КОД БУДУЩЕГО
обучение талантливых школьников 8 -11 классов современным языкам программирования

Минцифры России

НАЦИОНАЛЬНЫЕ ПРОЕКТЫ РОССИИ ЦИФРОВАЯ ЭКОНОМИКА

20.35 УНИВЕРСИТЕТ

Программирование на языке Python. Базовый уровень

Модуль 2. Строки и списки

Списки (занятие 3)

академия ПРОСВЕЩЕНИЕ

Функции по обработке списков

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

Функции по обработке списков

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

Поскольку данные функции являются методами класса «Список», то обращение к ним имеет вид:

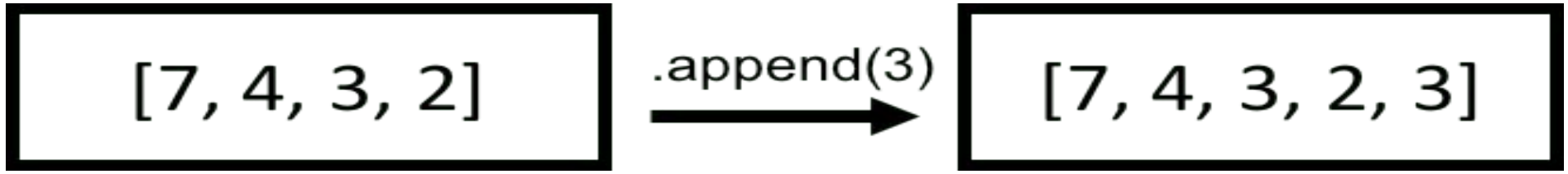
< имя списка>. <имя метода>

Пример.

```
l=[2,4,6,8,10,12,14,16]
print(l.index(4))
```

В результате на экран выводится индекс элемента, равного 4. В данном случае это 1.

Метод Append()



Пример.

```
all_types = [10, 3.14, 'Python', ['I', 'am', 'list']]
all_types.append(1024)
all_types.append('Hello world!')
all_types.append([1, 2, 3])
print(all_types)
```

Результат: [10, 3.14, 'Python', ['I', 'am', 'list'], 1024, 'Hello world!', [1, 2, 3]]

Метод clear()

Данный метод удаляет все элементы списка.

Пример.

```
lst = ['к', 'у', 'р', 'с']
lst.clear()
print(lst)
```

Результат: []

Метод copy()

Данный метод возвращает копию списка.

Пример.

```
a = [1, 7, 9]
b = a.copy()
print(a)
print(b)
```

Результат:

```
[1, 7, 9]
[1, 7, 9]
```

Метод count()

Данный метод считает, сколько раз указанное значение появляется в списке.

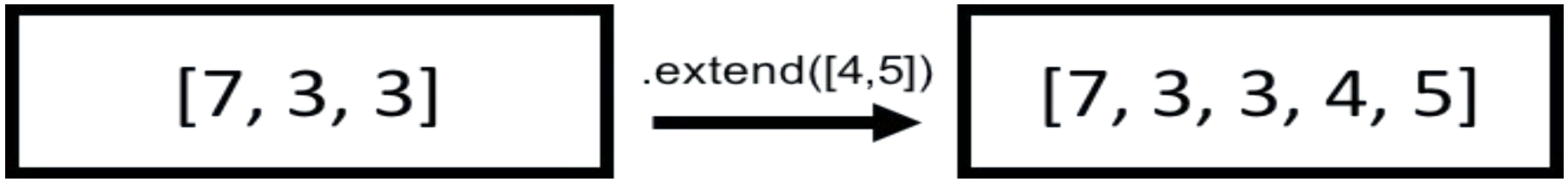
Пример.

```
numbers = [1, 5, 9, 6, 1, 2, 1]
print(numbers.count(1))
```

Результат: 3

```
# вывод 3, потому что единица встречается 3 раза
```

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 index()

z =	[4,	1,	5,	4,	10,	4]
index	0	1	2	3	4	5

Метод index возвращает положение первого индекса, со значением x. В указанном ниже коде, он возвращает назад 0.

 # Создайте список

```
z = [4, 1, 5, 4, 10, 4]
print(z.index(4))
```

Результат:

0