

Задачи

На этом занятии мы с вами познакомимся с основными методами, которые есть у списков. Что такое методы вам должно быть уже известно, об этом была речь, когда мы рассматривали методы строк. Кратко напомним, что список – это объект и с этим объектом связаны функции, которые и называются его методами.

Давайте предположим, что у нас имеется список из чисел:

```
a = [1, -54, 3, 23, 43, -45, 0]
```

и мы хотим в конец этого списка добавить еще одно значение. Это можно сделать с помощью метода:

```
a.append(100)
```

Данный метод `append` ничего не возвращает, а меняет сам список. Поэтому писать здесь конструкцию вида: `a = a.append(100)`

не следует, так как это приведет к потере данных. Этим методы списков отличаются от методов строк, когда мы записывали:

```
string="Hello"  
string = string.upper()
```

Здесь метод `upper` возвращает измененную строку, поэтому все работает как и ожидается. А метод `append` ничего не возвращает, и присваивать значение `None` переменной `a` не имеет смысла.

Учитывая, что список может содержать самые разные данные, то в методе `append` можно прописывать не только число, но, например, строку:

```
a.append("hello")
```

тогда в конец списка будет добавлен этот элемент. Или, булево значение:

```
a.append(True)    Или еще один список:    a.append([1,2,3])
```

И так далее. Главное, чтобы было указано одно конкретное значение. Например, вот так работать не будет: `a.append(1,2)`

Если нам нужно вставить новый элемент в произвольную позицию, то используется метод: `a.insert(3, -1000)`

Здесь мы указываем индекс вставляемого элемента и далее значение самого элемента.

Следующий метод `remove` удаляет элемент по значению: `a.remove(True)`
`a.remove('hello')`

Он находит первый подходящий элемент и удаляет его, остальные не трогает. Если же указывается несуществующий элемент:

`a.remove('hello2')` то возникает ошибка. Еще один метод для удаления `a.pop()`

выполняет удаление последнего элемента и при этом, возвращает его значение. В самом списке последний элемент пропадает. То есть, с помощью этого метода можно сохранять удаленный элемент в какой-либо переменной: `end = a.pop()`

Также в этом методе можно указывать индекс удаляемого элемента, например: `a.pop(3)`

Если нам нужно очистить весь список – удалить все элементы, то можно воспользоваться методом:

`a.clear()`

Получим пустой список. Следующий метод

```
a = [1, -54, 3, 23, 43, -45, 0]
c = a.copy()
```

возвращает копию списка. Это эквивалентно конструкции: `c = list(a)`

Следующий метод `count` позволяет найти число элементов с указанным значением:

```
c.count(1)
c.count(-45)
```

Если же нам нужен индекс определенного значения, то для этого используется метод `index`:

```
c.index(-45)
c.index(1)
```

возвратит 0, т.к. берется индекс только первого найденного элемента. Но, мы здесь можем указать стартовое значение для поиска:

```
c.index(1, 1)
```

Здесь поиск будет начинаться с индекса 1, то есть, со второго элемента. Или, так: `c.index(23, 1, 5)`

Ищем число 23 с 1-го индекса и по 5-й не включая его. Если элемент не находится `c.index(23, 1, 3)`

то метод приводит к ошибке. Чтобы этого избежать в своих программах, можно вначале проверить: существует ли такой элемент в нашем срезе:

23 in c[1:3]

и при значении True далее уже определять индекс этого элемента. Следующий метод `c.reverse()`

меняет порядок следования элементов на обратный. Ну и последний метод

`c.sort()` выполняет сортировку элементов списка по возрастанию. Для сортировки по убыванию, следует этот метод записать так:

`c.sort(reverse=True)`

Причем, этот метод работает и со строками: `lst = ["Москва", "Санкт-Петербург"]`
`lst.sort()`

Отличие метода `sort()` от ранее рассмотренной функции `sorted()` в том, что метод меняет сам список, а функция `sorted()` возвращает новый отсортированный список, не меняя начальный.

Метод	Описание
<code>append()</code>	Добавляет элемент в конец списка
<code>insert()</code>	Вставляет элемент в указанное место списка
<code>remove()</code>	Удаляет элемент по значению
<code>pop()</code>	Удаляет последний элемент, либо элемент с указанным индексом
<code>clear()</code>	Очищает список (удаляет все элементы)
<code>copy()</code>	Возвращает копию списка
<code>count()</code>	Возвращает число элементов с указанным значением
<code>index()</code>	Возвращает индекс первого найденного элемента
<code>reverse()</code>	Меняет порядок следования элементов на обратный
<code>sort()</code>	Сортирует элементы списка