

Списки

Список - упорядоченный изменяемый набор объектов произвольных типов.

Список может быть пустым

```
In [ ]: a = []  
a
```

```
Out[ ]: []
```

или содержать элементы произвольных типов

```
In [ ]: values = [1, 2, 'asd', 5, '6sda', 1.3]  
values
```

```
Out[ ]: [1, 2, 'asd', 5, '6sda', 1.3]
```

Возможно существование списка списков

```
In [ ]: b = [[1, 2, 3], [4, 5]]  
b
```

```
Out[ ]: [[1, 2, 3], [4, 5]]
```

Генераторы списков

Создадим два списка

```
In [ ]: a = ['a', 'b', 'c']  
b = [1, 2, 3]
```

На основе списка **a** создадим список **c**, элементы которого есть утроенные элементы **a**

```
In [ ]: c = [i * 3 for i in a]  
c
```

```
Out[ ]: ['aaa', 'bbb', 'ccc']
```

Также в генератор можно добавить условие

```
In [ ]: c = [i * 3 for i in a if i != 'b']  
c
```

```
Out[ ]: ['aaa', 'ccc']
```

Или использовать более сложные конструкции, например

```
In [ ]: c = [i + str(j) for i in a if i != 'b' for j in b if j != 3]
c
```

```
Out[ ]: ['a1', 'a2', 'c1', 'c2']
```

Действия со списками

Зададим три списка по аналогии со строками (из предыдущего занятия)

```
In [ ]: x = [1, 2, 3]
y = [4, 5, 6]
z = [7, 8, 9]
```

Конкатенация (сложение)

```
In [ ]: numbers = x+y+z

print(numbers)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Доступ по индексу. Нумерация элементов производится с нуля.

```
In [ ]: print(numbers[0])
print(numbers[1])
print(numbers[2])
print(numbers[3])
```

```
1
2
3
4
```

Дублирование списка

```
In [ ]: b = x*3
b
```

```
Out[ ]: [1, 2, 3, 1, 2, 3, 1, 2, 3]
```

Извлечение среза.

[с какого начать : перед каким остановиться : с каким шагом идти]

```
In [ ]: numbers[2:4]
```

```
Out[ ]: [3, 4]
```

Индекс "-1" указывает на последний элемент

```
In [ ]: numbers[2:-1]
```

```
Out[ ]: [3, 4, 5, 6, 7, 8]
```

Шаг "-1" указывает, что движение по списку происходит в обратном порядке

```
In [ ]: numbers[::-1]
```

```
Out[ ]: [9, 8, 7, 6, 5, 4, 3, 2, 1]
```

Извлекаем срез со второго (на самом деле третьего) элемента до четвёртого с конца элемента включительно, двигаемся с шагом 2

```
In [ ]: numbers[2:-3:2]
```

```
Out[ ]: [3, 5]
```

Важное отличие списков от строк

Все объекты в python имеют свой идентификатор. Вывести его можно с помощью команды **id()**

```
In [ ]: a = "abc"
        b = ["a", "b", "c"]

        print(id(a))
        print(id(b))
```

```
140707502759488
```

```
1876852502144
```

Создадим функцию, которая реализует удвоение списка/строки

```
In [ ]: def double(a):
        a *= 2 # a=a*2
        return a
```

Подеиствуем этой функцией на строку **a** и список **b**

```
In [ ]: aa = double(a)
        bb = double(b)
```

Выведем **id** и значения исходных и получившихся объектов

```
In [ ]: print(f'Строка:\nИсходная: {a}, id: {id(a)}\nНовая:      {aa}, id: {id(aa)}\n')
        print(f'Список:\nИсходный: {b}, id: {id(b)}\nНовый:      {bb}, id: {id(bb)}\n')
```

Строка:

Исходная: abc, id: 140707502759488

Новая: abcab, id: 1876852519856

Список:

Исходный: ['a', 'b', 'c', 'a', 'b', 'c'], id: 1876852502144

Новый: ['a', 'b', 'c', 'a', 'b', 'c'], id: 1876852502144

Так происходит потому, что списки - **изменяемый** тип данных, а строки - **неизменяемый**

Функции и методы списков

Длина списка (количество элементов) получается при использовании функции len()

```
In [ ]: print(values)
        len(values)
```

```
[1, 2, 'asd', 5, '6sda', 1.3]
```

```
Out[ ]: 6
```

Методы списков, в отличие от методов строк, изменяют сам список, поэтому результат не нужно записывать в отдельную переменную

Добавление элемента в конец списка

```
In [ ]: values.append([8, 7])
        values
```

```
Out[ ]: [1, 2, 'asd', 5, '6sda', 1.3, [8, 7]]
```

Расширение списка с помощью добавления в конец всех элементов другого списка

```
In [ ]: values.extend([8, 7])
        values
```

```
Out[ ]: [1, 2, 'asd', 5, '6sda', 1.3, [8, 7], 8, 7]
```

Вставка элемента по индексу

```
In [ ]: values.insert(2, 'Три')
        values
```

```
Out[ ]: [1, 2, 'Три', 'asd', 5, '6sda', 1.3, [8, 7], 8, 7]
```

Количество элементов со значением x

```
In [ ]: values.count('Три')
```

```
Out[ ]: 1
```

Сортировка списка

```
In [ ]: numbers = [1, 3, 6, 3, -1223, 24]
        numbers.sort()
        numbers
```

Out[]: [-1223, 1, 3, 3, 6, 24]

Копия списка

```
In [ ]: a = [1, 3, 5]
        b = a
        c = a.copy()

        print(f'a: {a}, id: {id(a)}\n')
        print(f'b: {b}, id: {id(b)}\n')
        print(f'c: {c}, id: {id(c)}\n')
```

a: [1, 3, 5], id: 1876852504640

b: [1, 3, 5], id: 1876852504640

c: [1, 3, 5], id: 1876852505152