ДЗ від 11.09.2021  
У поточному проекті у файлі students.py :

1. Змінити функцію print\_student(student) так, щоб вона виводила назви полів у людино-читаємому вигляді: кожне з великої літери, з пробілами замість "\_"
2. Написати функцію print\_students\_list() , яка виводитиме весь список студентів (викликатиме у циклі  print\_student(student) для кожного студента та візуально відокремлюватиме вивід інформації про кожного зі студентів)
3. Результат виконання залити у свій проект на гітхабі

У LMS:

* переглянути матеріали до  заняття № 10 (контестні менеджери та файли), пройти тести

Додатково:

* подивится на [CSV файли](https://en.wikipedia.org/wiki/Comma-separated_values) , [модуль у python для роботи з csv](https://docs.python.org/3.7/library/csv.html) .

Мій проект на гітхабі: <https://github.com/jane-at-beetroot/basics>

################################################################################

ДЗ від 13.09.2021:

1. у файлі lms.students.py змінити шляхи до файлів у всіх функціях так, щоб файли потрапляли не в кореневу папку проекту, а в папку lms/data.
2. закінчити функцію, яка виконує завантаження даних з файлу у форматі CSV.
3. у файлі lms/data/exam.json описати ще 4 питання по основам Python з трьому варінатами відповідей кожне (правильна відповідь для кожного питання тільки одна).
4. ~~подивитися теоретичний матеріал щодо класів у Beetroot LMS~~

################################################################################

ДЗ від 18.09.2021:

1. У проекті LMS створити класи Student (переймає всі атрибути зі словника student) та  Group(список студентів та назва групи). Розділити наявні функції на методи цих двох класів.
2. Документацію по класах <https://docs.python.org/3/tutorial/classes.html> читати/усвідомлювати до пункта 9.7 включно.
3. ~~У Beetroot LMS опрацювати уроки 12, 13.~~
4. Збережіть на майбутнє посилання на офіційний документ по Python MRO: <https://www.python.org/download/releases/2.3/mro/>

################################################################################

ДЗ від 22.09.2021:

1. У классі Student додати можливість порівняння студентів за віком
2. У классі Student додати збереження списку всіх створених об'єктів
3. У классі Student додати альтернативну можливість створення нового об'єкту зі словника
4. ~~У Beetroot LMS опрацювати наступне заняття "Функції як об’єкти першого класу"~~

################################################################################

ДЗ від 25.09.2021:

1. У классі Student додати перевірку, що в поле email неможливо записати щось, що не є валідною мейл адресою (валідна адреса - рядок, який обов'язково містить символ "@" , але не на першій чи останній позиції
2. Реалізувати класс Lesson, який буде зберігати в собі дані про відвідування студентом заняття та отриману оцінку для кожної дати
3. ~~У Beetroot LMS опрацювати заняття 15 та 16~~

################################################################################

ДЗ від 27.09.2021:

1. Подумати і відписати мені, над якою предметною областю вам було б попрацювати (створення серверу і примітивного сайту, веб-скраппер, простий data science, свій варіант)
2. Модифікувати створений на занятті декоратор так, щоб після виклику задекорованої функції він виводив інформацію про цей виклик: час виклику (<https://docs.python.org/3/library/time.html#time.localtime>), ім'я функції, яка виконувалася (func.\_\_name\_\_) та аргументи, які було передано задекоровній функції

################################################################################

Программа принимает в письменном виде через input что нужно делать

Cоздать нового студента, это запустить поля с вопросами имени, фамилии, мейла, возраста, адреса и пола. Все это записать в созданный экземпляр класса Студент, а экземпляр добавить в список студентов. Список тоже экземпляр класса.

Можно загрузить студентов из созданного заранее списка списков со всеми данными.

Если понадобится, то нужно напечатать студентов

Сохранять студентов в файлах JSON и CSV, и потом из них же и скачивать если поадобится

Программа должна уметь создавать список со студентами:

* с помощью создания в ручном режиме, вводя данные в ручную
* с помощью добавления их из тестового списка списков со студентами
* скачивая студентов из файла JSON
* скачивая студентов из файла CSV

20.09.2021

class Num():

​

def \_\_init\_\_(self, n):

self.n = n

​

​

class NumList():

​

nums = []

​

def add\_num(self, n):

self.nums.append(Num(n))

​

def print\_nums(self):

print(self.nums)

​

​

if \_\_name\_\_ == '\_\_main\_\_':

l = NumList()

for number in range(4):

l.add\_num(number)

l.print\_nums()

27.09.2021 от учительницы

Привет! Во-первых - совет: не думай об оптимальной структуре. В реальной разработке практически невозможно придумать структуру, которая будет обеспечивать все потребности проекта на протяжении всей его жизни. Лучше создать примитивную структуру и потом по необходимости изменять ее.  
По существу: как я говорила уже, иметь отдельный класс для регистрации объектов другого (в твоем случаем это StudentList) - это более сложное решение, чем оперировать одним классом. Поэтому я предлагаю класс StudentList удалить вообще.

У класса Student определить атрибут ALL\_STUDENTS = [] и добавлять туда каждый созданный объект этого класса (в init добавить строку Student.ALL\_STUDENTS.append(self)). Создание нового студента с использованием ввода можно определить как сlassmethod, в котором собирать все необходимые input и потом просто вызывать с этими данными обычный init. Все групповые методы (массовое создание из тестового списка, загрузка из файла, дамп в файл) в таком случае идут в класс Group, описывающий группу студентов.

Класс StudentList удалить

В классе Student определить атрибут ALL\_STUDENTS = [] и добавлять туда каждый созданный объект этого класса (в init добавить строку Student.ALL\_STUDENTS.append(self)).

Что бы создать нового студента через запрос к пользователю, input() можно через метод класса (через декоратор сlassmethod) в котором и собрать все необходимые input() запросы

Все групповые методы, такие как массовое создание из тестового списка, загрузка из файла, дамп и лоад в/из файла перенести в класс Group, описывающий группу студентов