

Типы данных

Для хранения данных в языке можно объявлять переменные. Каждая переменная имеет тип данных, который определяет поведение этой переменной при различных операциях над ней.

Python имеет несколько встроенных в язык типов данных:

Численные типы

`int` - целое число (e.g. 0, 1, 100, -99...)

Объявление:

```
a = 5
b = int('10')
print(a, b)          # будет выведено 5, 10
```

Операции и преобразования:

```
a = 4
b = 5

print(a + b)          # сложение          будет выведено 9
print(a - b)          # вычитание         будет выведено -1
print(a * b)          # умножение         будет выведено 20
print(a / b)          # деление           будет выведено 0.8
print(a // b)         # целочисленное деление будет выведено 0
print(a % b)          # остаток от деления будет выведено 4
print(a ** b)         # возведение в степень будет выведено 1024

flt = float(a)        # приведение к дроби   = 4.0
string = str(a)       # приведение к строке  = "4"
```

`float` - дробное число (e.g. 0.2, 15.4, -100.01...)

Объявление:

```
a = 9.5
b = float(10)
print(a, b)          # будет выведено 9.5, 10.0
```

Операции и преобразования:

```
a = 1.21
b = 3.355

print(a + b)          # сложение          будет выведено 4.565000
print(a - b)          # вычитание         будет выведено -2.145
print(a * b)          # умножение         будет выведено 4.05955
print(a / b)          # деление           будет выведено 0.36065573770491804
print(a // b)         # целочисленное деление будет выведено 0.0
print(a % b)          # остаток от деления будет выведено 1.21
print(a ** b)         # возведение в степень будет выведено 1.895592221929019

# округление до n знака после запятой
print(round(a, 1))    #          будет выведено 1.2
```

```
print(round(a, 0))    #          будет выведено 1.0

integer = int(a)      # приведение к целому ч.    = 1
string = str(a)       # приведение к строке      = "1.21"
```

Строки

`str` - строка, текст, символ (e.g. *"hello"*, *"cat"*, *"DOG"*, *"a"*, *"123"*)

Объявление:

```
a = "Cat"
b = str(123)
print(a, b)          # будет выведено Cat 123
```

Между объявлением строки с использованием одинарных кавычек `' '` и двойных `" "` нет принципиальной разницы. Однако, если строка содержит апострофы (например, *Don't give up, skeleton!*), то объявить ее можно будет только используя двойные кавычки:

```
string = "Don't give up, skeleton!"
```

Сложение строк:

```
a = "Hello"
b = "world"

# сложение строк
print(a + b)          # будет выведено Helloworld
print(a + " " + b)    # будет выведено Hello world
print(a + ", " + b + "!") # будет выведено Hello, world!
```

Сложение строк эквивалентно их последовательному "склеиванию" между собой.

Умножение строк:

```
a = "Fox"
b = 4

print(a * b)          # будет выведено FoxFoxFoxFox
```

Умножение строки на число N повторяет эту строку N раз.

Про другие методы и функции, применимые к строкам см. в разделе *"Строки"*, *"Форматирование"*.
Про *коллекции* и *логический* тип данных см. в соответствующих разделах.