

Основные методы и функции по работе со списками

l.append(x) – добавляет элемент x в конец списка l

```
l = ["red", "green", "blue"]
```

```
l.append("yellow")
```

```
# ["red", "green", "blue", "yellow"]
```

l.insert(index, x) – добавляет элемент x в список по индексу index из списка l

```
l = [1, 2, 3, 4]
```

```
l.insert(0, 6)
```

```
# [6, 1, 2, 3, 4]
```

l.remove(x) – удаляет элемент x из списка l. Удаляется только первое вхождение элемента. Если элемент не найден, генерирует исключение `ValueError`

```
l = [4, 9, 16, 25, 9]
```

```
l.remove(9)
```

```
# [4, 16, 25, 9]
```

l.pop(index) – удаляет из списка l и возвращает элемент по индексу index. Если индекс не передан, то просто удаляет последний элемент.

```
l = [1, 2, 3, 4, 5]
```

```
l.pop(0)
```

```
# [2, 3, 4, 5]
```

`l.index(x)` - возвращает индекс элемента `x`. Если элемент не найден, генерирует исключение `ValueError`

```
l = ["One", "Two", "three"]
```

```
l.index("Two")
```

```
# 2
```

`l.reverse()` - расставляет все элементы в списке `l` в обратном порядке

```
l = [1, 2, 3, 4, 5]
```

```
l.reverse()
```

```
# [5, 4, 3, 2, 1]
```

`l.clear()` - удаление всех элементов из списка `l`

```
l = [1, 2, 3, 4, 5]
```

```
l.clear()
```

```
# []
```

`len(l)` - возвращает длину списка `l`

```
l = [True, False, 2.1, -5, "hello"]
```

```
# 5
```

Основные методы и функции по работе со словарями

`d.items()` - возвращает пары (ключ, значение) словаря `d`.

```
d = {'a': 1, 'b': 2, 'c': 3}
```

```
d.items()
```

```
# dict_items([('a', 1), ('b', 2), ('c', 3)])
```

d.keys() - возвращает ключи в словаре d.

```
d = {'a': 1, 'b': 2, 'c': 3}
```

```
d.keys()
```

```
# dict_keys([('a', 1), ('b', 2), ('c', 3)])
```

d.values() - возвращает значения в словаре d.

```
d = {'a': 1, 'b': 2, 'c': 3}
```

```
d.keys()
```

```
# dict_values([1, 2, 3])
```

d.get(key[, default]) - возвращает значение ключа, но если его нет, не бросает исключение, а возвращает default (по умолчанию None).

```
d = {'a': 1, 'b': 2, 'c': 3}
```

```
print(d.get("a"))
```

```
# 1
```

d.pop(key[, default]) - удаляет ключ из словаря d и возвращает значение. Если ключа нет, возвращает default (по умолчанию бросает исключение).

```
d = {'a': 1, 'b': 2, 'c': 3}
```

```
print(d.pop("a"))
```

```
# {'b': 2, 'c': 3}
```

d.clear() - очищает словарь d.

```
d = {'a': 1, 'b': 2, 'c': 3}
```

```
d.clear()
```

```
#{}  

```

Основные методы и функции по работе со строками

s.find(s1, [start],[end]) - поиск подстроки s1 в строке s. Возвращает номер первого вхождения или -1, start, end – не обязательные параметры, указывающие с какого и по какой номер элемент производить поиск

s.rfind(s1, [start],[end]) – тоже самое, что и find, но поиск производится с конца строки s началу

```
s = "Somebody once told me"
```

```
s.find("o", 8, len(s)-1)
```

```
# 9 (Somebody once told me)
```

s.index(s1, [start],[end]) - поиск подстроки s1 в строке s. Возвращает номер первого вхождения или вызывает ValueError

s.rindex(s1, [start],[end]) - тоже самое, что и index, но поиск производится с конца строки s началу

```
s = "Somebody once told me"
```

```
s.index("m")
```

```
# 2 (Somembody once told me)
```

s.replace(x, y) – замена всех вхождений элемента x в строке s на y

```
s = "не сложно перепрыгнуть"
```

```
s.replace("не", "очень")
```

```
# очень сложно перепрыгнуть
```

s.split(symbol) - разбиение строки по разделителю symbol

```
s = "1*2*3*4*5*6"
```

```
s.split("*")
```

```
# ["1", "2", "3", "4", "5", "6"]
```

symbol.join(s) - сборка строки из списка s с разделителем symbol

```
l = ["r", "g", "b"]
```

```
"_".join(l)
```

```
# "r_g_b"
```

s.upper() - преобразование строки s к верхнему регистру

```
s = "some text"
```

```
# SOME TEXT
```

s.lower() - преобразование строки s к нижнему регистру

```
s = "SOME TEXT"
```

```
# some text
```

s.swapcase() - переводит символы нижнего регистра в верхний, а верхнего – в нижний

```
s = "SomE tEXt"
```

```
# sOMe TexT
```

s.capitalize() - переводит первый символ строки s в верхний регистр, а все остальные в нижний

```
s = "он закричал: «AAAA!»"
```

```
# Он закричал: «aaaa!»
```

`s.title()` - первую букву каждого слова переводит в верхний регистр, а все остальные в нижний

```
s = "rus, ua, kz"
```

```
# Rus, Ua, Kz
```