

## ☰ Swift 3 的第一印象

[⏮ 返回视频](#)[整数和浮点数 ⏭](#)[\(/series/swift-up-and-running\)](/series/swift-up-and-running)[https://www.boxueio.com/series/swift-up-and-running/ebook/2\)](https://www.boxueio.com/series/swift-up-and-running/ebook/2)

## 变量和常量

当我们学习一门语言的时候，无论是说还是写，也无论是外语或者编程语言，我们都会用自己已经掌握的语言中的元素去理解新语言中对应的部分。学习Swift也一样，作为一种编程语言，它有着和Objective-C，JavaScript或C#等其它编程语言共通的概念和类似的表达方式。如果你或多或少了解过一些编程语言，学习Swift将会是一个非常轻松的过程。

### Playground

Playground是Apple在2014年WWDC发布Swift的时候，和XCode一起发布的用于实时调试Swift代码的小工具。我们绝大部分讲Swift语言本身的视频，都会直接在Playground里完成。Playground会随着我们编写代码实时为我们完成编译，并自动为我们显示每一行代码的执行结果，无论是简单语句，循环，甚至是各种UI，Playground都可以完美胜任，是我们理解Swift中各种语言特性的利器。

我们打开XCode，选择Get started with a playground。

✕

[⏮ Back to series \(/series/s](#)

## Welcome to Xcode

Version 7.0.1 (7A1001)

- ⊕ 字号
- ⊖ 字号
- 🖌 默认主题
- 🖌 金色主题
- 🖌 暗色主题

**Get started with a playground**

Explore new ideas quickly and easily.

**Create a new Xcode project**

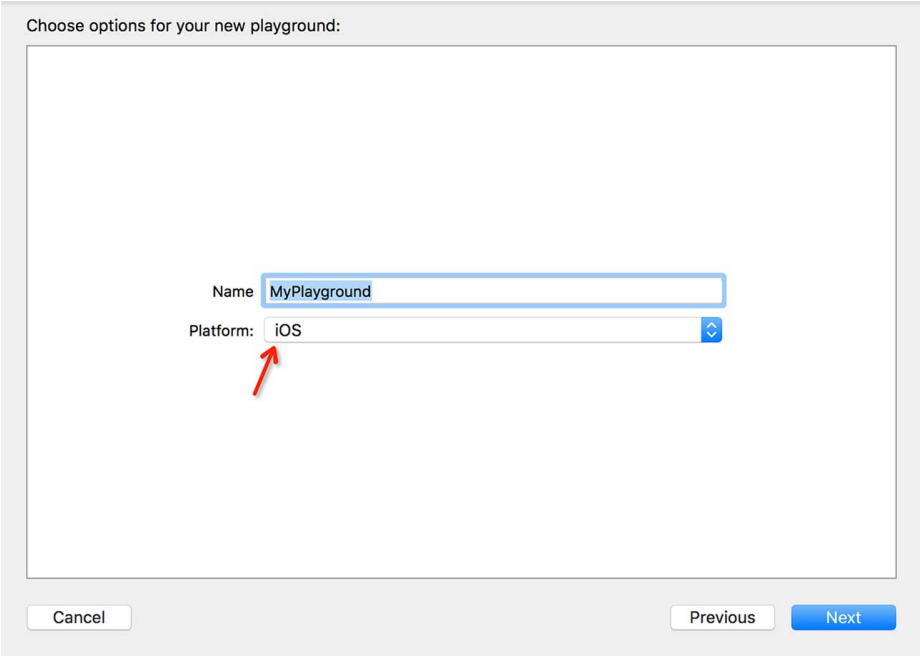
Start building a new iPhone, iPad or Mac application.

**Check out an existing project**

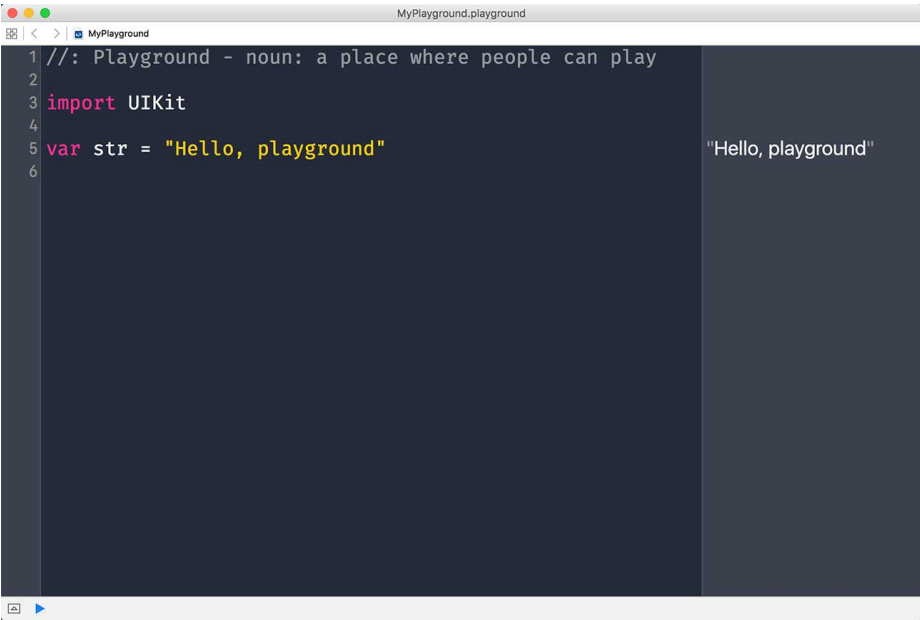
Start working on something from an SCM repository.

**Show this window when Xcode launches**

之后，Platform选择iOS，为自己的Playground写一个名字，例如我们用到的variables-and-constants。



点击Next按钮为Playground文件设置一个保存目录后， 就可以开始在Playground中编写代码了。



在上图中我们可以看到，Playground左边是我们编写代码的区域，右边则会显示出对应语句的执行结果。

## 从定义一个变量开始

变量是编程语言中最常用的语法元素，它们用于在程序中把一个名字（例如： `swiftString` ）和某种特定类型的值（例如： `"Swift is fun."` ）关联起来。

在Swift里，我们使用关键字 `var` 来定义各种不同类型的变量：

```
var Name = Value
```

例如，定义一些我们熟悉的基本类型变量：

```
//: #### Fundamental types

//: Integer
var hours = 24

//: Double
var PI = 3.14

//: Bool
var swiftIsFun = true

//: String
var boxue = "boxue.io"
```

除了这些基本类型之外，Swift还有一些自己特有的类型，例如：方便我们打包多个类型的Tuple

```
//: Tuple - Pack many types of values together
var me = ("Mars", 11, "11@boxue.io")
me.0
me.1
```

定义好一个Tuple之后，我们就可以使用示例代码中，`me.0`，`me.1`这样的方式来访问Tuple中的成员了。我们也可以像下面这样直接修改变量的值：

```
hours = 12
PI = 3.14159
```

如果我们不希望变量的值被修改，我们就需要定义一种特殊的变量。

### 常量是不能被修改的变量

定义常量的方法和变量类似，我们把关键字 `var`，变成关键字 `let` 就可以了。例如：

```
let minutes = 30
let fireIsHot = true
```

一旦我们定义常量之后，就不能再修改常量的值了，否则，编译器会告诉我们发生了错误。例如，我们修改常量 `fireIsHot`：

```
fireIsHot = false
```

这时，我们就可以在Playground中直接看到一个编译错误。



只要程序中的变量值不会被修改，就总应该使用 `let` 来定义它。这不仅可以防止变量被意外修改，编译器还可以针对常量做更多优化。

### Type inference / annotation

在前面的例子里，我们可以发现，无论是整数、浮点数、字符串还有Tuple，我们都没有明确的指定变量的类型，Swift编译器会根据我们为变量的赋值自动推导变量的类型，这个特性，在Swift里叫做**Type Inference**。

但有时，我们定义变量的时候，并不能确定变量的初始值，这个时候，我们就必须像下面这样明确告诉编译器我们需要的变量类型：`var Name:Type`。这种通过:Type指定变量类型的形式，在Swift中叫做**Type annotation**。例如：

```
var x: Int
var s: String
```

### What's next?

在这一节中，我们定义了一些包含不同类型值的常量和变量，并了解了Swift如何根据它们的值自动推导它们的类型。这些内容是Swift中最基本的语言要素，在下一节中，我们会进一步了解其中的一类变量：如何在Swift中使用各种形式的数字。



职场漂泊的你，每天多学一点。

从开发、测试到运维，让技术不再成为你成长的绊脚石。我们用打磨产品的精神去传播知识，把最新的移动开发技术，通过简单的图表，清晰的视频，简明的文字和切实可行的例子一一向你呈现。让学习不仅是一种需求，也是一种享受。

泊学动态

- 一个工作十年PM终创业的故事（二）(https://www.boxueio.com/after-the-full-upgrade-to-swift3)  
Mar 4, 2017
- 人生中第一次创业的"10有" (https://www.boxueio.com/founder-chat)  
Jan 9, 2016
- 猎云网采访报道泊学 (http://www.lieyunwang.com/archives/144329)  
Dec 31, 2015
- What most schools do not teach (https://www.boxueio.com/what-most-schools-do-not-teach)  
Dec 21, 2015
- 一个工作十年PM终创业的故事（一）(https://www.boxueio.com/founder-story)  
May 8, 2015

泊学相关

- 关于泊学 >
- 加入泊学 >
- 泊学用户隐私以及服务条款 (HTTPS://WWW.BOXUEIO.COM/TERMS-OF-SERVICE)
- 版权声明 (HTTPS://WWW.BOXUEIO.COM/COPYRIGHT-STATEMENT)

联系泊学

Email: 10[AT]boxue.io (mailto:10@boxue.io)

QQ: 2085489246