



python

# Лекция 8

## Словари



## Словари

**Словари** — коллекция объектов, где объекты хранятся парами. И каждому объекту - «ключу» - ставится в соответствие объект «значение». В последствии можно произвести получение «значения» по «ключу».



## Как себе представить, что такое словарь

1	→	«One»
2	→	«Two»
5	→	«Five»



**Ключ**



**Значение**

Графическое представление словаря приведено слева. Здесь цифры выступают ключами, а их названия прописью выступают в качестве значений. Однако стоит помнить, что в словаре не может быть двух одинаковых ключей, но может быть произвольное количество одинаковых значений.

**Важно!** В качестве ключа могут быть использованы переменные хешируемого типа.



## Хеш-код объекта

**Хеш (он же хеш-код)** — это, обычно, число, которое генерируется на основании содержимого объекта с помощью функции свертки.

**Возможность вычисления хеш-кода существует только для объектов неизменяемого типа.**

Т.е. для чисел, строк и т. д. хеш-код вычисляем. Однако для списков и подобных объектов хеш-код не вычисляем.

Стандартная функция Python для вычисления хеш-кода выглядит как :

**hash(объект)**



## Пример вычисления хеш-кода

```
text = "Hello world"
```

```
a = hash(text)
```

```
print(a)
```



**Вычисление хеш-кода строки**



## Как создать литерал словаря

Для создания литерала словаря используется следующая форма:

```
имя_словаря = {ключ_1:значение_1, ключ_2:значение_2}
```

Пример создания литерала словаря:

```
dict_one = {1:"one",5:"five"}
```



**Имя**



**Ключ**



**Значение**



## Как извлечь значение по ключу

Существует два способа извлечения значения по ключу:

1) имя\_словаря[ключ]

Пример:

```
dict_one = {1:"one",5:"five"}
```

```
a = dict_one[1]
```

```
print(a)
```

**Извлечение значения связанного  
с ключом «1»**



**Внимание!** Если вы укажете несуществующий ключ, то будет сгенерирована ошибка.



## Как извлечь значение по ключу

### 2) Использовать синтаксис

имя\_словаря.get(ключ)

Пример:

```
dict_one = {1:"one",5:"five"}  
a = dict_one.get(5)  
print(a)
```

**Внимание!** Если вы укажете несуществующий ключ, то будет возвращено значение **None**. Этот способ безопаснее. Также, можно задать значение по умолчанию, например так -  
`a = dict_one.get(2, "default")`





## Как создать пустой словарь

Пустой словарь создается с помощью синтаксиса

```
имя_словаря = {}
```

Пример:

```
dict_one = {}
```



## Как добавить пару ключ-значение в словарь

Для добавления пары ключ-значение в словарь используется синтаксис вида:

`имя_словаря[ключ] = значение`

Пример:

```
dict_one = {}
```

```
dict_one[2] = "two"
```

```
print(dict_one)
```

**Добавление в словарь на ключ 2,  
значения «two»**



## Вопросы о словарях

**В качестве ключей можно использовать только числа?**

Нет, можно использовать любой хешируемый тип данных.

**Могут ли в словаре одинаковые ключи?**

Нет, все ключи должны быть разными.

**Что будет, если добавить пару ключ-значение в словарь, в котором уже есть такой ключ?**

В таком случае значение, соответствующее такому ключу, будет заменено на новое значение.

**Что может выступать в качестве значения?**

Любой тип данных в Python.

## Методы словаря

Название метода	Описание
<code>clear()</code>	Очистка словаря
<code>copy()</code>	Возвращает копию словаря
<code>items()</code>	Возвращает пары словаря
<code>keys()</code>	Возвращает список ключей словаря
<code>get(key)</code>	Вернет значение связанное с <code>key</code>
<code>pop(key)</code>	Удаляет пару с ключом <code>key</code> и возвращает значение
<code>values()</code>	Вернет список значений словаря
....	И еще несколько дополнительных методов



## Пример использования методов словаря

```
dict_one = {}  
l = [1,2,3,6,3,1,7]  
for i in l:  
    count = dict_one.get(i)  
    if count:  
        dict_one[i] = count + 1  
    else:  
        dict_one[i] = 1  
print(dict_one)
```

Эта программа вернет вам словарь, в котором ключами являются элементы списка, а значениями то, сколько раз они встречались в списке. Можно сделать этот же функционал, но по-другому — через подсчет количества вхождений в список



## Как еще можно использовать словари

Очень часто словари используют для представления сложных наборов данных. Например, так можно описать студента:

```
student = {"name": "Alexander",  
           "lastname": "Ts",  
           "age": 36,  
           "group": "PN121"}
```

```
print(student.get("age"))
```

Такой способ использования словарей довольно распространен.



## Словарь в словаре

Иногда для описания сложных структур используется словарь, значением которого также является словарь. Например, нужно описать человека. Однако при описании нужно указать его адрес. При этом желательно, чтобы адрес можно было получить и обработать отдельно.

```
human = {"name": "Alexander",  
         "lastname": "Ts",  
         "age": 36,  
         "adress": {"street": "Lomonosova",  
                    "house": 87,  
                    "flat": 705}  
}
```

**Внутренний словарь для адреса**



## Список использованной литературы

- 1) Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011 - 137 - 144 с.
- 2) Саммерфилд М. Программирование на Python 3. Подробное руководство. - Пер. с англ. - СПб.:Символ-Плюс, 2009. - 152 - 162 с





## Домашнее задание

- 1)Используя словарь, напишите программу, которая выведет на экран название дня недели по его номеру. (1 - «Monday»)
- 2)Представьте описание кота (домашнее животное) на основе словаря.
- 3)Напишите программу которая считает строку текста с клавиатуры и выведет на экран статистику, сколько раз какая буква встречается в этой строке. Например, для строки «Hello world» эта статистика выглядит, как: «H» - 1 , «e» - 1, «l» - 3 и т. д.



## Дополнительное домашнее задание.

- 1) Ввести с клавиатуры число (до миллиона), которое обозначает количество долларов и центов пользователя. Вывести это количество прописью.

Например:

How much money do you have?

123,34

You have: one hundred twenty three dollars thirty four cents

- 2) Напишите программу, которая переведет целое число (от 1 до 100) из римской записи в обычные цифры.

Например: XXII -> 22

Подробнее: [https://en.wikipedia.org/wiki/Roman\\_numerals](https://en.wikipedia.org/wiki/Roman_numerals)