

# Оператор присваивания

Оператор присваивания ( $a = b$ ) инициализирует или изменяет значение переменной `a` на значение `b`:

```
let b = 10
var a = 5
a = b
// теперь a равно 10
```

Если левая часть выражения является кортежем с несколькими значениями, его элементам можно присвоить сразу несколько констант или переменных:

```
let (x, y) = (1, 2)
// x равно 1, а y равно 2
```

В отличие от `C` и `Objective-C` оператор присваивания в `Swift` не может возвращать значение. К примеру, следующее выражение недопустимо:

```
if x = y {
    // это неверно, так как x = y не возвращает никакого значения
}
```

Эта особенность не позволяет разработчику спутать оператор присваивания (`=`) с оператором проверки на равенство (`==`). Благодаря тому, что выражения типа `if x = y` некорректны, подобные ошибки при программировании на `Swift` не произойдут.

# Арифметические операторы

Язык *Swift* поддерживает четыре стандартных арифметических оператора для всех числовых типов:

- сложение (+)
- вычитание (-)
- умножение (\*)
- деление (/)

```
1 + 2 // равно 3
5 - 3 // равно 2
2 * 3 // равно 6
10.0 / 2.5 // равно 4.0
```

В отличие от *C* и *Objective-C* арифметические операторы *Swift* по умолчанию не допускают переполнения типа. Контроль переполнения типа включается в *Swift* специальными операторами (например, *a &+ b*). Подробнее см. в главе Операторы переполнения.

Оператор сложения служит также для конкатенации, или же склейки, строковых значений (тип *String*):

```
"hello, " + "world" // равно "hello, world"
```

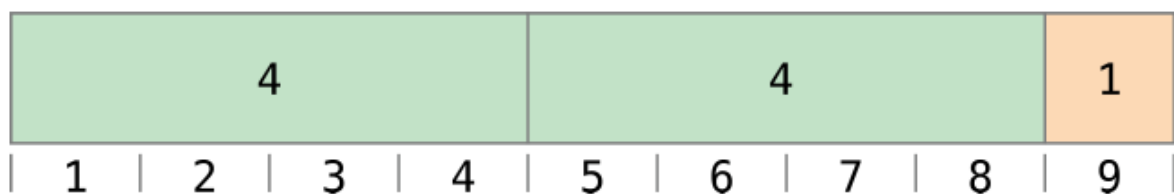
## Оператор целочисленного деления

Оператор целочисленного деления (*a % b*) показывает, какое количество *b* помещается внутри *a*, и возвращает остаток деления *a* на *b*.

*Заметка*

Оператор целочисленного деления (%) в некоторых языках называется оператором деления по модулю. Однако учитывая его действие над отрицательными числами в Swift, этот оператор, строго говоря, выполняет деление с остатком, а не по модулю.

Оператор целочисленного деления работает следующим образом. Для вычисления выражения  $9 \% 4$  сначала определяется, сколько четверок содержится в девятке:



В одной девятке содержатся две четверки, а остатком будет 1 (выделено оранжевым цветом).

На языке Swift это записывается так:

```
9 % 4 // равно 1
```

Чтобы получить результат деления  $a \% b$ , оператор % вычисляет следующее выражение и возвращает остаток:

$$a = (b \times \text{множитель}) + \text{остаток}$$

где множитель показывает, сколько целых раз  $b$  содержится в  $a$ .

Подставляя в это выражение 9 и 4, получим:

$$9 = (4 \times 2) + 1$$

Точно так же рассчитывается остаток, когда  $a$  отрицательно:

$$-9 \% 4 // \text{равно } -1$$

Подставляя в наше выражение -9 и 4, получим:

$$-9 = (4 \times -2) + -1$$

причем остаток будет равен -1.

Если  $b$  отрицательно, его знак отбрасывается. Это означает, что выражения  $a \% b$  и  $a \% -b$  всегда будут давать одинаковый результат.