

Общие правила синтаксиса Python

Команды, из которых состоит программа, должны быть написаны по определённым правилам. Компьютер должен однозначно распознавать команды. Он должен понимать, где заканчивается одна команда, а где начинается следующая. Для этого необходимо соблюдать следующие правила.

<i>Правило</i>	<i>Правильно</i>	<i>Неправильно</i>
Правило порядка Команды будут выполняться по очереди, если они написаны ровно друг под другом.	<pre>a = "Hello, " b = "world!" print(a+b)</pre>	<pre>a = "Hello, " b = "world!" print(a+b)</pre>
Правило начала Каждая новая команда пишется с новой строки.	<pre>a = "Hello, " b = "world!" print(a) print(b)</pre>	<pre>a = "Hello, " b = "world!" print(a)print(b)</pre>
Правило аккуратности Строчные (маленькие) буквы нельзя заменять заглавными (большими).	<pre>a = "Hello, " b = "world!" print(a+b)</pre>	<pre>a = "Hello, " b = "world!" PrInt(A+b)</pre>

Печать информации на экран





Команда:

```
print()
```

Описание:

Функция `print()` нужна для печати на экран того, что находятся внутри скобок. Это могут быть строки, числа, переменные и др. Чтобы напечатать какую-либо строку, её нужно записать в кавычках — `" "` или `' '` (одинаковые с обеих сторон) и вставить в `print()`. Чтобы напечатать число или переменную, нужно просто вставить её в `print()`. Если необходимо напечатать несколько аргументов (строку, число и переменную), то достаточно разделить их внутри скобок запятой.

Как использовать:





	   
<pre>1 #напечатать можно строку/число 2 print("Пример 1") 3 print("Сколько тебе лет?") 4 print("Мне",9) 5 6 #или целое выражение 7 print("Пример 2") 8 print("Что получится, если сложить 111 и 999?") 9 a = 111 10 b = 999 11 sum = a+b 12 print(sum)</pre>	<p>Пример 1 Сколько тебе лет? Мне 9</p> <p>Пример 2 Что получится, если сложить 111 и 999? 1110</p>

Переменные

Описание:

Переменная — это элемент данных, имеющий имя. Переменная нужна для хранения данных, которые могут меняться в программе. Чтобы использовать переменную, ей нужно придумать имя и задать начальное значение. Оператор присваивания “=” задаёт начальное значение переменной. В переменных могут храниться числа, строки и др. В имени переменной можно использовать только латинские (английские) символы. Только после того, как переменную создали и присвоили ей начальное значение, с ней можно работать.

Как использовать:

	   
<pre>1 #в переменных могут храниться числа 2 print("Пример 1") 3 a = 9 4 b = 10 5 sum = a + b 6 print("Сумма",sum) 7 8 #в переменных могут храниться и строки 9 print("Пример 2") 10 name = "Robert" 11 hi = "Hello" 12 s = name + ", " + hi 13 print(s) 14 name = "Anna" 15 s = hi + ", " + name 16 print(s)</pre>	<pre>Пример 1 Сумма 19 Пример 2 Robert, Hello Hello, Anna</pre>

Ввод данных в программу

Команда:

```
input()
```

Описание:

Функция `input()` нужна для передачи в программу данных от пользователя. Внутри скобок указывается сообщение для пользователя, призывающее ввести какую-либо информацию. Введённые данные записываются в переменную и выполняется следующая команда. Данная функция всегда в переменную записывает **строку**.

Как использовать:

```
1 #узнать у пользователя имя и
2 поприветствовать его
3 print("Пример 1")
4 name = input("Привет! Как тебя зовут?")
5 print("Рад познакомиться,",name,"!")
6
7 #в переменную всегда записывается строка
8 print("Пример 2")
9 num1 = input("Введите первое число:")
10 num2 = input("Введите второе число:")
11 print("Сумма чисел равна",num1+num2)
```



```
Пример 1
Привет! Как тебя зовут?
>>> Кристина
Рад познакомиться, Кристина !
Пример 2
Введите первое число:
>>> 12
Введите второе число:
>>> 35
Сумма чисел равна 1235
```

Типы данных

Описание:

Над разными типами данных можно выполнять разные операции. Например, над числами можно выполнять все арифметические операции, строки можно печатать или склеивать. Поэтому, чтобы компьютер однозначно понимал программиста, для каждого типа данных было придумано своё имя. Определить, к какому типу относится переменная, можно с помощью функции `type()`. Ниже приведены некоторые типы данных.

Тип данных	Что можно делать	Что нельзя делать
Числа: <ul style="list-style-type: none"> ● int — целое число, ● float — дробное число. 	Выполнять все арифметические и логические операции: <ul style="list-style-type: none"> ● сложение (+), ● умножение (*), ● деление (/), ● целочисленное деление (//), ● остаток от деления (%), ● возведение в степень (**), ● сравнение . 	Пытаться выполнить арифметические операции с другими типами данных: <ul style="list-style-type: none"> ● складывать число и строку, ● возводить строку в степень, ● аналогично с другими арифм. операциями.
Строки: <ul style="list-style-type: none"> ☐ str — строка. 	<ul style="list-style-type: none"> ☐ печатать на экран, ☐ склеивать (+), ☐ дублировать (*), ☐ сравнивать. 	<ul style="list-style-type: none"> ☐ выполнять арифм. операции над двумя строками.
Логические переменные: <ul style="list-style-type: none"> ◆ bool. 	<ul style="list-style-type: none"> ◆ сравнивать. 	



```
1 a = "Привет!" #тип str
2 b = 123 #тип int
3 c = 2.5 #тип float
4 d = True #тип bool
5 name = input("Как тебя зовут?") #тип str
6
7 print("Переменная a:",a)
8 print("Переменная b:",b)
9 print("Переменная c:",c)
10 print("Переменная d:",d)
11 print("Переменная name:",name)
```

Как тебя зовут?

>>> Пётр

Переменная a: Привет!

Переменная b: 123

Переменная c: 2.5

Переменная d: True

Переменная name: Пётр

Определение типа данных

Команда:

```
type()
```

Описание:

Функция `type()` нужна для определения типа данных. Чтобы узнать какого типа переменная — нужно указать её внутри скобок.

Как использовать:

```
1 #узнать типы переменных
2 a = "Привет!"
3 b = 123
4 c = 2.5
5 d = True
6 num1 = input("Введите число:")
7 print("Типы переменных:")
8 print("a:", a, "-", type(a))
9 print("b:", b, "-", type(b))
10 print("c:", c, "-", type(c))
11 print("d:", d, "-", type(d))
12 print("num1:", num1, "-", type(num1))
```



Введите число:

```
>>> 12
```

Типы переменных:

a: Привет! - <class 'str'>

b: 123 - <class 'int'>

c: 2.5 - <class 'float'>

d: True - <class 'bool'>

num1: 12 - <class 'str'>

Перевод из строки в число

Команда:

```
int()
```

Описание:

Функция `int()` преобразует строку в целое число. Чтобы превратить последовательность цифр (`str`) в число, нужно указать её внутри скобок. Нужно помнить, что слова в число данная функция превратить не может.

Как использовать:

```
1 #преобразовать в число и найти сумму
2 b = 3.5 #тип float!
3 a = input("Введите число:") #получили str!
4
5 #найдем сумму чисел a и b
6 #для суммы a нужно преобразовать в int
7 a = int(a)
8 sum = a + b #тип float
9 print("Сумма:",sum)
10
11 # из float в int
12 sum1 = int(sum)
13 print("Сумма:",sum1)
```



Введите число:

```
>>> 5
```

```
Сумма: 8.5
```

```
Сумма: 8
```


Перевод из числа в строку

Команда:

```
str()
```

Описание:

Функция `str()` преобразует любой тип данных в строку. Чтобы это сделать — нужно указать внутри скобок данные, которые необходимо преобразовать в строку.

Как использовать:

```
1 #найти периметр прямоугольника
2 dl = int(input("Введите ширину в см:"))
3 sh = int(input("Введите длину в см:"))
4 S = 2*(dl + sh)
5 S = str(S)
6 message = "Периметр прямоугольника "+ S +" см"
7 print(message)
```



Введите ширину в см:

```
>>> 15
```

Введите длину в см:

```
>>> 12
```

Периметр прямоугольника 54 см