1. 程序流程控制

Swift提供了多种流程控制结构，包括可以多次执行任务的while循环，基于特定条件选择执行不同代码分支的if、guard和switch语句，还有控制流程跳转到其他代码位置的break和continue语句。

Swift 还提供了for-in循环，用来更简单地遍历数组（Array），字典（Dictionary），区间（Range），字符串（String）和其他序列类型。

Swift 的switch语句比 C 语言中更加强大。case 还可以匹配很多不同的模式，包括范围匹配，元组（tuple）和特定类型匹配。switch语句的 case 中匹配的值可以声明为临时常量或变量，在 case 作用域内使用，也可以配合where来描述更复杂的匹配条件。

3.1 for-in 循环

如果你有其他语言的基础，如C++和JAVA，你一定会对大名鼎鼎的for循环非常熟悉。可是在Swift语言中，传统的for循环已经被移除了，取而代之的是for-in循环。

如果我们用for-in循环打印0到9

//代码段 3.1 for-in 打印 0...9

//代码段 0..<10

代码中的(0...9)和(0..<10)被称作区间操作符，其中(0...9)是闭区间操作符，表达的是一个左闭右闭的区间，而(0..<10)是半开区间操作符，表达了一个左闭右开的区间。索引index第一次被赋值为区间中的第一个数字0，然后通过print语句打印出0这个数字。随后，index被赋值为区间中的下一个数字1，再次打印，依次类推，打印0到9这10个数字。最后由于区间(0..<10)不包含10，所以不会打印10。

上面的例子中，index是每次循环遍历开始时自动被赋值的常量。这种情况下，index不需要使用let关键字进行声明。

有时，我们并需要index这个索引，我们可以用\_代替index，从而忽略这个值。

//代码段3.2 打印 3遍 hello world

swift中的区间操作符一定要是一个合法的区间，也就是左边界要小于等于右边界。如果我们要实现逆序打印该怎么做？我们可以调用reverse()方法。

//代码3.3 逆序打印 9-0

3.2 while 循环

While循环在实际使用中也是非常常用的循环。while循环会一直运行一段语句直到条件变成false。while循环非常适合使用在第一次迭代前，迭代次数未知的情况下。

Swift提供两种while循环的形式：

1. while循环，每次在执行语句之前判断条件是否为true
2. repeat-while循环，每次执行语句之后判断条件是否为true

while循环

while循环先判断再执行语句。如果条件为true，会重复运行一段一段语句，直到条件变为false。

//代码3.4 依旧是打印