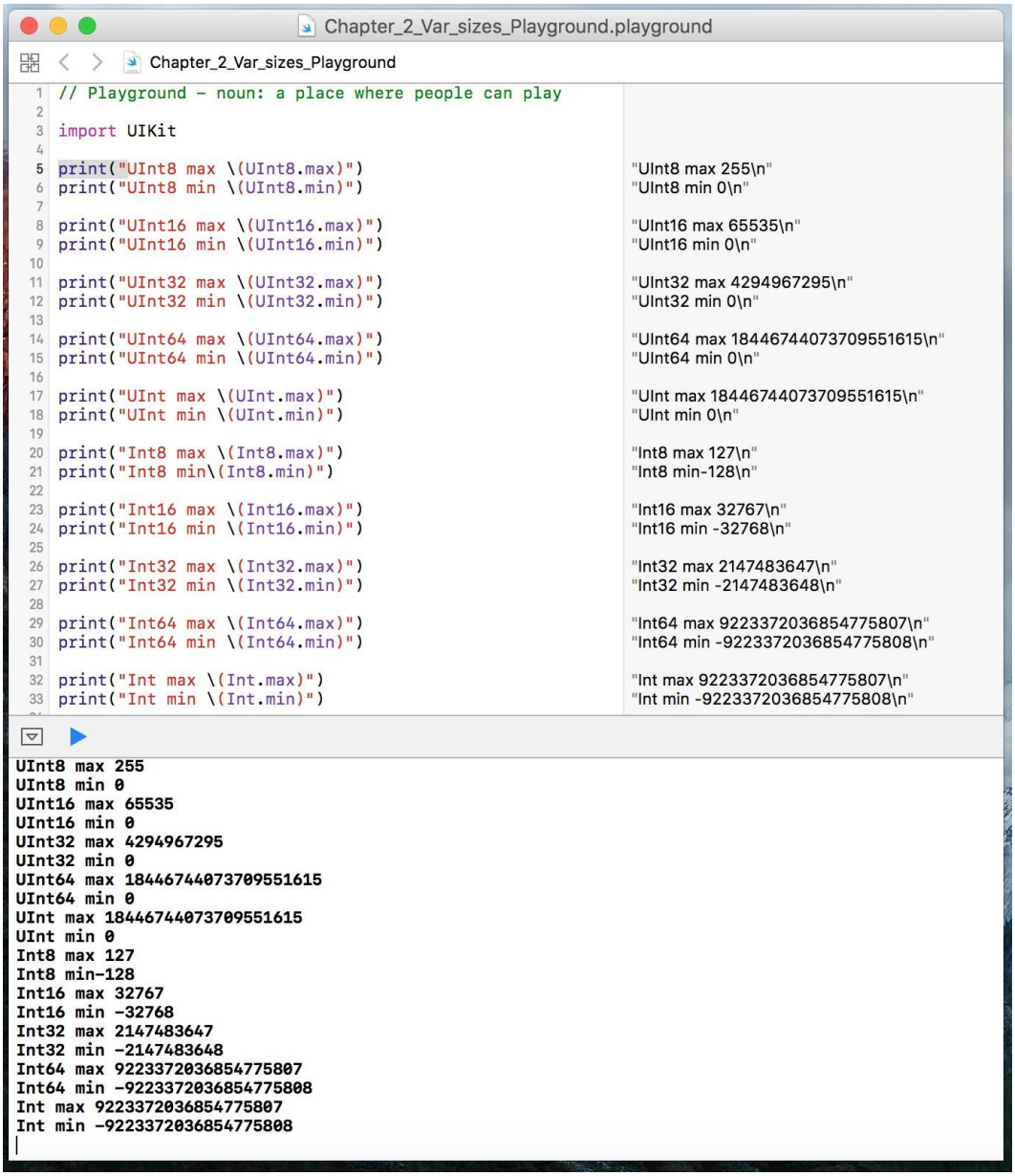

Ready | Today at 8:38 PM
MyPlayground2
//: Playground - noun: a place where people can play
import UIKit
var i = 1
1
if i = 1 {
Type '()' does not conform to protocol 'BooleanType'
print("Hello")
}
while i = 1 {
Type '()' does not conform to protocol 'BooleanType'
print("Hello")
}

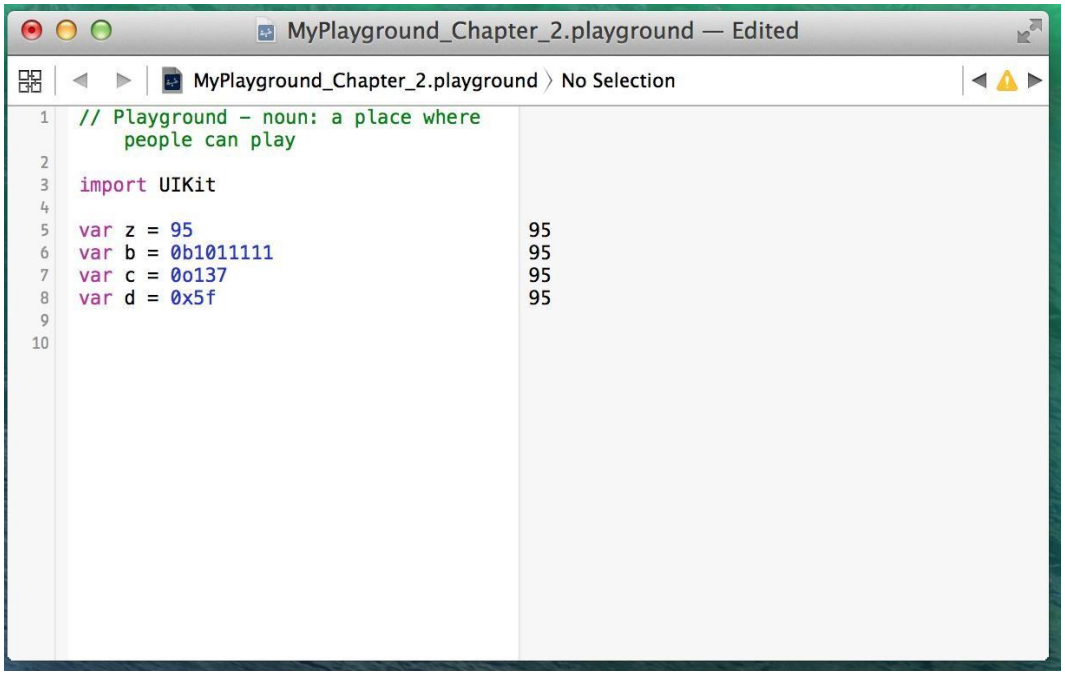

Chapter_1_white_space.playground
Chapter_1_white_space
// Playground - noun: a place where people can play
import UIKit
//The i block
var i=1
1
if i==1 {
print("HI")
"HI\n"
}
//The j block
var j = 1
1
if j == 1 {
print("HI")
"HI\n"
}

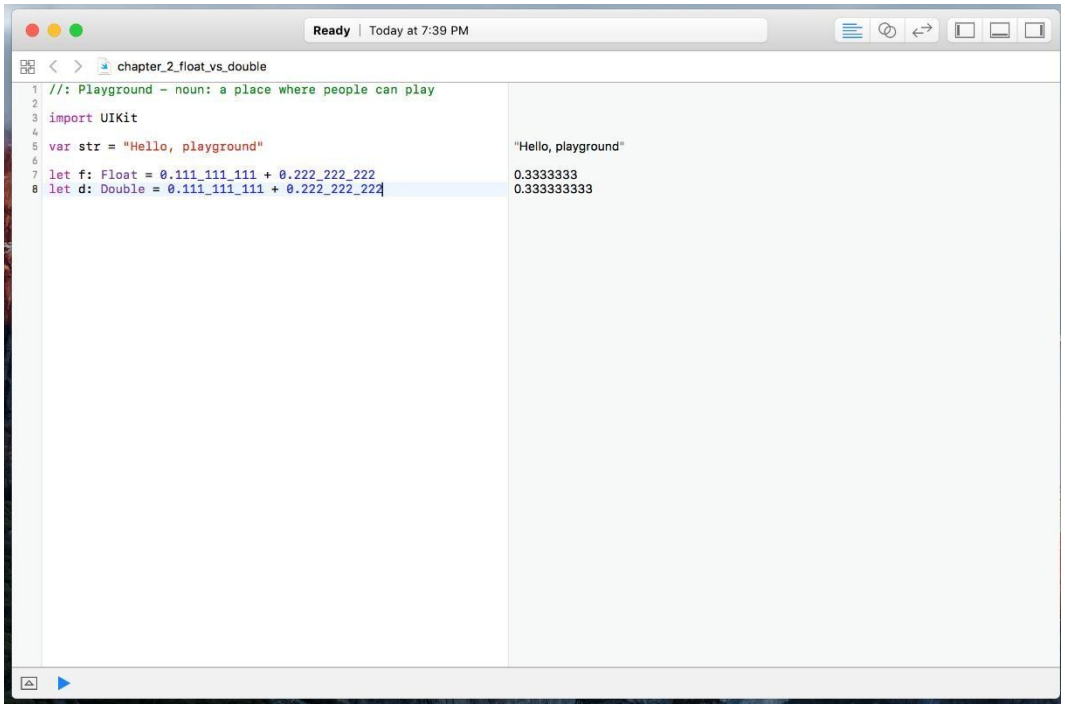
Chapter 2: Learning About Variables, Constants, Strings, and Operators

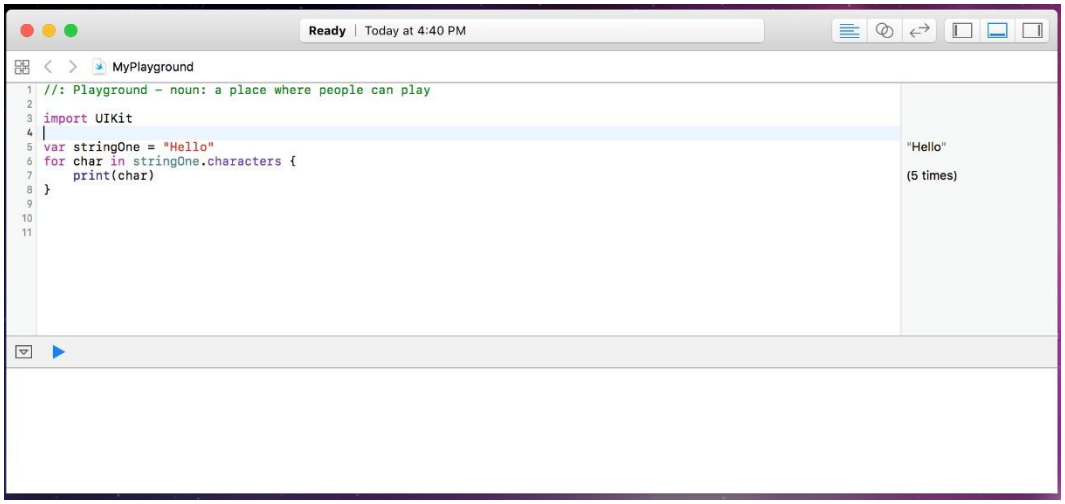


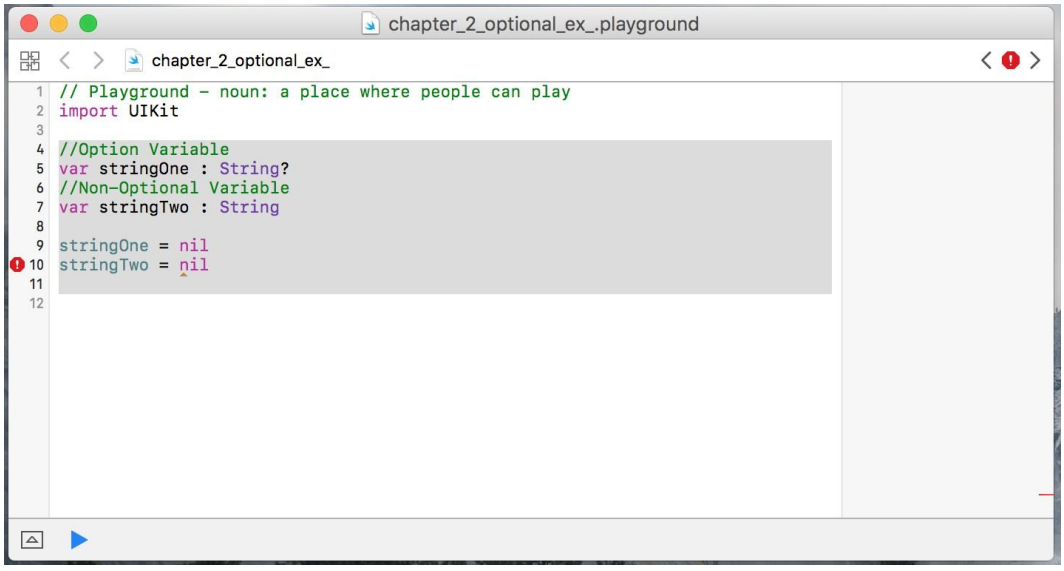

Chapter_2_Type_Safety.playground
Chapter_2_Type_Safety
1 // Playground - noun: a place where people can play
2
3 import UIKit
4
5 var integerVar = 10
6 integerVar = "My String"
7

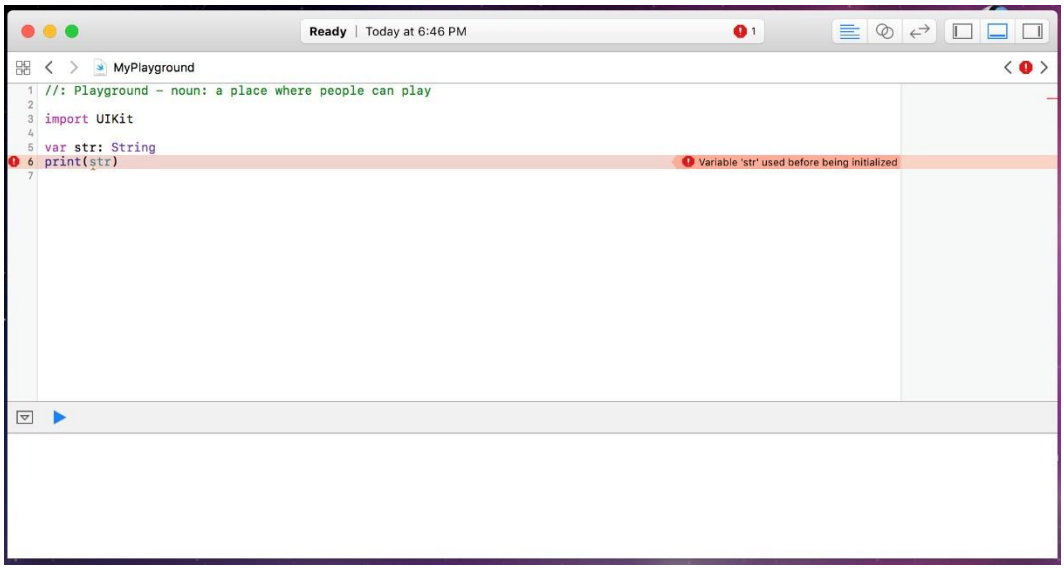

Chapter_2_Var_sizes_Playground.playground
Chapter_2_Var_sizes_Playground
// Playground - noun: a place where people can play
import UIKit
print("UInt8 max \(UInt8.max)")
print("UInt8 min \(UInt8.min)")
print("UInt16 max \(UInt16.max)")
print("UInt16 min \(UInt16.min)")
print("UInt32 max \(UInt32.max)")
print("UInt32 min \(UInt32.min)")
print("UInt64 max \(UInt64.max)")
print("UInt64 min \(UInt64.min)")
print("UInt max \(UInt.max)")
print("UInt min \(UInt.min)")
print("Int8 max \(Int8.max)")
print("Int8 min\(Int8.min)")
print("Int16 max \(Int16.max)")
print("Int16 min \(Int16.min)")
print("Int32 max \(Int32.max)")
print("Int32 min \(Int32.min)")
print("Int64 max \(Int64.max)")
print("Int64 min \(Int64.min)")
print("Int max \(Int.max)")
print("Int min \(Int.min)")
"UInt8 max 255\n"
"UInt8 min 0\n"
"UInt16 max 65535\n"
"UInt16 min 0\n"
"UInt32 max 4294967295\n"
"UInt32 min 0\n"
"UInt64 max 18446744073709551615\n"
"UInt64 min 0\n"
"UInt max 18446744073709551615\n"
"UInt min 0\n"
"Int8 max 127\n"
"Int8 min-128\n"
"Int16 max 32767\n"
"Int16 min -32768\n"
"Int32 max 2147483647\n"
"Int32 min -2147483648\n"
"Int64 max 9223372036854775807\n"
"Int64 min -9223372036854775808\n"
"Int max 9223372036854775807\n"
"Int min -9223372036854775808\n"
UInt8 max 255
UInt8 min 0
UInt16 max 65535
UInt16 min 0
UInt32 max 4294967295
UInt32 min 0
UInt64 max 18446744073709551615
UInt64 min 0
UInt max 18446744073709551615
UInt min 0
Int8 max 127
Int8 min-128
Int16 max 32767
Int16 min -32768
Int32 max 2147483647
Int32 min -2147483648
Int64 max 9223372036854775807
Int64 min -9223372036854775808
Int max 9223372036854775807
Int min -9223372036854775808

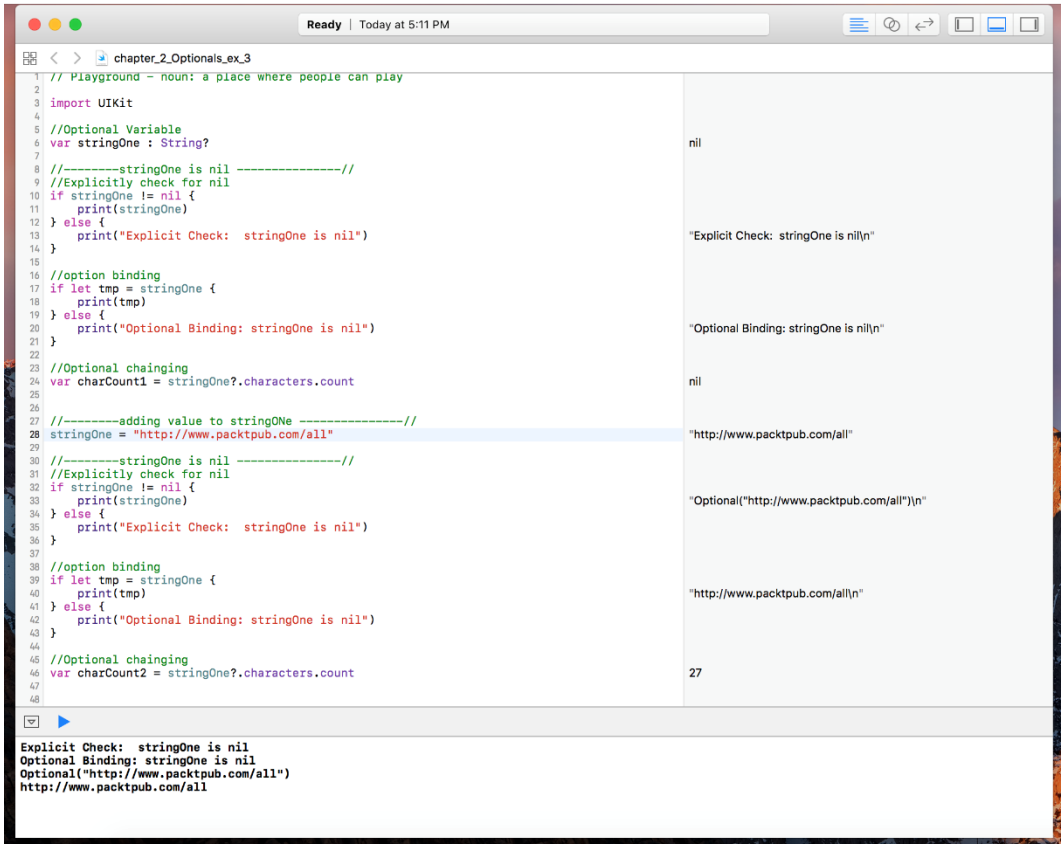

MyPlayground_Chapter_2.playground — Edited
MyPlayground_Chapter_2.playground
No Selection
// Playground - noun: a place where people can play
import UIKit
var z = 95
var b = 0b1011111
var c = 0o137
var d = 0x5f
95
95
95
95


Ready | Today at 7:39 PM
chapter_2_float_vs_double
//: Playground - noun: a place where people can play
import UIKit
var str = "Hello, playground"
let f: Float = 0.111_111_111 + 0.222_222_222
let d: Double = 0.111_111_111 + 0.222_222_222
"Hello, playground"
0.3333333
0.333333333

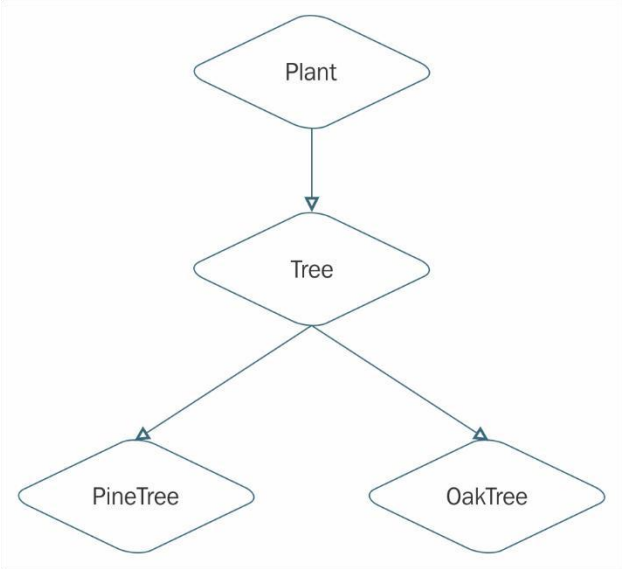

Ready | Today at 4:40 PM
MyPlayground
//: Playground - noun: a place where people can play
import UIKit
var stringOne = "Hello"
for char in stringOne.characters {
print(char)
}
"Hello"
(5 times)


chapter_2_optional_ex_.playground
chapter_2_optional_ex_
// Playground - noun: a place where people can play
import UIKit
//Option Variable
var stringOne : String?
//Non-Optional Variable
var stringTwo : String
stringOne = nil
stringTwo = nil

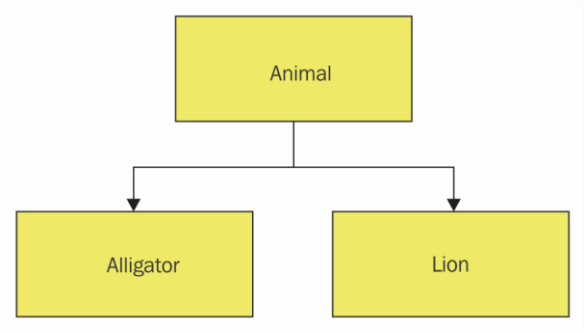

Ready | Today at 6:46 PM
MyPlayground
//: Playground - noun: a place where people can play
import UIKit
var str: String
print(str)
Variable 'str' used before being initialized

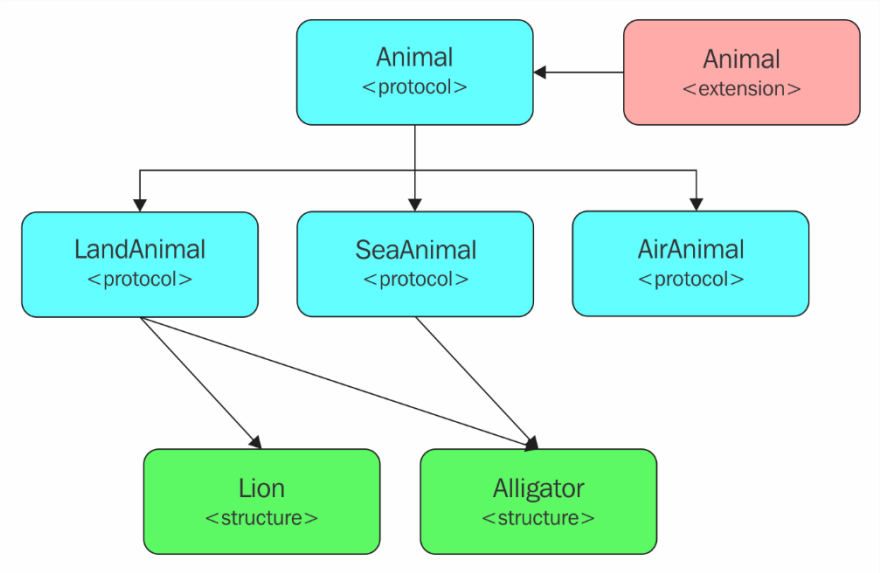

Ready | Today at 5:11 PM
chapter_2_Optionals_ex_3
// Playground - noun: a place where people can play
import UIKit
//Optional Variable
var stringOne : String?
nil
//--------stringOne is nil ---------------//
//Explicitly check for nil
if stringOne != nil {
print(stringOne)
} else {
print("Explicit Check: stringOne is nil")
"Explicit Check: stringOne is nil\n"
}
//option binding
if let tmp = stringOne {
print(tmp)
} else {
print("Optional Binding: stringOne is nil")
"Optional Binding: stringOne is nil\n"
}
//Optional chainging
var charCount1 = stringOne?.characters.count
nil
//--------adding value to stringONe ---------------//
stringOne = "http://www.packtpub.com/all"
"http://www.packtpub.com/all"
//--------stringOne is nil ---------------//
//Explicitly check for nil
if stringOne != nil {
print(stringOne)
"Optional("http://www.packtpub.com/all")\n"
} else {
print("Explicit Check: stringOne is nil")
}
//option binding
if let tmp = stringOne {
print(tmp)
"http://www.packtpub.com/all\n"
} else {
print("Optional Binding: stringOne is nil")
}
//Optional chainging
var charCount2 = stringOne?.characters.count
27
Explicit Check: stringOne is nil
Optional Binding: stringOne is nil
Optional("http://www.packtpub.com/all")
http://www.packtpub.com/all

Chapter 5: Classes and Structures

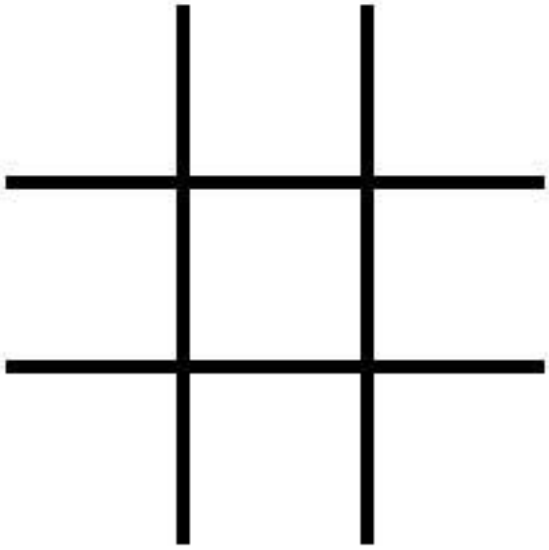


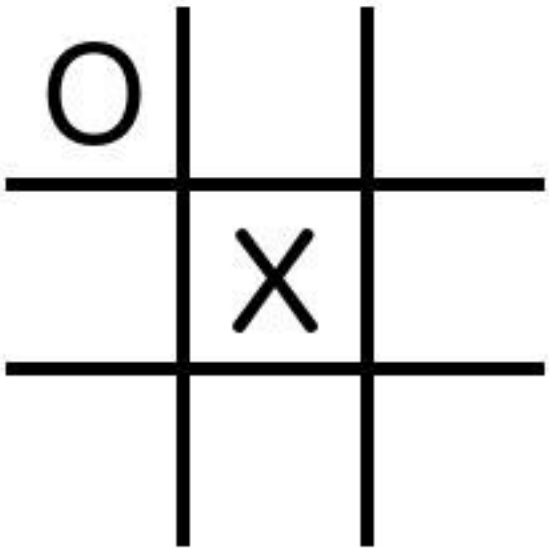
Chapter 7: Protocol-Oriented Design



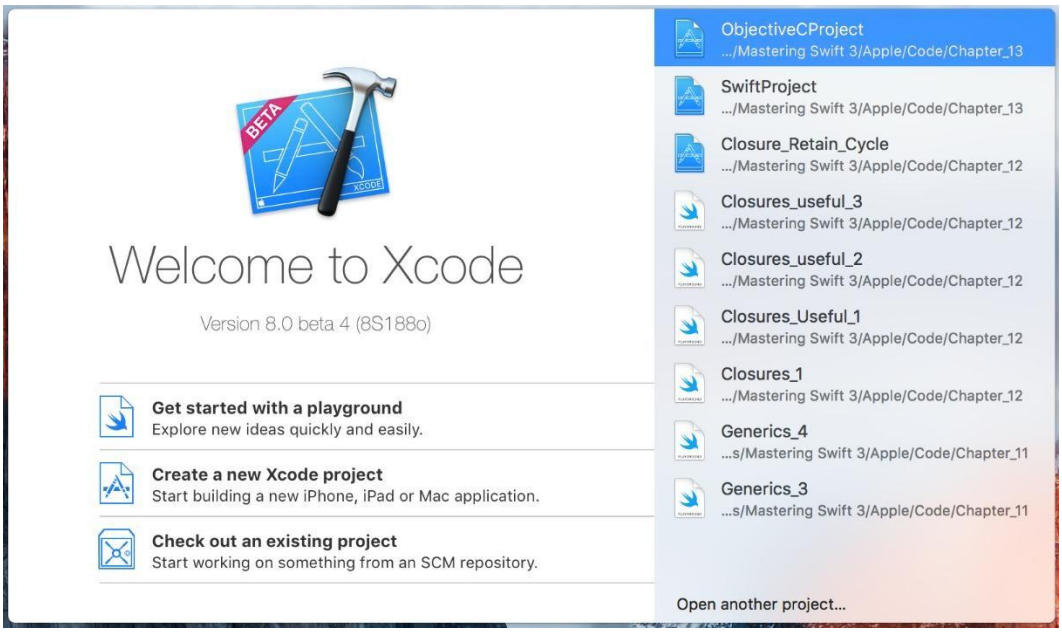


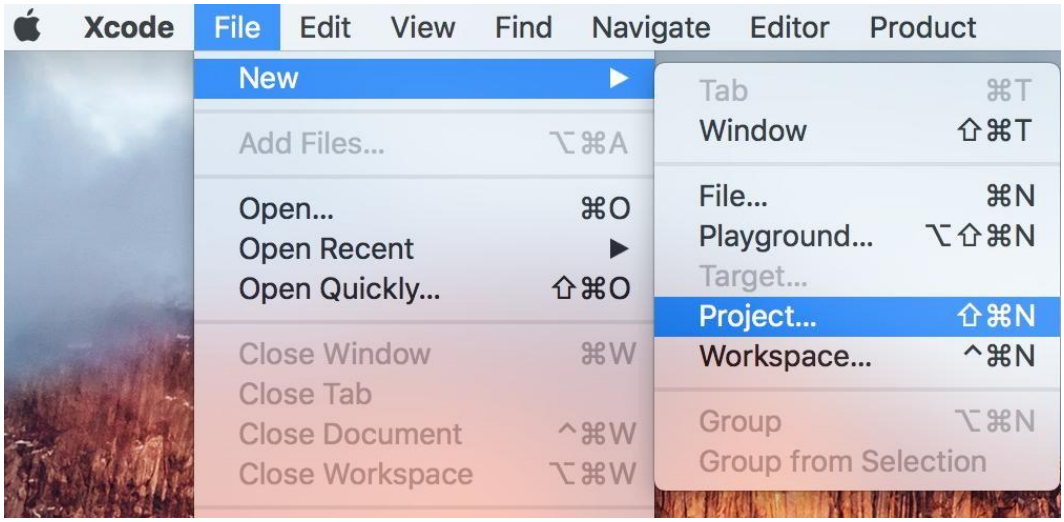
Chapter 9: Custom Subscripting

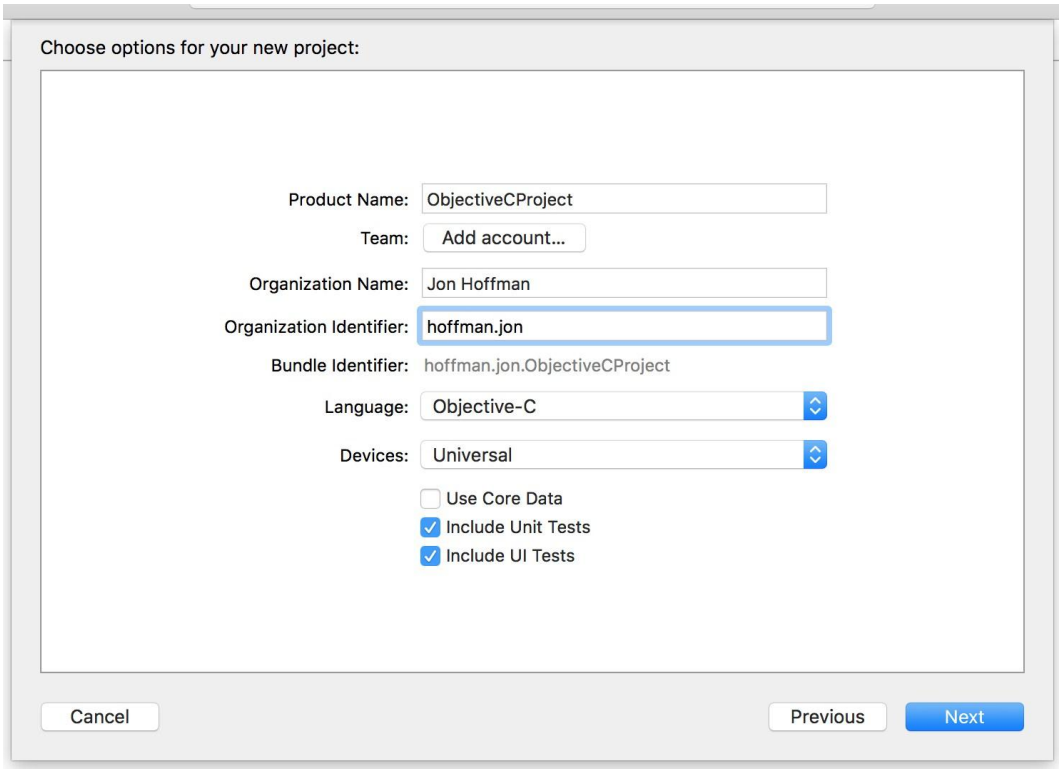


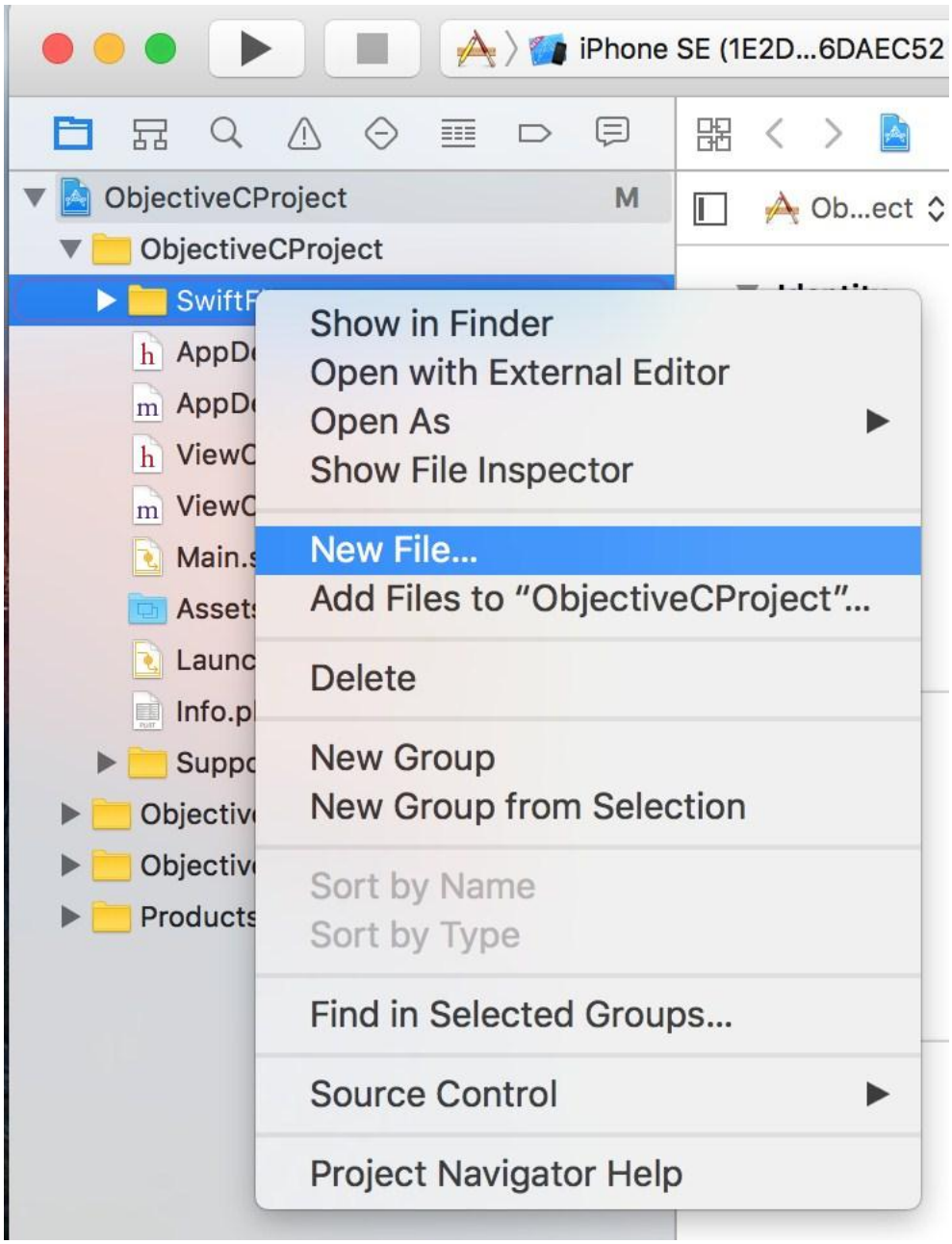

O
X

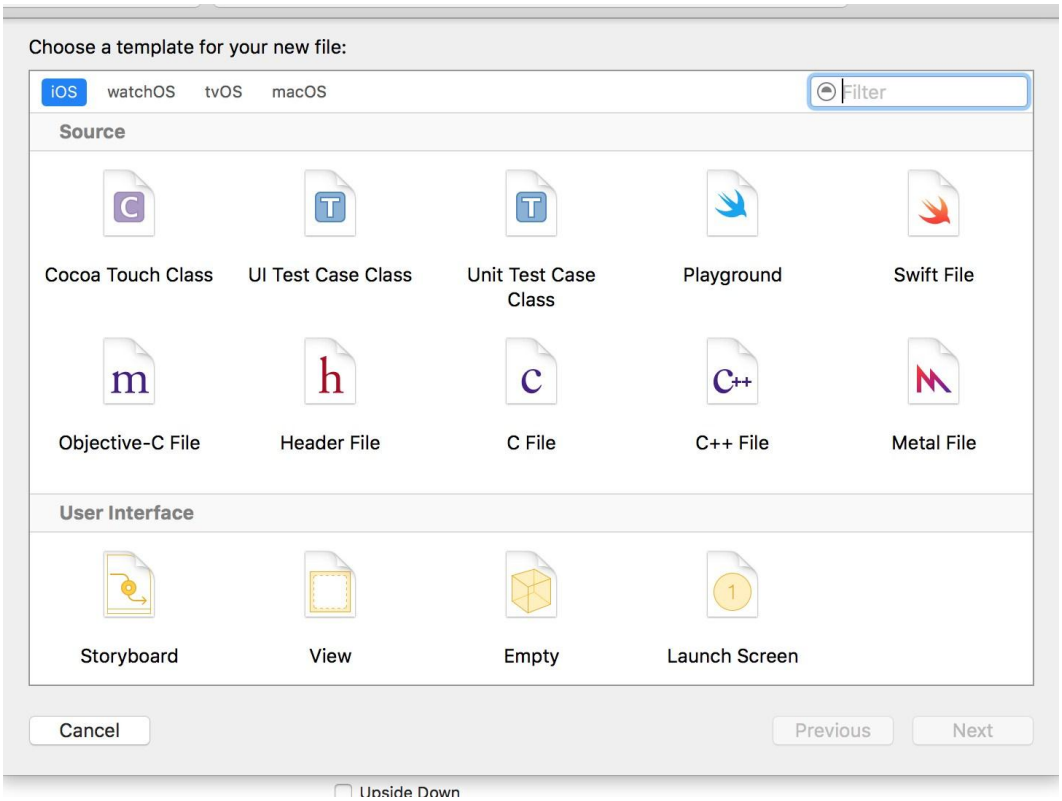
Chapter 13: Using Mix and Match

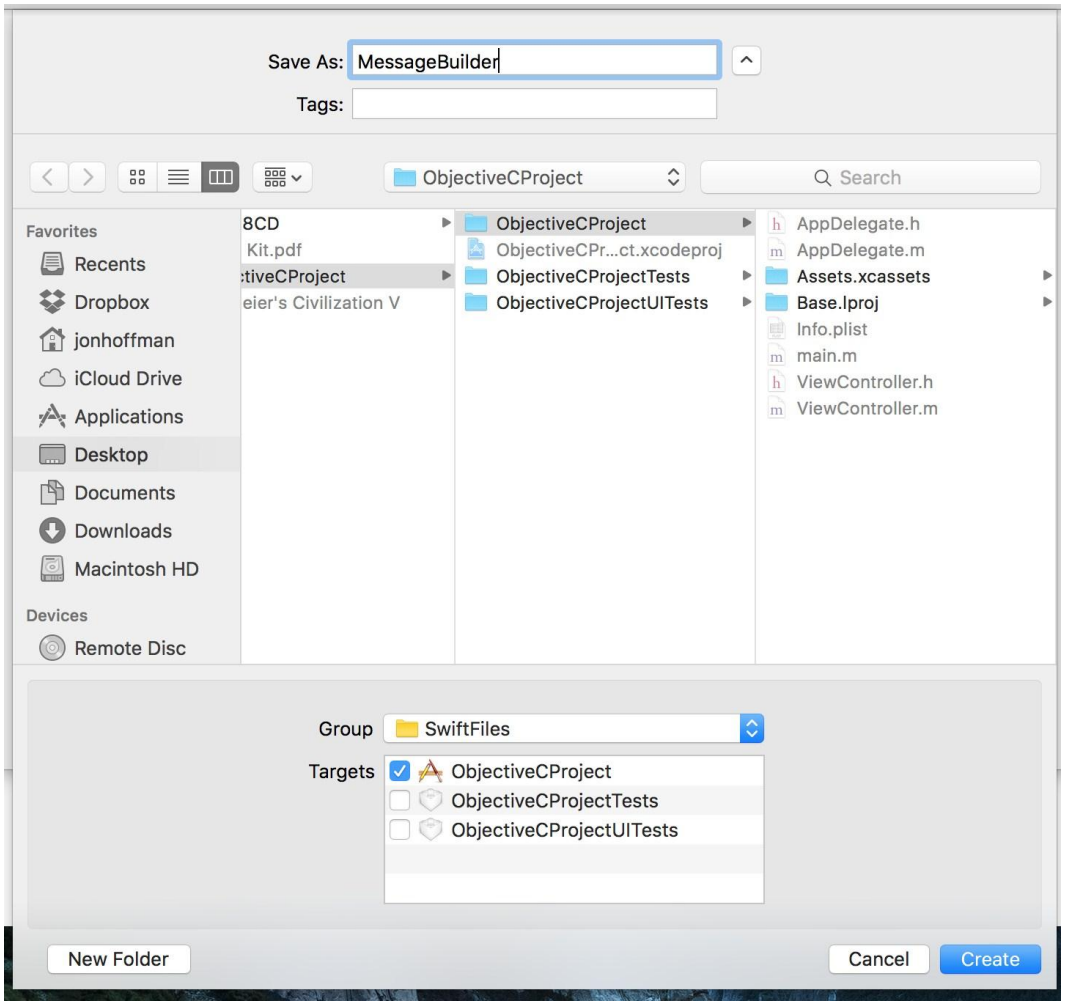


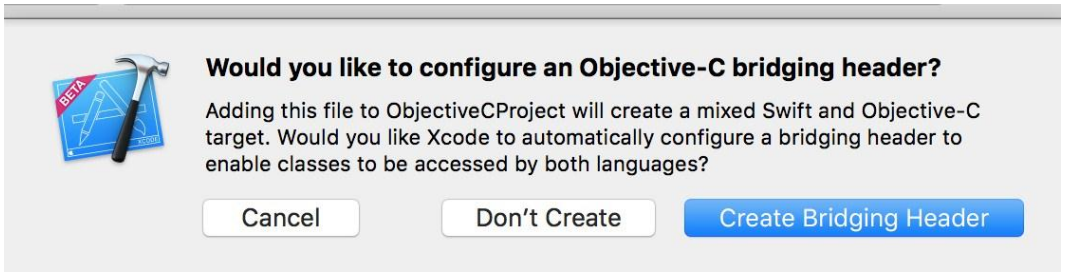

Xcode
File
Edit
View
Find
Navigate
Editor
Product
New
Add Files... ⌥⌘A
Open... ⌘O
Open Recent
Open Quickly... ⇧⌘O
Close Window ⌘W
Close Tab
Close Document ^⌘W
Close Workspace ⌥⌘W
Tab ⌘T
Window ⇧⌘T
File... ⌘N
Playground... ⌥⇧⌘N
Target...
Project... ⇧⌘N
Workspace... ^⌘N
Group ⌥⌘N
Group from Selection

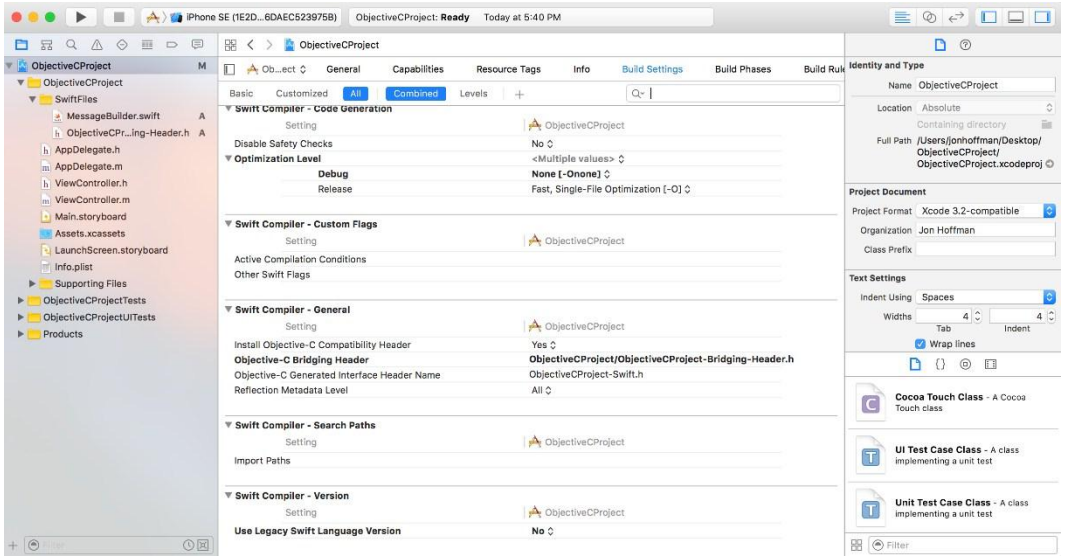

Choose options for your new project:
Product Name: ObjectiveCProject
Team: Add account...
Organization Name: Jon Hoffman
Organization Identifier: hoffman.jon
Bundle Identifier: hoffman.jon.ObjectiveCProject
Language: Objective-C
Devices: Universal
Use Core Data
Include Unit Tests
Include UI Tests
Cancel
Previous
Next

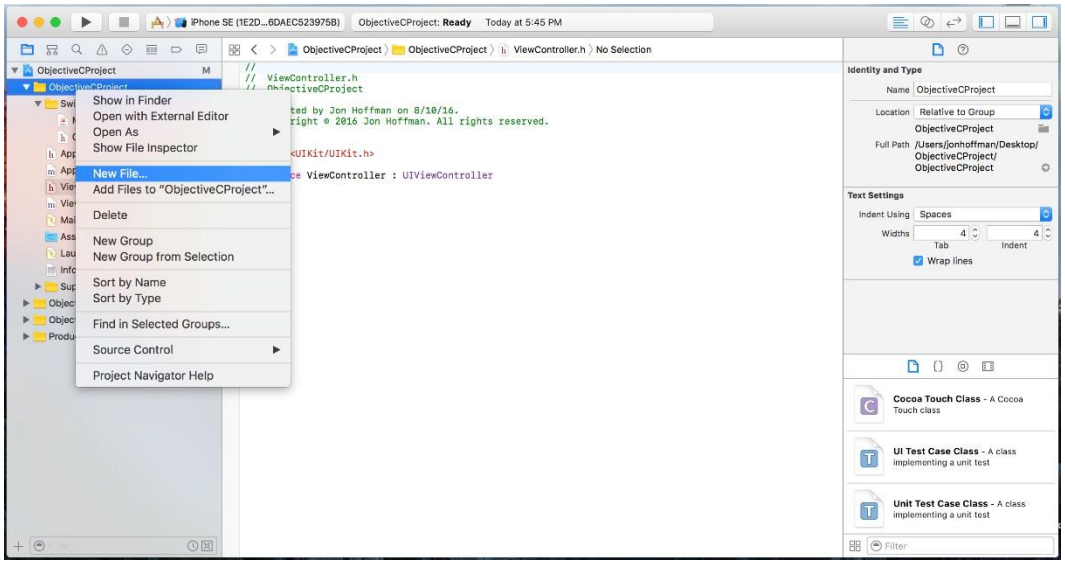

iPhone SE (1E2D...6DAEC52
ObjectiveCProject
M
ObjectiveCProject
Ob...ect
Show in Finder
Open with External Editor
Open As
Show File Inspector
New File...
Add Files to "ObjectiveCProject"...
Delete
New Group
New Group from Selection
Sort by Name
Sort by Type
Find in Selected Groups...
Source Control
Project Navigator Help

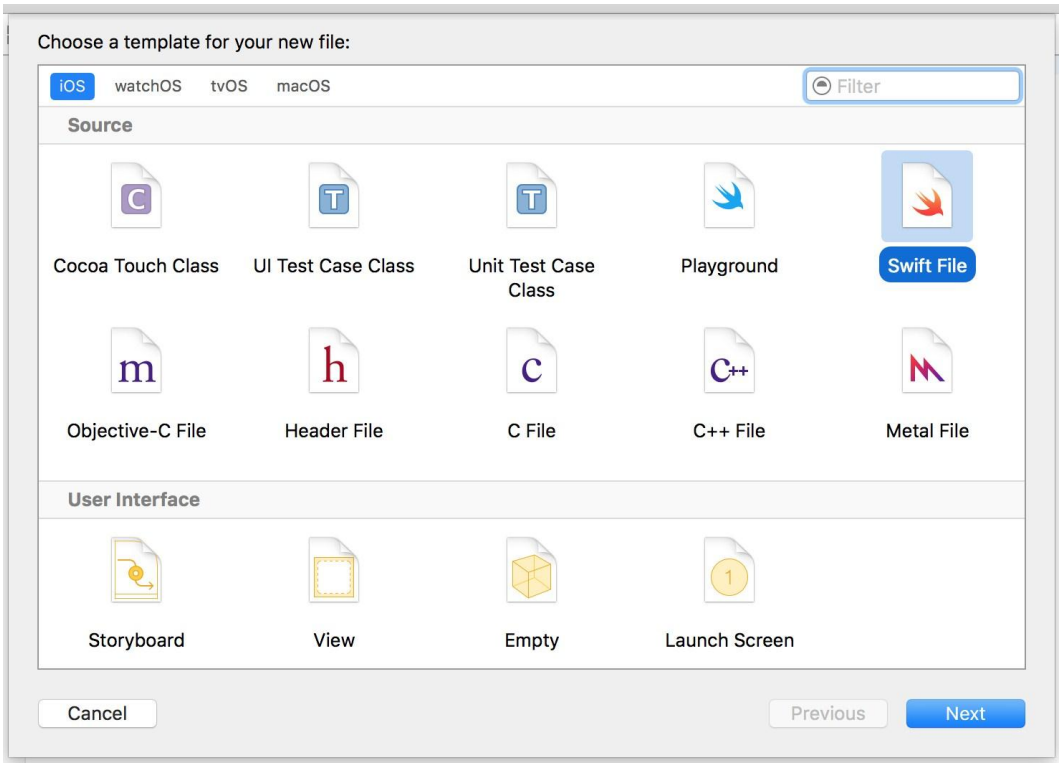

Choose a template for your new file:
iOS
watchOS
tvOS
macOS
Filter
Source
Cocoa Touch Class
UI Test Case Class
Unit Test Case Class
Playground
Swift File
Objective-C File
Header File
C File
C++ File
Metal File
User Interface
Storyboard
View
Empty
Launch Screen
Cancel
Previous
Next
Upside Down

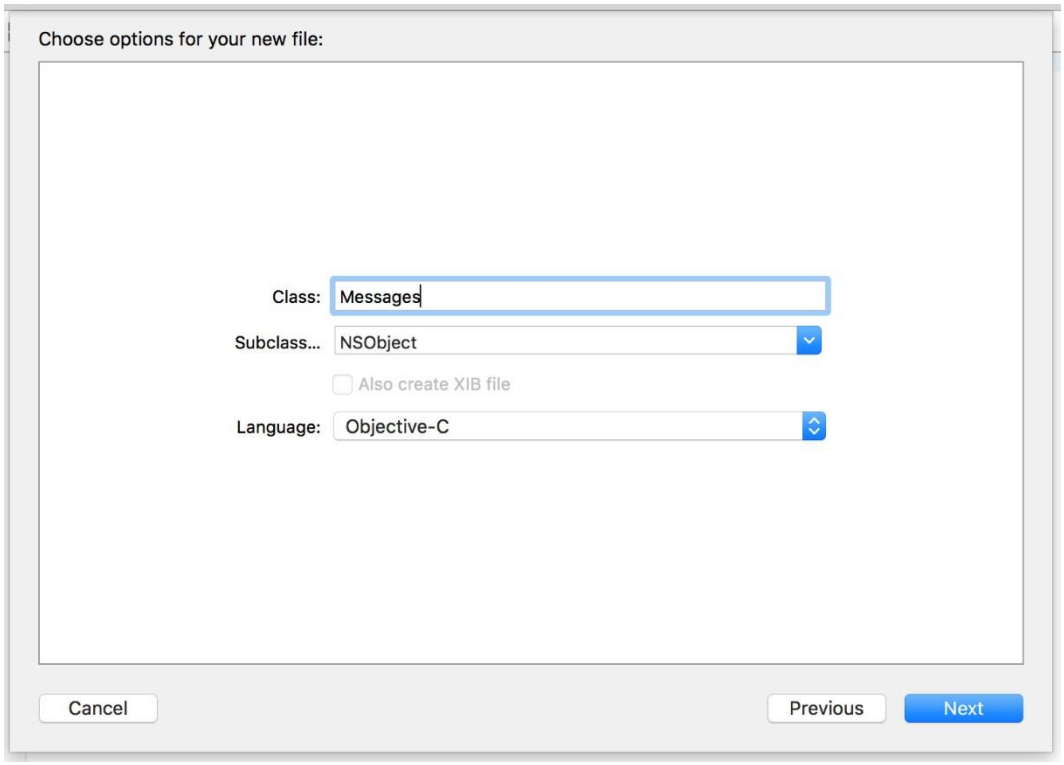
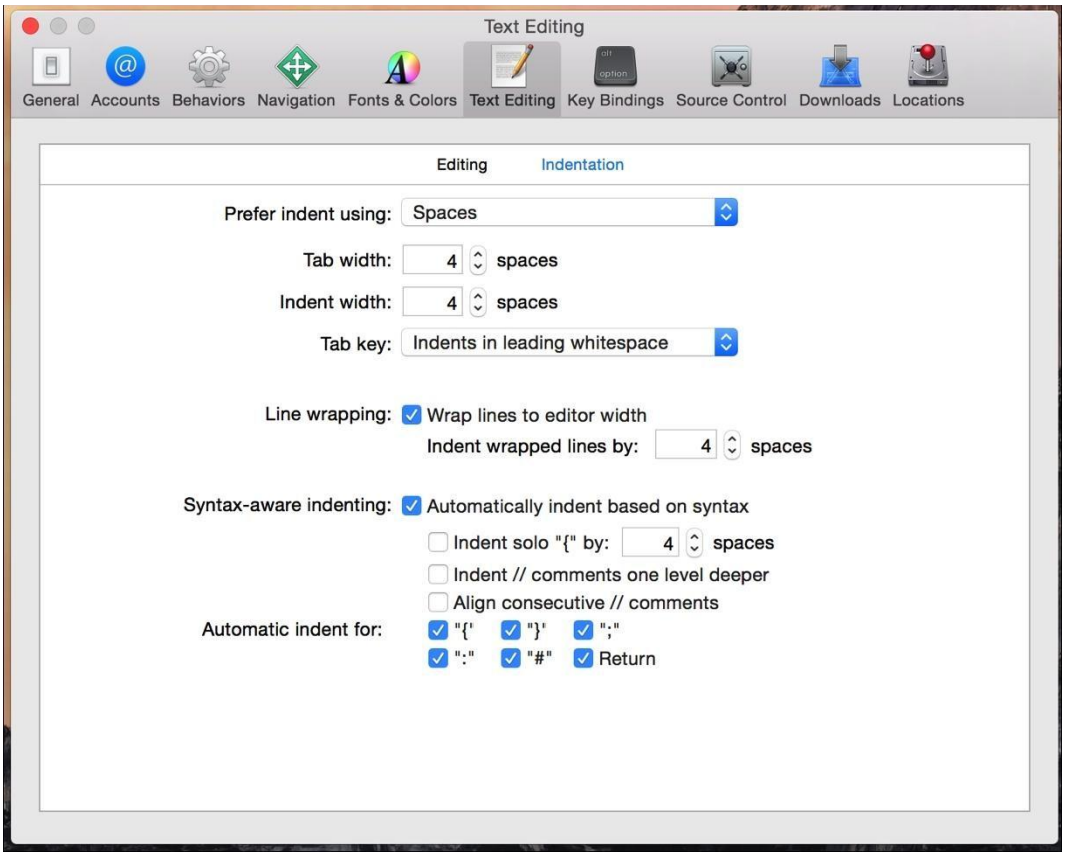

Save As: MessageBuilder
Tags:
ObjectiveCProject
Search
Favorites
Recents
Dropbox
jonhoffman
iCloud Drive
Applications
Desktop
Documents
Downloads
Macintosh HD
Devices
Remote Disc
8CD
Kit.pdf
tiveCProject
eier's Civilization V
ObjectiveCProject
ObjectiveCPr...ct.xcodeproj
ObjectiveCProjectTests
ObjectiveCProjectUITests
AppDelegate.h
AppDelegate.m
Assets.xcassets
Base.lproj
Info.plist
main.m
ViewController.h
ViewController.m
Group
SwiftFiles
Targets
ObjectiveCProject
ObjectiveCProjectTests
ObjectiveCProjectUITests
New Folder
Cancel
Create


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?
Cancel
Don't Create
Create Bridging Header


iPhone SE (1E2D...6DAEC523975B)
ObjectiveCProject: Ready
Today at 5:40 PM
ObjectiveCProject
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Basic
Customized
All
Combined
Levels
Swift Compiler - Code Generation
Setting
Disable Safety Checks
Optimization Level
Debug
Release
Swift Compiler - Custom Flags
Active Compilation Conditions
Other Swift Flags
Swift Compiler - General
Install Objective-C Compatibility Header
Objective-C Bridging Header
Objective-C Generated Interface Header Name
Reflection Metadata Level
Swift Compiler - Search Paths
Import Paths
Swift Compiler - Version
Use Legacy Swift Language Version
Identity and Type
Name
Location
Full Path
Project Document
Project Format
Organization
Class Prefix
Text Settings
Indent Using
Widths
Tab
Indent
Wrap lines
Cocoa Touch Class - A Cocoa Touch class
UI Test Case Class - A class implementing a unit test
Unit Test Case Class - A class implementing a unit test
Filter


iPhone SE (1E2D...6DAEC523975B)
ObjectiveCProject: Ready
Today at 5:45 PM
Show in Finder
Open with External Editor
Open As
Show File Inspector
New File...
Add Files to "ObjectiveCProject"...
Delete
New Group
New Group from Selection
Sort by Name
Sort by Type
Find in Selected Groups...
Source Control
Project Navigator Help
Identity and Type
Location
Relative to Group
Text Settings
Cocoa Touch Class - A Cocoa Touch class
UI Test Case Class - A class implementing a unit test
Unit Test Case Class - A class implementing a unit test
Filter


Choose a template for your new file:
iOS
watchOS
tvOS
macOS
Filter
Source
Cocoa Touch Class
UI Test Case Class
Unit Test Case Class
Playground
Swift File
Objective-C File
Header File
C File
C++ File
Metal File
User Interface
Storyboard
View
Empty
Launch Screen
Cancel
Previous
Next


Choose options for your new file:
Class:
Messages
Subclass...
NSObject
Also create XIB file
Language:
Objective-C
Cancel
Previous
Next

Chapter 15: Swift Formatting and Style Guide