

《Swift编程权威指南》阅读计划（图灵移动开发群第1期）

领读人	Story5
本书特色：	涵盖Swift知识点比较全面,且示例比较多
适合读者：	有一定的编程基础的新手入门Swift 有一定的OC基础的更容易上手
第一周答疑时间安排：	欢迎读者们来图灵读者小密圈打卡、贴图、发疑问。
总阅读时间长度（预估）：	每天阅读用时：2小时

版块划分：

第一部分 起步

第 1 章 起步

阅读时长:30分钟

内容概要:

第1章主要讲解iOS编程特有IDE-[Xcode](#)的简单实用.

- 对于OC转Swift有一定的Xcode的基础的同学,可以简单略过.
- 对于新手,这一部分可能只能简单的通过playground去练习Swift,涉及到项目编程,Xcode的使用,可能还需要更深入的了解.建议跟着书本步骤自己动手实践.

补充:

关于Xcode的更多的用法,可以参考 [《Welcome to Xcode》](#)

第 2 章 类型、常量和变量

阅读时长:1小时30分钟

内容概要:

第2张主要讲解编程语言基础概念,类型/常量/变量,以及其在Swift中的表示方式.

重点内容:

- 类型,贯彻类型的定义
- 常量 `let` 的表示
- 变量 `var` 的表示

难点内容:

- 字符串插值用法 `\()`,实现转义字符串,新手比较不喜欢这种表示方式,在跟读的过程中,尝试跟着示例编程使用该用法.

补充:

这些内容是编程的基础,地基要打好.一定要理解透彻.

第二部分 基础知识

第 3 章 条件语句

阅读时长:2小时

内容概要:

第3章主要介绍条件语句在Swift中的使用,条件语句在编程语言中是很重要的控制程序跳转的方式,该章要细化swift中条件语句的条件表达式形式.

重点内容:

- if-else的使用
- if-else的嵌套
- 三目运算符的使用

难点内容:

- 三目运算符 (条件)?(真表达式,条件为真执行):(否表达式,条件为否是执行)

第 4 章 数

阅读时长:2小时

内容概要:

第4章主要介绍了Swift中的整数,整数的四则预算(`+-*/`),以及浮点数表示

重点内容:

- 整数(`Int`)
- 整数的四则运算,加(`+`),减(`-`),乘(`*`),除(`/`)
- 浮点数

难点内容:

- `Int`在不同系统中的形式(`Int8`、`Int16`、`Int32`、`Int64`)
- 溢出操作符的理解,以及产生溢出的原因

第 5 章 switch语句

阅读时长:2小时

内容概要:

第5章介绍了在Swift中更灵活和强大的switch-case用法.switch-case同if-else一样是编程中很常用的程序跳转控制语句,需要理解并熟练使用.

重点内容:

- switch-case的用法
- 区间的表示
- 值绑定的用法
- where子句在switch-case的用法
- 元组和模式匹配

难点内容:

- 对于值绑定的理解
- 对于元祖的理解和元祖在switch-case中的匹配

第 6 章 循环

阅读时长:1小时30分钟

内容概要:

第6章介绍了编程常用的循环语句.循环在处理数组或字典元素时,很使用,避免了写很多重复代码.新手一定要领会循环的用法,以及应用场景

重点内容:

- for-in循环
- while循环
- repeat-while循环
- break/continue在循环中的使用

第 7 章 字符串

阅读时长:30分钟

内容概要:

第7章介绍了swift中的字符串表示以及国际字符串编码标准Unicode

重点内容:

- 字符串的表示
- Unicode的含义及应用

第 8 章 可空类型

阅读时长:2小时

内容概要:

第8章介绍了Swift中的可空类型的灵活使用,可空类型在Swift中是个比较难理解的概念,坑比较的多,特别是涉及到 `?` 以及 `!` 的用法,在Swift的整个编程过程中也很常用.大家一定要把这章作为重点,一定要理解到位再开始往下学习.

重点内容:

- 关于可空类型的定义
- 可空实例绑定
- 可空类型的隐式展开
- 可空链式调用

难点内容:

- `?` 和 `!` 的意义及使用

补充:

关于可空类型 `?` 和 `!` 的理解与使用是个难点.

下面几篇[onevcat](#)的文章能够有助于你理解其用法

- [《OPTIONAL CHAINING》](#)
- [《隐式解包 OPTIONAL》](#)
- [《多重 OPTIONAL》](#)
- [《OPTIONAL MAP》](#)

第三部分 容器和函数

第 9 章 数组

阅读时长:2小时

内容概要:

第9章介绍swift中数组的用法.大家跟读并按示例代码联系就可以了,基本没难点

重点内容:

- 数组
- 不可变数组

第 10 章 字典

阅读时长:2小时

内容概要:

第10章主要介绍swift中字典的定义以及增删遍历等操作

重点内容:

- 字典概念以及初始化
- 字典的增删
- 字典的遍历
- 字典转成数组

第 11 章 集合

阅读时长:2小时

内容概要:

第11章介绍了iOS中集合(Set)的概念以及用法.

重点内容:

- 集合的概念以及运用
- 集合的运算,并集/交集/不相交

第 12 章 函数

阅读时长:4小时

内容概要:

第12章介绍了程序的根本,函数的定义以及使用.我们编程的过程就是从 `main` 函数开始不断调用不同的函数,需要扎扎实实的学习并学会自定义函数.

重点内容:

- 函数的参数
- 函数的返回值

难点内容:

- 变长参数
- in-out参数
- 嵌套函数
- 元祖实现多个返回值

第 13 章 闭包

阅读时长:4小时

内容概要:

闭包,可以理解为OC中的代码块Block,第13章主要介绍闭包的定义及使用.且提及了一个概念函数式编程.关于闭包的概念比较难懂,大家在这里需要细读

重点内容:

- 闭包的用法
- 函数作为返回值
- 函数作为参数
- 闭包能捕获变量
- 闭包是引用类型
- 函数式编程

难点内容:

- 闭包能捕获变量
- 函数式编程
- 三个高阶函数的使用
 - `map(_:)`
 - `filter(_:)`
 - `reduce(_:_:)`

第四部分 枚举、结构体和类

第 14 章 枚举

阅读时长:4小时

内容概要:

第14章介绍了枚举类型在Swift中的使用.Swift中的枚举相当强大,大家在学习后要思考枚举的应用场景,来使自己的代码更简练.

重点内容:

- 基本枚举的定义以及枚举的调用
- 怎么定义整型枚举---原始值枚举
- 怎么讲方法与枚举关联
- Swift中更高级的枚举用法---关联值
- 递归枚举

难点内容:

- 递归枚举,涉及到递归,很多人思路就会乱,这一章需要细读理解

第 15 章 结构体和类

阅读时长:4小时

内容概要:

第15章讲解了Swift中结构体和类的概念,有过OC基础的同学,就会发现,在OC中有些类在Swift中不再是类的形式存在,而是以结构体形式.这一章大家需要学会Swift中结构体和类的定义以及他们之间的区别,不同的应用场景等.

重点内容:

- 结构体
- 类
- 类的继承

难点内容:

- mutating方法在结构体中的使用
- 类的继承

第 16 章 属性

阅读时长:4小时

内容概要:

第16章,属性分为2种,存储属性和计算属性,我们常把属性和类关联在一起.

重点内容:

- 属性定义,存储属性
- 嵌套类型的定义
- 惰性存储属性以及懒加载(lazy loading)
- 计算属性(getter和setter方法)
- 属性观察者
- 类型属性
- 访问控制(`open` , `public` , `internal` , `fileprivate` , `private`)

难点内容:

- 属性观察者,比较难理会,细读.

第 17 章 初始化

阅读时长:6小时

内容概要:

第17章,实例的创建都需要初始化,在这个工程中,系统会给每个存储属性分配内存,并赋予初始值.本章主要介绍了结构体和类的初始化方式

重点内容:

- 结构体初始化
- 类初始化
- 反初始化
- 可失败的初始化处理
- 初始化方法参数设置

难点内容:

- 反初始化
- 可失败的初始化处理

第 18 章 值类型与引用类型

阅读时长:4小时

内容概要:

本章以学习过的关于值类型(比如结构体)和引用类型(比如类的内容为基础,通过对比一系列场景中两者的不同行为来探索它们的区别。学习本章,需要对何时使用值类型、何时使用引用类型有一个比较好的理解。

重点内容:

- 值类型的定义以及应用
- 引用类型的定义以及应用
- 浅复制和深复制(又叫浅拷贝和深拷贝)

难点内容:

- 浅拷贝和深拷贝是个难点,需要对内存的概念相当清晰.特别是在数组中元素的拷贝时容易概念不清晰,不清楚到底是深拷贝还是浅拷贝

第五部分 Swift高级编程

第 19 章 协议

阅读时长:2小时

内容概要:

第19章,协议.协议是iOS编程中很重要的概念,最常用于反向传值.对于有OC基础的可以主要关注Swift协议的写法.而对于新手,该一概念比较难理解,新手需要仔细阅读协议的定义,实现以及作用.并跟着书中的示例进行练手操作.

重点内容:

- 协议的概念,以及如何声明一个协议
- 如何声明一个delegate遵循协议并实现协议方法.

难点内容:

- 协议继承,同OC不同,Swift支持了协议继承.

第 20 章 错误处理

阅读时长:2小时

内容概要:

第20章介绍了,如何进行错误处理.仔细阅读这一章,有助于我们在程序中进行的异常/错误处理,且能够帮我能更深一步的提升调试能力.毕竟实际编程过程中, 20%编码80%调试

重点内容:

- 词法分析
- 捕获错误
- 理解怎么处理错误

难点内容:

- 词法分析

第 21 章 扩展

阅读时长:2小时

内容概要:

第21章,扩展.扩展可以让你给已有类型添加你想要的功能.可以用来扩展类,结构体和枚举

重点内容:

- 扩展已有的类型
- 扩展自己的类型
- 扩展使类型符合协议

难点内容:

- 本章难点就在于扩展的写法,大家仔细研究书中的示例,并按其练习.

第 22 章 泛型

阅读时长:6小时

内容概要:

第22章介绍了Swift中泛型的用法,泛型让我们写出的类型和函数可以使用对于我们或编译器都未知的类型。

重点内容:

- 泛型数据结构,要理解泛型栈的概念
- 泛型函数和方法
- 类型约束的作用
- 关联类型协议
- 可空类型
- 参数多态

难点内容:

- 关联类型协议
- 可空类型
- 参数多态

第 23 章 协议扩展

阅读时长:2小时

内容概要:

第23章介绍协议扩展出现的缘由,已经协议扩展的实现.

重点内容:

- 掌握协议扩展的实现

第 24 章 内存管理和ARC

阅读时长:4小时

内容概要:

第24章,内存管理和ARC,本人觉得iOS编程中的重中之重.从ARC出现后,很多人很少去关注内存方面的问题.全部托由ARC去管理.但是学好内存管理,会让我们对于程序的内存优化,内存错误分析会有很大的帮助.

重点内容:

- 内存分配
- 循环强引用的出现缘由以及怎么解决
- 闭包中的循环引用
- 逃逸闭包和非逃逸闭包
- 深入学习:我能获取实例的引用计数吗

难点内容:

- 内存分配
- 逃逸闭包和非逃逸闭包

补充:

关于内存的更深入的研究可以看图灵出版的 [《高性能iOS应用开发》](#)

第 25 章 Equatable和Comparable

阅读时长:2小时

内容概要:

第25章,介绍了Swift如何实现自定义类型的比较

重点内容:

- 符合Equatable
- 符合Comparable

第六部分 事件驱动的应用

第六部分通过创建基本的应用形式,让我们熟悉iOS编程的过程.大家跟着书中的实例一步步练习熟悉.

第 26 章 第一个Cocoa应用

阅读时长:2小时

重点内容:

- 自动布局

补充:

想更深入的研究自动布局-autolayout可以看这本书 [《iOS Auto Layout开发秘籍\(第2版\)》](#)

第 27 章 第一个iOS应用

阅读时长:2小时

第 28 章 互操作

阅读时长:2小时

读书时间计划:

部分	时间	章节	小结
第一部分 起步	第1天	第1章 起步	
		第2章 类型、常量和变量	
第二部分 基础知识	第2天	第3章 条件语句	
	第3天	第4章 数	
	第4天	第5章 switch语句	
	第5天	第6章 循环	
		第7章 字符串	
	第6天	第8章 可空类型	
	第7天	第9章 数组	
	第8	第10章 字典	

第三部分 容器和函数	天		
	第9天	第11章 集合	
	第10天	第12章 函数	12.1 ~ 12.3
	第11天		12.4 ~ 12.11
	第12天	第13章 闭包	13.1 ~ 13.4
	第13天		13.5 ~ 13.10
第四部分 枚举、结构体和类	第14天	第14章 枚举	14.1 ~ 14.3
	第15天		14.4 ~ 14.7
	第16天	第15章 结构体和类	15.1 ~ 15.4
	第17天		15.5 ~ 15.10
	第18天	第16章 属性	16.1 ~ 16.4
	第19天		16.5 ~ 16.10

	第20天	第17章 初始化	17.1 ~ 17.2
	第21天		17.3
	第22天		17.4 ~ 17.8
	第23天	第18章 值类型与引用类型	18.1 ~ 18.4
	第23天		18.5 ~ 18.8
第五部分 Swift高级编程	第24天	第19章 协议	
	第25天	第20章 错误处理	
	第26天	第21章 扩展	
	第27天	第22章 泛型	22.1 ~ 22.2
	第28天		22.3 ~ 22.8
	第29天		22.9 ~ 22.10

	第30天	第23章 协议扩展	
	第31天	第24章 内存管理和ARC	24.1 ~ 24.3
	第32天		24.4 ~ 24.8
	第32天	第25章 Equatable和Comparable	
第六部分 事件驱动的应用	第33天	第26章 第一个Cocoa应用	
	第34天	第27章 第一个iOS应用	
	第35天	第28章 互操作	