



<https://dfedorov.spb.ru>

Типы данных в Python являются классами (аналог модуля)

```
>>> help(str)
>>> help(0)
```

Вызов методов производится по аналогии с вызовом функций из модуля.

Отличие методов: в качестве первого входного аргумента требуется строка (если строковый тип данных/класс).

Полная форма для строковых методов



```
>>> str.capitalize('hello')
```

The image shows a terminal window with a grey title bar and three window control buttons. Inside the terminal, the command `>>> str.capitalize('hello')` is entered. The prompt `>>>` is in red, `str.capitalize` is in purple, and `('hello')` is in green. Three dashed arrows point from explanatory text to parts of the command: a black arrow from the bottom left to `str`, a purple arrow from the bottom right to `capitalize`, and a green arrow from the top right to `('hello')`.

Первый аргумент
для строковых
методов - строка

Имя класса
(типа данных)

Метод возвращает копию строки, в
которой первый символ – в верхнем
регистре, остальные – в нижнем

Полная форма для строковых методов

```
>>> str.center('hello', 20)
```

Метод возвращает строку, центрированную по заданной длине. По умолчанию заполняется пробелами

Аргумент задает длину строки

имя_строкового_типа_данных.имя_метода (строка, аргументы...)

Сокращенная форма для вызова методов

```
>>> help(str.lower)
```

```
>>> 'hello'.capitalize()
```

Вынесли из аргумента

строковое_выражение . **имя_метода** (аргументы...)

A terminal window with a grey title bar containing three white circular window control buttons. The main area is white and contains a Python code snippet and its output. The code is on the first line, and the output is on the second line.

```
>>> ("TTA" + "G"*3).count("T")
```

```
2
```

```
>>> "spec".startswith("a")
```

```
False
```

```
>>> s = " \n MMMMMM \n "
```

```
>>> s.strip()
```

```
>>> "Hello".swapcase()
```



```
>>> s = '<[ у озера >]'
```

```
>>> s.strip('<>')[ ']
```



```
>>> x = 'www.python.org'
```

```
>>> x.strip('wrgo.')
```

```
'python'
```

```
>>> x.find('th')
```

```
6
```

```
>>> x.index('th')
```

```
6
```

```
>>> x.replace('p', 'j')
```

```
'www.jython.org'
```

```
>>>
```

В случае
отсутствия
возвращает **-1**

В случае
отсутствия
возникает
ошибка

```
>>> s = '2019.py'
```

```
>>> s.endswith('.py')
```

```
True
```

```
>>>
```

```
help(s.endswith)
```



```
"ПРИВЕТ".swapcase().endswith("т")
```

```
"привет".endswith("т")
```

True

Можно выполнять локальные арифметические действия:

```
>>> a = 5
```

```
>>> b = 10
```

```
>>> f'Пять плюс десять равняется {a + b}, а не {2 * (a + b)}.'
```

```
'Пять плюс десять равняется 15, а не 30.'
```

Популярные строковые методы

s - переменная, содержащая ссылку на произвольную строку

```
s.upper()
```

```
s.lower()
```

```
s.title()
```

```
s.find("вет", 2, 6)
```

```
s.count("e", 1, 5)
```

```
s.replace("a", "b")
```

```
s.isalpha()
```

```
s.isdigit()
```

```
s.isupper()
```

```
s.islower()
```

```
s.istitle()
```

```
s.isspace()
```

[Официальная документация](#)



<https://dfedorov.spb.ru>