



работа на уроке (24.12)

1. `with open(...) as ...: wget.download() .read()`
`.strip(string.punctuation) len() .split()`

Напишите программу которая будет принимать на вход имя файла, а возвращать среднюю длину слова в этом файле. При этом надо удалить пунктуацию и считать, что слова разделены пробельными символами.

В какой из песен (`you.txt` или `we.txt`) слова в среднем длиннее?

2. `with open(...) as ...: .read()/ .readlines() .write()/ .writelines()`
`sorted() .extend()`

У пользователя есть несколько файлов. Он хочет объединить их в один и **отсортировать** контент выходного файла по алфавиту. Напишите программу, которая спросит у пользователя все пути к файлам и соединит содержимое всех введенных файлов в один с именем, которое тоже нужно спросить у пользователя. Программа должна работать с любыми текстовыми файлами.

Пример:

Содержимое исходных файлов

file_one.txt

```
hello
hi
привет
```

file_two.txt

```
good bye
bye
пока
```

Содержимое полученного файла

merge_sorted_result.txt

```
bye
good bye
hello
hi
пока
привет
```

3. `with open(...) as ...: wget.download() .readlines() .writelines()`
`.split() ' '.join()`

Скачайте файл и откройте его. Замените все слова в нем, чья длина (вместе со знаками препинания) меньше 4 символов словом `BAD`, а слова, чья длина больше 8 символов, словом `NAUGHTY`. Сохраните полученный текст в файл `trick_or_treat.txt`.

Попробуйте читать текст построчно и не загружать его полностью в память.

4. `import random`
`with open(...) as ...: .readlines() wget.download()`

Запомните текстовый файл в список строк. Файл может быть любым, но, конечно, лучше, если в нём будет побольше строк. Например, можно что-то такое.

Используя библиотеку `random` ([официальная документация](#), [краткая выжимка по-русски](#)), напечатайте `N` случайных строк из этого файла, `N` спросите у пользователя. Теперь вы умеете гадать по онлайн-книжкам!