

# Списки

Список (`list`) - структура данных в языке Python, которая представляет собой индексированную последовательность. Списки не типизированы и позволяют произвольную вложенность.

Производительность:

- Время доступа к элементу списка не зависит от размера списка
- Время на добавление элемента в конец списка не зависит от размера списка
- Время на вставку элемента в список зависит от количества элементов правее него
- Время на удаление элемента из списка зависит от количества элементов правее него

Со списками, как и со строками, можно использовать срезы:

```
# Список [1, '23', True, 4]
l = list(1, '23', True, 4)
# Список [1, '23']
l1 = list[0:2]
```

Но, в отличие от строк список изменяемый:

```
# Список [1, '34', True, 4]
l[1] = '34'
```

А также изменяемый по срезу:

```
# Список [1, '34', False, 0.5]
l[2:4] = [False, 0.5]
```

## Операции со списками

### Создание копии списка

1) Через срез

```
l2 = l[:]
```

2) Через клонирование

```
l2 = list(l)
```

**Внимание!** Присваивание не создаёт нового списка, а просто создаёт "второе имя" (ссылку) уже существующего списка. Например:

```
l = [1, 2, '5']
l2 = l1
l2[1] = 3
```

```
print(l[1])      # На экран будет выведено число 3
```

## Сложение списков

Списки можно складывать. При сложении происходит конкатенация списков:

```
l = [1, 3]
l2 = [2, 4]
# Список [1, 2, 3, 4]
l3 = l + l2
```

## Повтор списка

Список можно умножать на целое число, при это будет произведён повтор списка:

```
l = [1, 2]
# Список [1, 2, 1, 2, 1, 2]
l2 = [1, 2]*3
```

## Генераторы списков

В Python есть особенный синтаксис конструирования списков. Выглядит он следующим образом: внутри квадратных скобок вместо обычных значений указывается цикл `for`. Например, сгенерируем список квадратов чисел от 1 до 9:

```
# Список [1, 4, 9, 16, 25, 36, 49, 64, 81]
l = [ i*i for i in range(1, 10) ]
```

Конструктор списков может содержать так же условия, например, если требуется вывести только чётные числа от 1 до 9:

```
# Список [2, 4, 6, 8]
l = [ i for i in range(1, 10) if i%2 == 0]
```

## Полезные методы

- `append(elem)` - добавление элемента `elem` в списоки
- `insert(i, elem)` - добавление элемента `elem` в список на место `i`
- `remove(elem)` - удаляет первый элемент в списке, имеющий значение `elem`
- `pop(i)` - удаляет элемент на месте `i` и возвращает его
- `count(elem)` - возвращает количество элементов, имеющих значнеие `elem`
- `sort()` - сортирует список
- `reverse()` - разворачивает список

- `clear()` - очищает список

## **Полезные функции, применяемые к спискам**

- `max(list)` - возвращает максимальный элемент в списке `list`
- `min(list)` - возвращает минимальный элемент в списке `list`
- `len(list)` - возвращает размер списка `list`