

{ EPIC SKILLS }

ПОЛУЧИ ПРАКТИЧЕСКИЕ НАВЫКИ
В СФЕРЕ IT И DIGITAL



ВЕБ-ДИЗАЙН



ВЕБ-РАЗРАБОТКА



ИНТЕРНЕТ-МАРКЕТИНГ

Урок 4

Python: основы

программирования

Дмитрий Федоров



Старший преподаватель кафедры вычислительных систем и программирования СПбГЭУ



ПОЛИТЕХ

Санкт-Петербургский
политехнический университет
Петра Великого



**Высшая школа
менеджмента**

Санкт-Петербургский
государственный университет

Резюме

 @haku_ru



{ Python для анализа данных }



@init_python



@init_python

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')
```

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 icon with a gray title bar and three white window control buttons (minimize, maximize, close) on the left. The main area is white and contains Python code and its output.

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

```
2
```



```
>>> name = "Иван"
>>> age = 6
>>> "Привет," + name + "! " + "В следующем году тебе будет " + str(age+1) + " лет."
```

```
>>> "Привет, {0}! В следующем году тебе будет {1} лет.".format(name, age+1)
```

```
>>> "{0}, {1}, {2}".format(10, 12.3, "str")  
'10, 12.3, str'
```

```
>>> "{model}, {color}".format(color='red', model='lada')  
'lada, red'
```

```
>>> "{color}, {0}".format(2015, color="red")  
'red, 2015'
```

```
>>> '{num:.3f}'.format(num=2/3)
'0.667'
```

f-строки или форматированные строковые литералы (*Formatted String Literals*)

```
>>> num = 2/3
>>> num
0.6666666666666666
>>> f'{num:.3f}'
'0.667'
```

Python 3.6 и выше

Можно выполнять локальные арифметические действия:

Python 3.6 и выше

```
>>> a = 5
>>> b = 10
>>> f'Пять плюс десять равняется {a + b}, а не {2 * (a + b)}.'
'Пять плюс десять равняется 15, а не 30.'
```

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

```
False
```

```
>>> s = " \n M M M M M M \n "
```

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

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

```
>>> s = '<[ у озера >]'
```

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

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

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

```
True
```

```
>>>
```

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



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

```
"привет".endswith("т")
```

True

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](#)

```
>>> 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

Специальные строковые методы

Объединение двух строк преобразуется в вызов спец. метода `__add__`

```
>>> 'TT' + 'rr'
>>> 'TT'.__add__('rr')
```

Как будет выглядеть полная форма?



```
>>> str.__add__("TT", 'rr')
```

Упражнение

s = "У лукоморья 123 дуб зеленый 456"

- 1) Определить, встречается ли в строке буква 'я'. Вывести на экран ее позицию (индекс) в строке.
- 2) Определить, сколько раз в строке встречается буква 'у'.
- 3) Определить, состоит ли строка из букв, ЕСЛИ нет, ТО вывести строку в верхнем регистре.
- 4) Определить длину строки. ЕСЛИ длина строки превышает 4 символа, ТО вывести строку в нижнем регистре.
- 5) Заменить в строке первый символ на 'О'. Результат вывести на экран

Упражнение

Написать функцию, которая на вход принимает два аргумента: строку (s) и целочисленное значение (n).

ЕСЛИ длина строки s превышает число n , ТО функция возвращает строку s в верхнем регистре, ИНАЧЕ возвращается исходная строка s .

{ Python для анализа данных }

python™



@init_python



@init_python