

< Teach  
Me  
Skills />

# Lesson 2

Базовый функционал

SWIFT



# Swift

Swift — это язык программирования от компании Apple с открытым исходным кодом. Предназначен для разработки приложений для iOS и macOS, реже используется в других проектах. Язык появился всего в 2014 году как альтернатива Objective-C. Сначала Swift был закрытым, потом Apple открыла его исходный код.



swift



# Swift

Swift - это строго типизированный язык, Что означает, что Swift запрещает дальнейшую работу параметров неправильного типа.

Swift - это статически типизированный язык, Это означает, что тип переменных в Swift проверяется во время компиляции. При компиляции компилятор изо всех сил старается найти связанные ошибки, включая ошибки типа. Например, добавляя String и Int к этому типу ошибки операции, компилятор может сообщить вам во время компиляции и не будет сообщать об ошибке во время выполнения.



Swift

```
var integer: Int = 10
```

```
integer = "11"
```



Cannot assign value of type 'String' to type 'Int'



# Типы данных. Int

Integer (целое число) - это число, не содержащее дробной части, например, как 42 и -23. Целые числа могут быть либо знаковыми (положительными, ноль или отрицательными) либо беззнаковыми (положительными или ноль).



```
var integer: Int = 10
```

# Типы данных. Числа с плавающей точкой

Типы с плавающей точкой могут представлять гораздо более широкий спектр значений, чем типы целых значений, и могут хранить числа намного больше (или меньше) чем может хранить Int. Swift предоставляет два знаковых типа с плавающей точкой:

**Double** - представляет собой 64-битное число с плавающей точкой. Используйте его когда число с плавающей точкой должно быть очень большим или чрезвычайно точным

**Float** - представляет собой 32-битное число с плавающей точкой. Используйте его, когда значение не нуждается в 64-битной точности.

```
var double: Double = 5.305274551439368
var float: Float = 9.443022
```



# Типы данных. Bool

В Swift есть простой логический тип Bool. Этот тип называют логическим, потому что он может быть только true или false. Swift предусматривает две логические константы, true и false соответственно:

```
var isNight: Bool = false  
var isDay: Bool = true
```



# Типы данных. String, Character

Строка представляет собой совокупность символов, например "hello, world" или "albatross". Строки в Swift представлены типом String. К значению типа String можно получить доступ различными способами, включая и коллекцию значений типа Character (символ).

```
let greeting = "Hello, world!"
```



# Переменные и константы

Используйте `let` для создания констант и `var` для объявления переменных. Значение константы не обязательно должно быть известно на момент компиляции, но оно должно присваиваться строго один раз. Это значит, что вы можете использовать константу для обозначения значения, определяемого единожды, но используемого во многих местах.

```
var myVariable = 42  
myVariable = 50  
let myConstant = 42
```





# Приведение типов

Приведение типов - это способ проверить тип экземпляра и/или способ обращения к экземпляру или же конвертация одного типа к другому.

Приведение типов в Swift реализуется с помощью операторов `is` и `as`. Эти два оператора предоставляют простой и выразительный способ проверки типа значения или преобразование значения к другому типу.

Так же можно преобразовывать типы данных через следующую конструкцию:  
`нужный_тип_данных(конвертируемый_тип_данных)`



```
let myInt = 10
let myDouble = 10.0
```

```
// Convert Double to Int
let myInt2 = Int(myDouble)
```

```
let sum = myInt + myInt2
```

```
// Convert Int to String
let str2 = String(myInt)
```

# Оператор ветвления if-else

В самой простой своей форме инструкция if имеет всего одно условие if. Эта инструкция выполняет установленные инструкции только в случае, когда условие true:

```
var temperatureInFahrenheit = 30
if temperatureInFahrenheit <= 32 {
    print ("It's very cold. Consider wearing a scarf.")
}
```

Инструкция if может предусматривать еще один дополнительный набор инструкций в ветке известной как оговорка else, которая нужна в случае, если условие false. Эти инструкции указываются через ключевое слово else:

```
if temperatureInFahrenheit <= 32 {
    print("It's very cold. Consider wearing a scarf.")
} else {
    print("It's not that cold. Wear a t-shirt.")
}
```



# Оператор ветвления switch-case

Инструкция switch подразумевает наличие какого-то значения, которое сравнивается с несколькими возможными шаблонами. После того как значение совпало с каким-либо шаблоном, выполняется код, соответствующий ответвлению этого шаблона, и больше сравнений уже не происходит. Switch представляет собой альтернативу инструкции if, отвечающей нескольким потенциальным значениям.

В самой простой форме в инструкции switch значение сравнивается с одним или более значений того же типа:



```
let someCharacter: Character = "z"
switch someCharacter {
case "a":
    print("The first letter of the alphabet")
case "z":
    print("The last letter of the alphabet")
default:
    print("Some other character")
}
// Выведет "The last letter of the alphabet"
```

Swift