



Основы проектной деятельности

Второе занятие. Продолжение темы списки

Список

Список представляет собой последовательность элементов, пронумерованных от 0, как символы в строке. Список можно задать перечислением элементов списка в квадратных скобках

ПРИМЕР:

```
numbers = [5, 7, 234, 123, 78, 3]
```

```
words = ['hello', 'ball', 'yellow']
```



Функция len

Длину списка *x*, то есть количество элементов в нем, можно узнать при помощи функции ***len(x)***

ПРИМЕР:

```
numbers = [5, 7, 234, 123, 78, 3]
```

```
print(len(numbers))
```



Вывод списка

Вывести элементы списка **a** можно одной инструкцией **print(a)**, при этом будут выведены квадратные скобки вокруг элементов списка и запятые между элементами списка. Такой вывод неудобен, чаще требуется просто вывести все элементы списка в одну строку или по одному элементу в строке.



Вывод списка

```
a = [1, 2, 3, 4, 5]
```

```
for i in range(len(a)):
```

```
    print(a[i])
```

Здесь в цикле меняется индекс элемента i , затем выводится элемент списка с индексом i .



Вывод списка

```
a = [1, 2, 3, 4, 5]
```

```
for elem in a:
```

```
    print(elem, end=' ')
```

В этом примере элементы списка выводятся в одну строку, разделенные пробелом, при этом в цикле меняется не индекс элемента списка, а само значение переменной



Методы split

Элементы списка могут вводиться по одному в строке, в этом случае строку целиком можно считать функцией **input()**.

После этого можно использовать метод строки **split()**, возвращающий список строк, которые получатся, если исходную строку разрезать на части по пробелам.

```
s = input() # s == '1 2 3'
```

```
a = s.split() # a == ['1', '2', '3']
```

Если при запуске этой программы ввести строку 1 2 3, то список а будет равен ['1', '2', '3']. Обратите внимание, что список будет состоять из строк, а не из чисел.



Методы join

```
a = ['red', 'green', 'blue']  
print(' '.join(a))
```

вернёт red green blue

```
print("".join(a))
```

вернёт redgreenblue

```
print('***'.join(a))
```

*# вернёт red***green***blue*

В Питоне можно вывести список строк при помощи однострочной команды. Для этого используется метод строки join. У этого метода один параметр: список строк. В результате возвращается строка, полученная соединением элементов переданного списка в одну строку, при этом между элементами списка вставляется разделитель, равный той строке, к которой применяется метод.



Метод join

```
a = [1, 2, 3]
```

```
print(' '.join([str(i) for i in a]))
```

Если же список состоит из чисел, то придется использовать еще тёмную магию генераторов. Вывести элементы списка чисел, разделяя их пробелами, можно так





Основы проектной деятельности

Второе занятие. Словари

Словари

Словарями в Python называются наборы значений аналогично спискам и кортежам. Отличие состоит в том, что каждому элементу словаря соответствуют ключ и связанное с ним значение.



Пример словаря

```
favorite_sports = {'Ральф Уильямс': 'Футбол',  
                  'Майкл Типпетт': 'Баскетбол',  
                  'Эдвард Элгар': 'Бейсбол',  
                  'Ребекка Кларк': 'Нетбол',  
                  'Этель Смит': 'Бадминтон',  
                  'Фрэнк Бридж': 'Регби' }
```

Для разделения каждой пары «ключ—значение» мы использовали двоеточие, записав при этом ключ и значение в одинарных кавычках. Также обратите внимание, что элементы словаря заключены в фигурные (а не круглые или квадратные) скобки.



Ключ	Значение
Ральф Уильямс	Футбол
Майкл Типпетт	Баскетбол
Эдвард Элгар	Бейсбол
Ребекка Кларк	Нетбол
Этель Смит	Бадминтон
Фрэнк Бридж	Регби

Таблица 3.1. Ключи и соответствующие им значения
в словаре любимых видов спорта

Словари

Теперь, чтобы узнать любимый вид спорта Ребекки Кларк, нужно обратиться к словарю `favorite_sports`, используя ее имя в качестве ключа:

```
print(favorite_sports['Ребекка Кларк'])
```

Нетбол



Словари

Чтобы удалить значение из словаря, тоже используется ключ. Например, удалим Этель Смит:

```
del favorite_sports['Этель Смит']
```



Словари

Ключ нужен и для замены значения в словаре:

```
avorite_sports['Ральф Уильямс'] = 'Хоккей на льду'
```



Функция №1: .keys()


.keys() — это удобный метод, который возвращает все ключи в словаре. Далее посмотрим на пример с использованием метода keys.



```
>>> p_ages = {"Андрей": 32, "Виктор": 29, "Максим": 18}
>>> print(p_ages.keys())
dict_keys(['Андрей', 'Виктор', 'Максим'])
```

Функция №2: .get()


.get() — полезный метод для получения значений из словаря по ключу. Получим доступ к возрасту с помощью метода .get().



```
>>> p Ages = {"Андрей": 32, "Виктор": 29, "Максим": 18}  
>>> print(p Ages.get("Андрей"))  
32
```

Функция №3: .clear()


Метод .clear() очищает словарь ото всех элементов.



```
>>> p_ages = {"Андрей": 32, "Виктор": 29, "Максим": 18}
>>> p_ages.clear()
>>> print(p_ages)
{}
```

Функция №4: .copy()


Метод .copy() возвращает копию словаря.



```
>>> p_pages = {"Андрей": 32, "Виктор": 29, "Максим": 18}  
>>> print(p_pages.copy())  
{"Андрей": 32, "Виктор": 29, "Максим": 18}
```

Функция №5: len()

Метод len() возвращает количество элементов словаря.



```
>>> p_ages = {"Андрей": 32, "Виктор": 29, "Максим": 18}  
>>> print(len(p_ages))  
3
```

Задача 1: Скрабл

A, E, I, O, U, L, N, S, T, R – 1 очко;

D, G – 2 очка;

B, C, M, P – 3 очка;

F, H, V, W, Y – 4 очка;

K – 5 очков;

J, X – 8 очков;

Q, Z – 10 очков.



Задача 1: Скрабл

```
d = {'A': 1, 'E': 1, 'I': 1, 'O': 1, 'U': 1, 'L': 1, 'N': 1, 'S': 1, 'T': 1, 'R': 1,  
     'D': 2, 'G': 2,  
     'B': 3, 'C': 3, 'M': 3, 'P': 3,  
     'F': 4, 'H': 4, 'V': 4, 'W': 4, 'Y': 4,  
     'K': 5,  
     'J': 8, 'Z': 8,  
     'Q': 10}
```

```
text = input().upper();  
res = 0;
```

```
for c in text:  
    res += d[c];  
    print(c, ' ', d[c], ' ', res)
```

```
print(res)
```





Основы проектной деятельности

Второе занятие. Работа с файлами

