

11.3 Методы списков. Часть 1

Шаг 1

Тема урока: методы добавления и удаления элементов

1. Метод добавления элемента `append()`
2. Метод расширения списка `extend()`
3. Оператор `del`
4. Решение задач

Аннотация. Добавление элементов в список. Оператор `del`, удаляющий элементы по заданному индексу.

Добавление элементов

Мы научились создавать статические списки, то есть списки, элементы которых известны на этапе создания. Следующий шаг – научиться добавлять элементы в уже существующие списки.

Метод `append()`

Для добавления нового элемента **в конец списка** используется метод `append()`.

Приведённый ниже код:

```
numbers = [1, 1, 2, 3, 5, 8, 13] # создаем список

numbers.append(21) # добавляем число 21 в конец списка
numbers.append(34) # добавляем число 34 в конец списка

print(numbers)
```

ВЫВОДИТ:

```
[1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Обратите внимание, для того чтобы использовать метод `append()`, нужно, чтобы список был создан (при этом он может быть пустым).

Приведённый ниже код:

```
numbers = [] # создаем пустой список

numbers.append(1)
numbers.append(2)
numbers.append(3)

print(numbers)
```

ВЫВОДИТ:

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



Важно: мы не можем использовать индексы для установки значений элементов списка, если список пустой.

Приведённый ниже код:

```
numbers = [] # создаем пустой список

numbers[0] = 1
numbers[1] = 2
numbers[2] = 3

print(numbers)
```

приводит к возникновению ошибки:

```
IndexError: list assignment index out of range
```

Метод `extend()`

Можно также расширить список другим списком путём вызова метода `extend()` .

Приведённый ниже код:

```
numbers = [0, 2, 4, 6, 8, 10]
odds = [1, 3, 5, 7]

numbers.extend(odds)
print(numbers)
```

ВЫВОДИТ:

```
[0, 2, 4, 6, 8, 10, 1, 3, 5, 7]
```

Метод `extend()` как бы расширяет один список, добавляя к нему элементы другого списка.

Отличие между методами `append()` и `extend()` проявляется при добавлении строки к списку.

Приведённый ниже код:

```
words1 = ['iq option', 'stepik', 'beegeek']
words2 = ['iq option', 'stepik', 'beegeek']

words1.append('python')
words2.extend('python')

print(words1)
print(words2)
```

ВЫВОДИТ:

```
['iq option', 'stepik', 'beegeek', 'python']
['iq option', 'stepik', 'beegeek', 'p', 'y', 't', 'h', 'o', 'n']
```

Метод `append()` добавляет строку `'python'` целиком к списку, а метод `extend()` разбивает строку `'python'` на символы `'p'`, `'y'`, `'t'`, `'h'`, `'o'`, `'n'` и их добавляет в качестве элементов списка.

Удаление элементов

С помощью оператора `del` можно удалять элементы списка по определенному индексу.

Приведённый ниже код:

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]
del numbers[5] # удаляем элемент, имеющий индекс 5

print(numbers)
```

ВЫВОДИТ:

```
[1, 2, 3, 4, 5, 7, 8, 9]
```

Элемент под указанным индексом удаляется, а список перестраивается.



Обратите внимание на синтаксис удаления, так как он отличается от обычного вызова метода. При удалении элементов не надо передавать аргумент внутри круглых скобок.

Оператор `del` работает и со срезами: мы можем удалить целый диапазон элементов списка.

Приведённый ниже код:

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]
del numbers[2:7]    # удаляем элементы с 2 по 6 включительно

print(numbers)
```

ВЫВОДИТ:

```
[1, 2, 8, 9]
```

Мы можем удалить все элементы на чётных позициях (`0` , `2` , `4` , ...) исходного списка.

Приведённый ниже код:

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]
del numbers[::2]

print(numbers)
```

ВЫВОДИТ:

```
[2, 4, 6, 8]
```

❤️ Happy Pythoning! 🐍

Шаг 2

Допустим, программа состоит из строки кода:

```
names = []
```

Какую из приведённых ниже инструкций следует применить для добавления в список по 0 индексу строкового значения `'Chromatic'` ?

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/2>

Шаг 3

Что покажет приведённый ниже код?

```
numbers = [4, 2, 8, 6, 5]
numbers.append(7)
numbers.append(1)

print(numbers)
```

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/3>

Шаг 4

Что покажет приведённый ниже код?

```
numbers = [4, 2]
numbers.extend([1, 2, 3])
numbers.extend([7, 17, 777])

print(numbers)
```

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/4>

Шаг 5

Что покажет приведённый ниже код?

```
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'purple', 'brown', 'magenta']
del colors[2]
del colors[6]

print(colors)
```

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/5>

Шаг 6

Все сразу 1 🌶️

Дополните приведённый ниже код, чтобы он:

1. Вывел длину списка;
2. Вывел последний элемент списка;
3. Вывел список в обратном порядке (вспоминаем срезы);
4. Вывел «YES» (без кавычек), если список содержит числа 5 и 17, или «NO» (без кавычек) в противном случае;
5. Вывел список с удалёнными первым и последним элементами.

Примечание. Каждый вывод осуществлять с новой строки.

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/6>

Шаг 7

Список строк

На вход программе подаются натуральное число n , а затем n строк. Напишите программу, которая создаёт из указанных строк список, а затем выводит его.

Формат входных данных

На вход программе подаются натуральное число n , а затем n строк, каждая на отдельной строке.

Формат выходных данных

Программа должна вывести список, состоящий из указанных строк.

► Тестовые данные 

Sample Input:

```
5
C#
C++
C
Python
F#
```

Sample Output:

```
['C#', 'C++', 'C', 'Python', 'F#']
```

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/7>

Шаг 8

Алфавит

Напишите программу, выводящую следующий список:

```
['a', 'bb', 'ccc', 'dddd', 'eeeee', 'ffffff', ...]
```

Формат выходных данных

Программа должна вывести указанный список.

Примечание. Последний элемент списка должен состоять из 26 символов `z`.

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/8>

Шаг 9

Список кубов

На вход программе подаются натуральное число n , а затем n целых чисел. Напишите программу, которая создаёт из указанных чисел список их кубов, а затем выводит его.

Формат входных данных

На вход программе подаются натуральное число n , а затем n целых чисел, каждое на отдельной строке.

Формат выходных данных

Программа должна вывести список, состоящий из кубов указанных чисел.

► Тестовые данные 

Sample Input 1:

```
5
1
2
3
4
5
```

Sample Output 1:

```
[1, 8, 27, 64, 125]
```

Sample Input 2:

```
2
-5
-2
```

Sample Output 2:

```
[-125, -8]
```

Sample Input 3:

```
1
100
```

Sample Output 3:

```
[1000000]
```

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/9>

Шаг 10

Список делителей


На вход программе подаётся натуральное число n . Напишите программу, которая создаёт список, состоящий из делителей введённого числа, а затем выводит его.

Формат входных данных

На вход программе подаётся натуральное число n .

Формат выходных данных

Программа должна вывести список, состоящий из делителей введённого числа.

► Тестовые данные 

Sample Input 1:

```
17
```

Sample Output 1:

```
[1, 17]
```

Sample Input 2:

```
25
```

Sample Output 2:

```
[1, 5, 25]
```

Sample Input 3:

```
36
```

Sample Output 3:

```
[1, 2, 3, 4, 6, 9, 12, 18, 36]
```

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/10>

Шаг 11

Суммы двух


На вход программе подаётся натуральное число n ($n \geq 2$). Затем поступают n целых чисел. Напишите программу, которая создаёт из указанных чисел список, состоящий из сумм соседних чисел (0 и 1, 1 и 2, 2 и 3 и так далее).

Формат входных данных

На вход программе подаются натуральное число n ($n \geq 2$), а затем n целых чисел, каждое на отдельной строке.

Формат выходных данных

Программа должна вывести список, состоящий из сумм соседних чисел.

► Тестовые данные 

Sample Input 1:

```
5
1
2
3
4
5
```

Sample Output 1:

```
[3, 5, 7, 9]
```

Sample Input 2:

```
2
10
9
```

Sample Output 2:

```
[19]
```

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/11>

Удалите нечётные индексы

На вход программе подаются натуральное число n , а затем n целых чисел. Напишите программу, которая создаёт из указанных чисел список, затем удаляет все элементы, стоящие по нечётным индексам, а затем выводит полученный список.

Формат входных данных

На вход программе подаются натуральное число n , а затем n целых чисел, каждое на отдельной строке.

Формат выходных данных

Программа должна вывести список в соответствии с условием задачи.

Примечание. Используйте оператор `del`.

► Тестовые данные 

Sample Input 1:

```
10
0
1
2
3
4
5
6
7
8
9
```

Sample Output 1:

```
[0, 2, 4, 6, 8]
```

Sample Input 2:

```
1
8
```

Sample Output 2:

```
[8]
```

Sample Input 3:

```
2
9
6
```

Sample Output 3:

```
[9]
```

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/12>

Шаг 13

k -ая буква слова 🌶

На вход программе подаются натуральное число n и n строк, а затем число k . Напишите программу, которая выводит k -ую букву из введённых строк на одной строке без пробелов.

Формат входных данных

На вход программе подаются натуральное число n , далее n строк, каждая на отдельной строке. В конце вводится натуральное число k – номер буквы (нумерация начинается с единицы).

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

Примечание. Если некоторые строки слишком короткие и в них нет символа с заданным номером, то такие строки при выводе нужно игнорировать.

► Тестовые данные ●

Sample Input 1:

```
5
abcdef
bcdefg
cdefgh
defghi
efghij
2
```

Sample Output 1:

```
bcdef
```

Sample Input 2:

```
5
aaaaa
bbbb
ccc
dd
e
3
```

Sample Output 2:

```
abc
```

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/13>

Шаг 14

Символы всех строк

На вход программе подаются натуральное число n , а затем n строк. Напишите программу, которая создает список из символов всех строк, а затем выводит его.

Формат входных данных

На вход программе подаются натуральное число n , а затем n строк, каждая на отдельной строке.

Формат выходных данных

Программа должна вывести список, состоящий из символов всех введённых строк.

Примечание. В результирующем списке могут содержаться одинаковые символы.

► Тестовые данные 

Sample Input:

```
3
abc
def
ghi
```

Sample Output:

```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i']
```

Чтобы решить это задание откройте <https://stepik.org/lesson/327207/step/14>