

Программирование на языке Python. Базовый уровень

Модуль 2. Строки и списки

Списки (занятие 4)



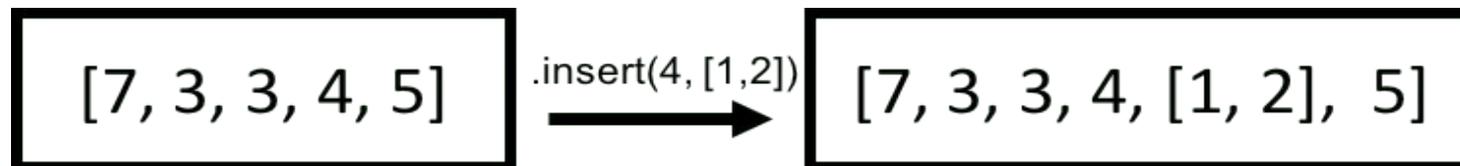
Функции по обработке списков

Наименование	Описание
<code>append(x)</code>	Добавляет элемент <code>x</code> в конец списка
<code>clear()</code>	Очищает список
<code>copy()</code>	Возвращает копию списка
<code>count(x)</code>	Возвращает количество элементов со значением <code>x</code> , входящих в список
<code>extend(L)</code>	Расширяет список через добавление в него всех элементов списка <code>L</code>
<code>index(x, [start [, end]])</code>	Возвращает индекс в списке первого элемента со значением <code>x</code> (при этом поиск ведется от <code>start</code> до <code>end</code>). Если <code>start</code> и <code>end</code> не указываются, то начиная с нулевой позиции

Функции по обработке списков

Наименование	Описание
<code>insert(i, x)</code>	Вставляет в список на i -ую позиция значение x
<code>pop([i])</code>	Удаляет из списка i -ый элемент и возвращает его. Если индекс не указан, удаляется последний элемент
<code>remove(x)</code>	Удаляет первый элемент в списке, имеющий значение x . Возвращает <code>ValueError</code> , если такого элемента не существует
<code>reverse()</code>	Переворачивает список
<code>sort()</code>	Сортирует список
<code>insert(i, x)</code>	Вставляет в список на i -ую позиция значение x

Метод insert()



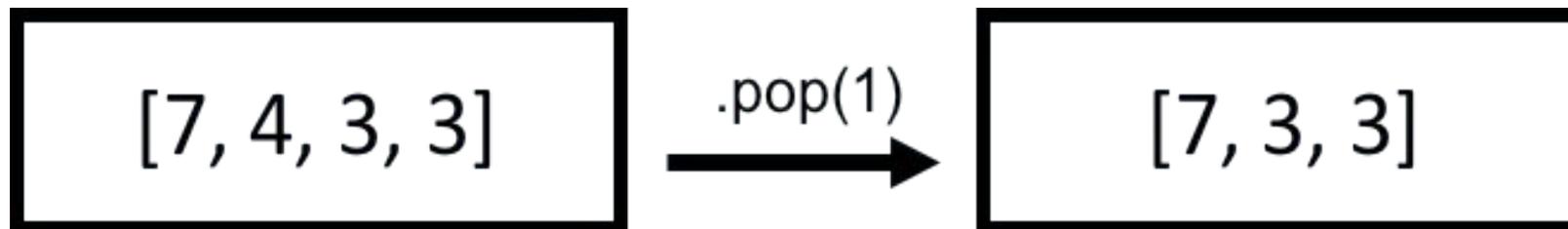
Метод добавляет элемент в список на произвольную позицию. `insert()` принимает в качестве первого аргумента позицию, на которую нужно вставить элемент, а вторым – сам элемент.

Пример.

```
z = [7, 3, 3, 4, 5]
z.insert(4, [1, 2])
print(z)
```

Результат: `[7, 3, 3, 4, [1, 2], 5]`

Метод pop()



Метод удаляет элемент из списка по его индексу.

Пример.

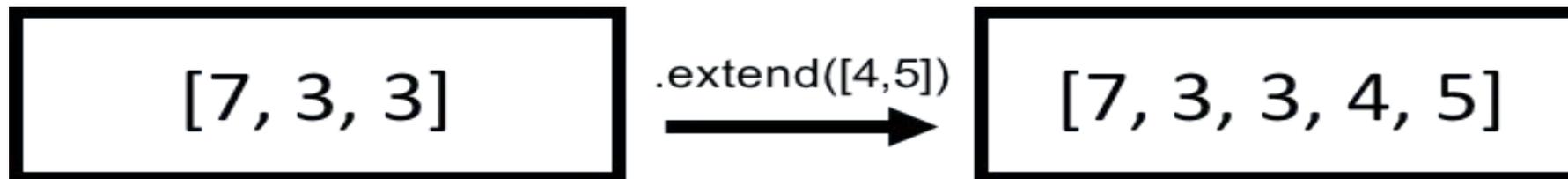
```
numbers = list(range(10))  
print(numbers)  
print(numbers.pop(0))  
print(numbers.pop())
```

Результат: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

0

9

Метод extend()



Т.е., подобно методу `append()`, добавляет элементы, но преимущество метода `extend()` в том, что он также позволяет добавлять списки.

Пример. Добавим `[4, 5]` в конец списка `z`:

```
z = [7, 3, 3]
```

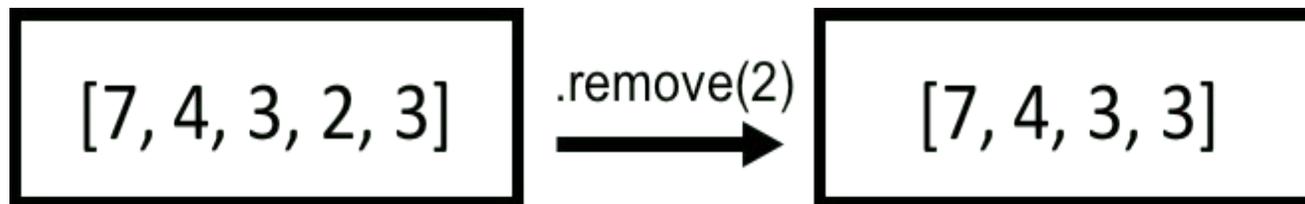
```
z.extend([4,5])
```

```
print(z)
```

```
# обновлённый список:
```

```
[7, 3, 3, 4, 5]
```

Метод `remove()`



Удаляет первое попавшееся вхождение элемента в списке.

Пример.

```
z = [7, 4, 3, 2, 3]
```

```
z.remove(2)
```

```
print(z)
```

Результат:

```
[7, 4, 3, 3]
```

Для удаления элемента из списка можно использовать ключевое слово **del** с названием списка после него. Также потребуется передать индекс того элемента, который нужно удалить.

Синтаксис: `del list[index]`.

Также можно выбрать элементы в определенном диапазоне и удалить их с помощью `del`. Для этого нужно передать начальное и конечное значение диапазона.

Синтаксис: `del list[start:stop]`.

Пример.

```
my_list = list(range(7))  
print("Исходный список",  
my_list)
```

Чтобы удалить первый элемент

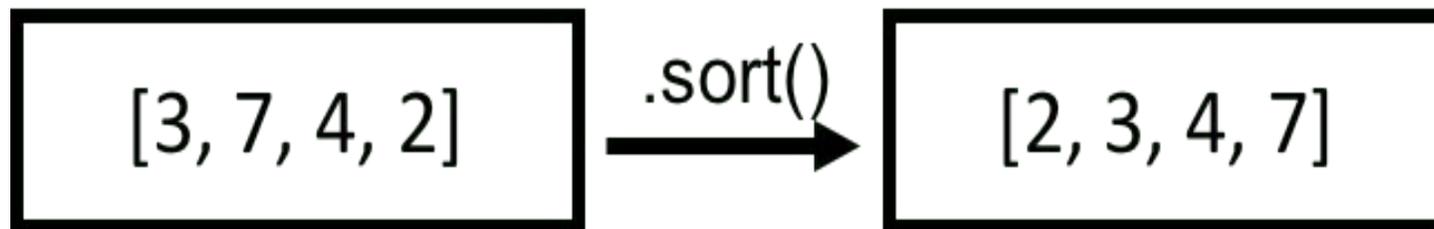
```
del my_list[0]  
print("После удаления первого  
элемента", my_list)
```

Результат:

```
Исходный список [0, 1, 2, 3, 4, 5,  
6]
```

```
После удаления первого
```

Метод `sort()`



Метод `sort()` сортирует список по возрастанию значений его элементов.

Пример.

```
z = [3, 7, 4, 2]
```

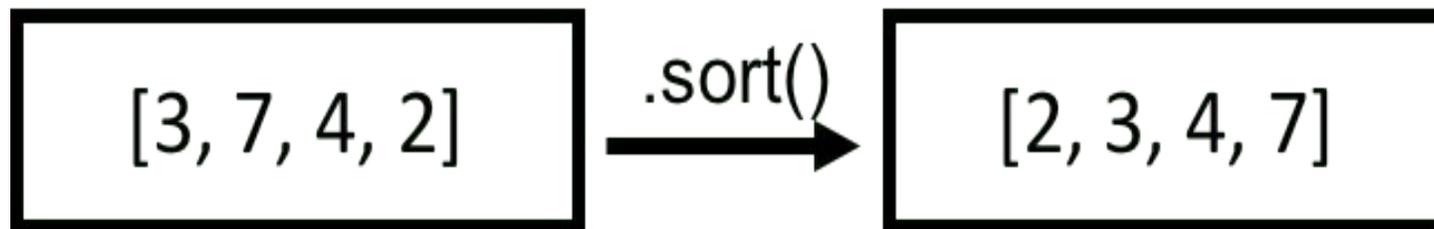
```
z.sort()
```

```
print(z)
```

Результат:

```
[2, 3, 4, 7]
```

Метод reverse()



Метод используется для перестановки элементов в обратном порядке.

Пример.

```
lst = ['м', 'о', 'н', 'и', 'т', 'о', 'р']
```

```
lst.reverse()
```

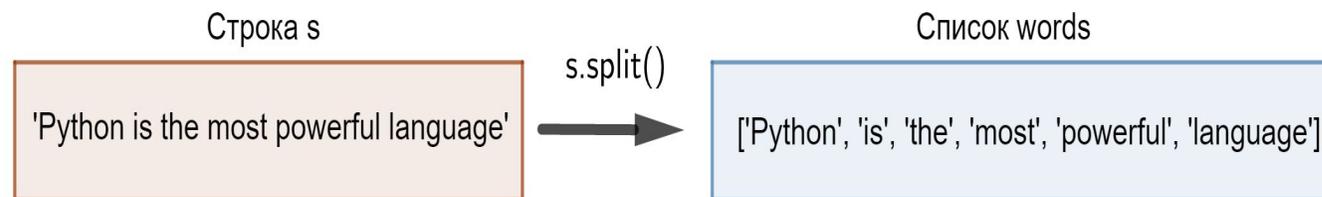
```
print(lst)
```

Результат:

```
['р', 'о', 'т', 'и', 'н', 'о', 'м']
```

Строковый метод `split()`

Метод `split()` разбивает строку на слова, используя в качестве разделителя последовательность пробельных символов.



У метода `split()` есть необязательный параметр, который определяет, какой набор символов будет использоваться в качестве разделителя между элементами списка. Например, вызов метода `split('.')` вернет список, полученный разделением исходной строки по символу '.'.

Строковый метод join()

Метод join() собирает строку из элементов списка, используя в качестве разделителя строку, к которой применяется метод.

Пример.

```
words = ['Python', 'is', 'the', 'most', 'powerful', 'language']  
s = ' '.join(words)  
print(s)
```

Результат:

Python is the most powerful language