**Что такое массив?**

Массив (в Python — список) — это структура данных, в которой можно хранить набор значений. Списки применяются, когда нужно работать с последовательностью данных (чисел, строк, объектов и т. д.), которые связаны общим смыслом и обрабатываются одинаковым образом, например, при поиске минимального или максимального элемента, подсчёте суммы и так далее. Список полезен при работе с данными, которые нужно динамически менять и обрабатывать в циклах.

**Списки в Python**

Список в Python — это упорядоченный набор элементов, где каждый элемент имеет свой индекс, начиная с 0. Важно, что список — это динамическая структура данных, его размер может изменяться во время выполнения программы. Элементы списка могут быть любого типа, а один список может содержать данные разного типа (например, числа и строки одновременно).

**Как создать список?**

В Python создать список можно несколькими способами:

1) Пустой список создается с помощью квадратных скобок:

***my_list = []***

2) Список с элементами создается с помощью квадратных скобок, в которые сразу добавляются значения:

***my_list = [3, 7, 2, 9, 10]***

**Как добавить элементы в список?**

Чтобы добавить элемент в конец списка, можно использовать метод append(). Он добавляет один элемент:

```
my_list = [1, 2, 3]
my_list.append(4)
print(my_list)  # [1, 2, 3, 4]
```

Для добавления всех элементов одного списка в другой используется метод extend():

```
list1 = [1, 2, 3]
list2 = [4, 5, 6]
list1.extend(list2)
print(list1)  # [1, 2, 3, 4, 5, 6]
```

При использовании append() весь второй список добавляется как один элемент:

```
list1 = [1, 2, 3]
list2 = [4, 5, 6]
list1.append(list2)
print(list1)  # [1, 2, 3, [4, 5, 6]]
```

**Удаление элементов из списка**

В Python существует несколько методов для удаления элементов из списка:

1) remove(значение) — удаляет первый элемент, равный указанному значению. Если указанного значения в списке нет, возникает ошибка:

```
my_list = [1, 2, 3, 2]
my_list.remove(2)
print(my_list)  # [1, 3, 2]

# Ошибка:
my_list.remove(5)  # ValueError: list.remove(x): x not in list
```

2) pop(индекс) — удаляет элемент по указанному индексу и возвращает его. Если индекс не указан, удаляется последний элемент:

```
my_list = [1, 2, 3]
my_list.pop()
print(my_list)  # [1, 2]
```

pop() не только удаляет элемент из списка, но и возвращает его, что позволяет сразу сохранить удалённый элемент в отдельную переменную для дальнейшего использования. Если указать индекс, pop(индекс) удаляет и возвращает элемент по этому индексу:

```
my_list = [5, 15, 25, 35]
second_element = my_list.pop(1)  # Удаляет элемент по индексу 1 и сохраняет его в переменную
print(second_element)  # 15
print(my_list)         # [5, 25, 35]
```

3) del список[индекс] — удаляет элемент по указанному индексу без возврата значения:

```
my_list = [1, 2, 3]
del my_list[1]
print(my_list)  # [1, 3]
```

```
# Ошибка:
del my_list[5]  # IndexError: list index out of range
```

**Использование списков в цикле for**

Важно помнить, что список — это динамическая структура данных, и если в цикле for изменять сам список, результат может оказаться неожиданным. Пример:

```
numbers = [2, 4, 6, 8, 10, 12]
for num in numbers:
    numbers.remove(num)
print(numbers)  # Оставшиеся числа: [4, 8, 12]
```

В данном случае изначальный список изменяется в процессе выполнения цикла, что приводит к пропуску некоторых элементов. Поэтому, если требуется фильтровать значения, лучше создать новый список и добавлять туда нужные элементы:

```
numbers = [2, 5, 7, 8, 10, 9]
even_numbers = []
for num in numbers:
    if num % 2 == 0:
        even_numbers.append(num)
print(even_numbers)  # [2, 8, 10]
```

**Алгоритмы работы со списками**

Сортировка (пузырьковая):

```
numbers = [5, 1, 8, 3, 2]
for i in range(len(numbers)):
    for j in range(len(numbers) - 1 - i):
        if numbers[j] > numbers[j + 1]:
            numbers[j], numbers[j + 1] = numbers[j + 1], numbers[j]
print(numbers)  # [1, 2, 3, 5, 8]
```

Поиск минимального и максимального значения:

```
numbers = [4, 2, 8, 1, 9]
min_value = numbers[0]
max_value = numbers[0]
for num in numbers:
    if num < min_value:
        min_value = num
    if num > max_value:
        max_value = num
print(min_value, max_value)  # 1, 9
```

Сумма чисел в списке:

```
numbers = [4, 7, 1, 3]
total_sum = 0
for num in numbers:
    total_sum += num
print(total_sum)  # 15
```

Поиск и учёт значений, удовлетворяющих условию:

```
numbers = [10, 17, 20, 25, 31]
divisible_by_5 = []
for num in numbers:
    if num % 5 == 0:
        divisible_by_5.append(num)
print(divisible_by_5)  # [10, 20, 25]
```