

Программирование на языке Python. Базовый уровень

Модуль 2. Строки и списки

Списки (занятие 2)



Списки и строки

С помощью `split()` можно разбить строку по определенным разделителям – по умолчанию это пробел.

Пример.

```
sp = 'Разработка ПО на языке Python'.split()  
print(sp)  
['Разработка', 'ПО', 'на', 'языке', 'Python']
```

Срезы списка

Срез позволяет используя индексы обратиться не к конкретному элементу, а к их набору. Для этого в квадратных скобках указываем индекс начального элемента, индекс элемента, до которого будем получать элементы. Третий аргумент – шаг выборки. Все аргументы могут быть как положительными, так и отрицательными, все элементы необязательны. Если указать отрицательным шаг, то элементы будут выданы в обратном порядке.

Пример.

```
список_1 = list(range(11))
print('список_1[0:5:-1]', список_1[0:5:-1]) # Элементы с первого по пятый в
обратном порядке
print('список_1[2:]',
список_1[2:]) # Все элементы, начиная с третьего
print('список_1[::2]', список_1[::2]) # Все чётные элементы
print('список_1[:]', список_1[:]) # Такая запись вернёт все элементы

# Но это копия, а не тот же список
список_2 = список_1
список_3 = список_1[:]
print('Равенство списков: ')
print('список_2 == список_1', список_2 == список_1)
```

Длиной списка называется количество его элементов. Для того, чтобы посчитать длину списка мы используем встроенную функцию `len()` (от слова `length` – длина).

Пример:

```
numbers = [2, 4, 6, 8, 10]
```

```
languages = ['Python', 'C#', 'C++', 'Java']
```

```
print(len(numbers))    # выводим длину списка numbers
```

```
print(len(languages))  # выводим длину списка languages
```

Операции над списками Python

`x in l` – true, если элемент `x` есть в списке `l`;

`x not in l` – true, если элемент `x` отсутствует в `l`;

`l1 + l2` – объединение двух списков;

`l * n`, `n * l` – копирует список `n` раз;

`len(l)` – количество элементов в `l`;

`min(l)` – наименьший элемент;

`max(l)` – наибольший элемент;

`sum(l)` – сумма чисел списка;

`for i in list()` – перебирает элементы слева направо.

Пример.

Введите элементы числового списка. Вычислите сумму минимального и максимального элементов списка, среднее арифметическое элементов списка.

```
numbers = [12.5, -5, 2.7, -8, 32, 3.75, 75.97]
print(min(numbers))
print(max(numbers))
print(min(numbers) + max(numbers))
print(sum(numbers)/len(numbers))
```

Результат представлен ниже.

-8

75.97

67.97

Вывод с помощью цикла for

Для вывода элементов списка каждого на отдельной строке можно использовать следующий код (если нужны индексы элементов):

```
numbers = [1, 2, 3, 4, 5]
for i in range(len(numbers)):
    print(numbers[i])
```

```
=====
1
2
3
|4
5
```

Распаковка списка через цикл for с индексом

Передаем в функцию range() длину списка len(numbers).

```
numbers = [1, 2, 3, 4, 5]
for num in range(len(numbers)):
    print(num)
```

```
=====
0
1
2
3
4
.
```

Распаковка через цикл for без индексов

Этот цикл пройдет по списку numbers, придавая переменной цикла num значение каждого элемента списка, в отличие от предыдущего цикла, в котором переменная цикла «бегала» по индексам списка.

```
numbers = [1, 2, 3, 4, 5]
for num in numbers:
    print(num, end = ' ')
```

```
=====
1 2 3 4 5
>>> |
```

Распаковка списка в строку через пробел

В Python есть удобный способ вывода элементов списка без использования цикла for.

1. Вывод элементов списка через один символ пробела:
2. Вывод элементов списка, каждого на отдельной строке

```
numbers = [1, 2, 3, 4, 5]
print(* numbers)
numbers = [1, 2, 3, 4, 5]
print(* numbers, sep='\n')
```

1 2 3 4 5

1

2

3

4

5