无限互联是国内唯一一家**专注**于iPhone和iPad软件开发培训机构，到目前为止为各大公司输送了一大批优秀的iOS高级软件研发人才。随着iOS7系统的发布，我们也在陆续发布**国内首套完整**的iOS开发的视频教程，手把手教您写代码，从入门到熟练再到精通。

高薪就业是检验一家培训机构质量的唯一标准，我们的学员高薪就业是对我们最好的肯定，也是我们前进的最强烈的动力，我们感谢同学们的努力，感谢你们对我们的支持！我们也将免费为你们提供最好的就业后的技术支持！

亲爱的同学们，你们的**高薪就业**才是我们最大的成功！

http://www.iphonetrain.com

Swift 开发入门

主讲：汪鸿俊

http://www.iphonetrain.com

無限互联
MOBILE STAR

本节内容

- Swift 简介（雨燕）
- Swift语言基础
- 字符、字符串
- 容器（数组、字典、元组）
- 可选量（？、！）

無限互联
MOBILE STAR

Welcome to Swift

- Swift 简介
  - WWDC 2014年6月3号 苹果开发者大会 发布，2010年7月开始开发
  - 基于 C 和 Objective-C语言，使用现有的Cocoa和Cocoa Touch框架，无缝兼容C、Objective-C语言
  - 兼具编译语言的高性能（Performance）和脚本语言的交互性（Interactive）
  - 支持Playground，它允许程序实时预览，无需频繁创建和运行App
  - 简洁、安全、容易、灵活、高效
- 开发环境
  - Mac OS X系统 10.10　或　Mac OS X系统 10.9.3
    - http://pan.baidu.com/s/1eQj94jg
  - Xcode6-Beta
    - http://pan.baidu.com/s/1kTBP9D9

無限互联 MOBILE STAR

Swift语言基础

- Swift官网
  - https://developer.apple.com/swift/
- 官方文档（The Swift Programming Language）
  - https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/Swift_Programming_Language/TheBasics.html#//apple_ref/doc/uid/TP40014097-CH5-XID_399
  - 从应用程序 iBooks 免费下载
    - https://itun.es/cn/jEUH0.l

- Hello World

```
println("Hello, World!")
```

無限互联 MOBILE STAR

语法基础

- 单行注释

```
//        单行注释
```

- 多行注释
  - Swift 多行注释可以嵌套，这点和其他语言不一样， 很实用的一个功能

```
/*
    // 单行注释
    /* 多行注释。需要注意的是，多行注释可以嵌套，这点和其他语言不一样，很实用的一个功能 */
*/
```

- 分号
  - 单个语句后面无需加分号，加了也行
  - 多个语句放在一行，需要加分号

無限互联 MOBILE STAR

数据类型

- 常用数据类型
  - Int：Int8、Int16、Int32、Int64
  - UInt：UInt8、UInt16、UInt32、Uint64
  - Double
  - Float
  - Bool
  - Character
  - String
  - Optional（可选类型）

無限互联 MOBILE STAR

变量与常量

- 变量
  - 值可以根据需要不断修改的量称为变量
  - 使用var声明变量

```
var myVariable = 42
```

- 常量
  - 值不能够被二次修改的量称为常量
  - 使用let声明常量

```
let myConstant = 12
```

- 类型推导:
  - 编译器在编译的时候 通过提供的初始化值 自动推导出 特定的表达式的类型

```
var myVariable = 42      //编译器自动推断其类型为Int
var int : Int = 10       //显示声明类型，语法: "变量 : 类型名"
```

命名规则

- 命名规则
  - Swift中可以使用几乎任何字符来作为常量和变量名，包括Unicode。但是不能含有数学符号、箭头、无效的Unicode、横线-、制表符，且不能以数字开头

```
var π = 3.14
var 无限互联 = "学习iOS"
let 🐶 = "汪"
let cat = "🐱";
```

- 整型表现形式
  - 二进制数，前缀为 0b
  - 八进制数，前缀为 0o
  - 十六进制数，前缀为 0x

```
let decimalInteger = 17
let binaryInteger = 0b10001     // 17 in binary notation
let octalInteger = 0o21         // 17 in octal notation
let hexadecimalInteger = 0x11   // 17 inhexadecimal notation
```

字符与字符串

- 字符 Character
  - 用 Character 定义单个字符，只能存放一个字符

```
//用 Character 定义单个字符:
let money: Character = "￥"
var face: Character = "😱"
println(money+face)
```

- 字符串 String
  - Swift中，字符串不是指针，而是实际的值

```
var emptyString = "" //空字符
var anotherEmptyString = String()

let someString = “ABC"
let dollarSign = "\x24" //单字节Unicode字符，两个十六进制

//通过isEmpty属性可以检查一个字符串是否为空
if emptyString.isEmpty {
    println("Nothing to see here")
}
```

字符串

- 字符串常用方法

```
//字符计数。Swift 中的字符在一个字符串中 并不一定占用相同的内存空间，需使用全局函数
countElements计算一个字符串中字符的数量
let unusualMenagerie = "Koala , Snail , Penguin , Dromedary "
println("unusualMenagerie has \(countElements(unusualMenagerie)) characters”)

let string1 = "hello"
let string2 = " there"
let character1: Character = "!"
let character2: Character = "?"
let stringPlusCharacter = string1 + character1
let stringPlusString = string1 + string2
let characterPlusString = character1 + string1
let characterPlusCharacter = character1 + character2

var instruction = "look over"
instruction += string2
var welcome = "good morning"
welcome += character1

let multiplier = 3
let message = "\(multiplier) times 2.5 is \(Double(multiplier) * 2.5)”

let label = "The width is “; let width = 56
var widthLabel = label + String(width)  //swift不支持隐式类型转换，需要显示类型转换
```

無限互联 MOBILE STAR

字符串比较

- 字符串比较
  - Swift提供三种方法比较字符串的值：字符串相等，前缀相等和后缀相等
- 字符串相等
  - 直接使用 == 来判断

```
//当两个字符串的包含完全相同的字符时，他们被判断为相等。
let quotation = "We're a lot alike, you and I."
let sameQuotation = "We're a lot alike, you and I."
if quotation == sameQuotation {
    println("These two strings are considered equal")
}
```

字符串比较

- 前缀（prefix）、后缀（hasSuffix）相等
  - 通过调用字符串的 hasPrefix、hasSuffix 方法来检查字符串是否拥有特定前缀、后缀

```
let animals = ["食肉:老虎", "食肉:狮子", "食草:牛群", "食草:羊群", "食草:马群"]
//hasPrefix
var aCount = 0
for animal in animals {
    if animal.hasPrefix("食肉") {
        ++aCount
    }
}
println("这有\(aCount)头食肉动物")

//hasSuffix
var bCount = 0
for animal in animals {
    if animal.hasSuffix("群") {
        ++bCount
    }
}
println("这有\(bCount)群食草动物")
```

無限互联 MOBILE STAR

字符串

- 大小写转换
  - 通过字符串的 uppercaseString 和 lowercaseString 来转换大小写

```
let normal = "Could you help me, please?"
let shouty = normal.uppercaseString
println("shouty: " + shouty)
let whispered = normal.lowercaseString
println("whispered: " + whispered)
```

- 字符串编码

```
let dogString = "Dog!🐶"
for codeUnit in dogString.utf8 {    //UTF-8. 得到字符的utf8编码的值
    print("\(codeUnit) ")
}
//Unicode 标量 (Unicode Scalars)你可以使用String类型的unicodeScalars属性遍历一个
Unicode标量编码的字符
for scalar in dogString.unicodeScalars {
    print("\(scalar.value) ")
//    println("\(scalar) ")
}
```

类型别名

- 类型别名
  - 使用typealias关键字定义类型别名，类似typedef

```
typealias NSInteger = Int
var value : NSInteger = 45
value = 12
println(value);
```

- 布尔类型

```
var tigerIsAnimal : Bool = true
var animalIsTiger = false
```

无限互联网站：　http://www.iphonetrain.com
老师E-mail：　junewhj@qq.com
老师Blog：　http://blog.csdn.net/jaywon

视频讲解过程中如有不妥之处，欢迎大家将信息反馈到我的Email中，我们会努力完善！谢谢各位的支持。

视频持续更新中... 敬请期待！

无限互联是国内唯一一家**专注**于iPhone和iPad软件开发培训机构，到目前为止为各大公司输送了一大批优秀的iOS高级软件研发人才。随着iOS7系统的发布，我们也在陆续发布**国内首套完整**的iOS开发的视频教程，手把手教您写代码，从入门到熟练再到精通。

高薪就业是检验一家培训机构质量的唯一标准，我们的学员高薪就业是对我们最好的肯定，也是我们前进的最强烈的动力，我们感谢同学们的努力，感谢你们对我们的支持！我们也将免费为你们提供最好的就业后的技术支持！

亲爱的同学们，你们的**高薪就业**才是我们最大的成功！

http://www.iphonetrain.com

数组与字典

- Array 数组
  - 使用[]操作符声明数组

```
let emptyArray1 = []          //声明空数组，可以插入任意类型的值
let emptyArray2 = String[]() //声明空数组，限定了值的类型为String
var shoppingList = ["catfish", "water", "tulips", "blue paint"]
println(shoppingList.count)
println(shoppingList[1])
shoppingList[1] = "bottle of water"
println(shoppingList[1])
```

- Dictionary 字典
  - 使用[key:value]操作符声明字典

```
let emptyDictionary1 = [:]
let emptyDictionary = Dictionary<String, Float>()
var occupations = [
    "Malcolm": "Captain",
    "Kaylee": "Mechanic",
]
println(occupations["Malcolm"])
occupations["Jayne"] = "Public Relations"
println("The occupations count is " + String(occupations.count))
```

無限互联 MOBILE STAR

元组（tuple）

- 元组（tuple）
  - 元组可以将任意数据类型组装成一个元素
  - 元组在作为函数返回值的时候特别适用，可以为函数返回更多的信息。
- 元组创建

```
let (x, y) = (1, 2)
let http200Status = (statusCode: 200, description: “OK")

let http404Error = (404, "Not Found")   //由一个Int和一个字符串String组成
let (statusCode, statusMessage) = http404Error
```

- 元组访问
  - 直接访问
  - 通过key访问value，类似 Dictionary
  - 序号访问方式，序号从0开始
  - 属性访问方式，点语法

無限互联 MOBILE STAR

可选类型（Optional）

- 可选类型 Optional（? 的用法）
  - 可选类型：这个值要么存在，并且等于x；要么根本不存在。
  - （变量名\常量名: 类型?）

```
var serverResponseCode: Int? = 404  //?的意思是 要么存在，且值为404；要么值不存在，为nil

let possibleNumber = “123”
//let possibleNumber = “Hello"

//由于toInt方法可能会失败，因此它会返回一个可选的Int类型，而不同于Int类型。一个可选的Int类型被记为Int?，不是Int。问号表明它的值是可选的，可能返回的是一个Int，或者返回的值不存在。
let convertedNumber: Int? = possibleNumber.toInt()
```

- nil
  - Swift 的nil不同于Object-C中的nil。Object-C中，nil是一个指针指向不存在的对象。Swift中，nil不是指针而是一个特定类型的空值。任何类型的可选变量都可以被设为nil，不光是指针。
  - nil 不能用于非可选类型。

解包

- 解包（！的用法）
  - 可选类型在每次访问的时候 都会提取并检测它的值是否存在，但有时候根据程序结构就可以推断 可选量在首次赋值后 必然存在值，这时候，就不需要每次验证值是否存在，如果确定一个可选量的值一定存在，那么我们使用 ！ 进行解包获取它的值，或者使用 Optional Binding。

```
let possibleString: String? = "An optional string."
println(possibleString!) // 解包，我确定possibleString的值一定存在，不需要每次验证值是否存在

let stringValue = possibleString!.hashValue    //解包，这里的!表示“我确定这里的的possibleString一定是非nil的，尽情调用吧”

//Optional Binding，等价于上面一行
if let value = possibleString{
    let stringValue = value.hashValue
}
```

隐式解包的可选类型

- 隐式解包的可选类型
  - 可以把 隐式解包可选类型 当成自动解包的可选类型。即不是每次使用的时候 在变量、常量后面加！，而是直接在定义的时候加！这些可选量定义为隐式解包的可选量（implicitly unwrapped optional）。
  - 隐式解包的可选量的声明格式为：在希望标为可选的类型名称后面，用感叹号 (!) 代替问号 (?)。
  - 隐式解包的可选类型主要用在一个变量/常量在定义瞬间完成之后值一定会存在的情况。主要用在类的初始化过程中。

```
//把 隐式解包可选类型 当成对每次使用的时候自动解包的可选类型。即不是每次使用的时候 在变量/常量后面加！，而是直接在定义的时候加!
let assumedString: String! = "An implicitly unwrapped optionalstring."
println(assumedString)                //访问其值时无需感叹号
println(assumedString.hashValue)  //访问其值时无需感叹号
```

无限互联网站：　http://www.iphonetrain.com
老师E-mail：　junewhj@qq.com
老师Blog：　http://blog.csdn.net/jaywon

视频讲解过程中如有不妥之处，欢迎大家将信息反馈到我的Email中，我们会努力完善！谢谢各位的支持。

视频持续更新中... 敬请期待！

无限互联是国内唯一一家**专注**于iPhone和iPad软件开发培训机构，到目前为止为各大公司输送了一大批优秀的iOS高级软件研发人才。随着iOS7系统的发布，我们也在陆续发布**国内首套完整**的iOS开发的视频教程，手把手教您写代码，从入门到熟练再到精通。

高薪就业是检验一家培训机构质量的唯一标准，我们的学员高薪就业是对我们最好的肯定，也是我们前进的最强烈的动力，我们感谢同学们的努力，感谢你们对我们的支持！我们也将免费为你们提供最好的就业后的技术支持！

亲爱的同学们，你们的**高薪就业**才是我们最大的成功！

http://www.iphonetrain.com