

Заняття №18. Файли

Корисні посилання:

<https://www.programiz.com/python-programming/file-operation>

Правила здачі:

- Назва файлу має відповідати назві завдання (програми).
- Форматування файлу має відповідати **PEP8**.
- Файли з рішеннями потрібно завантажити на ваш git репозиторій з назвою **itstep**, в окрему папку з назвою **lesson18**. Репозиторій має бути публічним, як це перевірити можна дізнатися в офіційній документації вашого git провайдера, або за допомогою Google.
- Програма ні в якому разі не має закінчуватися помилкою та завершитися будь-яким статусом окрім 0.
- Посилання на репозиторій помістіть в окремий файл під назвою **solution_lesson18_<your_login>** де **<your_login>** це ваш студентський логін та прикріпіть як рішення в mystat. Уважно перевірте, що ви прикріплюєте саме рішення до поточного заняття!
- всі завдання спираються на теми, що ми вже вивчали.
- за будь-яку невідповідність до вищезазначених пунктів оцінка знижується.

Завдання.

1. Напишіть програму **file_lines_amount.py**, яка вираховує кількість непорожніх рядків в файлі **lines.txt**, який знаходиться в одній директорії з програмою.

Приклад:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The terminal shows the command 'cat -e lines.txt' followed by the content of the file 'lines.txt' which consists of four lines of text, each ending with a dollar sign (\$) as an escape character. Below the file content, the command 'python3 file_lines_amount.py' is entered, and the output '4' is displayed, representing the number of non-empty lines in the file.

```
→ cat -e lines.txt
Despite the storms,$
$
beauty arrives like$
it was always going to.$
Despite the darkness,$

→ python3 file_lines_amount.py
4
```

2. Напишіть функцію **beautiful_list_generator**, що приймає список, символ (маркер) та назву файлу. Функція створить файл з назвою, що в неї передали і маркером та помістить цей список в сприйнятний для більшості формі.

Приклад:

```
→ cat beautiful_list_generator_example.py

def beautiful_list_generator(lst: list, marker: str, file_name: str) -> bool:
    ...

if __name__ == "__main__":
    dc_heroes = ["Batman",
                "Superman",
                "Flash",
                "Green Lantern",
                "Wonder Woman"]
    beautiful_list_generator(dc_heroes, "\u2713", "heroes.txt")

→ python3 beautiful_list_generator_example.py
→ cat heroes.txt
✓ Batman
✓ Superman
✓ Flash
✓ Green Lantern
✓ Wonder Woman
→
```

3. Напишіть програму **word_counter.py**, яка приймає як аргумент назву файлу, та роздруковує кількість слів в файлі.

Приклад:

```
→ python3 word_counter.py file_with_words
10
```

4. Дуже часто ми можемо забути про певну задачу чи навіть декілька з них. І як приємно відмітити, що ви щось виконали, досягли мети. В цьому завданні вам випаде нагода зробити маленьки ToDo list.

Основні можливості:

- користувач може додати нову задачу
- користувач може відредагувати задачу

- користувач може видалити задачу
- користувач може позначити задачу як виконану
- користувач може переглянути список виконаних та невиконаних задач і за виведенням розуміти яка з задач має який статус. Для того щоб інформація не втрачалася використовуйте файл для того щоб зберегти інформацію. В цьому завданні ви можете використовувати бібліотеку **csv**.