

Списки в Python: изменяемость, доступ к элементам

Список представляет собой последовательность в Python. Словарное значение списка — «**некоторое количество связанных элементов** или имен, написанных или напечатанных последовательно».

Мы часто используем списки и в повседневной жизни (например, составление списка покупок), и в программах (например, создание списка простых чисел). Мы составляем списки всякий раз, когда хотим сгруппировать определенные элементы по определенным критериям.

Список в Python предоставляет те же функциональные возможности, позволяющие группировать элементы с возможностью произвольного доступа к каждому элементу, изменения каждого элемента, добавления или удаления элемента из него.

```
list1 = [1,2,3,4,5]
```

```
>>> list1
```

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

```
>>> type(list1)
```

```
<type 'list'>
```

`type()` — это функция в Python, которая возвращает тип объекта переменной.

Создать список в Python очень просто. Вы можете сделать это двумя способами. Первый: заключить в квадратные скобки элементы, которые мы видели в приведенном выше примере. Другой — использовать функцию `list()`.

```
>> list1 = list([1,2,3,4,5])
```

```
>> list1
```

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

```
>>> type(list1)
```

```
<type 'list'>
```

У списков есть интересная особенность: они хранят ссылки на объекты или элементы в нем. Таким образом, при переназначении списка на другую переменную Python изменяет только ссылку, а не создает новый объект списка.

```
>>> id(list1)
```

```
4426491160
```

```
>>> list2 = list1
```

```
>>> id(list2)
```

```
4426491160
```

```
>>> list1
```

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

```
>>> print(list1[0])
```

```
1
```

```
>>> print(list1[2])
```

```
3
```

```
>>> print(list1[1])
```

```
2
```

Бывают ситуации, когда вы знаете, к какому элементу вы хотите получить доступ, но не знаете его индекс. В таких ситуациях вы можете найти этот элемент в соответствующем списке, а функция поиска (функция `index()` в случае списков) возвращает индекс этого элемента.

```
>>> list1
```

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

```
>>> list1.index(2)
```

```
1 // index of 2
```

Изменение элементов списка

```
>>> list1
[1, 2, 3, 4, 5]
>>> list1[0] = 2
>>> list1
[2, 2, 3, 4, 5]
```

Удаление элементов списка

```
>>> list1
[2, 2, 3, 4, 5]
>>> list1.remove(5) #remove element 5
>>> list1
[2, 2, 3, 4]
```

Вставка элемента

`list.insert(index, value)` принимает 2 аргумента. Первый — это индекс, по которому вы хотите добавить элемент, а второй — сам элемент или его значение.

```
>>> list1
[2, 2, 3, 4]
>>> list1.insert(4,89)
>>> list1
[2, 2, 3, 4, 89] #inserting 89 at 4th index.
```

Если передать индекс, который намного больше, чем размер списка, то в этом случае список добавит элемент в конец.

```
>>> list1
[2, 2, 3, 4, 89]
>>> list1.insert(100000,100)
>>> list1
[2, 2, 3, 4, 89, 100]
```

Можно использовать модуль `append()`, чтобы добавить элемент в конец списка.

```
>>> list1 = [2, 2, 3, 4, 89, 100]
>>> list1.append(67)
>>> list1
[2, 2, 3, 4, 89, 100, 67]
```