



ЧИСЛА

ЧИСЛА

```
val a = 3
```

```
val b = -5L
```

```
val c = 10.3
```

```
val d: Float = 10.3
```

```
val e = 1.03e1
```

ЧИСЛОВЫЕ ОПЕРАЦИИ

```
val x = -5
```

```
val y = 3
```

```
x + y    // Сложение
```

```
x - y    // Вычитание
```

```
x * y    // Умножение
```

```
x / y    // Деление (нацело)
```

```
x % y    // Остаток от деления
```

```
-x        // Противоположное число
```

БИНАРНЫЕ ОПЕРАЦИИ

```
val x = 0xF  
val y = 0xA1
```

```
x >> y // побитовый сдвиг вправо  
x << y // побитовый сдвиг влево  
x | y  // побитовое ИЛИ  
x & y  // побитовое И  
x ^ y  // побитовое искл. ИЛИ  
~x     // побитовая инверсия
```

ПОРЯДОК ОПЕРАЦИЙ

```
val x = -5
```

```
val y = 3
```

```
(x * y) + (2 * 3)
```

```
x * (y + 2) * 3
```

```
x * y + 2 * 3
```

- Можем определять порядок выполнения с помощью скобок, изменяя приоритет

ПОРЯДОК ОПЕРАЦИЙ

1. СИМВОЛЫ
2. |
3. ^
4. &
5. = !
6. < >
7. :
8. + -
9. * / %
10. все остальные знаки

БОЛЬШИЕ ЧИСЛА

```
val x = BigInt(10)

val y = BigDecimal(10)

val z = BigInt("10000000000000000000000000000000")
```

```
val x = BigInt(10)

val y = BigDecimal(10)

val z = BigInt("10000000000000000000000000000000")
```

```
val x = BigInt(10)

val y = BigDecimal(10)

val z = BigInt("10000000000000000000000000000000")
```

- Есть ссылочные типы больших чисел

БОЛЬШИЕ ЧИСЛА

```
val x = BigInt(10)
```

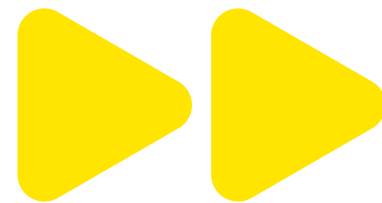
```
x pow 100
```

```
val y = BigInt(10)
```

```
x gcd y
```

- Операции над ними не дают переполнения

**Мы изучили
числа**



**В следующем разделе
изучим строки**