



ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON



Язык программирования

Язык программирования устанавливает набор правил, определяющих внешний вид программы и действия, которые выполнит исполнитель под её управлением

Гвидо ван Россум приступил к созданию языка **Python** в декабре 1989 года в центре математики и информатики в Нидерландах. Ван Россум является основным автором Python и продолжал выполнять центральную роль в принятии решений относительно развития языка вплоть до 12 июля 2018 года



Команда вывода

```
print ( 'Виват, Лицей!' )
```

По традиции, начавшейся в 1978 г. с примера из книги Брайана Кернигана и Дениса Ритчи «Язык программирования Си», первая программа на любом языке программирования должна просто выводить на экран приветствие миру

Команда вывода

Команда `print()` предназначена для **вывода данных**.

Она является функцией, аргументы которой – выводимые значения

Текст, который нужно вывести на экран, заключают в кавычки.
Кавычки могут быть одинарными или двойными

```
print('Виват, Лицей!')
```

```
print("Виват, Лицей!")
```



Кавычки на клавише «Э» !

Команда вывода

```
print('Значение')
```



```
graph TD; A["print('Значение')"] --> B["Если 'Значение', то текст не анализируется"]; A --> C["Если Значение без кавычек, то производится анализ данных"];
```

Если `'Значение'` , то текст не анализируется

Если `Значение` без кавычек, то производится анализ данных

Текст выводится на экран в том виде, в котором он записан. Его можно записать как на русском, так и на любом другом языке

```
print('2 + 2 * 2 =')
```

2 + 2 * 2 =

Если в скобках написать арифметическое выражение, то сначала вычисляется его значение, а затем выводится результат

```
print(2 + 2 * 2)
```

6

Команда вывода

| Пример кода | Пояснение | Вывод |
|------------------------------------------------------------|----------------------------------------------------------------------------------------------|-----------------------------------------------------|
| <pre>print ('Привет') print ('мир')</pre> | После того, как команда вывода была выполнена, курсор переводится на следующую строку | <pre>D:\progi\python\ Привет мир</pre> |
| <pre>print ('Привет', 'мир')</pre> | В одной команде можно выводить несколько значений, которые отделяются друг от друга запятыми | <pre>D:\progi\python\ Привет мир</pre> |
| <pre>print ('Привет') print () print ('мир')</pre> | Если функция используется без аргументов, то будет выведена пустая строка | <pre>D:\progi\python\ Привет мир</pre> |

Аргументы команды print()

| Название | Описание |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <code>sep</code> | задает текст, который будет служить разделителем при выводе различных значений в одной команде <code>print()</code> По умолчанию – пробел |
| <code>end</code> | задает текст, который будет выведен на экран после вывода всех значений одной команды <code>print()</code> По умолчанию – перевод строки |



перевод строки задается как `'\n'`. Если использовать `sep = '\n'`, то каждое выводимое значение одной команды `print()` будет расположено в новой строке

Пример

```
D:\progi\python\lesson1\venv\Scripts\python.exe D:/progi/python/1
```

```
Привет! Я компьютер!!!!
```

```
Я умею выполнять программы!
```

```
Сегодня ты написал свою первую программу и я ее выполнил.
```

```
Сейчас на экране - ее результат.
```

```
Ура! Ура! Ура!
```

```
print(' ура , ура , ура! , sep=' ' )
```



Что выведет программа

Понятие типа данных

Информацию, представленную в виде, пригодном для обработки на компьютере, называют **данными**

Переменная в программировании – это именованная ячейка памяти, хранящая значение переменной

Тип данных определяет способ хранения данных в памяти компьютера, диапазон возможных значений данных и операции, которые с этим типом данных можно выполнять

Имена переменных

Могут включать

- латинские буквы (A-Z, a-z)
- знак подчеркивания _
- цифры 0-9

НЕ могут включать

- русские буквы
- пробелы
- скобки, знаки +, =, !, ? и др.



Имя не может начинаться с цифры!

Числовые типы данных

Язык Python поддерживает работу с целыми и вещественными числами

```
a = 2  
b_1 = 7.5  
radius = 12.0
```

Тип переменной язык Python определяет по ее значению

Переменная `a` будет определена как целая, а переменные `b_1` и `radius` – как вещественные

```
a, b_1, radius = 2, 7.5, 12.0
```

Задавать значения можно и таким образом

Оператор присваивания

Оператор присваивания предназначен для того, чтобы

- задавать значения переменным
- вычислять значения арифметического выражения, результат вычисления которого будет записан как значение переменной

```
x = 9
x1 = 3.5
a_1 = 20 * (x + x1) - 32
y = 3
chastnoe = x / y
y = 7 + 2 * y ** 3
```

Примеры

| Математические операции | Запись в Python |
|-------------------------|-----------------|
| + (сложение) | + |
| - (вычитание) | - |
| · (умножение) | * |
| : (деление) | / |
| ВОЗВЕДЕНИЕ В СТЕПЕНЬ | ** |

ЗНАКИ
МАТЕМАТИЧЕСКИХ
ДЕЙСТВИЙ
(целые и
вещественные числа)

| Математические операции | Запись в Python |
|-------------------------|-----------------|
| целочисленное деление | // |
| нахождение остатка | % |

**ЗНАКИ
МАТЕМАТИЧЕСКИХ
ДЕЙСТВИЙ**
(только целые числа)

Результат операций // и %

| a | b | a // b |
|-----|----|--------|
| 17 | 3 | 5 |
| -17 | 3 | -6 |
| 17 | -3 | -6 |
| -17 | -3 | 5 |

При выполнении операции // знак результата определяется также как в математике:

- положительный, если исходные числа одного знака
- отрицательный, если – разного

Значение результата округляется до ближайшего целого в меньшую сторону

Результат операций // и %

| a | b | a // b | a % b |
|-----|----|--------|-------|
| 17 | 3 | 5 | 2 |
| -17 | 3 | -6 | 1 |
| 17 | -3 | -6 | -1 |
| -17 | -3 | 5 | -2 |

Результат операции % может быть отрицательным, хотя в математике под остатком понимают неотрицательное число.

Если остаток не равен нулю, то знак числа, которое является результатом операции %, определяется знаком делителя

$$a \% b = a - (a // b) * b$$

Порядок выполнения операций

- вычисление выражений в скобках
- умножение, деление, //, % слева направо
- сложение и вычитание слева направо

$$z = \frac{5ac + 3(c - d)}{ab} (b - c)$$



z = (5 * a * c + 3 * (c - d)) * (b - c) / (a * b) ;

Операции над строками

То, что заключается в кавычки, называется строками.

```
stroka = "Виват, Лицей!"  
print(stroka)
```

Виват, Лицей!

Строки можно складывать.

```
stroka = "Виват, "  
print(stroka + "Лицей!")
```

Виват, Лицей!

И можно умножать на число.

```
stroka = "Виват, Лицей! "  
print(stroka * 3)
```

Виват, Лицей! Виват, Лицей! Виват, Лицей!

F-строки

Иногда бывает удобно вставить значение переменной в строку. Перед строкой ставится буква `f`, а переменная пишется в фигурных скобках `{ }`.

```
name = "пользователь"
```

$$x = 17$$

```
stroka = f"Виват, {name} {x}!"
```

```
print (stroka)
```

Виват, пользователь 17!

Можно выводить вещественные числа с заданным числом знаков после запятой.

$$y = 10 / 3$$

```
print(f"10 / 3 = {y}")
```

```
print(f"10 / 3 = {y:.4f}")
```

$$10 / 3 = 3.3333333333333335$$
$$10 / 3 = 3.3333$$

Преобразование типов

Для преобразования строк в числа и наоборот используются следующие функции:

- **int()** — преобразует строку (или вещественное число) в целое число.
- **float()** — преобразует строку (или целое число) в вещественное число;
- **str()** — преобразует значения (в общем случае не только числовые) в строки.

```
a = "1"
b = "2"
print(a + b)
a = int(a)
b = int(b)
print(a + b)
```

12

3

Ввод данных

Команда `input()` предназначена для **ввода данных**. Для того, чтобы значение переменной вводилось с клавиатуры нужно присвоить ей значение функции `input()`

```
a = input()
```

Переменная `a` после ввода значения будет содержать текст (тот который мы набрали с помощью клавиатуры)

```
y = input()  
print('y =', y)
```

```
D:\progi\python\
```

```
Привет
```

```
y = Привет
```

Ввод данных

```
a = input()
```

Даже, если мы набирали цифры, то получим набор символов, с которыми нельзя выполнять арифметические действия

```
y = input()  
print('y =', y + 5)
```

```
D:\progi\python\lesson1\venv\Scripts\python.exe D:/progi/python/
```

```
25
```

```
Traceback (most recent call last):
```

```
File "D:\progi\python\lesson1\main.py", line 2, in <module>
```

```
    print('y =', y + 5)
```

```
TypeError: can only concatenate str (not "int") to str
```

Ввод данных

При вводе нужно указать в какой числовой тип данных мы хотим преобразовать вводимый текст

```
y = int (input ())  
print ('y =', y + 5)
```

```
D:\progi\python\T  
25  
y = 30
```

В качестве параметра функции `input()` можно задать текст, который будет служить подсказкой при вводе данных

```
x = int (input ('Введите целое число '))  
y = float (input ('Введите дробное число '))  
print ('x =', x)  
print ('y =', y)
```

```
D:\progi\python\lesson1\venv  
Введите целое число 51  
Введите дробное число 5.75  
x = 51  
y = 5.75
```

УСЛОВИЯ ЗАДАЧ

```
main.py x
1  # Задан квадрат с длиной стороны a.
2  # Найти его площадь и периметр
3  a = float(input('a = '))
4  S = a ** 2
5  P = 4 * a
6  print('площадь -', S)
7  print('периметр -', P)
8
```

Текст после символов #
считается комментарием

Условия обязательно
вставляются в текст
программы как
комментарий

main.py ×

main1.py ×

```
1  # Даны x, y, z. Напишем программу для вычисления значения выражения
2  #  $a = (2x + y - z) / (3 + x^2)$ 
3  x = float(input('x = '))
4  y = float(input('y = '))
5  z = float(input('z = '))
6  a = (2 * x + y - z) / (3 + x * x)
7  print('a =', a)
8
```

Задание 1а

Даны вещественные x , y , z . Написать программу для вычисления значения выражения $a = \frac{2x+3y-z}{3+x^2}$

```
# Даны x, y, z. Напишем программу для вычисления
# значения выражения a = (2x+3y-z) / (3+x^2)

x = float(input('x = '))
y = float(input('y = '))
z = float(input('z = '))

a = (2 * x + 3 * y - z) / (3 + x * x)
print('a =', a)
```

```
D:\progi\python\lesson1\
x = 2
y = 3
z = 1
a = 1.7142857142857142
```

Математические функции

Для подключения библиотеки используют команду `import math`

| Запись на Python | Описание |
|----------------------------|-------------------------------------------------------------------------------|
| <code>math.fabs(x)</code> | Находит модуль числа x |
| <code>math.sqrt(x)</code> | Находит корень квадратный из числа x . Результат — число вещественного типа |
| <code>math.ceil(x)</code> | Находит ближайшее целое число, которое не меньше, чем значение x . |
| <code>math.floor(x)</code> | Находит ближайшее целое число, которое не больше, чем x . |
| <code>math.pi</code> | $\pi = 3,1415926\dots$ |
| <code>math.sin(x)</code> | Вычисляет синус числа x . Число x задается в радианах |
| <code>math.cos(x)</code> | Вычисляет косинус числа x . Число x задается в радианах |
| <code>math.tan(x)</code> | Вычисляет тангенс числа x . Число x задается в радианах |

Задание 16

Заданы значения переменных x и y . Вычислить значение выражения $b = \frac{\sqrt{x}+4}{|y|-2} \sin x$

```
# Даны x, y. Напишем программу для вычисления значения выражения  
# b=(√x+4) / (|y|-2) sin(x)
```

```
import math
```

```
x = float(input('x = '))
```

```
y = float(input('y = '))
```

```
b = (math.sqrt(x) + 4) / (math.fabs(y) - 2) * math.sin(x)
```

```
print('b =', b)
```

```
D:\progi\python\lesson1\
```

```
x = 13
```

```
y = 7.7
```

```
b = 0.5606319197973305
```

Задание 2

Пусть таймер показывает время только в секундах. Напишем программу, которая переведет время в минуты и секунды

```
#Пусть таймер показывает время только в секундах.  
#Напишем программу, которая переведет время в минуты и секунды  
  
c = int(input('c = '))  
  
#минуты  
m = c // 60  
#секунды  
s = c % 60  
  
print(m, s, sep=':')
```

D:\progi\python\lesson1\

c = 137

2:17

Задание 2

С начала месяца прошло t часов. Определить какое сейчас число и сколько часов прошло с начала суток.



В каждом месяце 30 дней



Дни, месяцы и года исчисляются с 1



Не должно получиться ситуации:
0 число 5 часов 15 минут



Не должно получиться ситуации:
2 число 49 часов 75 минут

Задание 3

Дано целое число a . Написать программу, которая получит a^3 и a^{10} за четыре операции умножения (другие операции использовать нельзя)

```
# Дано число a. Написать программу,  
# которая получит a3 и a10 за четыре операции умножения  
# (другие операции использовать нельзя)
```

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

```
a2 = a * a
```

```
a3 = a2 * a
```

```
a5 = a3 * a2
```

```
a10 = a5 * a5
```

```
print(a3, a10, sep='; ')
```

Для сокращения количества умножений введем дополнительные переменные, которые будут хранить промежуточные значения

```
D:\progi\python\
```

```
a = 5
```

```
125; 9765625
```

Задание 4

Задан квадрат с длиной стороны a . Найти его площадь и периметр

```
# Задан квадрат с длиной стороны a.  
# Найти его площадь и периметр
```

```
a = float(input('a = '))
```

```
S = a ** 2
```

```
P = 4 * a
```

```
print('площадь -', S)
```

```
print('периметр -', P)
```

```
D:\progi\python\1
```

```
a = 2
```

```
площадь - 4.0
```

```
периметр - 8.0
```