

Операции с таблицами

Списки

Получить из списка диапазон

```
In digits_names = [  
    'ноль', 'один', 'два', 'три', 'четыре', 'пять', 'шесть', 'семь', 'восемь', 'девять'  
]  
  
# указываем обе границы диапазона (правая не включается)  
print(digits_names[4:7])  
  
# опускаем левую границу – идём с начала списка  
print(digits_names[:5])  
  
# опускаем правую границу – идём до конца списка  
print(digits_names[7:])
```

```
Out ['четыре', 'пять', 'шесть']  
     ['ноль', 'один', 'два', 'три', 'четыре']  
     ['семь', 'восемь', 'девять']
```

Добавить к списку элемент в конец

```
In emoji = ['Ухмыляюсь', 'Сияю от радости', 'Катаюсь от смеха', 'Слёзы радости']  
  
print(emoji)  
  
emoji.append('Подмигиваю')  
  
print(emoji)
```

```
Out ['Ухмыляюсь', 'Сияю от радости', 'Катаюсь от смеха', 'Слёзы радости']  
     ['Ухмыляюсь', 'Сияю от радости', 'Катаюсь от смеха', 'Слёзы радости', 'Подмигиваю']
```

Отсортировать таблицу (список списков) по столбцу

```
In data.sort(key=lambda row: row[1], reverse=True)
```

Циклы

Получить диапазон чисел или повторить код

```
In for element in range(5):  
    print(element)
```

```
Out 0  
     1  
     2  
     3  
     4
```

```
In for i in range(3):  
    print("*****")
```

```
Out *****  
     *****  
     *****
```

Изменить список в цикле

```
In ...  
  
for i in range(len(data)):  
    part = data[i][1]/emojixpress_total  
    data[i].append(part)  
  
...
```

Вывести на одной строке

```
In print('Анализ ', end='')  
    print('эмодзи')
```

```
Out Анализ эмодзи
```

Словарь

Именованный аргумент

Аргумент функции, для которого указывается его имя

Лямбда-функция

Безымянная функция со специальным синтаксисом

Нормирование

Приведение значений к общему масштабу

Метод

Функция, прикрепленная к типу данных и выполняющая работу над переменной этого типа. Вызов метода производится с использованием **точечной нотации**

Срез списка

Часть элементов списка в определённом диапазоне индексов