

String

1. OC的NSString
2. 字符串字面量
3. 空字符串: ""或者String()
4. 字符串拼接: + +=
5. 可变性: var/let
6. 字符串插值: ()
7. 判断相等: == !=
8. 截取
9. 前缀后缀
10. 遍历

Array

1. OC的NSArray
2. 空数组: [Int](#)
3. 字面量构造
4. 带默认值的数组
5. 数组相加
6. 数组元素数量: count、isEmpty
7. 追加: append、+=
8. 下标操作: 支持多下标
9. 插入和删除
10. 遍历: 只遍历元素、同时遍历索引和元素(enumerate方法)
11. 字符串和数组互相转化

Dictionary

1. OC的NSDictionary
2. 构造器创建
3. 字面量
4. 键值对数: count、isEmpty
5. 下标访问: 修改、添加、删除 (置为nil)
6. updateValue方法
7. removeValueForKey方法
8. 遍历: 遍历键值对 (元组)、单独遍历keys或者values
9. 使用keys或者values构造数组

练习

1. 练习字符串、数组、字典的基本操作

枚举

1. C和OC的枚举
2. 枚举定义
3. 赋值：类型确定时可省略枚举名
4. switch匹配：必须涵盖所有情况（可使用default分支）
5. 关联值
6. 原始值：自定义原始值、整型和字符串类型的枚举原始值
7. 构造器：原始值做参数，返回值为可选类型

结构体

1. OC的结构体
2. 结构体定义
3. 生成实例：默认构造器和逐一成员构造器
4. 访问属性：点语法
5. 直接设置类的结构体属性的子属性
6. 结构体是值类型
7. 类是引用类型
8. 恒等运算符：=== !==
9. 结构体和类的选择

函数

1. 最简单函数：无参数无返回值，对比C语言
2. swift命名空间：同一个项目下的文件属于同一命名空间，不同文件中的函数可以不用声明直接调用
3. 无参有返回值函数
4. 无参多返回值函数
5. 带参数函数：参数名称（外部和局部）：默认、指定、_忽略
6. 参数默认值
7. 可变参数
8. 常量参数和变量参数
9. 输入输出参数：inout
10. 函数作为类型：定义变量、作为参数类型、作为返回值类型（嵌套函数）

闭包

1. OC的block
2. 最简单闭包：无参无返回值
3. 单表达式隐式返回：省略return
4. 参数缩写：\$0、\$1。。。。
5. 尾随闭包：闭包是函数最后一个参数，且闭包比较长（数组的map方法）

6. 捕获值：捕获上下文的变量或者常量，即使超出作用域，也可使用
7. 闭包是引用类型

作业

1. 给出一组文件名，根据后缀名将文件分类，用一个字典将文件分类保存。

```
如 let files = ["OC常用语法总结.md", "main.swift", "第一天作业.pdf", "swift基本语法.md", "第二天作业.pdf"],  
最终得到字典：  
["md": ["OC常用语法总结", "swift基本语法"], "swift": ["main"], "pdf": ["第一天作业", "第二天作业"]]
```

2. 枚举和结构体练习：
 - 定义Month结构体，包含两个属性：工作日天数和周末天数
 - 定义WeekDay枚举，包含一周的七天
 - 计算2016年6月的工作日天数和周末天数（提示：需要定义数组，保存每天是周几）
 - 选做：打印出6月份的日历
3. 函数练习：编写一个函数，统一字符串中字母、数字、空格和其他字符的个数。（提示：返回值可用元组）
4. 闭包练习：给定一个Dictionary数组，Dictionary包含key值name和key值age，用map函数转换name为英文格式（如 zhang san转换为san zhang），最终得到一个英文名数组