

Язык программирования

Python

06 Списки и кортежи



Списки и кортежи

Списки и кортежи позволяют нам работать с последовательностями данных.

Список (list) — это тип данных, позволяющий нам работать с набором элементов.

Кортеж (tuple) — это тип данных, очень похожий на список, однако кортеж является неизменяемым, то есть изменения происходят не над имеющимся объектом, а над объектом, созданным на основе имеющегося объекта.

Создание списка

Существует несколько способов создания списка:

- Перечисление в квадратных скобках элементов списка

```
>>> numbers1 = [1, 2, 3]
```

```
>>> numbers1
```

```
[1, 2, 3]
```

- Создание с использованием конструктора

```
>>> numbers2 = list(numbers1)
```

Создать пустой список можно следующим образом:

```
>>> my_list = []
```

```
>>> my_list = list()
```

Создание списка

- Создание списка с помощью функции range

Если нам необходимо создать список, элементы которого будут составлять арифметическую прогрессию, то можно использовать функцию range.

range(stop) создаст последовательность целых чисел от 0 до stop не включительно;

range(start, stop) создаст последовательность целых чисел от start до stop не включительно;

range(start, stop, step) создаст последовательность целых чисел от start до stop не включительно с шагом step.

```
>>> nums = list(range(5)) # Создаст список [0, 1, 2, 3, 4]
```

```
>>> nums = list(range(1, 5)) # Создаст список [1, 2, 3, 4]
```

```
>>> nums = list(range(1, 5, 2)) # Создаст список [1, 3]
```

Элементы списка

Доступ к элементам списка происходит также, как и к элементам строк — по индексам, с помощью квадратных скобок. Индексы начинаются с нуля, а индекс последнего элемента равен длине списка минус 1.

```
>>> numbers = [1, 2, 3, 4, 5]
```

```
>>> numbers[0]
```

```
1
```

```
>>> numbers[4]
```

```
5
```

Последний элемент списка

Обратиться к последнему элементу списка, не зная точное количество элементов, можно следующим образом:

```
>>> numbers = [1, 2, 3, 4, 5]
>>> numbers[len(numbers) - 1]
5
```

Также к последнему элементу можно обратиться с помощью индексов с конца (перед индексом ставится знак минус).

```
>>> numbers[-1]
5
>>> numbers[-2] # Обращение к предпоследнему элементу
4
```

Содержимое списка

Список может содержать элементы различных типов данных, а не только одного типа, как это сделано в массивах, с которыми вы можете быть знакомы по другим языкам программирования.

Более того, элементом списка может быть другой список или даже сам список.

```
>>> my_list = [1, 2.3, 'four', True, [6, 7], (8, 9)]
>>> my_list
[1, 2.3, 'four', True, [6, 7], (8, 9)]
>>> my_list[4]
[6, 7]
>>> my_list[4][0]
6
>>> my_list += my_list
>>> my_list
[1, 2.3, 'four', True, [6, 7], (8, 9), 1, 2.3, 'four',
True, [6, 7], (8, 9)]
```

Операции над списками

Списки и кортежи можно, как и строки, объединять и умножать.

```
>>> a = [1, 2]
```

```
>>> b = [3, 4]
```

```
>>> a + b
```

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

```
>>> a * 2
```

```
[1, 2, 1, 2]
```

```
...
```


Операции над кортежами

Также и с кортежами, элементы которых «обнимаются» не квадратными скобками, а круглыми.

```
>>> a = (1, 2)
```

```
>>> b = (3, 4)
```

```
>>> a + b  
(1, 2, 3, 4)
```

```
>>> a * 2  
(1, 2, 1, 2)
```

```
...
```

Изменяемый объект

Список — это изменяемый объект.

Это значит, что все операции выполняются именно над этим объектом, а не над объектом, созданным на его основе. Такое положение вещей может привести к нежелательным результатам, поэтому за такими объектами (изменяемыми) надо следить.

```
>>> a = [1, 2]
>>> b = a # На объект [1, 2] теперь ссылается еще одно имя,
b. b — синоним.
>>> b += [3, ]
>>> a
[1, 2, 3]
```

Почему так получилось? Потому что `b = a` не поместило в `b` новый объект, `b` ссылается на тот же объект, что и `a` + объект изменяемый = мы изменили сам объект `[1, 2]`, объединив его со списком `[3,]`.

Неизменяемый объект

Кортеж и все рассмотренные нами типы до — это неизменяемые объекты.

Это значит, что все операции выполняются не над этим объектом, а над объектом, созданным на его основе.

```
>>> a = (1, 2)
```

```
>>> b = a # На объект (1, 2) теперь ссылается еще одно имя,  
b. b — синоним.
```

```
>>> b += (3, )
```

```
>>> a
```

```
(1, 2)
```

```
>>> b
```

```
(1, 2, 3) # Это другой объект, появившийся при вычислении  
выражения (1, 2) + (3, )
```

Итоги

Мы познакомились с такими понятиями, как список, кортеж, объединение списков/кортежей, повторение списков/кортежей, изменяемый объект, неизменяемый объект,

а также рассмотрели способы создания списка/кортежа

и попрактиковались в работе с интерпретатором в интерактивном режиме.