Chapter 1: Taking the First Steps with Swift
Choose options for your new file:

Name: MasteringSwift_Chapter_3

Platform: iOS

---

```
// Playground - noun: a place where people can play

import UIKit

var str = "Hello, playground"
```

"Hello, playground"
import UIKit

var str = "Hello, playground"
// Playground - noun: a place where people can play

import UIKit

define image = UIImage(named: "swift.png")

text 256 h 256
import UIKit

var j = 1
for i in 1...5 {
    j = j * i
}

func myAdd(first: Int, second: Int) -> Int {
    var results = first + second
    return results
}
import UIKit

/**
 * myAdd will take two integers, add them together and return the sum.
 * :param: first The first integer to add
 * :param: second The second integer to add
 * :returns: The sum of the two integers
 */
func myAdd(first: Int, second: Int) -> Int {
    var results = first + second
    return results
}

myAdd(5, 6)

// Playground - noun: a place where people can play
import UIKit

print("Hello from Swift")
print("Hello from Swift");
import UIKit

var x = 1

if x == 1 {
    print("x == 1")
}

if (x == 1) {
    print("x == 1")
}

// Single conditional statement, no parentheses
if x == 1 {
    print("X == 1")
}

// Multiple conditional statements, parentheses
if (x == 1) && (y == 1) && (z == 1) {
    print("All vars == 1")
}

// Multiple conditional statements, no Parentheses
if x == 1 && y == 1 && z == 1 {
    print("All vars == 1")
}
import UIKit

let x = 1

if x == 1 {
    print("x == 1")
}

if x == 1
    print("x == 1")

//: Playground — noun: a place where people can play

import UIKit

var i = 1

if i == 1 {
    print("Hello")
}

while i == 1 {
    print("Hello")
}
Chapter 2: Learning Variables, Constants, Strings, and Operators
// Playground - noun: a place where people can play

import UIKit

var integerVar = 10

integerVar = "My String" // Type 'Int' does not conform to protocol 'StringLiteralConvertible'

// Playground - noun: a place where people can play

import UIKit

var a = UInt8.max 255
var b = UInt8.min 0
var c = UInt16.max 65535
var d = UInt16.min 0
var e = UInt32.max 4294967295
var f = UInt32.min 0
var g = UInt64.max 18446744073709551615
var h = UInt64.min 0
var j = UInt.max 18446744073709551615
var k = UInt.min 0
var l = Int8.max 127
var m = Int8.min -128
var n = Int16.max 32767
var o = Int16.min -32768
var p = Int32.max 2147483647
var q = Int32.min -2147483648
var r = Int64.max 9223372036854775807
var s = Int64.min -9223372036854775808
var t = Int.max 9223372036854775807
var u = Int.min -9223372036854775808
// Playground — noun: a place where people can play

import UIKit

var z = 95
var b = 0b1011111
var c = 0o137
var d = 0xf

let f: Float = 0.111_111_111 + 0.222_222_222
let d: Double = 0.111_111_111 + 0.222_222_222
```swift
// Playground - noun: a place where people can play

import UIKit

var str = "Hello"
for char in str.characters {
    print(char)
}

"Hello" (5 times)

Hello

Playground execution failed: /var/folders/t7/rld8zyd6vzcwdw6z2fhpbdv00000gn/T/.lldb/1316/playground0.swift:9:13: error: cannot assign a value of type 'nil' to a value of type 'String'
stringTwo = nil
```
import UIKit

// Playground - noun: a place where people can play

var str: String

print(str)
Chapter 5: Classes and Structures
Chapter 9: Custom Subscripting
Chapter 13: Using Mix and Match
### Choose a template for your new file:

**iOS**
- **Source**
- **User Interface**
- **Core Data**
- **Apple Watch**
- **Resource**
- **Other**
  - **cocos2d**

**watchOS**
- **Source**
- **User Interface**
- **Core Data**
- **Resource**
- **Other**
  - **C++**

**OS X**
- **Source**
- **User Interface**
- **Core Data**

<table>
<thead>
<tr>
<th>Template</th>
<th>File Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cocoa Class</td>
<td>.swift</td>
</tr>
<tr>
<td>UI Test Case Class</td>
<td>.t</td>
</tr>
<tr>
<td>Unit Test Case Class</td>
<td>.h</td>
</tr>
<tr>
<td>Playground</td>
<td>.c</td>
</tr>
<tr>
<td>Swift File</td>
<td>.swift</td>
</tr>
<tr>
<td>Objective-C File</td>
<td>.m</td>
</tr>
<tr>
<td>Header File</td>
<td>.h</td>
</tr>
<tr>
<td>C File</td>
<td>.c</td>
</tr>
<tr>
<td>C++ File</td>
<td>.c++</td>
</tr>
<tr>
<td>Metal File</td>
<td>.metal</td>
</tr>
</tbody>
</table>

An empty Swift file.
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?

[Options: Cancel, Don't Create, Create Bridging Header]
Chapter 15: Swift Formatting and Style Guide