



# работа на уроке (03.12)

1. `dict[key]` `for-loop` `if / for key in dict:`

Дан словарь с парами имя пользователя-пароль:

```
passwords = {'Alex': '123', 'Batman': '5uP3rm4n_5t1Nk5', 'little_cat': 'meow111'}
```

Напишите код, который запрашивает от пользователя имя пользователя и пароль и сверяет их с данными в словаре. Если имя пользователя есть в словаре и пароль совпадает с соответствующим значением в словаре, то выводим на экран: "добро пожаловать!". Если пароль не совпадает, пишем: "неверный пароль". Если введённого имени пользователя нет в словаре, пишем: "неизвестное имя пользователя".

```
Имя пользователя: little_cat
Пароль: bow-wow
неверный пароль
```

```
Имя пользователя: little_dog
Пароль: bow-wow
неизвестное имя пользователя
```

```
Имя пользователя: little_cat
Пароль: meow111
добро пожаловать!
```

2. `for key in dict:` `dict.values()` `sum()` `dict[key]`

Для производного частотного словаря `dictionary` (в нем ключи — это что угодно, а значения — неотрицательные числа), в котором частоты — абсолютные, переведите значения частот в относительные.

dictionary	Вывод
<code>{'a': 2, 'b': 3}</code>	<code>{'a': 0.4, 'b': 0.6}</code>
<code>{'paper': 10, 'rock': 4, 'scissors': 2}</code>	<code>{'paper': 0.625, 'rock': 0.25, 'scissors': 0.125}</code>

3. `with open(...) as ...: .readlines() os.path.isfile()`

Файлы можно открывать в трёх режимах: чтение ( `'r'` ), дозапись ( `'a'` ) и перезапись ( `'w'` ). Пользуясь этими способами открыть файлы, про файл `cringe.txt` напишите такой код, который будет писать, сколько в нём строк, если этот файл есть, и писать *"нет такого файла"*, если этого файла нет (проверять существование файла можно через модуль `os` ).

4. `with open(...) as ...: .append() if key in dict: dict[key] .split()`

- Имеется файл с оценками: фамилия ученика и его оценка через пробел (у одного ученика может быть больше одной записи). Скачайте и прочитайте этот файл. Запишите его в словарь, где ключи — их фамилии, а значения — списки с их оценками (могут содержать одно значение).
- Пользователь через пробел вводит фамилию студента и его оценку. Если студент есть в словаре, добавить в его список оценок введенную оценку, если нет — добавить студента в словарь.
- Вывести пары: фамилия студента, его средняя оценка.
- Сделать предыдущий пункт, но в лексикографическом (=алфавитном) порядке.

5. `string.punctuation dict.fromkeys(keys, value) while-loop dict[key]`

Заведите словарь для хранения частот знаков препинания (их можно найти в модуле `string` ), где изначальный ключ — это знак, а значение — ноль.

Программа должна спрашивать у пользователя тексты, которые вводятся вручную (пока он не скажет *«отстань»*). При получении очередного текста обновляйте значения частот знаков препинания: увеличивайте соответствующее число на 1.

6. `string.punctuation if key in dict: while-loop dict[key] sorted() key=itemgetter(1)`

Заведите словарь для хранения частот знаков препинания. В отличие от первой задачи, мы не знаем наперёд, какие именно символы будем обрабатывать: словарь пустой.

Программа должна спрашивать у пользователя тексты, которые вводятся вручную (пока он не скажет *«отстань»*). При получении очередного текста обновляйте

значения частот знаков препинания: впервые встретив символ, запоминайте ему частоту 1, а когда увидите его снова, увеличивайте его запомненную частоту на +1.

Напечатайте получившийся словарь **по убыванию частотности символов**.