Ekow

The problem:
Present the core classes for the information science major in a presentable manner and work with inputs form the user to make sure all requirements are fulfilled. Verify if student chooses the available classes and enough classes to move on to the next semester.

Solution:
Use pandas to create a database with the required core classes, restrict them by the sequence in which they are to be taken (ie. make them mandatory) and use lists to store the classes the user chooses and update the list as they move through semesters.

Learning points:

- It's better to use dictionaries to better visualize how the data frame will look before converting it as opposed to importing csv files especially if data is not very large.
- Be sure to map out (physically) what the intended plan and get that verified before starting the script

My Code: Ekow.ipynb

Shahin

The problem:
Ask for the user's name and ask questions if they completed benchmark courses for their course planner.
The Solution:
Creating a list of dictionaries containing questions about each benchmark course required as part of InfoSci major, and it will record the user's response and proceed with the next question from the list. If one benchmark question is not passed, the user will be shown a message that this course will be added to their first-semester plan.
Plan:
To store these recorded data and map them with "course class" to cross-check each benchmark course from all available courses within the InfoSci list for the upcoming course planner function.

My Code: Shahin.ipynb

Sean

The problem:

Create an iterable object in python to hold information on InfoSci courses and their requirements for enrollment.

The Solution:
Using data available through the umd.io API repo, I've created a list of dictionaries containing information on InfoSci courses and their requirements for enrollment.

What I learned.

- How to open a JSON file in Python and store it in a Python object
- How to iterate over and target items within a list of dictionaries

My Code

File:
sean_check_in.ipynb / 202008.json

Instructions:
My code is contained entirely in the one Jupyter cell. It will need the JSON file will be needed as well as my function is working with data contained in this file. I've included a call of the function with parameters that should work as long as the JSON file is in the same directory as the Jupyter file. Otherwise, enter the path directory in the first parameter and any course code in the second parameter. The output should be a list of dictionaries.