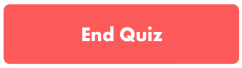
| Topic | Web Scraping 2 | |
|---|---|---|
| Class Description | **Students would be reworking the previously written code to scrape more data.** | |
| Class | **C128** | |
| Class time | **45 mins** | |
| Goal | ● Scrape more data about all the exoplanets | |
| Resources Required | ● Teacher Resources<br>○ Laptop with internet connectivity<br>○ Earphones with mic<br>○ Notebook and pen<br><br>● Student Resources<br>○ Laptop with internet connectivity<br>○ Earphones with mic<br>○ Notebook and pen | |
| Class structure | **Warm Up**<br>**Teacher-led Activity**<br>**Student-led Activity**<br>**Wrap up** | **5 mins**<br>**15 min**<br>**15 min**<br>**5 min** |

**CONTEXT**

- **Review the concepts learned in the earlier classes**

| Class Steps | Teacher Action | Student Action |
|---|---|---|
| **Step 1: Warm Up (5 mins)** | Hi <Student Name>!<br>In the last class, we scraped exoplanet's data from NASA's website. Can you recall all the tools that we used in the last class? | **ESR:**<br>- Selenium<br>- BeautifulSoup |

| | | |
|---|---|---|
| | 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 so that when we perform analysis later, we can better predict the planets, for instance, to see if they are likely habitable, etc.<br><br>Are you excited? | **ESR:**<br>"Yes!" |
| | Before we start I have an exciting quiz question for you! Are you ready to answer this question?<br><br>Teacher click on the   button on the bottom right corner of your screen to start the In-Class Quiz.<br><br>A quiz will be visible to both you and the student.<br><br>Encourage the student to answer the quiz question.<br><br>The student may choose the wrong option, help the student to think correctly about the question and then answer again.<br><br>After the student selects the correct option, the   button will start appearing on your screen.<br><br>Click the End quiz to close the quiz pop-up and continue the class. | ESR:<br>Yes! |

| | | |
|---|---|---|
| | Let's get started! | |
| **Teacher Initiates Screen Share** | | |
| **<u>CHALLENGE</u>**<br>• **Scraping more data from the website and letting students lead the development this time.** | | |
| **Step 2: Teacher-led Activity (15 min)** | *Teacher opens the same website that we scraped in the last class.*<br><br>*<Teacher opens the link from Teacher activity 1>*<br>https://exoplanets.nasa.gov/exoplanet-catalog/ | |
| | Let's look at this page again.<br><br>Here, if we look closely, we can see that the name of these exo-planets is a hyperlink. | |

1 of 428 Per page 10

| NAME ↑ | LIGHT-YEARS FROM EARTH | PLANET MASS | STELLAR MAGNITUDE | DISCOVERY DATE |
|---|---|---|---|---|
| 11 Comae Berenices b | 305 | 19.4 Jupiters | 4.74 | 2007 |
| 11 Ursae Minoris b | 410 | 14.74 Jupiters | 5.016 | 2009 |
| 14 Andromedae b | 247 | 4.8 Jupiters | 5.227 | 2008 |
| 14 Herculis b | 59 | 4.66 Jupiters | 6.61 | 2002 |
| 16 Cygni B b | 69 | 1.78 Jupiters | 6.25 | 1996 |
| 18 Delphini b | 249 | 10.3 Jupiters | 5.506 | 2008 |
| 1RXS J160929.1-210524 b | 473 | 8 Jupiters | 12.057 | 2008 |
| 24 Bootis b | 314 | 0.91 Jupiters | 5.58 | 2018 |
| 24 Sextantis b | 236 | 1.99 Jupiters | 6.441 | 2010 |
| 24 Sextantis c | 236 | 0.86 Jupiters | 6.441 | 2010 |

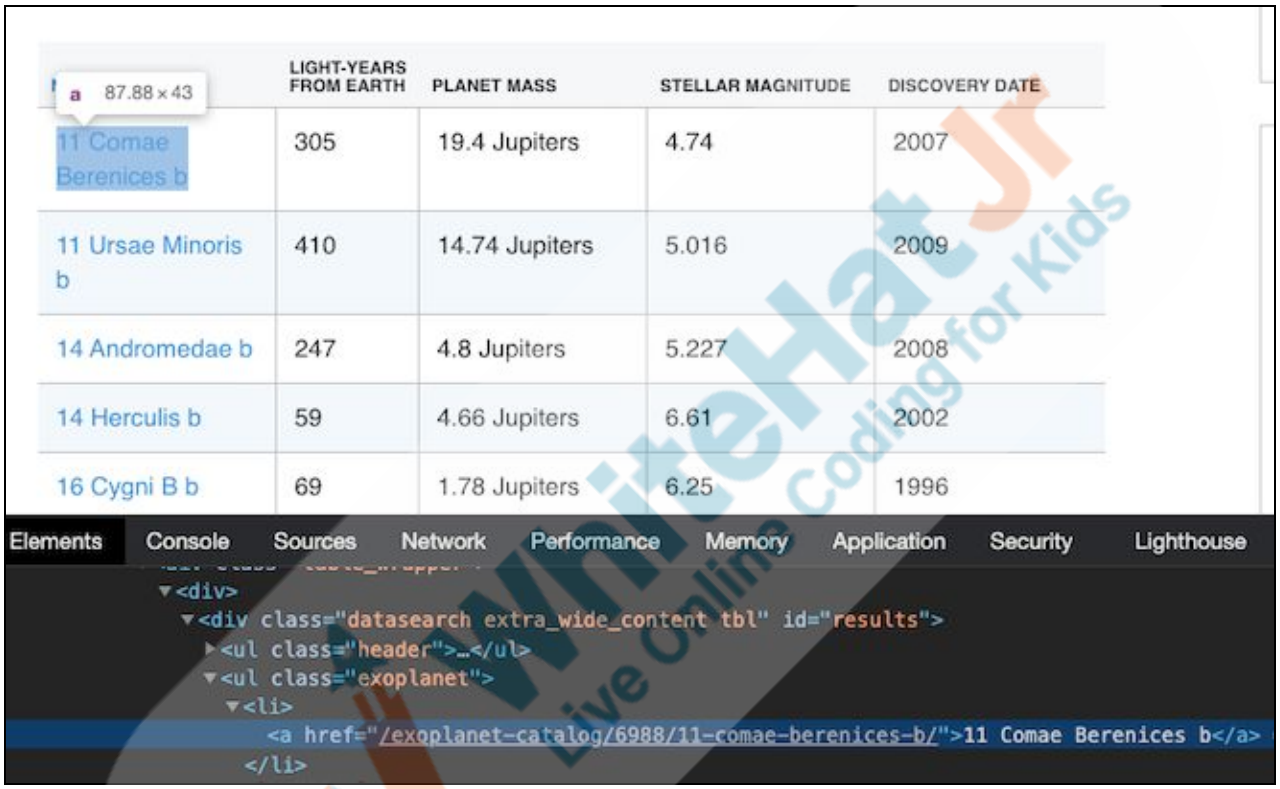
1 of 428 Back to top

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


| | | |
|---|---|---|
| | 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 make in our previous code? | **ESR:**<br>We need to save the hyperlink's href in our CSV. |
| | That's great! Let's get started.<br><br>We will add a new column in our header. Our header variable would now look like this:<br><br>`headers = ["name", "light_years_from_earth", "planet_mass", "stellar_magnitude", "discovery_date", "hyperlink"]`<br><br>We have added an extra hyperlink into our header list. Now, we also | |

need to add this into the temp_list variable list, before we append into the planet_data.

Before we do that, let's investigate the href url in these hyperlinks:




Here, we can see that these links do not have https://exoplanets.nasa.gov before them. We will have to add them.

Now to achieve this, we will do the following:

scrapper_2.py

```python
from selenium import webdriver
from bs4 import BeautifulSoup
import time
import csv

START_URL = "https://exoplanets.nasa.gov/exoplanet-catalog/"
browser = webdriver.Chrome("/Users/apoorvelous/Downloads/chromedriver")
browser.get(START_URL)
time.sleep(10)

def scrape():
    headers = ["name", "light_years_from_earth", "planet_mass", "stellar_magnitude", "discovery_date", "hyperlink"]
    planet_data = []
    for i in range(0, 428):
        soup = BeautifulSoup(browser.page_source, "html.parser")
        for ul_tag in soup.find_all("ul", attrs={"class", "exoplanet"}):
            li_tags = ul_tag.find_all("li")
            temp_list = []
            for index, li_tag in enumerate(li_tags):
                if index == 0:
                    temp_list.append(li_tag.find_all("a")[0].contents[0])
                else:
                    try:
                        temp_list.append(li_tag.contents[0])
                    except:
                        temp_list.append("")
            hyperlink_li_tag = li_tags[0]
            temp_list.append("https://exoplanets.nasa.gov"+hyperlink_li_tag.find_all("a", href=True)[0]["href"])
            planet_data.append(temp_list)
        browser.find_element_by_xpath('//*[@id="primary_column"]/footer/div/div/div/nav/span[2]/a').click()
    with open("scrapper_2.csv", "w") as f:
        csvwriter = csv.writer(f)
        csvwriter.writerow(headers)
        csvwriter.writerows(planet_data)

scrape()
```

We have added:

```python
hyperlink_li_tag = li_tags[0]

temp_list.append("https://exoplanets.nasa.gov"+hyperlink_li_tag.find_all("a", href=True)[0]["href"])
```

Here, first we are creating a variable hyperlink_li_tag and then we are using this variable to find all the anchor tag with href, take the first anchor tag (since we know there's only one anchor tag in all li tags) and then we are taking out the href from it.

**ESR:**

- We'll scrape data by using these links!

| | | |
|---|---|---|
| | 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 that will take these hyperlinks one by one, get the HTML and then we will scrape the data.<br><br>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 with the browser, therefore we can do this without selenium.<br><br>Let's get started! | |
| **Teacher Stops Screen Share** | | |
| | Now it's your turn. Please share your screen with me. | |
| • **Ask Student to press ESC key to come back to panel**<br>• **Guide Student to start Screen Share**<br>• **Teacher gets into Fullscreen** | | |
| **ACTIVITY**<br>• **Student creates a new function to use all the hyperlinks one by one and scrape data from there** | | |
| **Step 3: Student-Led Activity (15 min)** | *Ask the student to move the variables* ***headers*** *and* ***planet_data*** *to the global scope, i.e, below time.sleep(10) line.*<br><br>This is because we now would want to access these variables in multiple functions. | *The student moves the variables.* |

| | | |
|---|---|---|
| scrapper_2.py<br>`from selenium import webdriver`<br>`from bs4 import BeautifulSoup`<br>`import time`<br>`import csv`<br><br>`START_URL = "https://exoplanets.nasa.gov/exoplanet-catalog/"`<br>`browser = webdriver.Chrome("/Users/apoorvelous/Downloads/chromedriver")`<br>`browser.get(START_URL)`<br>`time.sleep(10)`<br>`headers = ["name", "light_years_from_earth", "planet_mass", "stellar_magnitude", "discovery_date", "hyperlink"]`<br>`planet_data = []` | | |
| | Great! Now, let's add the new headers, that is, the new data that is available on the new page we just discovered.<br><br>`headers = ["name", "light_years_from_earth", "planet_mass", "stellar_magnitude", "discovery_date", "hyperlink", "planet_type", "planet_radius", "orbital_radius", "orbital_period", "eccentricity"]` | *The student adds more headers.* |
| | Great, now let's create a new function and call that function. We will call the function in loop and pass the hyperlink we saved with the earlier function into this function.<br><br>Also, let's comment out the CSV saving code. We want to save a csv with half the data, right? | *The student creates a new function.* |

| | | |
|---|---|---|
| | ```<br>31 # with open("scrapper_2.csv", "w") as f:<br>32 # csvwriter = csv.writer(f)<br>33 # csvwriter.writerow(headers)<br>34 # csvwriter.writerows(planet_data)<br>35<br>36 def scrape_more_data(hyperlink):<br>37 pass<br>38<br>39 scrape()<br>40 for data in planet_data:<br>41 scrape_more_data(data[5])<br>```<br>Okay, now earlier, we created a soup object where we passed the browser's page source and parsed it as html. This time, since we are not going to use selenium, how can we do it? | **ESR:**<br>We can get the page's HTML by making a GET request. |
| | That's right! For that, we will import requests module<br>```<br>import requests<br>```<br>And we will write the following code inside the new function we created:<br>```<br>page = requests.get(hyperlink)<br>soup = BeautifulSoup(page.content, "html.parser")<br>```<br>Here, we are first getting the page, and then we are parsing the contents of the page as HTML. | The student follows instructions. |

| | | |
|---|---|---|
| | *Ask the student to create a new list **new_planet_data** to save data from these new pages, and ask them to scrape the data like before.*<br><br>*Help the student if required.*<br>*The code should look something like this:* | |

```
def scrape_more_data(hyperlink):
    page = requests.get(hyperlink)
    soup = BeautifulSoup(page.content, "html.parser")
    for tr_tag in soup.find_all("tr", attrs={"class": "fact_row"}):
        td_tags = tr_tag.find_all("td")
        temp_list = []
        for td_tag in td_tags:
            try:
                temp_list.append(td_tag.find_all("div", attrs={"class": "value"})[0].contents[0])
            except:
                temp_list.append("")
        new_planet_data.append(temp_list)
```

| | | |
|---|---|---|
| | Great job! Now we have 2 lists, **planet_data** and **new_planet_data**.<br><br>What we want to do is, we want to merge this data. Adding 2 lists creates 1 final list with elements from both the lists in the same order. | *The student merges the data.* |

```
final_planet_data = []

for index, data in enumerate(planet_data):
    final_planet_data.append(data + final_planet_data[index])
```

| | | |
|---|---|---|
| | Finally, we will create a csv with our **headers** and **final_planet_data**.<br><br>Our final code looks something like this: | *The student creates a CSV.* |

scrapper_2.py

```python
from selenium import webdriver
from bs4 import BeautifulSoup
import requests
import time
import csv

START_URL = "https://exoplanets.nasa.gov/exoplanet-catalog/"
browser = webdriver.Chrome("/Users/apoorvelous/Downloads/chromedriver")
browser.get(START_URL)
time.sleep(10)
headers = ["name", "light_years_from_earth", "planet_mass", "stellar_magnitude", "discovery_date", "hyperlink", "planet_type", "planet_r
planet_data = []
new_planet_data = []

def scrape():
    for i in range(0, 428):
        soup = BeautifulSoup(browser.page_source, "html.parser")
        for ul_tag in soup.find_all("ul", attrs={"class", "exoplanet"}):
            li_tags = ul_tag.find_all("li")
            temp_list = []
            for index, li_tag in enumerate(li_tags):
                if index == 0:
                    temp_list.append(li_tag.find_all("a")[0].contents[0])
                else:
                    try:
                        temp_list.append(li_tag.contents[0])
                    except:
                        temp_list.append("")
            hyperlink_li_tag = li_tags[0]
            temp_list.append("https://exoplanets.nasa.gov"+hyperlink_li_tag.find_all("a", href=True)[0]["href"])
            planet_data.append(temp_list)
        browser.find_element_by_xpath('//*[@id="primary_column"]/footer/div/div/div/nav/span[2]/a').click()
```

```python
def scrape_more_data(hyperlink):
    page = requests.get(hyperlink)
    soup = BeautifulSoup(page.content, "html.parser")
    for tr_tag in soup.find_all("tr", attrs={"class": "fact_row"}):
        td_tags = tr_tag.find_all("td")
        temp_list = []
        for td_tag in td_tags:
            try:
                temp_list.append(td_tag.find_all("div", attrs={"class": "value"})[0].contents[0])
            except:
                temp_list.append("")
        new_planet_data.append(temp_list)

scrape()
for data in planet_data:
    scrape_more_data(data[5])

final_planet_data = []

for index, data in enumerate(planet_data):
    final_planet_data.append(data + final_planet_data[index])

with open("final.csv", "w") as f:
        csvwriter = csv.writer(f)
        csvwriter.writerow(headers)
        csvwriter.writerows(final_planet_data)
```

| | | |
|---|---|---|
| | Let's run this code to see if it works fine and generates the desired result. | *Student runs the code.* |
| | Although it is the running version of the code, scraping can take a lot of time sometimes (like for scraping 4,277 pages in this case) therefore we'll provide you the final csv. <Student Activity 1><br><br>If you want you can also try running your code after the class to check the output | Student runs the code after class to get the output or downloads the csv from Student Activity 1 |

| Teacher Guides Student to Stop Screen Share | | |
|---|---|---|
| **FEEDBACK**<br>• **Appreciate the student for their efforts**<br>• **Identify 2 strengths and 1 area of progress for the student** | | |
| **Step 4: Wrap-Up (5 min)** | So, in this project class we revisited the concepts from the previous class and you did the majority of the scraping yourself! Congratulations! | **ESR:** Thanks! |
| | Next class, we will be learning new concepts and building new projects. | - |
| **Teacher Clicks** ✖ End Class | | |

| Activity | Activity Name | Links |
|---|---|---|
| Teacher activity 1 | solution | https://github.com/whitehatjr/web-scrapping-2 |
| Student Activity 1 | final csv | https://raw.githubusercontent.com/whitehatjr/web-scrapping-2/master/final.csv |