

# Справочные материалы по языку Python

## Некоторые типы данных

Название	Функция преобразования	Возможные значения
Целые	<code>int(s)</code>	целое число
Вещественные	<code>float(s)</code>	действительное число
Логический (булев)	<code>bool(s)</code>	False   True
Строка	<code>str(...)</code>	строка
	<code>chr(n)</code>	символ (с кодом n в Unicode); обратное преобразование – <code>ord(c)</code>
Список	<code>list(s)</code> [список элементов]	пронумерованная последовательность элементов (нумерация с нуля)
Множество	<code>set(s)</code> {список элементов}	неупорядоченная последовательность уникальных элементов
Словарь	<code>dict(s)</code> {ключ: значение, ...}	множество пар «ключ–значение»

Другие типы: `complex`, `tuple`, `frozenset`

## Операции

арифметические операции		операции сравнения		логические операции	
+	сложение	<	меньше	and	«и» (конъюнкция)
-	вычитание	>	больше	or	«или» (дизъюнкция)
*	умножение	==	равно	not	«не» (отрицание)
/	деление	!=	не равно		
//	деление нацело	<=	меньше либо равно		
%	остаток от деления	>=	больше либо равно		
**	возведение в степень	in	вхождение		

## Работа со строками

+	конкатенация (склейка)	строка[n]	выбор n-го символа
*	повторение	строка[n:m]	выбор символов с n до m (не включая m)

## Некоторые функции

### Встроенные

<code>abs(x)</code>	$ x $	модуль	<code>max(L)</code>	максимальный элемент
<code>pow(x, y)</code>	$x^y$	возведение в степень	<code>min(L)</code>	минимальный элемент
<code>round(x)</code>		округление	<code>sorted(L)</code>	сортированный список
<code>round(x, n)</code>		округление до n знаков	<code>list(reversed(L))</code>	в обратном порядке
			<code>len(s)</code>	длина строки, списка и т. д.

### Модули `math`, `random`

<code>math.sqrt(x)</code>	$\sqrt{x}$	квадратный корень	<code>random.random()</code>	случайное число от 0.0 до 1.0
<code>math.sin(x)</code>	$\sin x$	синус*	<code>random.sample(список, n)</code>	выбор n случайных элементов из списка
<code>math.cos(x)</code>	$\cos x$	косинус*		
<code>math.trunc(x)</code>		целая часть		

\* Аргумент тригонометрических функций выражается в радианах ( $\pi$  рад =  $180^\circ$ )

## Некоторые операторы

### Присваивание

переменная = выражение

```
d = b*b-4*a*c
```

Переменная d получает значение, равное результату заданного выражения.

переменная += выражение  
(также -= \*= /= //= %=)

```
n += 1
```

Значение переменной n увеличивается на единицу.

### Ввод

```
input('приглашение')
```

```
a = int(input('a='))
```

Компьютер выводит на экран «a=», ожидает ввода значения с клавиатуры, затем преобразует его в целое и присваивает переменной a.

### Вывод

```
print(выражения)
```

```
print('a=', a)
```

На экран выводится «a=», через пробел – значение переменной a, затем переводится строка.

```
print(выражения,  
sep='разделитель',  
end='завершение')
```

```
print(a+b,c, sep='\t', end='')
```

На экран выводится результат вычисления выражения a + b, через табуляцию – c. Строка не переводится.

### Условный оператор

```
if условие :  
    операторы  
elif условие :  
    операторы  
else :  
    операторы
```

```
if d >= 0 :  
    x1 = (-b-sd)/(2*a)  
    x2 = (-b+sd)/(2*a)  
    print(x1, x2)  
elif d == 0 :  
    x1 = -b/(2*a)  
    print(x1)  
else  
    print('Корней нет')
```

Если переменная d больше либо равна нулю, выполняются вычисления x1 и x2 и вывод их значений. Иначе, если d равна нулю, вычисляется и выводится значение x1. В противном случае выводится сообщение «Корней нет».

### Цикл с условием (в Python есть только цикл с предусловием)

```
while условие :  
    операторы
```

```
while z > e :  
    z = z/n  
    n += 1
```

Пока значение переменной z больше e, повторяются действия, предписанные операторами тела цикла.

### Цикл с параметром (цикл со счетчиком)

```
for перем in список :  
    операторы
```

```
for n in range(1,11) :  
    a *= n  
    print(n,a,sep='\t')
```

Для каждого значения переменной n, изменяющейся от 1 до 10, повторяются действия, предписанные операторами тела цикла.

```
for перем in range(b, e, s) :  
    операторы
```

```
for i in range(0,-9,-1) :  
    print(i,x)  
    x /= 2
```

Для каждого значения переменной i, изменяющейся от 0 до -8, повторяются действия тела цикла.

### Прерывание цикла

```
break
```

```
while true :  
    n *= 2  
    if n > x : break
```

Выполнение цикла будет прервано, когда значение переменной n превысит x.

```
continue
```

```
for i in range(10) :  
    if i == 5 : continue  
    print(i, i**3)
```

Для значения переменной i равного 5 выполнение оператора print будет пропущено.