

Модуль 2 - урок 1:

Строки

Строки

```
let greeting = "Привет"  
var otherGreeting = "Салют"
```

```
let joke = ""  
    Вопрос: Зачем курица переходит дорогу?  
    Ответ: Чтобы попасть на другую сторону!  
    ""  
print(joke)
```

```
Вопрос: Зачем курица переходит дорогу?  
Ответ: Чтобы попасть на другую сторону!
```

Строки. Основа

Управляющие последовательности

```
let greeting = "В программировании есть традиция писать \"Здравствуй, мир!\""
```

Управляющие символы	Описание
\"	Двойные кавычки
\\	Обратный слеш
\t	Табуляция
\n	Перевод строки
\r	Возврат каретки в начало строки

Строки. Основы.

Пустые строки

```
var myString = ""  
  
if myString.isEmpty {  
    print("Строка пуста")  
}
```

Строка пуста

Строки. Основы.

Символы

```
let a = "a" // 'a' – строка (String)
let b: Character = "b" // 'b' – символ (Character)
```

Конкатинация (сцепление)

```
let string1 = "Здравствуй"  
let string2 = ", мир!"  
var myString = string1 + string2 // "Здравствуй, мир!"
```

```
myString += " Здравствуй!" // "Здравствуй, мир! Здравствуй!"
```

Интерполяция (вставка)

```
let name = "Костя"  
let age = 30  
print("Имя: \(name). Полных лет: \(age)")
```

```
Имя: Костя. Полных лет: 30
```

Интерполяция

Выражение

```
let a = 4  
let b = 5  
print("Если A равно \({a}), а B равно \({b}), тогда A + B будет равно \({a} + {b})")
```

Если A равно 4, а B равно 5, тогда A + B будет равно 9

Равенство и сравнение строк

```
let month = "Январь"  
let otherMonth = "Январь"  
let lowercaseMonth = "январь"  
  
if month == otherMonth {  
    print("Строки равны")  
}  
  
if month != lowercaseMonth {  
    print("Строки не равны")  
}
```

Строки равны

Строки не равны

Равенство и сравнение строк

Игнорирование регистра

```
let name = "Дядя Стёпа"  
if name.toLowerCase() == "дядя Стёпа".toLowerCase() {  
    print("Имена совпадают")  
}
```

Имена совпадают

Равенство и сравнение строк

Префикс (приставка) и суффикс (окончание)

```
let greeting = "Здравствуй, мир!"  
  
print(greeting.hasPrefix("Здравствуй"))  
print(greeting.hasSuffix("мир!"))  
print(greeting.hasSuffix("Мир!"))
```

```
true  
true  
false
```

Равенство и сравнение строк

Поиск подстрок

```
let greeting = "Привет, Костя! Меня зовут Аня."  
if greeting.contains("Меня зовут") {  
    print("Первое знакомство")  
}
```

Первое знакомство

Равенство и сравнение строк

Проверка длины

```
let name = "Константин Константинопольский"
let count = name.count
let newPassword = "1234"

if newPassword.count < 8 {
    print("Этот пароль слишком короткий. Пароли должны содержать минимум 8 символов.")
}
```

Этот пароль слишком короткий. Пароли должны содержать минимум 8 символов.

Равенство и сравнение строк

Использование switch

```
let character: Character = "з"    // буква

switch character {
case "а", "е", "ё", "и", "о", "у", "ы", "э", "ю", "я":
    print("\(character) – гласная буква.")
case "б", "в", "г", "д", "ж", "з", "й", "л", "м", "н", "р":
    print("\(character) – звонкая согласная.")
case "к", "п", "с", "т", "ф", "х", "ц", "ч", "ш", "щ":
    print("\(character) – глухая согласная.")
default:
    print("\(character) – не буква.")
}
```

з – звонкая согласная.

Юникод

```
let cow = "🐮"  
let credentials = "résumé"  
let myBook = "私の本"  
let writer = "Пушкин"  
print("∞".count)
```

Модуль 2 - урок 2: Функции

Функции

```
tieMyShoes()
```

```
makeBreakfast(food: "яичница-болтунья", drink: "апельсиновый сок")
```

Функции

Определение функции

```
func имяФункции (параметры) -> ВозвращаемыйТип {  
    // Тело функции  
}
```

```
func displayPi() {  
    print("3.1415926535")  
}  
  
displayPi()
```

```
3.1415926535
```

Параметры

```
func triple(value: Int) {  
    let result = value * 3  
    print("Если умножить \(value) на 3, получится \(result).")  
}  
  
triple(value: 10)
```

Если умножить 10 на 3, получится 30.

Параметры

Несколько параметров

```
func multiply(firstNumber: Int, secondNumber: Int) {  
    let result = firstNumber * secondNumber  
    print("Результат: \(result).")  
}
```

```
multiply(firstNumber: 10, secondNumber: 5)
```

Результат: 50.

Возвращаемые значения

```
func multiply(firstNumber: Int, secondNumber: Int) -> Int {  
    let result = firstNumber * secondNumber  
    return result  
}
```

Возвращаемые значения

```
func multiply(firstNumber: Int, secondNumber: Int) -> Int {  
    let result = firstNumber * secondNumber  
    return result  
}
```

```
let myResult = multiply(firstNumber: 10, secondNumber: 5)  
print("10 * 5 будет \ (myResult)")
```

```
print("10 * 5 будет \ (multiply(firstNumber: 10, secondNumber: 5))")
```

Параметры функции

```
func sayHello(firstName: String) {  
    print("Привет, \(firstName)!")  
}
```

```
sayHello(firstName: "Аня")
```

Параметры функции

```
func sayHello(to: String, and: String) {  
    print("Привет, \ (to) и \ (and)")  
}  
  
sayHello(to: "Иван", and: "Сергей")
```


Параметры

Имена параметров

```
func sayHello(to person: String, and anotherPerson: String) {  
    print("Привет, \(person) и \(anotherPerson)")  
}  
  
sayHello(to: "Иван", and: "Сергей")
```

Параметры

Пропуск аргументов

```
func add(_ firstNumber: Int, to secondNumber: Int) -> Int {  
    return firstNumber + secondNumber  
}
```

```
let total = add(14, to: 6)
```

Значения по умолчанию для параметров

```
func display(teamName: String, score: Int = 0) {  
    print("\(teamName): \(score)")  
}
```

```
display(teamName: "Факел", score: 100)  
display(teamName: "Факел")
```

Факел: 100

Факел: 0