In [1]:
```python
from bs4 import BeautifulSoup
import requests
```

In [2]:
```python
!pip install bs4
!pip install requests
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: bs4 in c:\users\sattu\appdata\roaming\python\python39\site-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in c:\programdata\anaconda3\lib\site-packages (from bs4) (4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\programdata\anaconda3\lib\site-packages (from beautifulsoup4->bs4) (2.3.1)
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-packages (2.28.1)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests) (2022.9.14)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests) (1.26.11)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests) (2.0.4)
```

In [3]:
```python
import selenium
from selenium import webdriver
import pandas as pd
from selenium.webdriver.common.by import By
import warnings
warnings.filterwarnings("ignore")
import time
```

4. Write a python program to search for a smartphone(e.g.: Oneplus Nord, pixel 4A, etc.) on www.flipkart.com and scrape following details for all the search results displayed on 1st page.

In [101]:
```python
driver = webdriver.Chrome()
```

In [102]:
```python
url4="https://www.flipkart.com/search?q=smartphone&otracker=search&otracker1=search&marketplace=FLIPKART&as-show=on&as=off"
driver.get(url4)
```

In [103]:
```python
Brand_Name=[]
Colour=[]
Storage_RAM_ROM=[]
P_F_Camera=[]
Display_size_Resolution=[]
ProcessorAndCores=[]
Battery=[]
Price=[]
Product_URL=[]
```

In [104]:
```python
BName=driver.find_elements(By.XPATH,'//div[@class="_4rR01T"]')
for i in BName:
    if i.text is None :
        Brand_Name.append("--")
    else:
        Brand_Name.append(i.text)
print(len(Brand_Name),Brand_Name)
```

```
24 ['Infinix HOT 20 Play (Aurora Green, 64 GB)', 'POCO M6 Pro 5G (Power Black, 128 GB)', 'Infinix HOT 20 Play (Racing Black, 64 GB)', 'POCO C51 (Power Black,
64 GB)', 'REDMI 12 (Moonstone Silver, 128 GB)', 'POCO C51 (Royal Blue, 64 GB)', 'POCO C50 (Royal Blue, 32 GB)', 'POCO C55 (Cool Blue, 128 GB)', 'Infinix HOT 2
0 Play (Luna Blue, 64 GB)', 'realme 11x 5G (Midnight Black, 128 GB)', 'realme 11x 5G (Purple Dawn, 128 GB)', 'realme 11x 5G (Purple Dawn, 128 GB)', 'realme 11
x 5G (Midnight Black, 128 GB)', 'POCO C55 (Power Black, 128 GB)', 'Infinix HOT 20 Play (Fantasy Purple, 64 GB)', 'POCO M6 Pro 5G (Power Black, 64 GB)', 'POCO
C55 (Forest Green, 128 GB)', 'POCO C50 (Country Green, 32 GB)', 'MOTOROLA e13 (Aurora Green, 64 GB)', 'POCO M6 Pro 5G (Forest Green, 64 GB)', 'Infinix SMART 7
(Emerald Green, 64 GB)', 'Infinix SMART 7 (Azure Blue, 64 GB)', 'REDMI 12 (Jade Black, 128 GB)', 'Infinix SMART 7 (Emerald Green, 128 GB)']
```

In [105]:
```python
ram=driver.find_elements(By.XPATH,"//ul[@class='_1xgFaf']//li[1]")
for i in ram:
    if i.text is None :
        Storage_RAM_ROM.append("--")
    else:
        Storage_RAM_ROM.append(i.text)
print(len(Storage_RAM_ROM),Storage_RAM_ROM)
```

```
24 ['4 GB RAM | 64 GB ROM | Expandable Upto 256 GB', '6 GB RAM | 128 GB ROM | Expandable Upto 1 TB', '4 GB RAM | 64 GB ROM | Expandable Upto 256 GB', '4 GB RA
M | 64 GB ROM | Expandable Upto 1 TB', '4 GB RAM | 128 GB ROM | Expandable Upto 1 TB', '4 GB RAM | 64 GB ROM | Expandable Upto 1 TB', '2 GB RAM | 32 GB ROM |
Expandable Upto 512 GB', '6 GB RAM | 128 GB ROM | Expandable Upto 1 TB', '4 GB RAM | 64 GB ROM | Expandable Upto 256 GB', '8 GB RAM | 128 GB ROM | Expandable
Upto 2 TB', '8 GB RAM | 128 GB ROM | Expandable Upto 2 TB', '6 GB RAM | 128 GB ROM | Expandable Upto 2 TB', '6 GB RAM | 128 GB ROM | Expandable Upto 2 TB', '6
GB RAM | 128 GB ROM | Expandable Upto 1 TB', '4 GB RAM | 64 GB ROM | Expandable Upto 256 GB', '4 GB RAM | 64 GB ROM | Expandable Upto 1 TB', '6 GB RAM | 128 G
B ROM | Expandable Upto 1 TB', '2 GB RAM | 32 GB ROM | Expandable Upto 512 GB', '4 GB RAM | 64 GB ROM | Expandable Upto 1 TB', '4 GB RAM | 64 GB ROM | Expanda
ble Upto 1 TB', '4 GB RAM | 64 GB ROM | Expandable Upto 2 TB', '4 GB RAM | 64 GB ROM | Expandable Upto 2 TB', '4 GB RAM | 128 GB ROM | Expandable Upto 1 TB',
'4 GB RAM | 128 GB ROM | Expandable Upto 512 GB']
```

In [106]:
```python
PC=driver.find_elements(By.XPATH,"//ul[@class='_1xgFaf']//li[3]")
for i in PC:
    if i.text is None :
        P_F_Camera.append("--")
    else:
        P_F_Camera.append(i.text)
print(len(P_F_Camera),P_F_Camera)
```

```
24 ['13MP + AI Lens | 8MP Front Camera', '50MP + 2MP | 8MP Front Camera', '13MP + AI Lens | 8MP Front Camera', '8MP Dual Rear Camera | 5MP Front Camera', '50M
P + 8MP + 2MP | 8MP Front Camera', '8MP Dual Rear Camera | 5MP Front Camera', '8MP Dual Camera | 5MP Front Camera', '50MP Dual Rear Camera | 5MP Front Camer
a', '13MP + AI Lens | 8MP Front Camera', '64MP + 2MP | 8MP Front Camera', '64MP + 2MP | 8MP Front Camera', '64MP + 2MP | 8MP Front Camera', '64MP + 2MP | 8MP
Front Camera', '50MP Dual Rear Camera | 5MP Front Camera', '13MP + AI Lens | 8MP Front Camera', '50MP + 2MP | 8MP Front Camera', '50MP Dual Rear Camera | 5MP
Front Camera', '8MP Dual Camera | 5MP Front Camera', '13MP Rear Camera | 5MP Front Camera', '50MP + 2MP | 8MP Front Camera', '13MP + AI Lens | 5MP Front Camer
a', '13MP + AI Lens | 5MP Front Camera', '50MP + 8MP + 2MP | 8MP Front Camera', '13MP + AI Lens | 5MP Front Camera']
```

In [107]:
```python
DS=driver.find_elements(By.XPATH,"//ul[@class='_1xgFaf']//li[2]")
for i in DS:
    if i.text is None :
        Display_size_Resolution.append("--")
    else:
        Display_size_Resolution.append(i.text)
print(len(Display_size_Resolution),Display_size_Resolution)
```

```
24 ['17.32 cm (6.82 inch) HD+ Display', '17.25 cm (6.79 inch) Full HD+ Display', '17.32 cm (6.82 inch) HD+ Display', '16.56 cm (6.52 inch) HD+ Display', '17.2
5 cm (6.79 inch) Full HD+ Display', '16.56 cm (6.52 inch) HD+ Display', '16.56 cm (6.52 inch) HD+ Display', '17.04 cm (6.71 inch) HD+ Display', '17.32 cm (6.8
2 inch) HD+ Display', '17.07 cm (6.72 inch) Full HD+ Display', '17.07 cm (6.72 inch) Full HD+ Display', '17.07 cm (6.72 inch) Full HD+ Display', '17.07 cm (6.
72 inch) Full HD+ Display', '17.04 cm (6.71 inch) HD+ Display', '17.32 cm (6.82 inch) HD+ Display', '17.25 cm (6.79 inch) Full HD+ Display', '17.04 cm (6.71 i
nch) HD+ Display', '16.56 cm (6.52 inch) HD+ Display', '16.51 cm (6.5 inch) HD+ Display', '17.25 cm (6.79 inch) Full HD+ Display', '16.76 cm (6.6 inch) HD+ Di
splay', '16.76 cm (6.6 inch) HD+ Display', '17.25 cm (6.79 inch) Full HD+ Display', '16.76 cm (6.6 inch) HD+ Display']
```

In [108]:
```python
P=driver.find_elements(By.XPATH,"//ul[@class='_1xgFaf']//li[5]")
for i in P:
    if i.text is None :
        ProcessorAndCores.append("--")
    else:
        ProcessorAndCores.append(i.text)
print(len(ProcessorAndCores),ProcessorAndCores)
```

```
24 ['MediaTek G37 Processor', 'Snapdragon 4 Gen 2 Processor', 'MediaTek G37 Processor', 'Helio G36 Processor', 'Helio G88 Processor', 'Helio G36 Processor',
'Mediatek Helio A22 Processor, Upto 2.0 GHz Processor', 'Mediatek Helio G85 Processor', 'MediaTek G37 Processor', 'Dimensity 6100+ Processor', 'Dimensity 6100
+ Processor', 'Dimensity 6100+ Processor', 'Dimensity 6100+ Processor', 'Mediatek Helio G85 Processor', 'MediaTek G37 Processor', 'Snapdragon 4 Gen 2 Processo
r', 'Mediatek Helio G85 Processor', 'Mediatek Helio A22 Processor, Upto 2.0 GHz Processor', 'Unisoc T606 Processor', 'Snapdragon 4 Gen 2 Processor', 'Unisoc S
preadtrum SC9863A1 Processor', 'Unisoc Spreadtrum SC9863A1 Processor', 'Helio G88 Processor', 'Unisoc Spreadtrum SC9863A1 Processor']
```

In [109]:
```python
B=driver.find_elements(By.XPATH,"//ul[@class='_1xgFaf']//li[4]")
for i in B:
    if i.text is None :
        Battery.append("--")
    else:
        Battery.append(i.text)
print(len(Battery),Battery)
```

```
24 ['6000 mAh Li-ion Polymer Battery', '5000 mAh Battery', '6000 mAh Li-ion Polymer Battery', '5000 mAh Battery', '5000 mAh Battery', '5000 mAh Battery', '500
0 mAh Lithium Polymer Battery', '5000 mAh Battery', '6000 mAh Li-ion Polymer Battery', '5000 mAh Battery', '5000 mAh Battery', '5000 mAh Battery', '5000 mAh B
attery', '5000 mAh Battery', '6000 mAh Li-ion Polymer Battery', '5000 mAh Battery', '5000 mAh Battery', '5000 mAh Lithium Polymer Battery', '5000 mAh Batter
y', '5000 mAh Battery', '6000 mAh Battery', '6000 mAh Battery', '5000 mAh Battery', '6000 mAh Battery']
```

In [110]:
```python
price=driver.find_elements(By.XPATH,"//div[@class='_30jeq3 _1_WHN1']")
for i in price:
    if i.text is None :
        Price.append("--")
    else:
        Price.append(i.text)
print(len(Price),Price)
```

```
24 ['₹7,999', '₹12,999', '₹7,999', '₹6,499', '₹9,999', '₹6,499', '₹5,499', '₹8,499', '₹7,999', '₹15,999', '₹15,999', '₹14,999', '₹14,999', '₹8,499', '₹7,999',
'₹10,999', '₹8,499', '₹5,499', '₹7,299', '₹10,999', '₹6,999', '₹6,999', '₹9,999', '₹7,599']
```

In [111]:
```python
FlipKart=pd.DataFrame([])
FlipKart['Brand_Name']=Brand_Name
FlipKart['Storage_RAM_ROM']=Storage_RAM_ROM
FlipKart['Amount P_F_Camera']=P_F_Camera
FlipKart['Display_size_Resolution']=Display_size_Resolution
FlipKart['ProcessorAndCores']=ProcessorAndCores
FlipKart['Battery']=Battery
FlipKart['Price']=Price

FlipKart
```

Out[111]:

	Brand_Name	Storage_RAM_ROM	Amount P_F_Camera	Display_size_Resolution	ProcessorAndCores	Battery	Price
0	Infinix HOT 20 Play (Aurora Green, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 256 GB	13MP + AI Lens | 8MP Front Camera	17.32 cm (6.82 inch) HD+ Display	MediaTek G37 Processor	6000 mAh Li-ion Polymer Battery	₹7,999
1	POCO M6 Pro 5G (Power Black, 128 GB)	6 GB RAM | 128 GB ROM | Expandable Upto 1 TB	50MP + 2MP | 8MP Front Camera	17.25 cm (6.79 inch) Full HD+ Display	Snapdragon 4 Gen 2 Processor	5000 mAh Battery	₹12,999
2	Infinix HOT 20 Play (Racing Black, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 256 GB	13MP + AI Lens | 8MP Front Camera	17.32 cm (6.82 inch) HD+ Display	MediaTek G37 Processor	6000 mAh Li-ion Polymer Battery	₹7,999
3	POCO C51 (Power Black, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 1 TB	8MP Dual Rear Camera | 5MP Front Camera	16.56 cm (6.52 inch) HD+ Display	Helio G36 Processor	5000 mAh Battery	₹6,499
4	REDMI 12 (Moonstone Silver, 128 GB)	4 GB RAM | 128 GB ROM | Expandable Upto 1 TB	50MP + 8MP + 2MP | 8MP Front Camera	17.25 cm (6.79 inch) Full HD+ Display	Helio G88 Processor	5000 mAh Battery	₹9,999
5	POCO C51 (Royal Blue, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 1 TB	8MP Dual Rear Camera | 5MP Front Camera	16.56 cm (6.52 inch) HD+ Display	Helio G36 Processor	5000 mAh Battery	₹6,499
6	POCO C50 (Royal Blue, 32 GB)	2 GB RAM | 32 GB ROM | Expandable Upto 512 GB	8MP Dual Camera | 5MP Front Camera	16.56 cm (6.52 inch) HD+ Display	Mediatek Helio A22 Processor, Upto 2.0 GHz Pro...	5000 mAh Lithium Polymer Battery	₹5,499
7	POCO C55 (Cool Blue, 128 GB)	6 GB RAM | 128 GB ROM | Expandable Upto 1 TB	50MP Dual Rear Camera | 5MP Front Camera	17.04 cm (6.71 inch) HD+ Display	Mediatek Helio G85 Processor	5000 mAh Battery	₹8,499
8	Infinix HOT 20 Play (Luna Blue, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 256 GB	13MP + AI Lens | 8MP Front Camera	17.32 cm (6.82 inch) HD+ Display	MediaTek G37 Processor	6000 mAh Li-ion Polymer Battery	₹7,999
9	realme 11x 5G (Midnight Black, 128 GB)	8 GB RAM | 128 GB ROM | Expandable Upto 2 TB	64MP + 2MP | 8MP Front Camera	17.07 cm (6.72 inch) Full HD+ Display	Dimensity 6100+ Processor	5000 mAh Battery	₹15,999
10	realme 11x 5G (Purple Dawn, 128 GB)	8 GB RAM | 128 GB ROM | Expandable Upto 2 TB	64MP + 2MP | 8MP Front Camera	17.07 cm (6.72 inch) Full HD+ Display	Dimensity 6100+ Processor	5000 mAh Battery	₹15,999
11	realme 11x 5G (Purple Dawn, 128 GB)	6 GB RAM | 128 GB ROM | Expandable Upto 2 TB	64MP + 2MP | 8MP Front Camera	17.07 cm (6.72 inch) Full HD+ Display	Dimensity 6100+ Processor	5000 mAh Battery	₹14,999
12	realme 11x 5G (Midnight Black, 128 GB)	6 GB RAM | 128 GB ROM | Expandable Upto 2 TB	64MP + 2MP | 8MP Front Camera	17.07 cm (6.72 inch) Full HD+ Display	Dimensity 6100+ Processor	5000 mAh Battery	₹14,999
13	POCO C55 (Power Black, 128 GB)	6 GB RAM | 128 GB ROM | Expandable Upto 1 TB	50MP Dual Rear Camera | 5MP Front Camera	17.04 cm (6.71 inch) HD+ Display	Mediatek Helio G85 Processor	5000 mAh Battery	₹8,499
14	Infinix HOT 20 Play (Fantasy Purple, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 256 GB	13MP + AI Lens | 8MP Front Camera	17.32 cm (6.82 inch) HD+ Display	MediaTek G37 Processor	6000 mAh Li-ion Polymer Battery	₹7,999
15	POCO M6 Pro 5G (Power Black, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 1 TB	50MP + 2MP | 8MP Front Camera	17.25 cm (6.79 inch) Full HD+ Display	Snapdragon 4 Gen 2 Processor	5000 mAh Battery	₹10,999
16	POCO C55 (Forest Green, 128 GB)	6 GB RAM | 128 GB ROM | Expandable Upto 1 TB	50MP Dual Rear Camera | 5MP Front Camera	17.04 cm (6.71 inch) HD+ Display	Mediatek Helio G85 Processor	5000 mAh Battery	₹8,499
17	POCO C50 (Country Green, 32 GB)	2 GB RAM | 32 GB ROM | Expandable Upto 512 GB	8MP Dual Camera | 5MP Front Camera	16.56 cm (6.52 inch) HD+ Display	Mediatek Helio A22 Processor, Upto 2.0 GHz Pro...	5000 mAh Lithium Polymer Battery	₹5,499
18	MOTOROLA e13 (Aurora Green, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 1 TB	13MP Rear Camera | 5MP Front Camera	16.51 cm (6.5 inch) HD+ Display	Unisoc T606 Processor	5000 mAh Battery	₹7,299
19	POCO M6 Pro 5G (Forest Green, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 1 TB	50MP + 2MP | 8MP Front Camera	17.25 cm (6.79 inch) Full HD+ Display	Snapdragon 4 Gen 2 Processor	5000 mAh Battery	₹10,999
20	Infinix SMART 7 (Emerald Green, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 2 TB	13MP + AI Lens | 5MP Front Camera	16.76 cm (6.6 inch) HD+ Display	Unisoc Spreadtrum SC9863A1 Processor	6000 mAh Battery	₹6,999
21	Infinix SMART 7 (Azure Blue, 64 GB)	4 GB RAM | 64 GB ROM | Expandable Upto 2 TB	13MP + AI Lens | 5MP Front Camera	16.76 cm (6.6 inch) HD+ Display	Unisoc Spreadtrum SC9863A1 Processor	6000 mAh Battery	₹6,999
22	REDMI 12 (Jade Black, 128 GB)	4 GB RAM | 128 GB ROM | Expandable Upto 1 TB	50MP + 8MP + 2MP | 8MP Front Camera	17.25 cm (6.79 inch) Full HD+ Display	Helio G88 Processor	5000 mAh Battery	₹9,999
23	Infinix SMART 7 (Emerald Green, 128 GB)	4 GB RAM | 128 GB ROM | Expandable Upto 512 GB	13MP + AI Lens | 5MP Front Camera	16.76 cm (6.6 inch) HD+ Display	Unisoc Spreadtrum SC9863A1 Processor	6000 mAh Battery	₹7,599

6. Write a program to scrap all the available details of top 10 gaming laptops from digit.in.

In [112]:
```python
driver = webdriver.Chrome()
```

In [ ]:
```python
url="https://www.digit.in"
driver.get(url) class="TopNumbeHeading active"
```

In [113]:
```python
url="https://www.digit.in/top-products/best-gaming-laptops-40.html"
```

In [114]:
```python
driver.get(url)
```

In [115]:
```python
Brands=[]
Specification=[]
Price=[]
```

In [120]:
```python
br=driver.find_elements(By.XPATH,"//span[@class='datahreflink']/h3")
len(br)
```

Out[120]:
```
7
```

In [121]:
```python
for i in br:

    Brands.append(str(i.text).replace("\n",""))
Brands
```

Out[121]:
```
['HP OMEN 17 (2023)',
 'MSI Titan GT77 12UHS',
 'Lenovo Legion 5i Pro',
 'ASUS ROG Strix Scar 18 2023',
 'Acer Predator Helios Neo 16',
 'ASUS ROG Zephyrus G14',
 'MSI Cyborg 15']
```

In [122]:
```python
sp=driver.find_elements(By.XPATH,"//div[@class='Specs-Wrap']")
len(sp)
```

Out[122]:
```
7
```

In [123]:
```python
for i in sp:

    Specification.append(str(i.text).replace("\n",""))
Specification
```

Out[123]:
```
['Windows 11 Home17.3" (2560 x 1440)13th Gen Intel Core i7-13700HX | 5.0 GHz1 TB SSD/16 GB DDR5',
 'Windows 11 Home17.3" (3840 x 2160)12th Gen Intel Core i9-12900HX | 3.6 GHz2 TB SSD/64 GB DDR5',
 'Windows 11 Home16" (2560 x 1600)12th Gen Intel Core i7-12700H | 4.7 GHz1 TB SSD/32 GB DDR5',
 'Windows 11 Home18" (1920 x 1200)13th Gen Intel Core i9-13980HX | 2.2 GHz1 TB SSD/32 GB DDR5',
 'Windows 11 Home16" (2560 x 1600)13th Gen Intel Core i7-13700HX | NA1 TB SSD/16 GB DDR5',
 'Windows 11 Home14" (1920 x 1200)AMD Ryzen 9-6900HS | 4.9 GHz1 TB SSD/16 GB DDR5',
 'Windows 11 Home15.6" (1920 x 1080)12th Gen Intel Core i7-12650H | 4.7 GHz1 TB SSD/16 GB DDR5']
```

In [124]:
```python
pri=driver.find_elements(By.XPATH,"//td[@class='smprice']")
len(pri)
```

Out[124]:
```
7
```

In [125]:
```python
for i in pri:

    Price.append(str(i.text).replace("\n",""))
Price
```

Out[125]:
```
['₹ 269,777',
 '₹ 499,990',
 '₹ 179,990',
 '₹ 279,990',
 '₹ 149,990',
 '₹ 156,990',
 '₹ 125,000']
```

In [126]:
```python
digit_lap=pd.DataFrame([])
digit_lap['Brands']=Brands[0:10]
digit_lap['Price']=Price[0:10]
digit_lap['Specification']=Specification[0:10]
digit_lap
```

Out[126]:

	Brands	Price	Specification
0	HP OMEN 17 (2023)	₹ 269,777	Windows 11 Home17.3" (2560 x 1440)13th Gen Int...
1	MSI Titan GT77 12UHS	₹ 499,990	Windows 11 Home17.3" (3840 x 2160)12th Gen Int...
2	Lenovo Legion 5i Pro	₹ 179,990	Windows 11 Home16" (2560 x 1600)12th Gen Intel...
3	ASUS ROG Strix Scar 18 2023	₹ 279,990	Windows 11 Home18" (1920 x 1200)13th Gen Intel...
4	Acer Predator Helios Neo 16	₹ 149,990	Windows 11 Home16" (2560 x 1600)13th Gen Intel...
5	ASUS ROG Zephyrus G14	₹ 156,990	Windows 11 Home14" (1920 x 1200)AMD Ryzen 9-69...
6	MSI Cyborg 15	₹ 125,000	Windows 11 Home15.6" (1920 x 1080)12th Gen Int...

5. Write a program to scrap geospatial coordinates (latitude, longitude) of a city searched on google maps.

In [127]:
```python
import time
import re
```

In [128]:
```python
driver = webdriver.Chrome()
```

In [129]:
```python
driver.get("https://www.google.co.in/maps")
time.sleep(3)
```

In [130]:
```python
designation=driver.find_element(By.CLASS_NAME,"searchboxinput")
designation.send_keys("Raipur")
```

In [131]:
```python
city = input('Enter City Name : ')
search = driver.find_element(By.CLASS_NAME,"searchboxinput")
search.clear()
time.sleep(2)
search.send_keys(city)
button = driver.find_element(By.ID,"searchbox-searchbutton")
button.click()
time.sleep(3)

try:
    url_string = driver.current_url
    print("URL Extracted: ", url_string)
    lat_lng = re.findall(r'@(.*)data',url_string)
    if len(lat_lng):
        lat_lng_list = lat_lng[0].split(",")
        if len(lat_lng_list)>=2:
            lat = lat_lng_list[0]
            lng = lat_lng_list[1]
        print("Latitude = {}, Longitude = {}".format(lat, lng))

except Exception as e:
        print("Error: ", str(e))
```

```
Enter City Name : Raipur
URL Extracted:  https://www.google.co.in/maps/place/Raipur,+Chhattisgarh/@21.2618855,81.5366704,12z/data=!4m6!3m5!1s0x3a28dda23be28229:0x163ee1204ff9e240!8m2!
3d21.2513844!4d81.6296413!16zL20vMDJ0NXM0?entry=ttu
Latitude = 21.2618855, Longitude = 81.5366704
```

Questions 1

In [ ]:
```python
driver = webdriver.Chrome()
```

In [ ]:
```python
driver.get('https://www.amazon.in/')
```

In [ ]:
```python
inputU = input('please enter product here--->')
```

In [ ]:

In [ ]:

Question - 2

In [ ]:
```python
start_page = 0
end_page = 3
urls = []
for page in range(start_page,end_page+1):
    try:
        page_urls = driver.find_elements(By.XPATH,'//a[@class="a-link-normal s-no-outline"]')

        for url in page_urls:
            url = url.get_attribute('href')
            if url[0:4]=='http':
                urls.append(url)
        print("Product urls of page {} has been scraped.".format(page+1))

        nxt_button = driver.find_elements(By.XPATH,'//li[@class="a-last"]/a')
        if nxt_button.text == 'Next→':
            nxt_button.click()
            time.sleep(5)

        elif driver.find_element(By.XPATH,'//li[@class="a-disabled a-last"]/a').text == 'Next→':
            print("No new pages exist. Breaking the loop")
            break

    except StaleElementReferenceException as e:
        print("Stale Exception")
        next_page = nxt_button.get_attribute('href')
        driver.get(next_page)
```

In [ ]:
```python
prod_dict = {}
prod_dict['Brand']=[]
prod_dict['Name']=[]
prod_dict['Rating']=[]
prod_dict['No. of ratings']=[]
prod_dict['Price']=[]
prod_dict['Return/Exchange']=[]
prod_dict['Expected Delivery']=[]
prod_dict['Availability']=[]
prod_dict['Other Details']=[]
prod_dict['URL']=[]
```

7. Write a python program to scrape the details for all billionaires from www.forbes.com. Details to be scrapped:

"Rank", "Name", "Net worth", "Age", "Citizenship", "Source", "Industry"

In [ ]:
```python
driver = webdriver.Chrome()
```

In [ ]:
```python
driver.get('https://www.forbes.com/')
```

In [ ]:
```python
Rank=[]
Name=[]
Net_worth=[]
Age=[]
Citizenship=[]
Source=[]
Industry=[]
```

In [ ]:
```python
Rank_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][1]')
for i in Rank_tags[0:200]:
    Rank=i.text
    Rank_name.append(Rank)

Name_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][2]')
for i in Name_tags[0:200]:
    Name=i.text
    Name.append(Name)

Net_worth_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][3]')
for i in Net_worth_tags[0:200]:
    Net_worth=i.text
    Net_worth.append(Net_worth)

Age_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][4]')
for i in Age_tags[0:200]:
    Age=i.text
    Age.append(Age)

Citizenship_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][5]')
for i in Citizenship_tags[0:200]:
    Citizenship=i.text
    Citizenship_name.append(Citizenship)

Source_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][6]')
for i in Source_tags[0:200]:
    Source=i.text
    Source_name.append(Source)

Industry_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][7]')
for i in Industry_tags[0:200]:
    Industry=i.text
    Industry_name.append(Industry)
```

In [ ]:
```python
Rank_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][1]')
for i in Rank_tags[0:200]:
    Rank=i.text
    Rank_name.append(Rank)

Name_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][2]')
for i in Name_tags[0:200]:
    Name=i.text
    Name.append(Name)

Net_worth_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][3]')
for i in Net_worth_tags[0:200]:
    Net_worth=i.text
    Net_worth.append(Net_worth)

Age_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][4]')
for i in Age_tags[0:200]:
    Age=i.text
    Age.append(Age)

Citizenship_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][5]')
for i in Citizenship_tags[0:200]:
    Citizenship=i.text
    Citizenship_name.append(Citizenship)

Source_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][6]')
for i in Source_tags[0:200]:
    Source=i.text
    Source_name.append(Source)

Industry_tags=driver.find_elements(By.XPATH,'//div[@class="TableRow_cell__db-hv Table_cell__houv9"][7]')
for i in Industry_tags[0:200]:
    Industry=i.text
    Industry_name.append(Industry)
```

Q.3

Fruits

In [6]:
```python
driver = webdriver.Chrome()
```

In [7]:
```python
driver.get('https://images.google.com/')
```

In [10]:
```python
designation=driver.find_element(By.CLASS_NAME,"gLFyf")
designation.send_keys("Fruits")
```

In [17]:
```python
print("start scrolling to generate more images on the page...")
for _ in range(20):
    driver.execute_script("window.scrollBy(0,100)")
```

```
start scrolling to generate more images on the page...
```

In [18]:
```python
 images = driver.find_elements(By.XPATH,'//img[@class="rg_i Q4LuWd"]')
```

In [19]:
```python
img_urls = []
for image in images:
    source= image.get_attribute('src')
    if source is not None:
        if(source[0:4] == 'http'):
            img_urls.append(source)
```

In [23]:
```python
for i in range(len(img_urls)):
    if i >= 10:
        break
    print("Downloading {0} of {1} images" .format(i, 10))
    response= requests.get(img_urls[i])
    file = open(r"C:\Users\sattu\Desktop\Fliprobo"+str(i)+".jpg", "wb")
    file.write(response.content)
```

```
Downloading 0 of 10 images
Downloading 1 of 10 images
Downloading 2 of 10 images
Downloading 3 of 10 images
Downloading 4 of 10 images
Downloading 5 of 10 images
Downloading 6 of 10 images
Downloading 7 of 10 images
Downloading 8 of 10 images
Downloading 9 of 10 images
```

Cars

In [50]:
```python
driver = webdriver.Chrome()
```

In [51]:
```python
driver.get('https://images.google.com/')
```

In [52]:
```python
designation=driver.find_element(By.CLASS_NAME,"gLFyf")
designation.send_keys("Cars")
```

In [53]:
```python
print("start scrolling to generate more images on the page...")
for _ in range(20):
    driver.execute_script("window.scrollBy(0,100)")
```

```
start scrolling to generate more images on the page...
```

In [54]:
```python
 images = driver.find_elements(By.XPATH,'//img[@class="rg_i Q4LuWd"]')
```

In [55]:
```python
img_urls = []
for image in images:
    source= image.get_attribute('src')
    if source is not None:
        if(source[0:4] == 'http'):
            img_urls.append(source)
```

In [56]:
```python
for i in range(len(img_urls)):
    if i >= 10:
        break
    print("Downloading {0} of {1} images" .format(i, 10))
    response= requests.get(img_urls[i])
    file = open(r"C:\Users\sattu\Desktop\Fliprobo"+str(i)+".jpg", "wb")
    file.write(response.content)
```

```
Downloading 0 of 10 images
Downloading 1 of 10 images
Downloading 2 of 10 images
Downloading 3 of 10 images
Downloading 4 of 10 images
Downloading 5 of 10 images
Downloading 6 of 10 images
Downloading 7 of 10 images
Downloading 8 of 10 images
Downloading 9 of 10 images
```

Machine Learning

driver = webdriver.Chrome()

In [58]:
```python
driver.get('https://images.google.com/')
```

In [59]:
```python
designation=driver.find_element(By.CLASS_NAME,"gLFyf")
designation.send_keys("Machine Learning")
```

In [62]:
```python
print("start scrolling to generate more images on the page...")
for _ in range(20):
    driver.execute_script("window.scrollBy(0,100)")
```

```
start scrolling to generate more images on the page...
```

In [64]:
```python
images = driver.find_elements(By.XPATH,'//img[@class="rg_i Q4LuWd"]')
```

In [65]:
```python
img_urls = []
for image in images:
    source= image.get_attribute('src')
    if source is not None:
        if(source[0:4] == 'http'):
            img_urls.append(source)
```

In [66]:
```python
for i in range(len(img_urls)):
    if i >= 10:
        break
    print("Downloading {0} of {1} images" .format(i, 10))
    response= requests.get(img_urls[i])
    file = open(r"C:\Users\sattu\Desktop\Fliprobo"+str(i)+".jpg", "wb")
    file.write(response.content)
```

```
Downloading 0 of 10 images
Downloading 1 of 10 images
Downloading 2 of 10 images
Downloading 3 of 10 images
Downloading 4 of 10 images
Downloading 5 of 10 images
Downloading 6 of 10 images
Downloading 7 of 10 images
Downloading 8 of 10 images
Downloading 9 of 10 images
```

Guitar

In [93]:
```python
driver = webdriver.Chrome()
```

In [94]:
```python
driver.get('https://images.google.com/')
```

In [95]:
```python
designation=driver.find_element(By.CLASS_NAME,"gLFyf")
designation.send_keys("Cakes")
```

In [73]:
```python
print("start scrolling to generate more images on the page...")
for _ in range(20):
    driver.execute_script("window.scrollBy(0,100)")
```

```
start scrolling to generate more images on the page...
```

In [74]:
```python
images = driver.find_elements(By.XPATH,'//img[@class="rg_i Q4LuWd"]')
```

In [75]:
```python
img_urls = []
for image in images:
    source= image.get_attribute('src')
    if source is not None:
        if(source[0:4] == 'http'):
            img_urls.append(source)
```

In [76]:
```python
for i in range(len(img_urls)):
    if i >= 10:
        break
    print("Downloading {0} of {1} images" .format(i, 10))
    response= requests.get(img_urls[i])
    file = open(r"C:\Users\sattu\Desktop\Fliprobo"+str(i)+".jpg", "wb")
    file.write(response.content)
```

```
Downloading 0 of 10 images
Downloading 1 of 10 images
Downloading 2 of 10 images
Downloading 3 of 10 images
Downloading 4 of 10 images
Downloading 5 of 10 images
Downloading 6 of 10 images
Downloading 7 of 10 images
Downloading 8 of 10 images
Downloading 9 of 10 images
```

Cakes

In [77]:
```python
driver = webdriver.Chrome()
```

In [78]:
```python
driver.get('https://images.google.com/')
```

In [79]:
```python
designation=driver.find_element(By.CLASS_NAME,"gLFyf")
designation.send_keys("Cakes")
```

In [89]:
```python
print("start scrolling to generate more images on the page...")
for _ in range(20):
    driver.execute_script("window.scrollBy(0,100)")
```

```
start scrolling to generate more images on the page...
```

In [90]:
```python
images = driver.find_elements(By.XPATH,'//img[@class="rg_i Q4LuWd"]')
```

In [91]:
```python
img_urls = []
for image in images:
    source= image.get_attribute('src')
    if source is not None:
        if(source[0:4] == 'http'):
            img_urls.append(source)
```

In [92]:
```python
for i in range(len(img_urls)):
    if i >= 10:
        break
    print("Downloading {0} of {1} images" .format(i, 10))
    response= requests.get(img_urls[i])
    file = open(r"C:\Users\sattu\Desktop\Fliprobo"+str(i)+".jpg", "wb")
    file.write(response.content)
```

```
Downloading 0 of 10 images
Downloading 1 of 10 images
Downloading 2 of 10 images
Downloading 3 of 10 images
Downloading 4 of 10 images
Downloading 5 of 10 images
Downloading 6 of 10 images
Downloading 7 of 10 images
Downloading 8 of 10 images
Downloading 9 of 10 images
```

In [ ]: