



Списки в Python.

Списки

Списки в Python – это структура данных, которая содержит упорядоченный набор элементов, т. е. хранит последовательность элементов. Изменяемые списки создаются через квадратные скобки. Элементы могут повторяться и меняться. Доступ к отдельному элементу тоже через квадратные скобки. Отсчёт с 0.

Создание списков

```
number = [1, 2, 3, 9, 7, 8, 0, -42]
```

```
letters = ["A", "Б", "B", "Г"]
```

```
logics = [True, True, False]
```

```
nothings = []
```

```
nothings_v2 = list()
```

Типы хранимых данных

Внутри списков можно хранить любые типы данных будь то целые числа, числа с плавающей точкой, строки, булевыe данные и даже другие списки.

```
values = ["A", 2, True, None, [], -99.45]
```

```
values = [  
    "A", 2, True, -999,  
    None, "\n\t", "",  
    -42, "Список", False  
]  
  
print(values)
```

```
['A', 2, True, -999, None, '\n\t', '', -42, 'Список', False]
```

Элементы списка

Для обращения к элементам списка надо использовать индексы, которые представляют номер элемента в списка. **Индексы начинаются с нуля.** То есть первый элемент будет иметь индекс 0, второй элемент - индекс 1 и так далее. Для обращения к элементам с конца можно использовать отрицательные индексы, начиная с -1. То есть у последнего элемента будет индекс -1, у предпоследнего - -2 и так далее.

```
number = [1, 2, 3, 4, 5]
people = ["Tom", "Sam", "Bob"]
to_buy = ["хлеб", "кабачки", "масло", "молоко"]

num = number[3] # 4
print(people[-2]) # Sam
print(to_buy[5]) # ошибка
```

Элементы вложенного списка

Элементом списка может быть и другой список. Такие списки называются вложенными.

```
values = [  
    [1, 2, 3],  
    ["a", "b", "c"],  
    [True, False, True]  
]  
  
print(values[0][1]) # 2  
print(values[-2][2]) # c
```

Изменение элемента списка

```
items = ["яблоки", "кабачки", "груши", "апельсины"]  
items[1] = "бананы"  
items[-1] = "мандарины"  
  
print(items) # ["яблоки", "бананы", "груши", "мандарины"]
```

Методы и функции по работе со списками

<code>len(list)</code>	Возвращает длину списка
<code>list.append(item)</code>	Добавляет элемент <code>item</code> в конец списка
<code>list.insert(index, item)</code>	Добавляет элемент <code>item</code> в список по индексу <code>index</code>
<code>list.remove(item)</code>	Удаляет элемент <code>item</code> . Удаляется только первое вхождение элемента. Если элемент не найден, генерирует исключение <code>ValueError</code>
<code>list.index(item)</code>	Возвращает индекс элемента <code>item</code> . Если элемент не найден, генерирует исключение <code>ValueError</code>
<code>list.pop(index)</code>	Удаляет и возвращает элемент по индексу <code>index</code> . Если элемент не найден, то просто удаляет последний элемент.
<code>list.clear()</code>	Удаление всех элементов из списка

<code>list.count(item)</code>	Возвращает количество вхождений элемента <code>item</code> в список
<code>list.reverse()</code>	Расставляет все элементы в списке в обратном порядке
<code>list.copy()</code>	Копирует список
<code>list.sort()</code>	Сортирует список по возрастанию
<code>min(list)</code>	Возвращает наименьший элемент списка
<code>max(list)</code>	Возвращает наибольший элемент списка
<code>item in list</code>	Проверяет есть ли в списке <code>list</code> элемент <code>item</code>

Пример

```
items = ["яблоки", "кабачки", "груши", "апельсины"]  
new_item = "бананы"
```

```
if new_item not in items:  
    items.append(new_item)
```

```
items.pop(1)
```

```
i = 0  
while i < len(items) - 1:  
    item = items[i]  
    print(item)  
    i += 1
```

Задания для выполнения

1. Найти среднее арифметическое из списка чисел [5,3,4].
2. Отсортировать произвольный список чисел по возрастанию.
3. Найти сумму всех четных чисел в произвольном списке чисел.
4. Создать и заполнить список. Размер и значения списка должен вводить пользователь.

Домашняя работа

- Найти частное, методом нахождения среди списка максимального числа и деление его на длину того же списка
- Дополнить задание №3 из прошлого слайда. Ограничить ввод только четными числами. При вводе нечетных чисел выводить ошибку и повторять ввод.
- Повторить ранее пройденный материал