Topic	WEB SCRAPING - 2	
Class Description	**Students would be reworking the previously written code to scrape more data.**	
Class	**PRO C128**	
Class time	**45 mins**	
Goal	• Scrape more data about all the exoplanets using planets hyperlinks on all webpage • Learn to visiting hyperlink through Selenium browser click automation • Learn to request module to get web page source • Use BeautifulSoup4 to extract the web page content • Create a CSV file to store data	

Resources Required	• Teacher Resources:
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen
	  ○ Smartphone
	• Student Resources:
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen	

Class structure	**Warm-Up** **Teacher-Led Activity 1** **Student-Led Activity 1** **Wrap-Up**	**10 mins** **10 mins** **20 mins** **05 mins**
Credit & Permissions:	**Exoplanet Exploration by NASA** **BeautifulSoup by Crummy (webspace of Leonard Richardson)** **Selenium under Apache 2.0 License**	
WARM-UP SESSION - 10 mins		

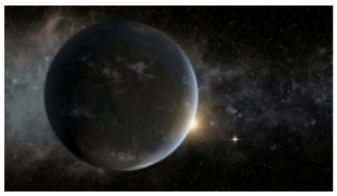
Teacher Starts Slideshow  **Slide 1 to 4** Refer to speaker notes and follow the instructions on each slide.	
Teacher Action	**Student Action**
Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today? ***Note**: Encourage the student to give answers and be more involved in the discussion.* **Following are the WARM-UP session deliverables:** • Greet the student. • Revision of previous class activities. • Quizzes.	**ESR**: Hi, thanks! Yes, I am excited about it! Click on the slide show tab and present the slides
WARM-UP QUIZ Click on In-Class Quiz	
Continue WARM-UP Session  **Slide 5 to 10**	
Activity Details **Following are the session deliverables:** • Appreciate the student. • Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.	
Teacher Action	**Student Action**
In the last class, we scraped the exoplanet data from NASA's website. Can you recall all the tools that we used in the last class?	**ESR:** - Selenium - BeautifulSoup

Note: *Encourage the student to give answers and connect the answer with today's topic.* Great! Now, in today's class, we will scrape some more data from the same website. We got some data like distance from earth, planet size, etc. but today we will scrape more data. This is important because we have to perform analysis later. Thus, we can better predict the planets, for instance, to see if they are likely habitable, etc. Are you excited?	**ESR:** Yes
Teacher Ends Slideshow 	
TEACHER-LED ACTIVITY - 10 mins	
Teacher Initiates Screen Share	
ACTIVITY • **Scraping more data from the website and letting students lead the development this time.**	
Teacher Action	**Student Action**
Open Teacher Activity 1 to show the website to the student. Do you remember this exoplanet website that we saw in the previous class? Great!! Here, if we look closely, we can see that the name of these exo-planets is a hyperlink. Let's click on the link and see what kind of data we can find?	**ESR:** We scraped the data from this website.

Search All Exoplanet Discov

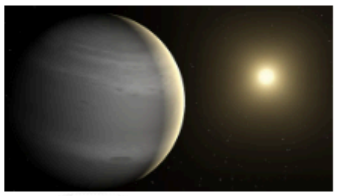
TOI-1347 b

Light-Years From Earth: 147.476
Planet Mass: 11.1 Earths
Stellar Magnitude: 11.168
Discovery Date: 2024



TOI-1347 c

Light-Years From Earth: 147.476
Planet Mass: 6.4 Earths
Stellar Magnitude: 11.168
Discovery Date: 2024



TOI-1135 b

Light-Years From Earth: 114.048
Planet Mass: 0.162 Jupiters
Stellar Magnitude: 9.54
Discovery Date: 2024

Note: *The total number of exoplanets may change due to the timely updation of data on the website. Open the link and mention the number accordingly.*

Let's click on the link and see what kind of data we can find?

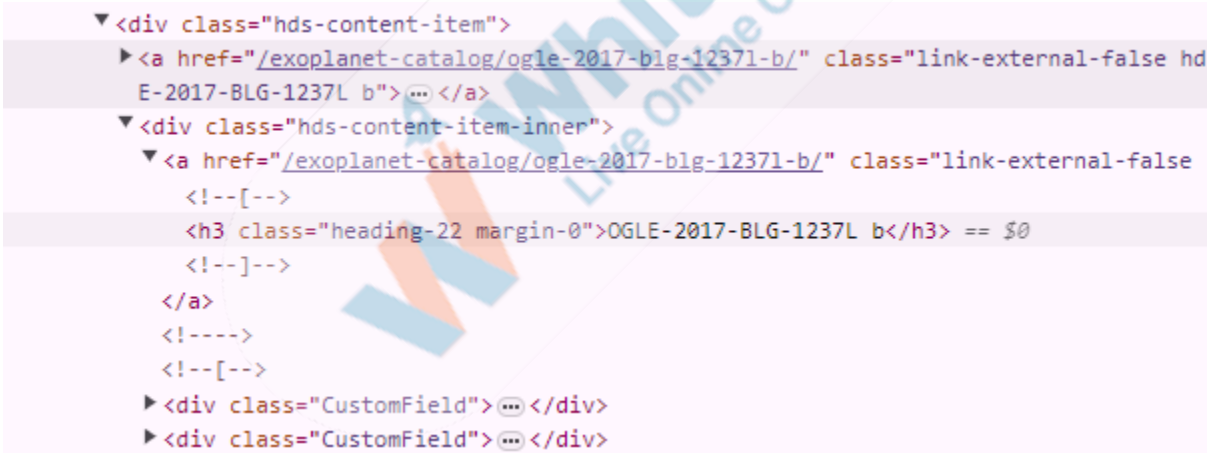
Planet Radius: 1.8 x Earth	**Planet Type:** Super Earth	**Discovery Method:** Transit	**Planet Mass:** 11.1 Earths
Discovery Date: 2024	**Orbital Radius:** Unknown	**Orbital Period:** 0.8 days	**Eccentricity:** 0.0

Great! Now, let's say we want to scrape this data as well.

Can you tell me what's the first change that we'll have to

<table>
<tr><td>make in our previous code?

That's great! Let's get started.
Open Teacher Activity 2 for boilerplate code.</td><td>ESR:
We need to save the hyperlink's href in our CSV.</td></tr>
<tr><td>Let's make some changes to our scrape() function.</td><td></td></tr>
<tr><td>We have added an extra hyperlink to our header list. Now, we also need to add this into the planet_info before we append it into the planet_data. Thus, we have to make a small change.

Before we do that, let's investigate the href for URLs in these hyperlinks.
Right-click on the hyperlink and click on inspect.
Inside the <a> tag, a link is given which gives us more detail about the exoplanet.</td><td></td></tr>
<tr><td colspan="2"> 
</td></tr>
<tr><td>Here, we can see that these links do not have https://science.nasa.gov before them.

We will have to add them.

Now to achieve this, we will do the following:
1. First, Create a variable llink and then we are using this variable to find the <a> tag with href.</td><td></td></tr>
</table>

2. Since we have to give the full URL of the page, we are adding **'https://exoplanets.nasa.gov'** to the hyperlink.
3. Then append it into **planet_info**.
4. This list is then appended in the **planet_data** list. Thus **planet_data** is a **list of lists.**

```
        information_to_extract = [ Light-Years From Earth ,  Planet Mass ,
                                   "Stellar Magnitude", "Discovery Date"]

        for info_name in information_to_extract:
            try:
                planet_info.append(planet.select_one(f'span:-soup-contains("{info_name}")')
                                   .find_next_sibling('span').text.strip())
            except:
                planet_info.append('Unknown')

        # Extract link
        link = 'https://science.nasa.gov'+ planet.find('a')['href']

        # Add link to  planet_info
        planet_info.append(link)

        planets_data.append(planet_info)

    try:
        time.sleep(2)
        next_button = WebDriverWait(browser, 10).until(EC.element_to_be_clickable((By.XPATH,
```

5. Since we are adding hyperlinks to our data, update the header with **hyperlink** and save the updated CSV file by name **updated_scraped_data.csv**

```
# Calling Method
scrape()

# Define Header
headers = ["name", "light_years_from_earth", "planet_mass", "stellar_magnitude", "discovery_date", "hyperlink"]

# Define pandas DataFrame
planet_df_1 = pd.DataFrame(planets_data, columns=headers)

# Convert to CSV
planet_df_1.to_csv('updated_scraped_data.csv',index=True, index_label="id")
```

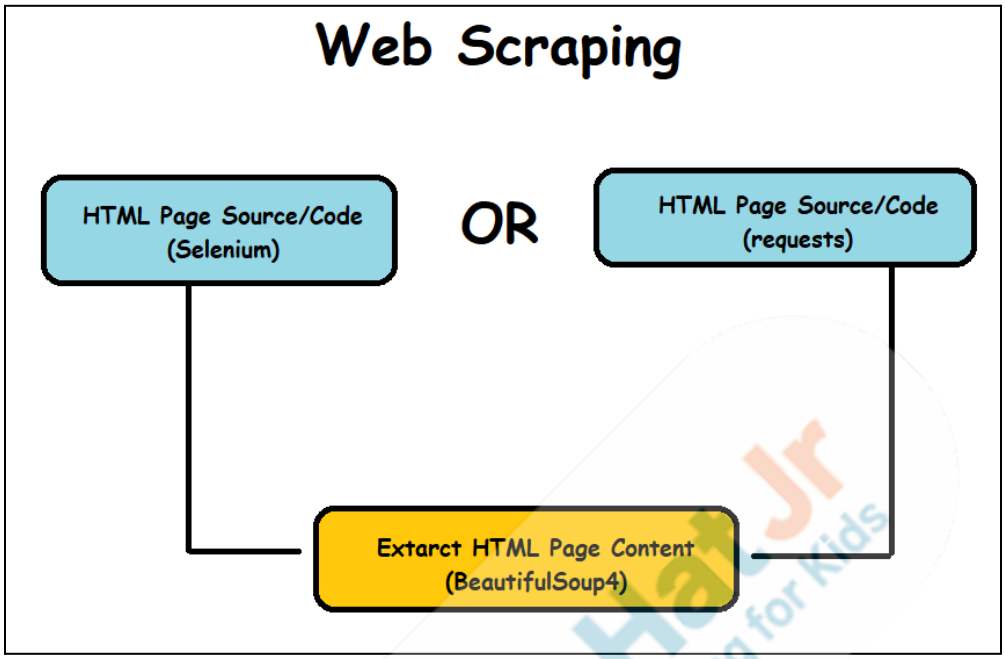
Now that we have the links in **planet_data**, can you tell me what should be our next steps?

Perfect, we will create a new function called **scraper_more_data()** function that will take these

ESR: We'll scrape data by using these links!

hyperlinks one by one, get the HTML, and then we will scrape the data. Next I'll help you to write the function to scrape data from these links. Earlier, we used selenium because we wanted to click a button on the page (next button) but this time, we do not want to interact (as we did by clicking on the webpage) with the browser, therefore we can do this without selenium.	
Teacher Stops Screen Share	
Please share your screen with me.	
Teacher Starts Slideshow **Slide 11 to 12** Refer to speaker notes and follow the instructions on each slide.	
We have one more class challenge for you. Can you solve it? Let's try. I will guide you through it.	
Teacher Ends Slideshow	
STUDENT-LED ACTIVITY - 20 mins	
• **Ask the student to press the ESC key to come back to the panel.** • **Guide the student to start Screen Share.** • **The teacher gets into Full Screen.**	
Student Initiates Screen Share	
ACTIVITY • **Create a new function to use all the hyperlinks one by one and scrape data from there**	

Teacher Action	Student Action
Open Student Activity 1 to start coding. Download the files. Open **new_scraper.py.** Since we scrapped all hyperlinks in one CSV, we only need to visit these links and scrape data from the web page of these hyperlinks using Python Script. **Extract HTML Page Content:** We use bs4 to extract data from HTML page code. To scrape data from a webpage using BeautifulSoup4, first we need to get page data. **Get HTML page code:** Earlier we used the Selenium attribute page_source to get the HTML page code. Now we will use the Python **requests** module to get the page code. ***Note**: The request module is another way of getting HTML page code before we extract data using the bs4 module.*	




```
new_scraped_data.py > scrape_more_data
1  from selenium import webdriver
2  from selenium.webdriver.common.by import By
3  from bs4 import BeautifulSoup
4  import requests
5  import time
6  import pandas as pd
7
```

In this file **scrape_more_data()** function is given to scrape data from **hyperlinks.**

A new list called **new_planets_data** is created to store new data of planets from hyperlinks.

To get the data:

1. Create a variable **page** and get the content of the HTML page for the given **hyperlink** using the **get()** method of the **requests** module.
2. Then, create a **BeautifulSoup** object called **soup** to get the HTML page code using **page.content**

attribute of **requests** module as first argument and **“html.parser”** as the second argument.

3. Also, create an empty list called **temp_list** to store the data temporarily.

```
19  def scrape_more_data(hyperlink):
20      try:
21          page = requests.get(hyperlink)
22
23          soup = BeautifulSoup(page.content, "html.parser")
24
25          temp_list = []
26
```

4. Inspect the page to find the grid of information we are looking for.

5. Here, you can see we have to access a **div** with the specific **text**. And the value is present in the next **span,** we’ll be accessing **values** only.

Note: *The teacher can refer to the previous class code as a reference code. Here instead of find_next_sibling() will directly use find_next()*

6. After getting the data into **temp_list**, we'll append it to the **new_planets_data** list.

```
    page = requests.get(hyperlink)

    soup = BeautifulSoup(page.content, "html.parser")

    temp_list = []

    information_to_extract = ["Planet Type: ", "Discovery Date: ", "Planet Mass: ","Planet Radius: ",
                              "Orbital Radius: ", "Orbital Period: ",  "Discovery Method: ",
                                ]

    for info_name in information_to_extract:
            try:
                value= soup.find('div', text=info_name).find_next('span').text.strip()
                print(value)
                temp_list.append(value)
            except:
                temp_list.append('Unknown')

    new_planets_data.append(temp_list)
```

7. **Planet_df_1** is used to store the updated CSV file in the form of DataFrame.

8. Use a **for** loop to get data from the hyperlink. Use the **iterrow()** method of pandas to iterate through the rows of the data frame and get the hyperlink.

9. Call **scrape_more_data()** and pass the hyperlink. It will scrape more data from the hyperlink.

10. To check the data, print 10 elements of **new_planets_data**.

```
43  planet_df_1 = pd.read_csv("updated_scraped_data.csv")
44
45  for index, row in planet_df_1.iterrows():
46      print(row['hyperlink'])
47      scrape_more_data(row['hyperlink'])
48      print(f"Data Scraping at hyperlink {index+1} completed")
49
50  print(new_planets_data[0:10])
51
```

Save it and run the file using the command prompt.

```
C:\Whitehat_jr\PRO-C128-Reference-Code-main\PRO-C128-Reference-Code-main>pyth
C:\Whitehat_jr\PRO-C128-Reference-Code-main\PRO-C128-Reference-Code-main\new_
e object
  browser = webdriver.Edge("C:/Whitehat_jr/PRO-127-130/msedgedriver.exe")

DevTools listening on ws://127.0.0.1:51357/devtools/browser/b4354f5b-f623-4f2
[20516:3220:0415/130949.921:ERROR:fallback_task_provider.cc(124)] Every rende
 task is shown, it is a bug. If you have repro steps, please file a new bug a
https://exoplanets.nasa.gov/exoplanet-catalog/6988/11-comae-berenices-b/
Data Scraping at hyperlink 1 completed
https://exoplanets.nasa.gov/exoplanet-catalog/6989/11-ursae-minoris-b/
Data Scraping at hyperlink 2 completed
https://exoplanets.nasa.gov/exoplanet-catalog/6990/14-andromedae-b/
Data Scraping at hyperlink 3 completed
https://exoplanets.nasa.gov/exoplanet-catalog/6991/14-herculis-b/
Data Scraping at hyperlink 4 completed
https://exoplanets.nasa.gov/exoplanet-catalog/6992/16-cygni-b-b/
Data Scraping at hyperlink 5 completed
```

Great job! Now we have **scrapped_data** without any special character.

11. A list called headers is created to save the data into a CSV file.
12. Using these headers, scrapped_data is converted into Dataframe.
13. This DataFrame is then saved into the **new_scraped_data.csv** file. The id will be the name of the first column with serial numbers.

```
63
64  headers = ["planet_type","discovery_date", "mass", "planet_radius", "orbital_radius",
65             "orbital_period", "eccentricity", "detection_method"]
66
67  new_planet_df_1 = pd.DataFrame(scrapped_data,columns = headers)
68  new_planet_df_1.to_csv('new_scraped_data.csv',index=True, index_label="id")
69
```

14. Save and run this file to check the CSV file. It opens the browser and starts scraping the data and

displaying the message each time we are scraping the data.

Note: *Guide the student to add the webdriver to the current directory and copy the path as done in the previous class. Run the Python file using a virtual environment.*

new_scraped_data.csv

```
id,planet_type,discovery_date,mass,planet_radius,orbital_radius,orbital_period,eccentricity,detection_method
0,Gas Giant,2007,19.4 Jupiters,1.08 x Jupiter,1.29 AU,326 days,0.23,
1,Gas Giant,2009,14.74 Jupiters,1.09 x Jupiter,1.53 AU,1.4 years,0.08,
2,Gas Giant,2008,4.8 Jupiters,1.15 x Jupiter,0.83 AU,185.8 days,0.0,
3,Gas Giant,2002,4.66 Jupiters,1.15 x Jupiter,2.93 AU,4.9 years,0.37,
4,Gas Giant,1996,1.78 Jupiters,1.2 x Jupiter,1.66 AU,2.2 years,0.68,
5,Gas Giant,2020,4.32 Jupiters,1.15 x Jupiter,1.45 AU,1.6 years,0.06,
6,Gas Giant,2008,10.3 Jupiters,1.11 x Jupiter,2.6 AU,2.7 years,0.08,
7,Gas Giant,2008,8 Jupiters,1.664 x Jupiter,330.0 AU,6505.9 years,0.0,
8,Gas Giant,2018,0.91 Jupiters,1.24 x Jupiter,0.19 AU,30.4 days,0.04,
```

Also, check the saved file in the directory.

	A	B	C	D	E	F	G	H	I
1	id	planet_type	discovery_date	mass	planet_radius	orbital_radiu	orbital_period	eccentricity	detection_method
2	0	Gas Giant	2007	19.4 Jupiters	1.08 x Jupiter	1.29 AU	326 days	0.23	
3	1	Gas Giant	2009	14.74 Jupiter	1.09 x Jupiter	1.53 AU	1.4 years	0.08	
4	2	Gas Giant	2008	4.8 Jupiters	1.15 x Jupiter	0.83 AU	185.8 days	0	
5	3	Gas Giant	2002	4.66 Jupiters	1.15 x Jupiter	2.93 AU	4.9 years	0.37	
6	4	Gas Giant	1996	1.78 Jupiters	1.2 x Jupiter	1.66 AU	2.2 years	0.68	
7	5	Gas Giant	2020	4.32 Jupiters	1.15 x Jupiter	1.45 AU	1.6 years	0.06	
8	6	Gas Giant	2008	10.3 Jupiters	1.11 x Jupiter	2.6 AU	2.7 years	0.08	
9	7	Gas Giant	2008	8 Jupiters	1.664 x Jupiter	330.0 AU	6505.9 years	0	
10	8	Gas Giant	2018	0.91 Jupiters	1.24 x Jupiter	0.19 AU	30.4 days	0.04	

Great work!!

We scraped exoplanet data from NASA's website. This data can be used to find useful insights.

Teacher Guides Student to Stop Screen Share

WRAP-UP SESSION - 05 mins

Teacher Starts Slideshow
Slide 13 to 18

Activity details

Following are the WRAP-UP session deliverables:

- Appreciate the student.
- Revise the current class activities.
- Discuss the quizzes.

WRAP-UP QUIZ
Click on In-Class Quiz

Continue WRAP-UP Session
Slide 19 to 24

Activity Details

Following are the session deliverables:

- Explain the facts and trivia
- Next class challenge
- Project for the day
- Additional Activity (Optional)

FEEDBACK

- **Appreciate and compliment the student for trying to learn a difficult concept.**
- **Get to know how they are feeling after the session.**
- **Review and check their understanding.**

Teacher Action	Student Action
So, in this project class we revisited the concepts from the previous class and you did the majority of the scraping yourself! Congratulations! You get “hats-off” for your excellent work!	*Make sure you have given at least 2 hats-off during the class for:*  

In the next class, we will be downloading more data and preprocessing it for further analysis.	
PROJECT OVERVIEW DISCUSSION Refer the document below in Activity Links Sections	
Teacher Clicks 	

ACTIVITY LINKS		
Activity Name	**Description**	**Links**
Teacher Activity 1	Exoplanet Exploration	https://exoplanets.nasa.gov/discovery/exoplanet-catalog/
Teacher Activity 2	Boilerplate Code	https://github.com/React-Native-Frontier/1-1-V3-C128-TA
Teacher Activity 3	Reference Code	https://github.com/React-Native-Frontier/1-1-V3-C128-TA
Teacher Reference 1	Project	https://s3-whjr-curriculum-uploads.whjr.online/def32133-27cf-4d33-b49b-cea2606ce399.pdf
Teacher Reference 2	Project Solution	https://github.com/procodingclass/PRO-C128-Project-Solution
Teacher Reference 3	Visual-Aid	Will be added after VA creation
Teacher Reference 4	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/20562587-c9b9-4474-8394-0e41a33d1f69.pdf
Student Activity 1	Boilerplate Code	https://github.com/React-Native-Fro

		ntier/1-1-V3-C128-SA