Module 2: Web Scraping

JHU EP 605.256 – Modern Software Concepts in Python

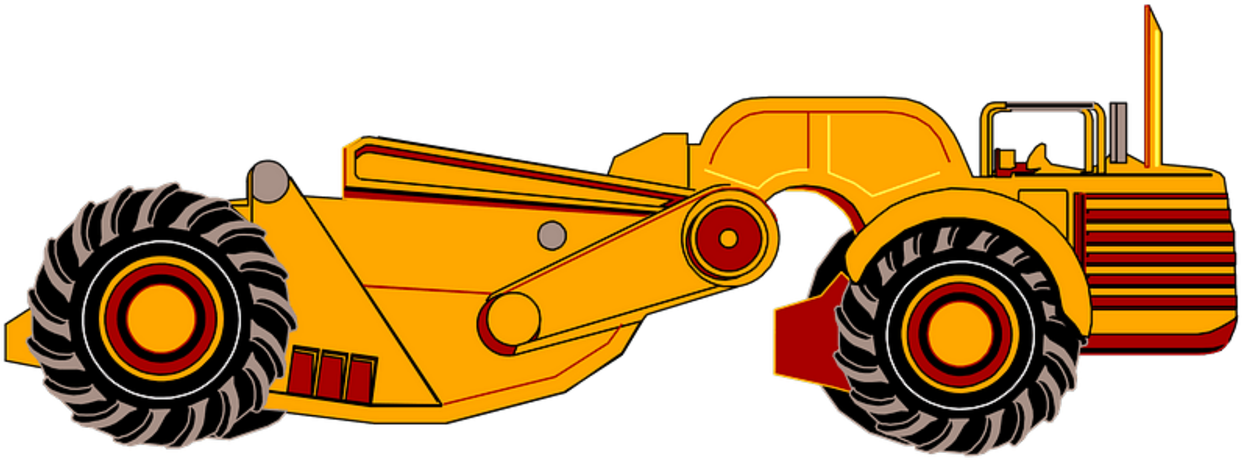
Introduction

This assignment is practices scraping data from websites. The site grad cafe allows people to upload information about programs they have been accepted / rejected / waitlisted from and basic applicant information, including date, start term, degree, program, and academic performance metrics. In this assignment, student developers (you!) are going to gather a copy of those freely uploaded student metrics and save it to be used in future requirements.

This will require you to manipulate urls, control HTML objects, and search within the structured data using several of the string methods taught in class this week. Your resultant output should clean dictionaries of information covering 1000nds of student's graduate school acceptance statistics – all ready to be analyzed in future assignments!

Skills: URL Management, HTML Searching Methods, Data Cleaning, JSON Data Object Storage

THE GRADCAFE



Assignment Overview

In this assignment you will:

1. Write a web scraper for grad cafe data about recent applicants and parse applicant data:
    a. Confirm the `robot.txt` file permits scraping.
    b. Use `urllib3` to request data from grad cafe.
    c. Use `beautifulSoup/regex/string` search methods to find admissions data.
    d. Structure the data as a clean, formatted `json` data object.
2. Submit associated deliverables on-time and with the appropriate structure.

After this assignment, you will have a structured json object containing thousands of students uploaded data in an easy-to-use format. This data will be used for analysis on program statistics and admissions in later course modules.

1. Programming Assignment Requirements

A common way to organize requirements is into a **SHALL**, **SHOULD, SHALL NOT** list. Requirements for your homework assignment are under this structure and give you development freedom over how to implement – if it conforms to the Shall/Should/Shall Not list.

SHALL: A high priority requirement that must be implemented as an essential part of this requirement and will otherwise result in an unsatisfactory "program correctness" grade within our assignment matrix within the syllabus.

SHOULD: A low priority requirement that will result in a good/excellent "program correctness" grade if implemented.

SHALL NOT: A forbidden component of this assignment. This list does not include items specifically mentioned on the syllabus under academic integrity, but the expectation is that those areas are similarly respected.

For this assignment your solution:

- **SHALL** programmatically pull data from grad cafe using python.
- **SHALL** only rely upon libraries explicitly covered within module 2 lecture.
- The data categories pulled **SHALL** include:
    - Program Name
    - University
    - Comments (if available)
    - Date of Information Added to Grad Café

  - URL link to applicant entry
  - Applicant Status
    - If Accepted: Acceptance Date
    - If Rejected: Rejection Date
  - Semester and Year of Program Start (if available)
  - International / American Student (if available)
  - GRE Score (if available)
  - GRE V Score (if available)
  - Masters or PhD (if available)
  - GPA (if available)
  - GRE AW (if available)
- **SHALL** use `urllib3` to carry out url management.
- **SHALL** use `json` to store data under file `applicant_data.json` with reasonable object keys.
- **SHALL** include at least 10,000 grad applicant entries.
- **SHALL** include a README.
- **SHALL** be available on Github within a private repository called `jhu_software_concepts` within a folder named `module_1.`
- **SHALL** comply with robots.txt (and include screenshot.jpg + README evidence that robots.txt was checked).
- **SHALL** include a requirements.txt file that allows complete reconstruction of your environment.
- **SHALL** use python 3.10+
- **SHOULD** use beautifulSoup / string methods / regex to find necessary data components
- **SHOULD** be written using either functions or class methods:
  - `scrape_data()`: pulls data from grad cafe
  - `clean_data():` converts data into structured format
  - `save_data()`: saves cleaned data into json file
  - `load_data()`:loads cleaned data from json file
  - Other private methods can be used and should be indicated using `_<private>()` with an underscore in front of the private method.

- **SHOULD** carry out scraping under file `scrape.py` and carry out data cleaning under file `clean.py.`
- **SHOULD** ensure data does not include any remnant HTML.
- **SHOULD** ensure unavailable data is maintained in a consistent format i.e. `None` or "“"
- **SHOULD** handle removal of unexpected/ messy information.
- **SHOULD** be accurate information that is true to the website.
- **SHOULD** be well commented, clear, with appropriately named variables.

- **SHOULD NOT** use find/search methods that cannot be found within beautifulSoup / string methods / regex.

2. Programming Assignment Requirements

readme.txt

So-called “read me” files are a common way for developers to leave high-level notes about their applications. Here’s an example of a README file for the Apache Spark project. They usually contain details about required software versions, installation instructions, contact information, etc. For our purposes, your readme.txt file will be a way for you to describe the approach you took to complete the assignment so that, in the event you may not quite get your solution working correctly, we can still award credit based on what you were trying to do. Think of it as the verbalization of what your code does (or is supposed to do). Your readme.txt file should contain the following:

1. **Name**: Your name and JHED ID
2. **Module Info**: The Module name/number along with the title of the assignment and its due date
3. **Approach**: a detailed description of the approach you implemented to solving the assignment. Be as specific as possible. If you are sorting a list of 2D points in a plane, describe the class you used to represent a point, the data structures you used to store them, and the algorithm you used to sort them, for example. The more descriptive you are, the more credit we can award in the event your solution doesn’t fully work.
4. **Known Bugs**: describe the areas, if any, where your code has any known bugs. If you’re asked to write a function to do a computation but you know your function returns an incorrect result, this should be noted here. Please also state how you would go about fixing the bug. If your code produces results correctly you do not have to include this section.

Please post to your github ahead of deadline. Github files should also be submitted through Canvas so graders can use the timestamp of the submitted files. Instructors will check date / time of final push to confirm all materials were submitted on time.

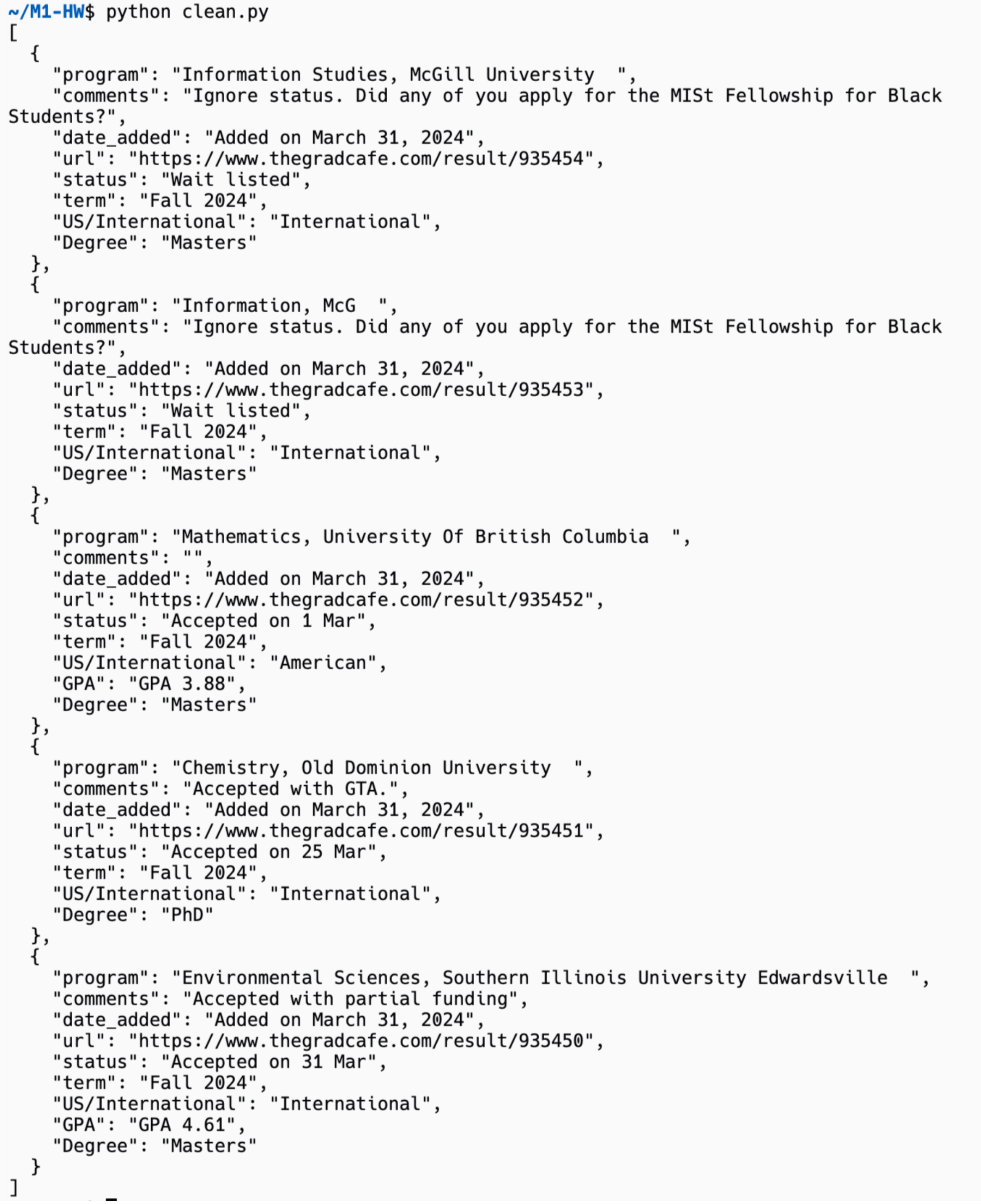
Recap:

1. The SSH URL to your GitHub repository

2. scrape.py under **`module_2`**
3. clean.py under **`module_2`**
4. applicant_data.json under **`module_2`**
5. robots.txt screenshot under **`module_2`**
6. README under **`module_2`**
7. requirements.txt under **`module_2`**

Please let us know if you have any questions via Teams or email!

Sample Cleaned Data Output



```
~/M1-HW$ python clean.py
[
  {
    "program": "Information Studies, McGill University  ",
    "comments": "Ignore status. Did any of you apply for the MISt Fellowship for Black
Students?",
    "date_added": "Added on March 31, 2024",
    "url": "https://www.thegradcafe.com/result/935454",
    "status": "Wait listed",
    "term": "Fall 2024",
    "US/International": "International",
    "Degree": "Masters"
  },
  {
    "program": "Information, McG  ",
    "comments": "Ignore status. Did any of you apply for the MISt Fellowship for Black
Students?",
    "date_added": "Added on March 31, 2024",
    "url": "https://www.thegradcafe.com/result/935453",
    "status": "Wait listed",
    "term": "Fall 2024",
    "US/International": "International",
    "Degree": "Masters"
  },
  {
    "program": "Mathematics, University Of British Columbia  ",
    "comments": "",
    "date_added": "Added on March 31, 2024",
    "url": "https://www.thegradcafe.com/result/935452",
    "status": "Accepted on 1 Mar",
    "term": "Fall 2024",
    "US/International": "American",
    "GPA": "GPA 3.88",
    "Degree": "Masters"
  },
  {
    "program": "Chemistry, Old Dominion University  ",
    "comments": "Accepted with GTA.",
    "date_added": "Added on March 31, 2024",
    "url": "https://www.thegradcafe.com/result/935451",
    "status": "Accepted on 25 Mar",
    "term": "Fall 2024",
    "US/International": "International",
    "Degree": "PhD"
  },
  {
    "program": "Environmental Sciences, Southern Illinois University Edwardsville  ",
    "comments": "Accepted with partial funding",
    "date_added": "Added on March 31, 2024",
    "url": "https://www.thegradcafe.com/result/935450",
    "status": "Accepted on 31 Mar",
    "term": "Fall 2024",
    "US/International": "International",
    "GPA": "GPA 4.61",
    "Degree": "Masters"
  }
]
```