Swift带来的改变和启示

唐平麟

Agenda

- Swift介绍
- LLVM-Clang
- Swift特性
- SwiftyJSON
- 函数式编程

Swift介绍

- Swift是在WWDC2014发布的全新开发语言
- Swift与Objective-C的关系

Chris Lattner

Chris Lattner（是 LLVM 项目的主要发起人与作者之一，Clang 编译器的作者。他现在是苹果公司『开发者工具』部门的主管，领导 Xcode、Instruments 和 编译器团队，从 2010 年 7 月开始主导开发 Swift 编程语言。

Clang-LLVM

- 传统的编译器通常分为三个部分，前端(frontEnd)，优化器(Optimizer)和后端(backEnd)。
- Clang
- LLVM

Swift简单吗？

- println("Hello World!")
- ["Hello ", " World", "!"].reduce(""){$0 + $1}

Swift特性

- 闭包
- 命名空间
- 类型推导
- 元组
- 范型
- 多返回值
- Optional types
- 运算符重载
- 面向对象
- 函数式编程
- 迭代器

Any和AnyObject

- Objective中的id
- AnyObject代表任意class的实例
- Any代表任意类型

条件编译

- Swift并没有宏
- Swift中任然存在简单的条件编译的功能

os() : OSX, iOS

arch() : x86_64, arm, arm64, i386

#if os(OSX)

typealias View = NSView

#else

typealias View = UIView

#endif

在Build Setting中的Swift Compiler - Custom Flags的Other Flag中可以添加自定义的编译符号

#if os(Testing)

baseURL = “http://codoon.com/api”

#else

baseURL = “http://test.codoon.com/api”

#endif

SwiftyJSON

- App会因为API的数据而不稳定
- 用Swift怎么才能安全的取出数据

我们有这样一个JSON数据：

```
[
  {
    "user": {"name": "Jack"}
  }
  {
    "user": {"name": "Tom"}
  }
]
```

按照Swift 1.2的方式来安全的取数据：

```
if let statusesArray = object as? [AnyObject],

    let status = statusesArray[0] as? [String: AnyObject],

    let user = status["user"] as? [String: AnyObject],

    let username = user["name"] as? String {

        println(username)

}
```

再简单一点的方法：

```
if let username = (((object as? [AnyObject])?[0] as? [String: AnyObject])?["user"] as? [String: AnyObject])?["name"] as? String {

    println(username)

}
```

使用SwiftyJSON后呢?

```
if let userName = json[0]["user"]["name"].string{
    println(username)
}
```

再简单一点呢?

```
if let userName = json[0, "user", "name"].string {
    println(username)
}
```

Chained calls

```
Alamofire.request(.GET, “http://abc.com/get”, parameters: ["foo": "bar"])

    .response { (request, response, data, error) in
      println(request)

      println(response)

      println(error)
    }
```

```
var button = UIButton(frame:buttonFrame)
    .now {button in
        button.setTitle("Button", forState: UIControlState.Normal)
    }.when(UIControlEvents.TouchUpInside) {
        $0.backgroundColor = UIColor.blackColor()
}
```

函数式编程

- 函数式编程是一种编程范式。

取10以内的奇数

传统做法:

```
var odds = [Int]()

for i in 1...10 {

    if i % 2 != 0 {

        odds.append(i)

    }

}

println(odds)
```

函数式编程：

```
func odd(number: Int) -> Bool {
    return number % 2 == 0
}
var odds = Array(1...10).filter(odd)
println(odds)
```

还可以这样：

```
var odds = Array(1...10).filter{ $0 % 2 == 0 }
println(odds)
```

取10以内的奇数的和

传统做法：

```
var sum = 0

var odds = [Int]()

for i in 1...10 {

    if i % 2 == 1 {

        odds.append(i)

        sum += i

    }

}

println(sum)
```

函数式编程:

```
let sum = Array(1...10)
    .filter {$0 % 2 == 1}
    .reduce(0) {$0 + $1}
println(sum)
```

@Github

- Alamofire
- SwiftyJSON
- Design-Patterns-In-Swift
- Dollar.swift
- ReactiveCocoa

Swift的不足

- 不需要使用Objective-C的API
- 需要使用到Objective-C的API
- UnsafePointer<T> （相当于C中的const char *）

Q&A

谢谢！