swift语言

雨燕、敏捷、快速

Swift语言

新一代苹果平台（iOS/macOS/watchOS/tvOS）的APP及游戏开发语言。

2014年6月推出第一个版本，由苹果员工Chris Lattern历时四年发明。

Swift已开源，正逐渐替代Objective-C语言，成为iOS开发的主流语言。

创新

增加可选变量

使用协议替代继承

环境

macOS ≥10

Xcode≥8.0

或者在线https://swiftlang.ng.bluemix.net

常量

常量的定义

格式：let 常量名 = 值

举例：let pi = 3.1415926

举例：let 工资 = 5000

变量

格式：var 变量名

举例：var age:Int = 3

举例：var age = 3（类型推断）

类型安全

变量一旦定义，其类型不可更改

举例：var 价格 = 3

举例：价格 =“五毛”　出错

注释

代码注释：有助于自己和他人理解

// 行注释

/*段注释*/

元组（Tuple）：定义变量的一个组合

形式（一般）：(5,”天”,”swift”)

形式（前缀）：(day:5, unit:”天”, lang:”swift”)

var title = (day:5, unit:”天”, lang:”swift”)

可选类型(Optional)

可选类型：代表变量可能有值的情况。

格式：var 变量名: 类型?　　默认是无值(nil)

举例：var addr : String?

addr = “徐州”

或者：var addr : String? = “徐州”

字符串

String是字符串类型，Character是字符类型

可通过字符串插值合成一个长字符串

```
let sentence = “iOS应用开发技术”
for word in sentences.characters {
    print(word)
}
```

字符串

判断字符串是否为空　.isEmpty

连接字符串　　+

let a = “Hello”

let b = “World”

let c = a + b

向字符串添加字符　　用append方法

let d : Character = “1”

c . append(d)

字符串

字符串插值

```
let name = "Nancy"

let classTime = 32

let expTime = 32

print("\(name)学习了iOS应用开发技术，一共\(classTime + expTime)学时")
```

数组、集合、字典

有序可重复-数组（Array）

无序不重复-集合（Set）

无序可重复，但每一个值有唯一的键（Key）-字典（Dictionary）

数组

定义：Array<类型>或[类型]

let array : [Int]

array = [int](repeatElement(3, count: 10))　　数组里面有10个3

范围值定义

let array2 = Array(1…10)　　数组里面有从1到10的10个数

let places = [“泉山区”, ”云龙区”, ”铜山区”]

数组

元素计数：count

空否：.isEmpty

添加一个元素：.append　　places.append(“鼓楼区”)

let otherPlaces = [“玄武区”, ”白下区”]

places = places + otherPlaces 或者 places += otherPlaces

插入：insert

places.insert(“Paris”, at: 4)

删除：remove

places.remove(at: 8)

字典

定义：Dictionary<键类型，值类型>，或[键类型：值类型]

常用方法：var airports = ["PVG":"Shanghai pudong","CHU":"Dalian","DUB":"DUBLIN Airport"]

字典是否为空　　airports.isEmpty

字典计数　　airports.count

添加或更新：字典变量[键] = 值

airports["SHQ"] = "Hongqiao Airport"

airports["CHU"] = "大连周水子机场"

字典

获取，可以用下标　　airports["DUB"]

移除，用下标把值设为nil　　airports["DUB"] = nil

循环一个字典for in，因为键值对有2个元素，用元组变量

```
for (key, value) in airports{
    print(key,value)
}
```

字典

单独使用其中键或值，使用keys或values(可使用for in)

```
for key in airports.keys{
    print(key)
}
```

把键值对分类成数组，用[数组类型]（字典变量.keys），[数组类型]（字典变量.values）

```
let codes = [String](airports.keys)
let airportFullname= [String](airports.values )
```

函数

形式：func 函数名（参数1：类型，参数2：类型，…）->返回结果的类型语句{执行语句}

调用：var 变量名称 = 函数名（变量1，变量2，…）

参数和返回值：可以无参数无返回值

多返回值：使用元组

函数类型：包含参数和返回类型的简写形式，可以像普通变量那样使用

函数

使用参数标签

为每一个参数提供标签，并且这些标签命名应该是唯一的，并且有意义。

```
func retangleArea(w width: Double, h height: Double) -> {
    let area = width * height
    return area
}
retangleArea(w: 20, h: 30)
```