Document highlight:

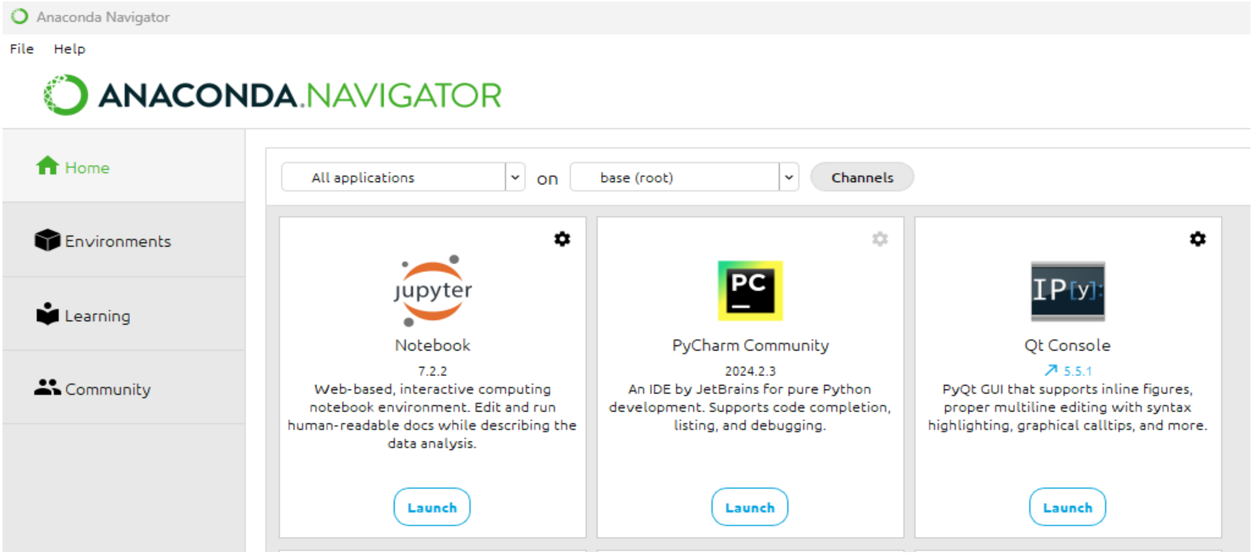
- Tool Installation
    - Python
    - Jupyter Notebook
- Open a new notebook, import the libraries to be used for the project
- Connect to a sampling website and pull in data
- Create a csv file, import sampling scraped data
- Automate and monitor a sampling scenario of a (ie., daily) price check
- In monitoring the sampling scenario (price check), have an email setup to be sent if the price is below a certain point

1. Tool Installation

Download and install (follow installation guide):

Python
https://www.python.org/downloads/

Jupyter Notebook
https://www.anaconda.com/download/success



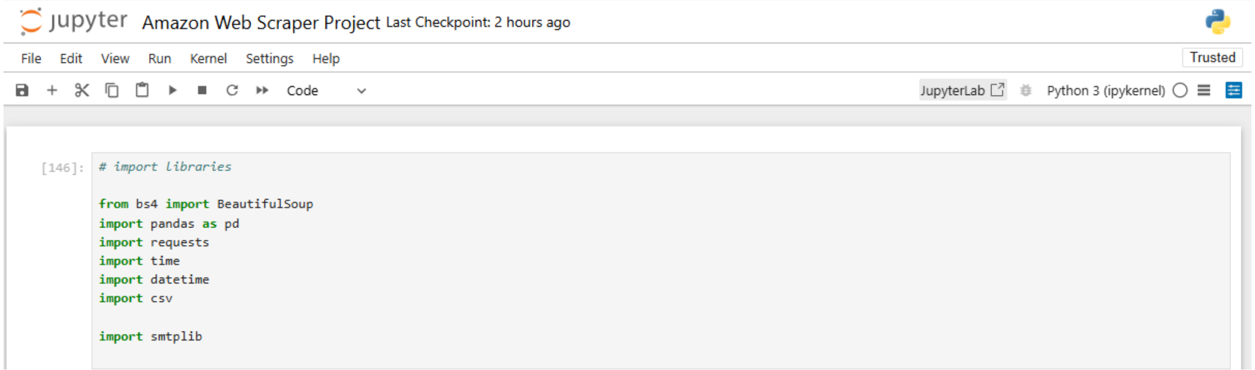
Notebook
7.2.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

PyCharm Community
2024.2.3
An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.

Qt Console
↗ 5.5.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

2. Open a new notebook, import the libraries to be used for the project

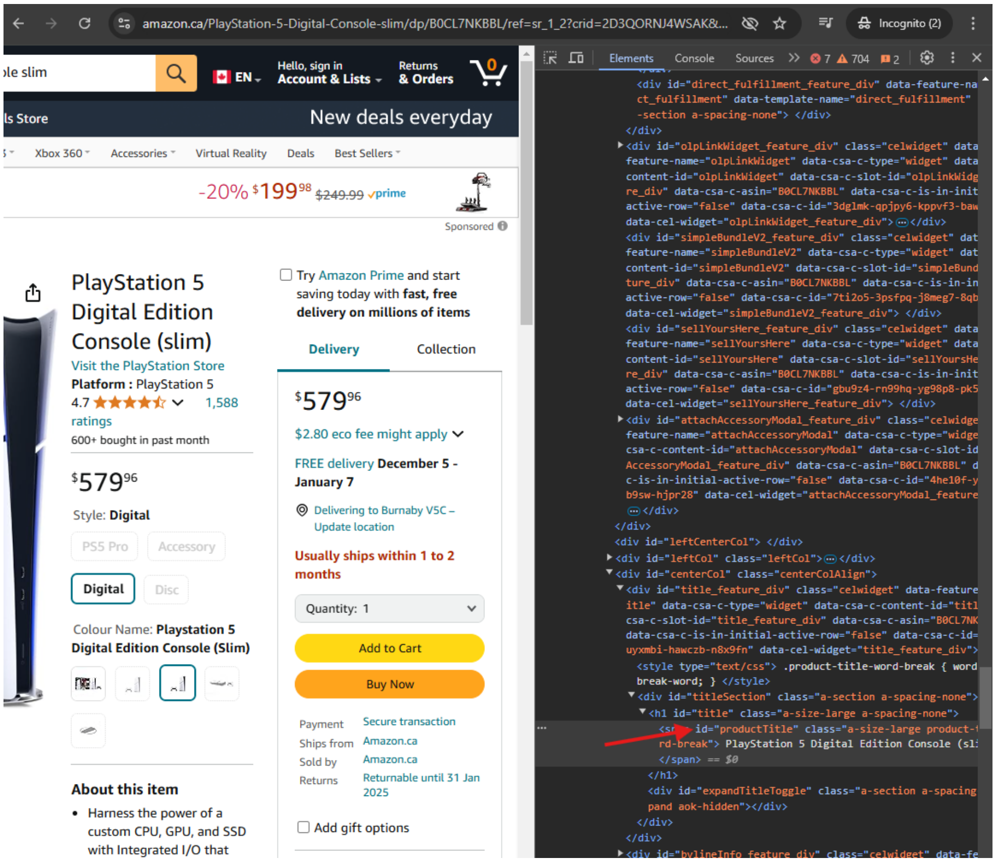
Import the libraries and other one-time configuration setup



```
# import libraries

from bs4 import BeautifulSoup
import pandas as pd
import requests
import time
import datetime
import csv

import smtplib
```

3. Connect to a sampling website and pull in data

From the page html



PlayStation 5 Digital Edition Console (slim)

Visit the PlayStation Store

Platform : PlayStation 5

4.7 ★★★★☆ 1,588 ratings

600+ bought in past month

$579.96

Style: **Digital**

PS5 Pro | Accessory | **Digital** | Disc

Colour Name: **Playstation 5 Digital Edition Console (Slim)**

About this item

- Harness the power of a custom CPU, GPU, and SSD with Integrated I/O that

□ Try Amazon Prime and start saving today with **fast, free delivery on millions of items**

Delivery | Collection

$579.96

$2.80 eco fee might apply

FREE delivery **December 5 - January 7**

Delivering to Burnaby V5C – Update location

Usually ships within 1 to 2 months

Quantity: 1

Add to Cart

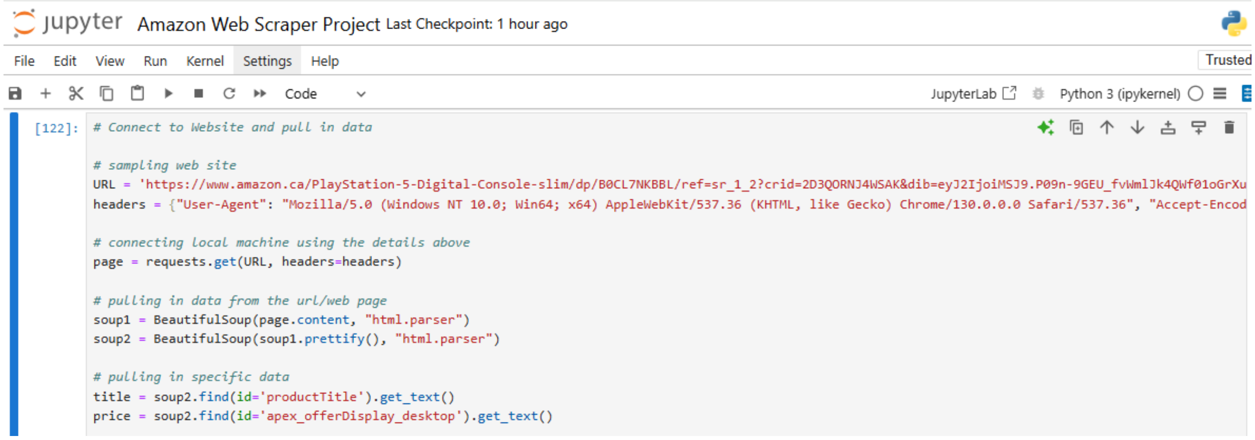
Buy Now

Payment	Secure transaction
Ships from	Amazon.ca
Sold by	Amazon.ca
Returns	Returnable until 31 Jan 2025

□ Add gift options

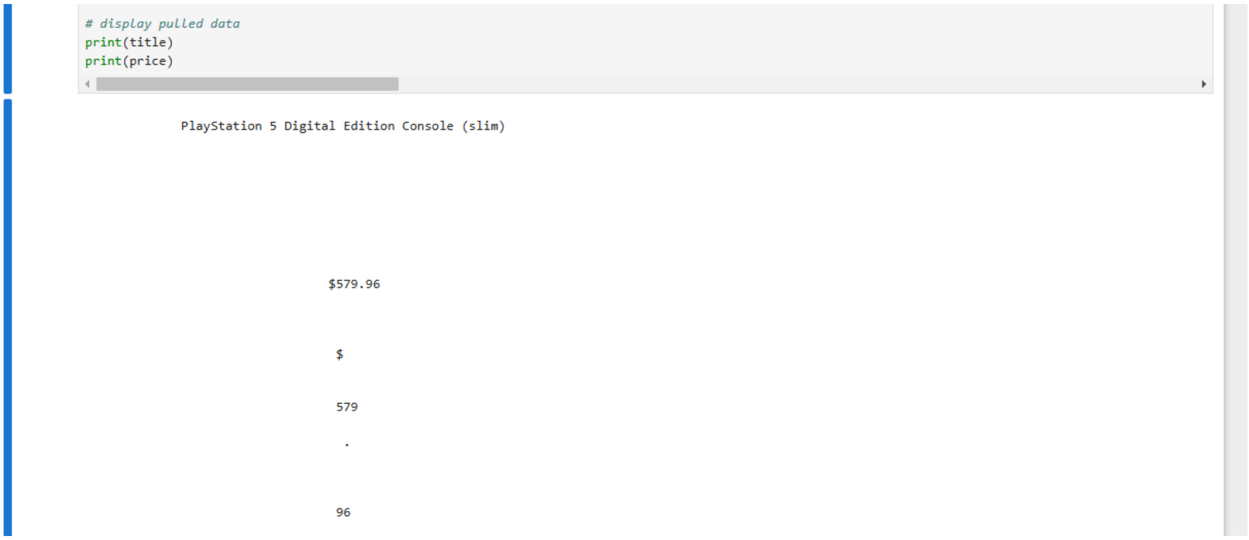
```
      <div id="direct_fulfillment_feature_div" data-feature-na
      ct_fulfillment" data-template-name="direct_fulfillment"
      -section a-spacing-none"> </div>
    </div>
   ▸<div id="olpLinkWidget_feature_div" class="celwidget" data
    feature-name="olpLinkWidget" data-csa-c-type="widget" data
    content-id="olpLinkWidget" data-csa-c-slot-id="olpLinkWidg
    re_div" data-csa-c-asin="B0CL7NKBBL" data-csa-c-is-in-init
    active-row="false" data-csa-c-id="3dglmk-qpjpy6-kppvf3-baw
    data-cel-widget="olpLinkWidget_feature_div">⋯</div>
    <div id="simpleBundleV2_feature_div" class="celwidget" dat
    feature-name="simpleBundleV2" data-csa-c-type="widget" dat
    content-id="simpleBundleV2" data-csa-c-slot-id="simpleBund
    ture_div" data-csa-c-asin="B0CL7NKBBL" data-csa-c-is-in-in
    active-row="false" data-csa-c-id="7ti2o5-3psfpq-j8meg7-8qb
    data-cel-widget="simpleBundleV2_feature_div"> </div>
    <div id="sellYoursHere_feature_div" class="celwidget" data
    feature-name="sellYoursHere" data-csa-c-type="widget" data
    content-id="sellYoursHere" data-csa-c-slot-id="sellYoursHe
    re_div" data-csa-c-asin="B0CL7NKBBL" data-csa-c-is-in-init
    active-row="false" data-csa-c-id="gbu9z4-rn99hq-yg98p8-pk5
    data-cel-widget="sellYoursHere_feature_div"> </div>
   ▸<div id="attachAccessoryModal_feature_div" class="celwidge
    feature-name="attachAccessoryModal" data-csa-c-type="widge
    csa-c-content-id="attachAccessoryModal" data-csa-c-slot-id
    AccessoryModal_feature_div" data-csa-c-asin="B0CL7NKBBL" d
    c-is-in-initial-active-row="false" data-csa-c-id="4he10f-y
    b9sw-hjpr28" data-cel-widget="attachAccessoryModal_feature
    ⋯</div>
  </div>
  <div id="leftCenterCol"> </div>
 ▸<div id="leftCol" class="leftCol">⋯</div>
 ▾<div id="centerCol" class="centerColAlign">
   ▾<div id="title_feature_div" class="celwidget" data-feature
    itle" data-csa-c-type="widget" data-csa-c-content-id="titl
    csa-c-slot-id="title_feature_div" data-csa-c-asin="B0CL7NK
    data-csa-c-is-in-initial-active-row="false" data-csa-c-id=
    uyxmbi-hawczb-n8x9fn" data-cel-widget="title_feature_div">
      <style type="text/css"> .product-title-word-break { word
      break-word; } </style>
     ▾<div id="titleSection" class="a-section a-spacing-none">
       ▾<h1 id="title" class="a-size-large a-spacing-none">
          <s   id="productTitle" class="a-size-large product-t
          rd-break"> PlayStation 5 Digital Edition Console (sli
          </span> == $0
        </h1>
        <div id="expandTitleToggle" class="a-section a-spacing
        pand aok-hidden"></div>
      </div>
    </div>
   ▸<div id="bylineInfo_feature_div" class="celwidget" data-fe
```

In the notebook



```
[122]: # Connect to Website and pull in data

       # sampling web site
       URL = 'https://www.amazon.ca/PlayStation-5-Digital-Console-slim/dp/B0CL7NKBBL/ref=sr_1_2?crid=2D3QORNJ4WSAK&dib=eyJ2IjoiMSJ9.P09n-9GEU_fvWmlJk4QWf01oGrXu
       headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36", "Accept-Encod

       # connecting local machine using the details above
       page = requests.get(URL, headers=headers)

       # pulling in data from the url/web page
       soup1 = BeautifulSoup(page.content, "html.parser")
       soup2 = BeautifulSoup(soup1.prettify(), "html.parser")

       # pulling in specific data
       title = soup2.find(id='productTitle').get_text()
       price = soup2.find(id='apex_offerDisplay_desktop').get_text()
```

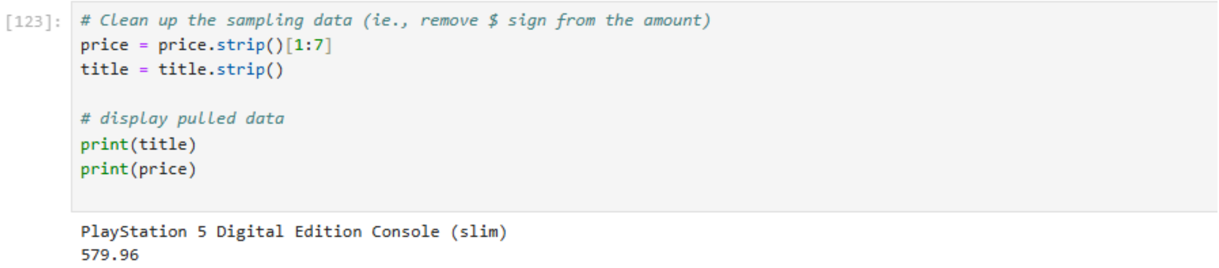
Display the data



```
# display pulled data
print(title)
print(price)
```

```
          PlayStation 5 Digital Edition Console (slim)






                             $579.96


                              $

                              579

                               .


                              96
```

Cleanup the sampling data

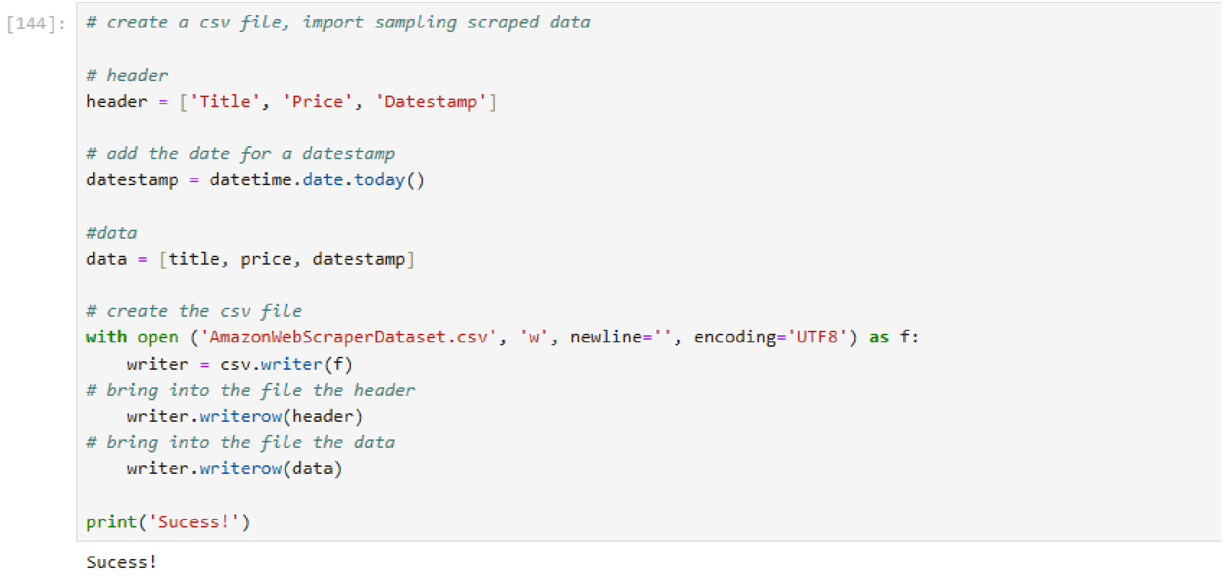


```
[123]: # Clean up the sampling data (ie., remove $ sign from the amount)
       price = price.strip()[1:7]
       title = title.strip()

       # display pulled data
       print(title)
       print(price)
```

```
PlayStation 5 Digital Edition Console (slim)
579.96
```

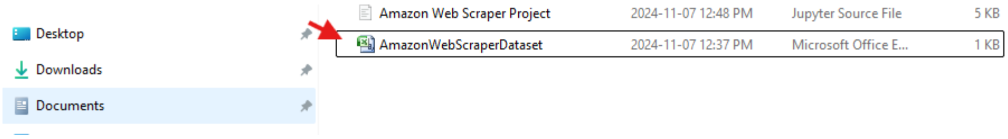
4. Create a csv file, import sampling scraped data

Run the code



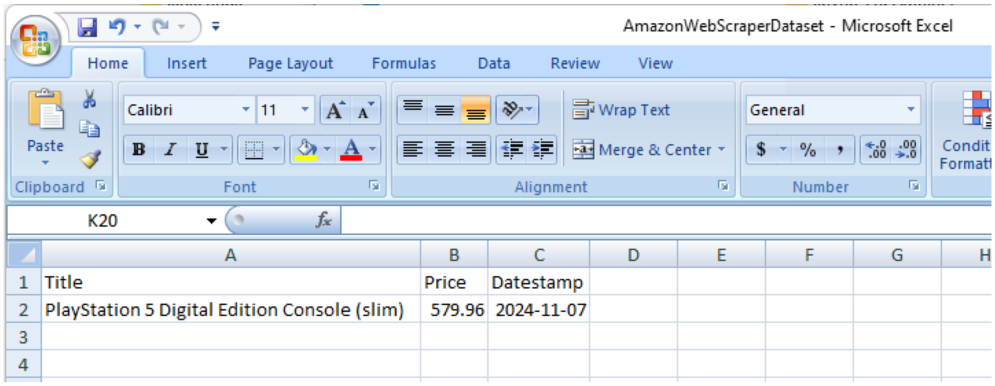
```
[144]: # create a csv file, import sampling scraped data

       # header
       header = ['Title', 'Price', 'Datestamp']

       # add the date for a datestamp
       datestamp = datetime.date.today()

       #data
       data = [title, price, datestamp]

       # create the csv file
       with open ('AmazonWebScraperDataset.csv', 'w', newline='', encoding='UTF8') as f:
           writer = csv.writer(f)
       # bring into the file the header
           writer.writerow(header)
       # bring into the file the data
           writer.writerow(data)

       print('Sucess!')

       Sucess!
```

Check the file



Amazon Web Scraper Project	2024-11-07 12:48 PM	Jupyter Source File	5 KB
AmazonWebScraperDataset	2024-11-07 12:37 PM	Microsoft Office E...	1 KB

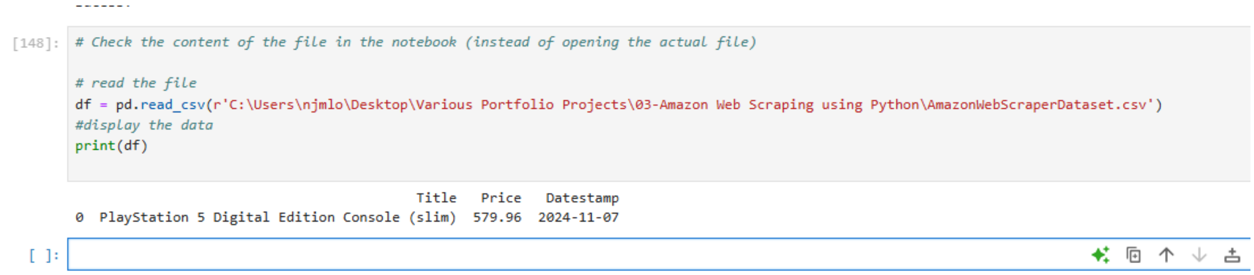
Open the file



AmazonWebScraperDataset - Microsoft Excel

	A	B	C	D	E	F	G	H
1	Title	Price	Datestamp					
2	PlayStation 5 Digital Edition Console (slim)	579.96	2024-11-07					
3								
4								

Check the content of the file in the notebook (instead of opening the actual file)

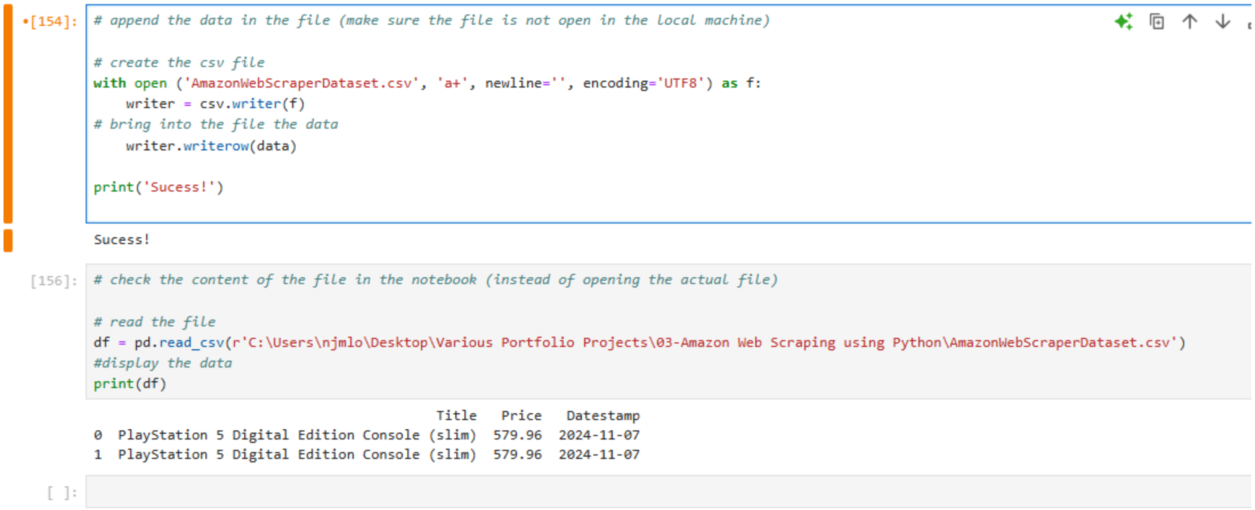


```
[148]: # Check the content of the file in the notebook (instead of opening the actual file)

       # read the file
       df = pd.read_csv(r'C:\Users\njmlo\Desktop\Various Portfolio Projects\03-Amazon Web Scraping using Python\AmazonWebScraperDataset.csv')
       #display the data
       print(df)
```

```
                                     Title   Price   Datestamp
0  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
```

```
[ ]:
```

Append (and view) the data from the file



```
•[154]: # append the data in the file (make sure the file is not open in the local machine)

        # create the csv file
        with open ('AmazonWebScraperDataset.csv', 'a+', newline='', encoding='UTF8') as f:
            writer = csv.writer(f)
        # bring into the file the data
            writer.writerow(data)

        print('Sucess!')
```

```
Sucess!
```

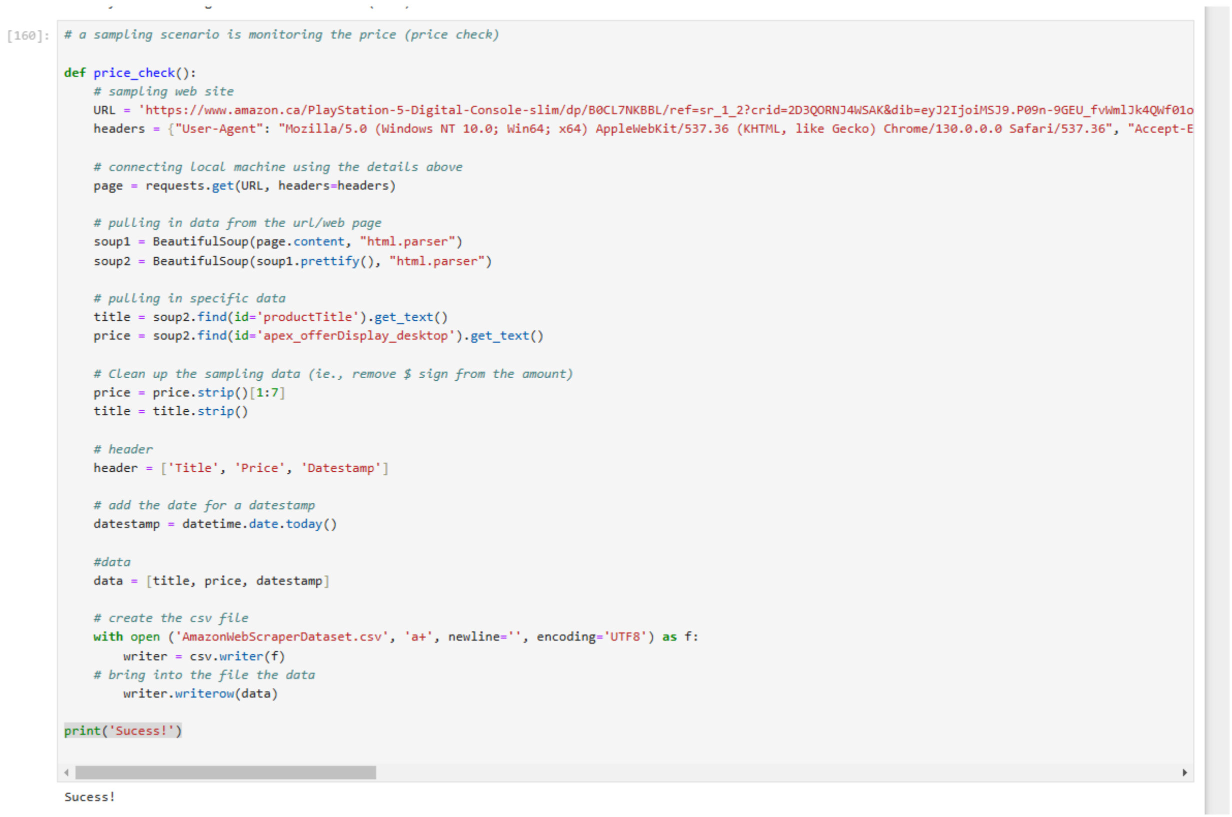
```
[156]: # check the content of the file in the notebook (instead of opening the actual file)

       # read the file
       df = pd.read_csv(r'C:\Users\njmlo\Desktop\Various Portfolio Projects\03-Amazon Web Scraping using Python\AmazonWebScraperDataset.csv')
       #display the data
       print(df)
```

```
                                     Title   Price   Datestamp
0  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
1  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
```

```
[ ]:
```

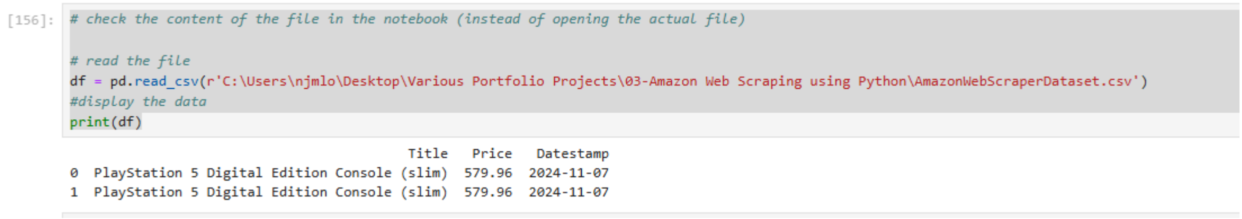
5. Automate and monitor a sampling scenario of a (ie., daily) price check

Define a function to use



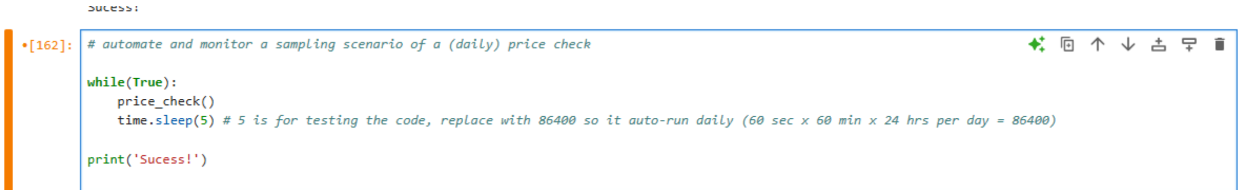
```
[160]: # a sampling scenario is monitoring the price (price check)

       def price_check():
           # sampling web site
           URL = 'https://www.amazon.ca/PlayStation-5-Digital-Console-slim/dp/B0CL7NKBBL/ref=sr_1_2?crid=2D3QORNJ4WSAK&dib=eyJ2IjoiMSJ9.P09n-9GEU_fvWmlJk4QWf01o
           headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36", "Accept-E

           # connecting local machine using the details above
           page = requests.get(URL, headers=headers)

           # pulling in data from the url/web page
           soup1 = BeautifulSoup(page.content, "html.parser")
           soup2 = BeautifulSoup(soup1.prettify(), "html.parser")

           # pulling in specific data
           title = soup2.find(id='productTitle').get_text()
           price = soup2.find(id='apex_offerDisplay_desktop').get_text()

           # Clean up the sampling data (ie., remove $ sign from the amount)
           price = price.strip()[1:7]
           title = title.strip()

           # header
           header = ['Title', 'Price', 'Datestamp']

           # add the date for a datestamp
           datestamp = datetime.date.today()

           #data
           data = [title, price, datestamp]

           # create the csv file
           with open ('AmazonWebScraperDataset.csv', 'a+', newline='', encoding='UTF8') as f:
               writer = csv.writer(f)
           # bring into the file the data
               writer.writerow(data)

       print('Sucess!')

       Sucess!
```

Before code run



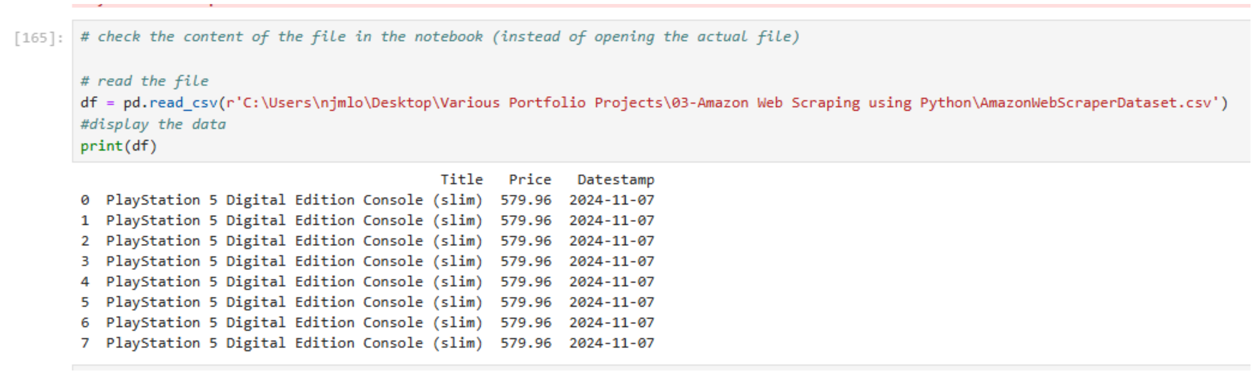
```
[156]: # check the content of the file in the notebook (instead of opening the actual file)

       # read the file
       df = pd.read_csv(r'C:\Users\njmlo\Desktop\Various Portfolio Projects\03-Amazon Web Scraping using Python\AmazonWebScraperDataset.csv')
       #display the data
       print(df)

                                          Title   Price   Datestamp
       0  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
       1  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
```

Run the code



```
       Sucess!

•[162]: # automate and monitor a sampling scenario of a (daily) price check

        while(True):
            price_check()
            time.sleep(5) # 5 is for testing the code, replace with 86400 so it auto-run daily (60 sec x 60 min x 24 hrs per day = 86400)

        print('Sucess!')
```

After code run

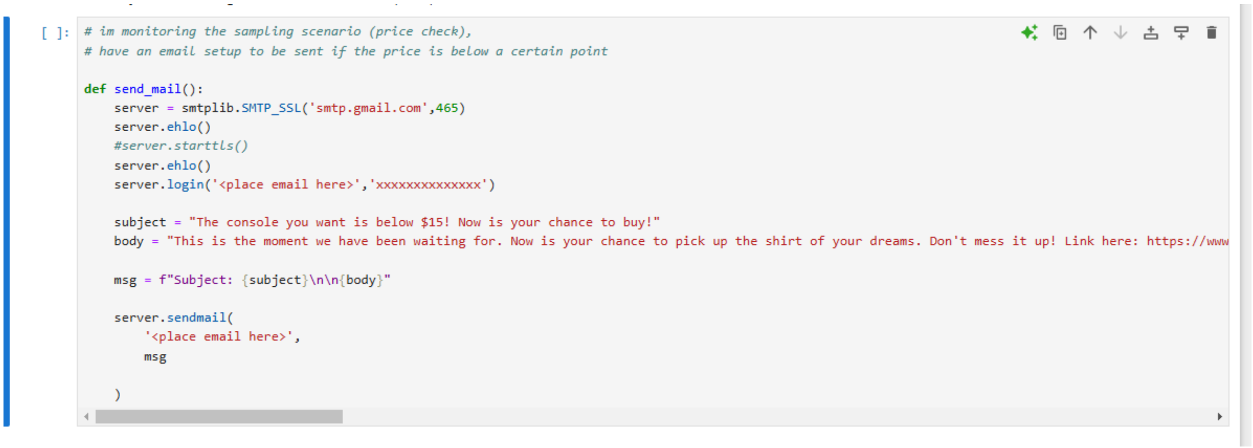


[165]:
```
# check the content of the file in the notebook (instead of opening the actual file)

# read the file
df = pd.read_csv(r'C:\Users\njmlo\Desktop\Various Portfolio Projects\03-Amazon Web Scraping using Python\AmazonWebScraperDataset.csv')
#display the data
print(df)
```

```
                                       Title   Price   Datestamp
0  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
1  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
2  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
3  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
4  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
5  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
6  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
7  PlayStation 5 Digital Edition Console (slim)  579.96  2024-11-07
```

6. In monitoring the sampling scenario (price check), have an email setup to be sent if the price is below a certain point

Code



[ ]:
```
# im monitoring the sampling scenario (price check),
# have an email setup to be sent if the price is below a certain point

def send_mail():
    server = smtplib.SMTP_SSL('smtp.gmail.com',465)
    server.ehlo()
    #server.starttls()
    server.ehlo()
    server.login('<place email here>','xxxxxxxxxxxxxx')

    subject = "The console you want is below $15! Now is your chance to buy!"
    body = "This is the moment we have been waiting for. Now is your chance to pick up the shirt of your dreams. Don't mess it up! Link here: https://www

    msg = f"Subject: {subject}\n\n{body}"

    server.sendmail(
        '<place email here>',
        msg

    )
```

END