

## Типы данных

Переменная хранит данные одного из типов данных. В Python существует множество различных типов данных. В данном случае рассмотрим только самые базовые типы: bool, int, float, complex и str.

### *Целые числа*

Тип int представляет целое число, например, 1, 4, 8, 50. Пример

```
age = 21
print("Возраст:", age) # Возраст: 21

count = 15
print("Количество:", count) # Количество: 15
```

По умолчанию стандартные числа расцениваются как числа в десятичной системе. Но Python также поддерживает числа в двоичной, восьмеричной и шестнадцатеричной системах.

Для указания, что число представляет двоичную систему, перед числом ставится префикс 0b:

```
a = 0b11
b = 0b1011
c = 0b100001
print(a) # 3 в десятичной системе
print(b) # 11 в десятичной системе
print(c) # 33 в десятичной системе
```

Для указания, что число представляет восьмеричную систему, перед числом ставится префикс 0o:

```
a = 0o7
b = 0o11
c = 0o17
print(a) # 7 в десятичной системе
print(b) # 9 в десятичной системе
print(c) # 15 в десятичной системе
```

Для указания, что число представляет шестнадцатеричную систему, перед числом ставится префикс 0x:

```
a = 0x0A
b = 0xFF
c = 0xA1
print(a) # 10 в десятичной системе
print(b) # 255 в десятичной системе
print(c) # 161 в десятичной системе
```

Стоит отметить, что в какой-бы системе мы не передали число в функцию print для вывода на консоль, оно по умолчанию будет выводиться в десятичной системе.

### *Дробные числа*

Тип float представляет число с плавающей точкой, например, 1.2 или 34.76. В качестве разделителя целой и дробной частей используется точка.

```
height = 1.68
pi = 3.14
weight = 68.
print(height) # 1.68
print(pi)     # 3.14
print(weight) # 68.0
```

Число с плавающей точкой можно определять в экспоненциальной записи:

```
x = 3.9e3
print(x) # 3900.0
```

```
x = 3.9e-3
print(x) # 0.0039
```

Число float может иметь только 18 значимых символов. Так, в данном случае используются только два символа - 3.9. И если число слишком велико или слишком мало, то мы можем записывать число в подобной нотации, используя экспоненту. Число после экспоненты указывает степень числа 10, на которое надо умножить основное число - 3.9.

### *Комплексные числа*

Тип complex представляет комплексные числа в формате вещественная\_часть+мнимая\_частьj - после мнимой части указывается суффикс j

```
complexNumber = 1+2j
print(complexNumber) # (1+2j)
```

### *Строки*

Тип str представляет строки. Строка представляет последовательность символов, заключенную в одинарные или двойные кавычки, например "hello" и 'hello'. В Python 3.x строки представляют набор символов в кодировке Unicode

```
message = "Hello World!"
print(message) # Hello World!
```

```
name = 'Tom'
print(name) # Tom
```

При этом если строка имеет много символов, ее можем разбить ее на части и разместить их на разных строках кода. В этом случае вся строка заключается в круглые скобки, а ее отдельные части - в кавычки:

```
text = ("Laudate omnes gentes laudate "
        "Magnificat in secula ")
print(text)
```

Если же мы хотим определить многострочный текст, то такой текст заключается в тройные двойные или одинарные кавычки:

```
text = "Laudate omnes gentes laudate  
Magnificat in secula  
Et anima mea laudate  
Magnificat in secula  
"  
print(text)
```

При использовании тройных одинарных кавычек не стоит путать их с комментариями: если текст в тройных одинарных кавычках присваивается переменной, то это строка, а не комментарий.

#### *Логические значения*

Тип `bool` представляет два логических значения: `True` (верно, истина) или `False` (неверно, ложь). Значение `True` служит для того, чтобы показать, что что-то истинно. Тогда как значение `False`, наоборот, показывает, что что-то ложно.

Источник: <https://metanit.com/python/tutorial/2.2.php>