

Программирование на языке Python. Базовый уровень

Модуль 2. Строки и списки

Списки (занятие 1)



Список

Список (list) – это структура данных для хранения объектов различных типов.

Размер списка не статичен, его можно изменять.

Список является изменяемым типом данных.

В Python списки записываются в квадратных скобках.

Пример.

```
thislist = ["яблоко", "банан", "вишня"] print(thislist)
```

Вывод:

```
["яблоко", "банан", "вишня"]
```

Создание списка

Первый способ: указание значений списка вручную.

Для создания списка применяются квадратные скобки [], внутри которых через запятую перечисляются элементы списка.

Пример.

```
list1 = [0, 1, 2, 3, 4, 5]
```

```
list2 = ['p', 'y', 't', 'h', 'o', 'n']
```

Также для создания списка можно использовать функцию-конструктор list():

```
list1 = []
```

```
list2 = list()
```



Создание списка

Второй способ: преобразование других типов данных в список.

Функция `list()` позволяет преобразовать строку в список:

```
my_lst = list('Python')
```

```
print(my_lst)
```

Результат будет следующим:

```
['P', 'y', 't', 'h', 'o', 'n']
```

Создание списка

Третий способ: генерация списков.

Самый простой метод – использовать встроенную функцию `range()`, которая обеспечит автоматическое создание списка, включающего значения из определенного диапазона.

Пример.

```
my_lst = list(range(5))
```

```
print(my_lst)
```

ВЫВОД

```
[0, 1, 2, 3, 4]
```

Пример.

```
from random import randint
```

```
a = [randint(-10, 10) for i in range(20)]
```

```
print(a)
```

ВЫВОД

```
[1, 5, 0, 9, -6, 0, 10, -3, 6, 7, -3, -10, 3, 6, 2, 7,  
-9, 3, 1, -3]
```

Создание списка

Если необходимо создать список, в котором повторяется одно и то же значение несколько раз, то можно использовать символ «*». Этот символ фактически применить операцию умножения к уже существующему списку:

```
numbers = [5] * 6 # 6 раз повторяем 5
```

```
print(numbers) # [5, 5, 5, 5, 5, 5]
```

Обращение к элементам списка

Значения, из которых состоит список, называются **элементами списка**.

К элементам списка можно обращаться по индексам (нумерация, как и в случае со строками, всегда начинается с 0), отрицательная нумерация также поддерживается:

Пример.

```
list_1= [' p ', 'y', 't', 'h', 'o', 'n']
```

```
print(list_1 [2])
```

```
print(list_1 [-3])
```

#ОТВЕТ

t

h

Обращение к элементам списка

Список L	8	-2	6	1	0	2
Индекс	0	1	2	3	4	5
Отрицательный индекс	-6	-5	-4	-3	-2	-1

Например, элемент $L[3]$ равен 1, а элемент $L[-4]$ равен 6.