CIT 255 - Web Scraping Training Module

Brie Ellis, Nolan Yanick, Tom Farrer, Andrew Lopez

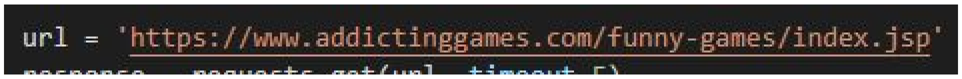
Overview:

This training module will focus on creating a simple script that is capable of extracting data from a public website and then storing that data into an external file. The level of training for our module would be placed at or around a beginner's level. Our module will run participants through a step by step process to create the necessary code, only basic HTML and CSS knowledge would be helpful, but not required. We have created a script for scraping publicly and openly available data and information from websites and storing them into a file (either JSON, .csv, or .txt). This script was built with a strong focus on the Python language, the BeautifulSoup and Requests libraries, and JSON.

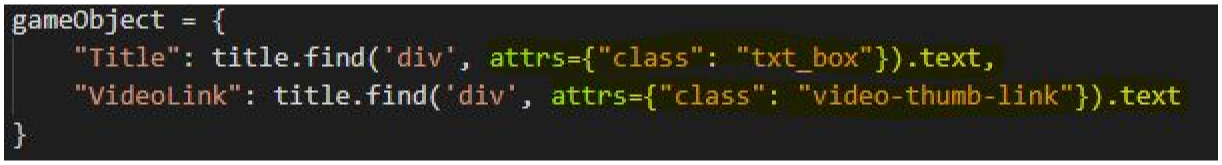
Learning Outcomes:

- Basics of Python
- The basics of the Beautiful Soup and Requests libraries
- How to write and save data to an external json file
- How to extract data from a website using Python

This instruction set is meant for use with the web scraping training module provided within this repository (https://github.com/briekellis/WebScraper).

1. Start by downloading the ZIP file and extracting the contents to the desktop.
2. Next, open Visual Studio Code.
3. Open the Explorer tab on the left hand side.
4. Expand the Folders tab and select "Open Folder". Use the File Explorer to navigate to the Documents folder and select the *project* folder. Click "Select Folder". This will open the folder in editor space.
5. Click on the 'scraper.py' file to open it in the editing window enter the following code where necessary.
    a. 

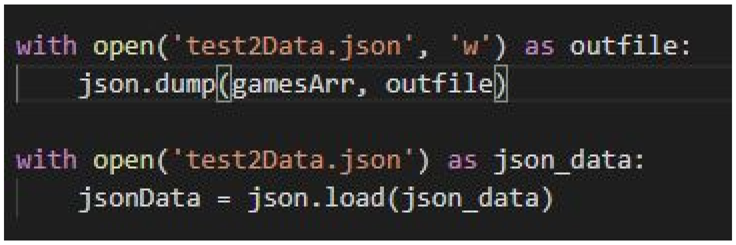
```
url = 'https://www.addictinggames.com/funny-games/index.jsp'
```



```
gameObject = {
    "Title": title.find('div', attrs={"class": "txt_box"}).text,
    "VideoLink": title.find('div', attrs={"class": "video-thumb-link"}).text
}
```

    b. For this demonstration, we have selected a website that we know will work. However, this script can be used for other websites as well. To do this, change the URL between the quotations after the "url = " variable. You will also need to change the gameObject classes that are elements within the CSS of the webpage in use. *For the sake of staying within a reasonable time frame, we will not be demonstrating how to do this during our in class lab tutorial*
    c. **Web scraping is a last resort. Use APIs if they are available.**

6. There will be a pop-up window at the bottom right of the screen for installing a Python extension. Click “Install” for both extensions and wait for the installation to complete.
7. Next, verify that the saved file will have a valid name and save any changes. Make sure the *outfile* and *json_data* have the same name.



```
with open('test2Data.json', 'w') as outfile:
    json.dump(gamesArr, outfile)

with open('test2Data.json') as json_data:
    jsonData = json.load(json_data)
```

8. From there, navigate back to the folder explorer window on the left and right click on the ‘scraper.py’ file. Select the “Run Python File in Terminal” option. Wait for the script to run.
9. Once the script has completed its cycle, click the refresh button on the folder explorer.
10. Verify that your json file has been created and is available in the explorer tab.
11. Click on one of your json file to open it in the editor window. The raw data will be available for use.
    a. A text file (data.txt) will also be available to view data in a more readable format.