

## Заняття №19. Серіалізація та десеріалізація.

### Персональна частина

Корисні посилання:

<https://docs.python.org/3/library/json.html>

[https://www.w3schools.com/python/gloss\\_python\\_convert\\_into\\_JSON.asp](https://www.w3schools.com/python/gloss_python_convert_into_JSON.asp)

Правила здачі:

- Назва файлу має відповідати назві завдання (програми).
- Форматування файлу має відповідати **PEP8**.
- Файли з рішеннями потрібно запушити на ваш git репозиторій з назвою **itstep**, в окрему папку з назвою **lesson19**. 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\_lesson19\_<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 **shopping\_ultra.py**, що буде надавати можливість додавати нові списки, видаляти вже існуючі, знаходити найдорожчий і найдешевший список з покупками, зчитувати дані при першому вході з файлу (де дані серіалізовані) та після завершення роботи серіалізувати та зберігати у файл (формат серіалізації json). Більше інформації в файлі з практичною за посиланням

<https://docs.google.com/document/d/1L2RBjjXfA7i56UrbY0uQcAXB5QnEp4nD-8HvYXpqzGA/edit?usp=sharing>

## Заняття №19. Сериалізація та десериалізація.

### Командна частина

Корисні посилання:

[<Натисни на мене>](#)

Правила задачі:

- Назва файлу має відповідати назві завдання (програми).
- Форматування файлу має відповідати **PEP8**.
- Ви можете розподілити задачі по коду як вирішите в команді, але кожен з вас має написати певну частину і розумітися з всім кодом цілком.
- Файли з рішенням потрібно запушити на git репозиторій одного з учасників команди під назвою **better\_than\_spotify**, в окрему папку з назвою **project**. 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 **team\_lesson19\_<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.
- Є шанс що перевірка проекту буде додатково проведена з викладачем особисто, тому будьте готові пояснити як працює кожна частина програми.
- Про склад вашої команди вас проінформують окремо.

- Перед тим як приступити до виконання, поставте собі питання: Як працювати в команді? Як вірно комунікувати? Як розділити обов'язки? Як бути ввічливим? Що важливе в командній роботі? І тільки після того як відповісте собі чи знайдете відповідь на них приступайте до виконання.

#### Завдання:

В цьому завданні ви будете реалізовувати інтерактивну музичну бібліотеку. Бібліотека має містити виконавців (Назва, Учасники, Рік заснування та Альбоми). Альбомів в свою чергу може бути декілька. Ваша програма має зчитувати дані з файлу з серіалізованими даними та зберігати їх в цей файл під час додавання або зміни інформації про виконавця чи Альбома.

Програма має також давати змогу, додавати/змінювати/видаляти виконавця та альбоми в ньому.

Виконавцем може бути як сольний артист так і група, тому учасників у виконавці може бути декілька.

Вся структура файлів та їх назви за вами.