



# работа на уроке (28.11)

## 1. `for-loop` `list[start:stop:step]`

Вводится последовательность символов. Проверить есть ли в строке символьная биграмма (последовательность из двух символов), встречающаяся дважды. Записать ответ в булевом типе данных в переменную `occurred_twice`, вывести ее значение на экран.

Решите задание двумя способами: при помощи **с** и **без** вложенных циклов.

Ввод	Вывод
гаплоглогия	True
контрфактивность	False

## 2. `set()`

Даны два слова:

`'превысокомногорассмотрительствующий'` и `'водогрязеторфопарафинолечение'`.

- Посчитать, сколько различных букв в слове `'превысокомногорассмотрительствующий'`.
- Какие различные буквы имеют слова `'превысокомногорассмотрительствующий'` и `'водогрязеторфопарафинолечение'` ?
- Какие общие буквы имеют слова `'превысокомногорассмотрительствующий'` и `'водогрязеторфопарафинолечение'` ?
- Есть ли буквы в слове `'превысокомногорассмотрительствующий'`, которых нет в слове `'водогрязеторфопарафинолечение'` ? А наоборот?

## 3. `.items()` `sorted()`

Создайте два списка, в которых будут содержаться ключи и значения будущего словаря соответственно. Пройдите циклом по индексам этих списков и создайте словарь. Далее необходимо вывести на экран пары кортежей (ключ, значение) из этого словаря в алфавитном порядке.

Списки	Вывод
<code>['o', 'и', 'е', 'а', 'у', 'я', 'ы', 'ю', 'э', 'ё']</code> <code>[10.97, 7.35, 8.45, 8.01, 2.62, 2.01, 1.90, 0.64, 0.32, 0.04]</code>	<code>('а', 8.01)</code> <code>('е', 8.45)</code> <code>('ё', 0.04)</code> <code>('и', 7.35)</code> <code>('о', 10.97)</code> <code>('у', 2.62)</code> <code>('ы', 1.90)</code> <code>('э', 0.32)</code> <code>('ю', 0.64)</code> <code>('я', 2.01)</code>

4. `.values()` `tuple()`

В переменной `my_dictionary` находится словарь (например, `{'a': 'а', 'e': 'е', 'ё': 'о', 'и': 'и', 'o': 'о', 'y': 'у', 'ы': 'у', 'э': 'е', 'ю': 'у', 'я': 'а'}`). С клавиатуры вводится значение из этого словаря. Выведите ключ для данного значения. Если их несколько, вернуть кортеж. Если таких значений в словаре нет — вывести `None`.

Ввод	Вывод
у	ы
и	(у, ю)
q	None

5. `.split()` `'''text'''` `.count()` `.append()`

Дан текст.

законы физики рядом с черной дырой уже не работают в том представлении в котором мы привыкли  
пространство время искривляется и все события протекают уже совсем по другому  
словно пылесос черная дыра поглощает все что находится около нее планеты астероиды свет и прочее  
ранее считалось что черная дыра ничего не излучает но как доказал стивен хокинг черная дыра излучает антивещество  
то есть поедает вещество и выделяет антивещество  
ранее думали что если кинуть в черную дыру человека то ему труба порвет на субатомы  
но как оказалось по некоторым уравнениям есть определенные траектории путешествия сквозь черную дыру чтоб чувствовать себя нормально  
правда не ясно что будет за ней другой мир или ничего  
область вокруг черной дыры которая интересна называется горизонтом событий  
если туда полететь не зная волшебное уравнение то будет конечно не очень  
наблюдатель будет видеть как космический корабль влетает в горизонт событий и крайне медленно потом отдаляется пока не застынет в центре  
у самого же космонавта дела будут идти совсем по другому  
искривленное пространство будет лепить из него как из пластилина различные формы пока наконец не разорвет все на субатомы  
но для внешнего наблюдателя космонавт навсегда останется улыбающимся и махающим в иллюминатор застывшим изображением

- a. Запишите его в переменную. Обратите внимание, что текст многострочный -- переносы строки нужно сохранить!
- b. Разбейте на слова и запишите результат (список уникальных слов) в другую переменную.
- c. Пользователь вводит с клавиатуры какое-нибудь слово.
- d. Проверяем, есть ли слово в массиве: если нет, добавляем его туда и печатаем "Слово x добавлено", а если есть, то печатаем ответ "Слово x в тексте n раз".

Ввод	Вывод
лингвистика	Слово лингвистика добавлено
не	Слово не в тексте 7 раз

6. string[index] sum()

Дана матрица  $3 \times 3$ , дополните ее таким образом, чтобы последний элемент каждой строки равнялся сумме предшествующих ему элементов этой строки, а каждый нижний элемент столбца равнялся сумме предшествующих ему элементов этого столбца.

```
matrix = [[34, 67, 23],
          [20, 71, 80],
          [26, 49, 51]]
```

Таким образом, нужно программным способом получить:

```
[[34, 67, 23, 124],
 [20, 71, 80, 171],
 [26, 49, 51, 126],
 [80, 187, 154]]
```

(со звездочкой) попробуйте сделать так, чтобы эта программа работала не только с матрицей  $3 \times 3$ , но и с любыми матрицами  $n \times n$ .