



ЛОГИЧЕСКИЕ ТИПЫ И ОПЕРАЦИИ С НИМИ

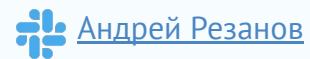


АНДРЕЙ РЕЗАНОВ



АНДРЕЙ РЕЗАНОВ

Руководитель мобильной разработки, МТС





План занятия

1. [Логические типы](#)
2. [Операции сравнения](#)
3. [Логические операции](#)



Вспоминаем прошлые занятия

- Как называется возможность в Xcode изучать код, не создавая проект для разработки?



Вспоминаем прошлые занятия

- Как инициализировать переменную, константу?
- Какая между ними разница?



Вспоминаем прошлые занятия

- Какие типы данных вы изучили на прошлом занятии?
- Есть ли вопросы по домашнему заданию к прошлой лекции?

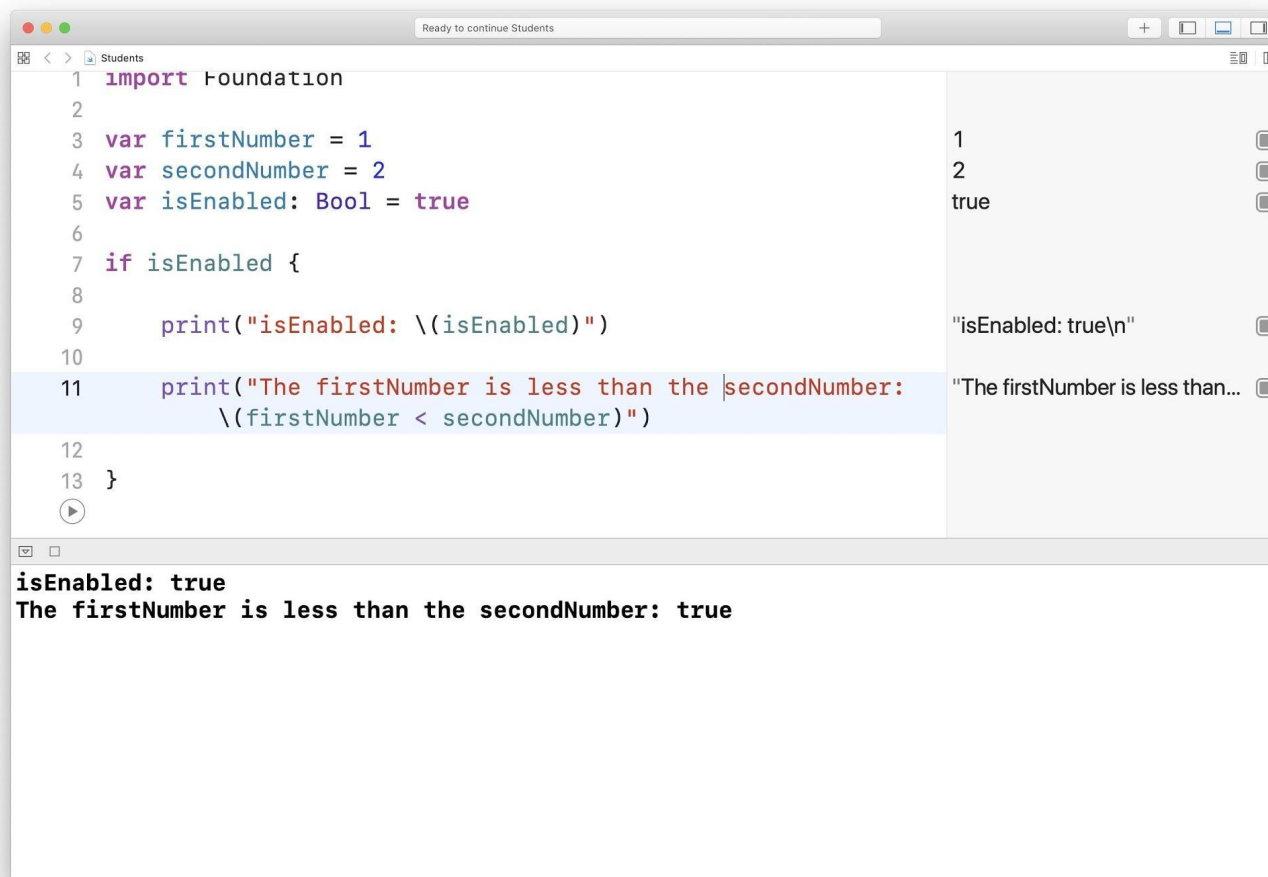
Логические значения (Bool)

Тип `Bool` представляет логическое значение `true` (истина) или `false` (ложь). Объект `Bool` может находиться в двух состояниях:

```
var isEnabled: Bool = true  
isEnabled = false
```

Объекты типа `Bool` нередко являются результатом условных выражений, т.е. таких выражений, которые представляют некоторое условие, и в зависимости от истинности условия возвращают `true` или `false`: `true` — если условие истинно и `false` — если условие ложно.

Пример



```
1 import Foundation
2
3 var firstNumber = 1
4 var secondNumber = 2
5 var isEnabled: Bool = true
6
7 if isEnabled {
8
9     print("isEnabled: \(isEnabled)")
10
11     print("The firstNumber is less than the secondNumber:
12         \(firstNumber < secondNumber)")
13 }
```

1
2
true

"isEnabled: true\n"

"The firstNumber is less than..."

isEnabled: true
The firstNumber is less than the secondNumber: true

Операции сравнения

Сравнивая значения, мы получим либо истину, либо ложь. Следуя этому высказыванию, мы понимаем, что результат сравнения – это переменная типа `Bool`.

- `==` Операция равенства. Если левое и правое значение равны, вернет `true`, иначе `false`:

```
var a = 20
var b = 20
var c = a == b
print(c) // true, так как a равно b
var d = 8
c = a == d
print(c) // false, так как a не равно d
```

Операции сравнения

- `!=` Операция неравенства. Если левое и правое значение не равны, вернет `true`, иначе `false`:

```
var a = 20
var b = 20
var c = a != b
print(c) // false, так как a равно b
var d = 8
c = a != d
print(c) // true, так как a не равно d
```

- `>` Если левое значение больше правого, вернет `true`, иначе `false`:

```
var a = 25
var b = 20
var c = a > b
print(c) // true, так как a больше чем b
var d = 8
c = d > a
print(c) // false, так как d меньше чем a
```

Операции сравнения

- < Если правое значение больше левого, вернет `true`, иначе `false`:

```
var a = 25
var b = 20
var c = a < b
print(c) // false, так как a больше чем b
var d = 8
c = d < a
print(c) // true, так как d меньше чем a
```

- >= Если левое значение больше либо равно правому, вернет `true`, иначе `false`:

```
var a = 20
var b = 20
var c = a >= b
print(c) // true, так как a равно b
var d = 8
c = d >= a
print(c) // false, так как d меньше чем a
```

Операции сравнения

- `<=` Если правое значение больше либо равно левому, вернет `true`, иначе `false`:

```
var a = 25
var b = 20
var c = a <= b
print(c) // false, так как a больше чем b
var d = 9
c = d <= a
print(c) // true, так как d меньше a
```

Логические операции

Логические операции позволяют преобразовывать логические значения. Данные операции работают с переменными типа `Bool` и возвращают тот же тип.

- `!` Отрицание (НЕ). Меняет значение `Bool` переменной на обратное:

```
var isEnabled: Bool = true  
var result = !isEnabled // false
```

Логические операции

- `&&` Логическое Умножение (И). Если оба значения будут равны `true`, тогда вернет `true`, иначе `false`:

```
let isEnabled: Bool = true
let isAlive = true
let result = isEnabled && isAlive
// true - так как оба операнда равны true
```

```
let a: Bool = true
let b: Bool = false
let c: Bool = true
let d = a && b && c // false, так как b = false
```

Логические операции

- `||` Логическое сложение (ИЛИ). Если хотя бы одно значение равно `true`, вернет `true`, иначе `false`:

```
var isEnabled: Bool = true
var isAlive = false
isEnabled || isAlive
// true, так как isEnabled равен true
```

```
var a: Bool = true
var b: Bool = false
var c: Bool = false
a || b || c // true, так как a = true
```

Логические операции

Также логические операции можно объединить с операциями сравнения:

```
let a = 10
let b = 12
let c = a > 8 && b < 10
let d = a > 8 || b < 10
print(c)// false
print(d)// true
```




Итоги

- Узнали о логических типах и как их использовать;
- Посмотрели операции сравнения (равно-не равно, больше-меньше и другие);
- Попробовали логические операции из булевой алгебры.



Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задаём в чате Slack!
- Задачи можно сдавать по частям.
- Зачёт по домашней работе проставляется после того, как приняты **все задачи**.



Задавайте вопросы и напишите отзыв о лекции!

АНДРЕЙ РЕЗАНОВ

 [Андрей Резанов](#)