

Целые числа (int)

```
In [12]: num = 13
         print(num)

         num = 0
         print(num)

         num = -10
         print(num)

13
0
-10

In [9]: num = 100_000_000
         print(num)

100000000
```

Встроенная функция **type**

```
In [6]: num = 13
         print(type(num))

<class 'int'>
```

Вещественные числа (float)

```
In [8]: num = 13.4
         print(num)

         num = 0.0
         print(num)

         num = -10.2
         print(num)

13.4
0.0
-10.2

In [9]: num = 100_000.000_001
         print(num)

100000.000001

In [5]: # 1.5 умножить на 10 в степени 2
         num = 1.5e2
         print(num)

150.0
```

Конвертация типов:

```
In [1]: num = 150.2
         print(type(num))

<class 'float'>

In [2]: num = int(num)
         print(num, type(num))

         num = float(num)
         print(num, type(num))

150 <class 'int'>
150.0 <class 'float'>
```

Комплексные числа (complex)

```
In [4]: num = 14 + 1j

         print(type(num))
         print(num.real)
         print(num.imag)

<class 'complex'>
14.0
1.0
```

Модуль **decimal** для работы с вещественными числами с фиксированной точностью

Модуль **fractions** для работы с рациональными числами

Основные операции с числами

Сложение:

```
In [7]: 1 + 1

Out[7]: 2

In [8]: 1 + 2.0

Out[8]: 3.0
```

Вычитание:

```
In [19]: 10 - 1

Out[19]: 9

In [20]: 4.2 - 1

Out[20]: 3.2
```

Деление:

```
In [10]: 10 / 2

Out[10]: 5.0
```

Делить на 0 нельзя:

```
In [11]: 2 / 0

-----
ZeroDivisionError                                Traceback (most recent call last)
<ipython-input-11-ae0c5d243292> in <module>()
----> 1 2 / 0

ZeroDivisionError: division by zero
```

Умножение:

```
In [12]: 4 * 5.25

Out[12]: 21.0
```

Возведение числа в степень:

```
In [21]: 2 ** 4

Out[21]: 16
```

Целочисленное деление:

```
In [14]: 10 // 3

Out[14]: 3
```

Остаток от деления:

```
In [15]: 10 % 3

Out[15]: 1
```

Порядок операций в выражениях с числами:

```
In [7]: print(10 * 3 + 3)
         print(10 * (3 + 3))

33
60
```

Побитовые операции:

```
In [18]: x = 4
         y = 3

         print("Побитовое или:", x | y)
         print("Побитовое исключающее или:", x ^ y)
         print("Побитовое и:", x & y)
         print("Битовый сдвиг влево:", x << 3)
         print("Битовый сдвиг вправо:", x >> 1)
         print("Инверсия битов:", ~x)

Побитовое или: 7
Побитовое исключающее или: 7
Побитовое и: 0
Битовый сдвиг влево: 32
Битовый сдвиг вправо: 2
Инверсия битов: -5
```

Задача: найти расстояние между двумя точками в декартовых координатах.

 distance

Решение:

```
In [8]: x1, y1 = 0, 0
         x2 = 3
         y2 = 4

         distance = ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 0.5
         print(distance)

5.0
```

Меняем местами значения 2-х переменных:

```
In [22]: a = 100
         b = 200
         print(a, b)

         a, b = b, a
         print(a, b)

100 200
200 100
```

Вместо x, y = 0, 0

```
In [25]: x = y = 0
         x += 1

         print(x)
         print(y)

1
0
```

Но нужно помнить об отличии изменяемых (mutable) и неизменяемых (immutable) типов:

```
In [26]: x = y = []
         x.append(1)
         x.append(2)

         print(x)
         print(y)

[1, 2]
[1, 2]
```

В этом видео:

- Поговорили о базовых численных типах в Python
- Рассмотрели математические операции с численными типами
- Узнали о конвертации типов
- Затронули тему изменяемых и неизменяемых объектов в Python

```
In [ ]:
```