

5



**Python**

## Множества (set)

**Множество** в python - "контейнер", содержащий не повторяющиеся элементы в случайном порядке.

```
>>> a = set()
>>> a
set()
>>> a = set('hello')
>>> a
{'h', 'o', 'l', 'e'}
>>> a = {'a', 'b', 'c', 'd'}
>>> a
{'b', 'c', 'a', 'd'}
>>> a = {i ** 2 for i in range(10)} # генератор множеств
>>> a
{0, 1, 4, 81, 64, 9, 16, 49, 25, 36}
>>> a = {} # А так нельзя!
>>> type(a)
<class 'dict'>
```

# Строки

```
1 print('Isn\'t', she said.)
2 # "Isn't", she said.
3 print("Hello\nworld!")
4 # Hello
5 # world!
6 print('C:\some\name')
7 # C:\some
8 # ame
9 print(r'C:\some\name')
10 # C:\Windows\Users
11
```

# Строки

```
1 print("""\
2 Usage: thingy [OPTIONS]
3     -h                Display this usage message
4     -H hostname       Hostname to connect to
5 """)
6
7 # Usage: thingy [OPTIONS]
8 #     -h                Display this usage message
9 #     -H hostname       Hostname to connect to
10
```

# Конкатенация строк

```
1  str1 = 'Python' + 'Python' + 'Python'
2  print(str1)
3  # PythonPythonPython
4
5  str2 = "Python"*3
6  print(str2)
7  # PythonPythonPython
8
```

# Строки: срезы

```
1 word = 'Python'
2
3 assert word[0] == 'P'
4 assert word[5] == 'n'
5 assert word[-1] == 'n'
6 assert word[-2] == 'o'
7 assert word[-6] == 'P'
8 assert word[0:2] == 'Py'
9 assert word[2:5] == 'tho'
10 assert word[:2] + word[2:] == 'Python'
11 assert word[:4] + word[4:] == 'Python'
12 assert word[:2] == 'Py'
13 assert word[4:] == 'on'
14 assert word[-2:] == 'on'
15 assert word[4:42] == 'on'
16 assert word[42:] == ''
```

# Методы

**format()**

**split()**

**startswith(str)**

# Задача

Написать функцию, которая принимает строку и символ.

Возвращает количество заданных символов в строке, учитывая регистр.

```
count_char("Hello world", "o") == 2
```

```
count_char("Hello world", "d") == 1
```

```
count_char("Hello world", "z") == 0
```



## Списки: копирование, отражение

```
1 list1 = [1, 2, 3, 4, 5]
2 list2 = list1[:]
3
4 assert list1 == list2
5 assert list1 is not list2
6
7 list3 = list1[::-1]
8 assert list3 == [5, 4, 3, 2, 1]
9
```

# Списки: методы

**Метод** - это функция класса, которая первым своим параметром принимает объект этого класса и записывается через точку после переменной, которая ссылается на объект.

**.append** - добавляет элемент в конец списка, возвращает None

**.pop** - удаляет последний элемент списка и возвращает его

**.sort** - сортирует список и возвращает None

```
1 my_list = [1, 3, 5, 4, 2]
2
3 my_list.append(6)
4 assert my_list == [1, 3, 5, 4, 2, 6]
5
6 assert my_list.append(37) == None
7 assert my_list == [1, 3, 5, 4, 2, 6, 37]
8 assert my_list.pop() == 37
9 assert my_list == [1, 3, 5, 4, 2, 6]
10 assert my_list.sort() == None
11 assert my_list == [1, 2, 3, 4, 5, 6]
```