Web Scraping Assignment - 3

Importing necessary libraries

In [1]:
```
import pandas as pd
import selenium
from bs4 import BeautifulSoup
import time
from selenium import webdriver
import requests
import re
from selenium.common.exceptions import NoSuchElementException, StaleElementReferenceException
from selenium.webdriver.support.ui import WebDriverWait
```

Q1: Write a python program which searches all the product under a particular product vertical from www.amazon.in. The product verticals to be searched will be taken as input from user. For e.g. If user input is ‘guitar’. Then search for guitars.

In [2]:
```
# connecting to the webdriver
driver=webdriver.Chrome(r"C:/Users/HP/Downloads/chromedriver_win32 (1)/chromedriver.exe")
```

In [3]:
```
# getting the webpage of mentioned url
url = "https://www.amazon.in/"
driver.get(url)
```

In [4]:
```
# entering the product that we want to search
user_input = input('Enter the product that we want to search : ')
```

```
Enter the product that we want to search : Guitar
```

In [5]:
```
# searching the web element for user input
search = driver.find_element_by_id("twotabsearchtextbox")
search

# sending the user input to search bar
search.send_keys(user_input)

# locating the search button using xpath
search_btn = driver.find_element_by_xpath("//div[@class='nav-search-submit nav-sprite']/span/input")

# clicking on search button
search_btn.click()
```

In [ ]:
```

```

In [ ]:
```

```

Q2 : In the above question, now scrape the following details of each product listed in first 3 pages of your search results and save it in a data frame and csv. In case if any product vertical has less than 3 pages in search results then scrape all the products available under that product vertical. Details to be scraped are: "Brand Name", "Name of the Product", "Rating", "No. of Ratings", "Price", "Return/Exchange", "Expected Delivery", "Availability", "Other Details" and “Product URL”. In case, if any of the details are missing for any of the product then replace it by “-“.

In [6]:
```
# fetching URLs to open the pages
urls = []          # empty list
for i in range(0,3):      # for loop to scrape 3 pages
    page_url = driver.find_elements_by_xpath("//a[@class='a-link-normal a-text-normal']")
    for i in page_url:
        urls.append(i.get_attribute("href"))
        next_btn = driver.find_element_by_xpath("//li[@class='a-last']/a")
        time.sleep(3)
```

In [7]:
```
len(urls)
```

Out[7]:
```
186
```

In [8]:
```
# making empty list and fetching required data
brand_name = []
product_name = []
ratings = []
num_ratings = []
prices = []
exchange = []
exp_delivery = []
availability = []
other_details = []

for i in urls:
    driver.get(i)
```

```
    time.sleep(3)


    #fetching brand name
    try:
        brand = driver.find_element_by_xpath("//a[@id='bylineInfo']")
        brand_name.append(brand.text)
    except NoSuchElementException:
        brand_name.append('-')


    # fetching Name of the Product
    try:
        product = driver.find_element_by_xpath("//span[@id='productTitle']")
        product_name.append(product.text)
    except NoSuchElementException:
        product_name.append('-')



     #fetching ratings
    try:
        rating = driver.find_element_by_xpath("//span[@class='a-size-base a-nowrap']/span")
        ratings.append(rating.text)
    except NoSuchElementException:
        ratings.append('-')


    #fetching  no of ratings
    try:
        num_rating = driver.find_element_by_xpath("//span[@id='acrCustomerReviewText']")
        num_ratings.append(num_rating.text)
    except NoSuchElementException:
        num_ratings.append('-')


    #fetching price of the product
    try:
        price = driver.find_element_by_xpath("//td[@class='a-span12']")
        prices.append(price.text)
    except NoSuchElementException:
        prices.append('-')


    #fetching return/exchange
    try:
        exch = driver.find_element_by_xpath("//span[@class='a-declarative']/div/a")
        exchange.append(exch.text)
    except NoSuchElementException:
        exchange.append('-')


    #fetching expected delivery
    try:
        delivery = driver.find_element_by_xpath("//div[@class='a-section a-spacing-mini']/b")
        exp_delivery.append(delivery.text)
    except NoSuchElementException:
        exp_delivery.append('-')


    #fetching availability information
    try:
        avail = driver.find_element_by_xpath("//span[@class='a-size-medium a-color-success']")
        availability.append(avail.text)
    except NoSuchElementException:
        availability.append('-')

    #other details
    try:
        oth_det = driver.find_element_by_xpath("//ul[@class='a-unordered-list a-vertical a-spacing-mini']")
        other_details.append(oth_det.text)
    except NoSuchElementException:
        other_details.append('-')
```

In [9]:

```
print(len(brand_name),
len(product_name),
len(ratings),
len(num_ratings),
len(prices),
len(exchange),
len(exp_delivery),
len(availability),
len(other_details))
```

```
186 186 186 186 186 186 186 186 186
```

In [11]:

```
# Creating the DataFrame for the scraped data

guitar = pd.DataFrame({})
guitar['Brand Name'] = brand_name
guitar['Name of the Product'] = product_name
guitar['Rating'] = ratings
guitar['No. of Ratings'] = num_ratings
guitar['Price'] = prices
guitar['Return/Exchange'] = exchange
guitar['Expected Delivery'] = exp_delivery
guitar['Availability'] = availability
guitar['Other Details'] = other_details
guitar['Product URL'] = urls
guitar
```

Out[11]:

	Brand Name	Name of the Product	Rating	No. of Ratings	Price	Return/Exchange	Expected Delivery	Availability	Other Details	Product URL
0	Brand: KETOSTICS	Ketostics Givson Venus Special Guitar Combo (R...	4 out of 5	38 ratings	₹3,999.00	7 Days Replacement	Sunday, Oct 31	In stock.	6-Strings| Acoustic Guitar | Right-Handed| Red...	https://www.amazon.in/gp/slredirect/picassoRed...

1	Brand: KETOSTICS	Ketostics Givson Venus Special Guitar Combo (B...	5 out of 5	1 rating	₹4,199.00	7 Days Replacement	Sunday, Oct 31	In stock.	Venus Special 6 Strings | Acoustic Guitar | Co...	https://www.amazon.in/gp/slredirect/picassoRed...
2	Brand: KETOSTICS	Ketostics Givson Venus Special Guitar (RED) VS...	4 out of 5	1 rating	₹3,999.00	7 Days Replacement	Sunday, Oct 31	In stock.	6-Strings|41" Acoustic Guitar | Right-Handed| ...	https://www.amazon.in/gp/slredirect/picassoRed...
3	Visit the Guitar Bro Store	GUITAR BRO - COMBO (Black Acoustic Guitar for ...	3.7 out of 5	28 ratings	₹6,499.00	7 Days Replacement	Oct 29 - Nov 2	In stock.	GUITAR BRO +20 mins FREE VIDEO Demo - is a lea...	https://www.amazon.in/gp/slredirect/picassoRed...
4	Visit the JUAREZ Store	Juârez Acoustic Guitar, 38 Inch Cutaway, 038C ...	4 out of 5	13,329 ratings	₹2,199.00	7 Days Replacement	Tuesday, Oct 26	In stock.	Black Glossy Finish, Number of Frets: 18, Acou...	https://www.amazon.in/Juarez-Acoustic-Cutaway-...
...	...	...	...	...	...	...	...	...	...	...
181	Brand: TECHBLAZE	TECHBLAZE Acoustic Guitar Strings, Guitar Wall...	5 out of 5	1 rating	₹449.00	7 Days Replacement	Saturday, Oct 30	In stock.	Guitar Capo.- Easy to clip on guitar and quick...	https://www.amazon.in/TECHBLAZE-Acoustic-Guita...
182	Brand: Honestum	Honestum 4-string acoustic guitar learning kid...	3.9 out of 5	3 ratings	₹999.00	7 Days Replacement	Wednesday, Oct 27	In stock.	Very Crisp and Clear Sound of Strings\nThis gu...	https://www.amazon.in/Honestum-4-string-acoust...
183	Brand: Crizer	Crizer 4 String Guitar Children's Musical Inst...	3.7 out of 5	91 ratings	₹1,099.00	7 Days Replacement	Wednesday, Oct 27	In stock.	PROVIDES SCREEN FREE FUN: A junior scale 4 str...	https://www.amazon.in/Crizer-Childrens-Instrum...
184	Brand: HRB MUSICALS	HRB MUSICALS GUITAR FLOOR STAND +GUITAR CAPO +...	3.8 out of 5	7 ratings	₹749.00	7 Days Replacement	Saturday, Oct 30	In stock.	For guitar floor stand its solid body ,string ...	https://www.amazon.in/gp/slredirect/picassoRed...
185	Brand: HRB musicals	HRB MUSICALS combo of guitar String Winder, gu...	3.7 out of 5	7 ratings	₹399.00	7 Days Replacement	Saturday, Oct 30	In stock.	GUITAR ALICE STRING (SET OF 6 STRING)\nGUITAR ...	https://www.amazon.in/gp/slredirect/picassoRed...

186 rows × 10 columns

In [12]:
```
#saving the data in csv
guitar.to_csv("Guitar.csv")
```

In [13]:
```
driver.close()
```

In [ ]:

In [ ]:

Q3 : Write a python program to access the search bar and search button on images.google.com and scrape 100 images each for keywords ‘fruits’, ‘cars’ and ‘Machine Learning’.

In [14]:
```
# connecting to the webdriver
driver=webdriver.Chrome(r"C:/Users/HP/Downloads/chromedriver_win32 (1)/chromedriver.exe")
```

In [17]:
```
# geting the webpage of mentioned url
url = "http://images.google.com/"

# creating empty list
urls = []
data = []

search_item = ["Fruits","Cars","Machine Learning"]
for item in search_item:
    driver.get(url)
    time.sleep(5)

    # finding webelement for search_bar
    search_bar = driver.find_element_by_tag_name("input")

    # sending keys to get the keyword for search bar
    search_bar.send_keys(str(item))

    # clicking on search button
    search_button = driver.find_element_by_xpath("//button[@class='Tg7LZd']").click()

    # scroling down the webpage to get some more images
    for _ in range(500):
        driver.execute_script("window.scrollBy(0,100)")

        imgs = driver.find_elements_by_xpath("//img[@class='rg_i Q4LuWd']")
    img_url = []
    for image in imgs:
        source = image.get_attribute('src')
        if source is not None:
            if(source[0:4] == 'http'):
                img_url.append(source)
    for i in img_url[:100]:
        urls.append(i)

for i in range(len(urls)):
    if i >= 300:
        break
    print("Doenloading {0} of {1} images" .format(i,300))
    response = requests.get(urls[i])

    file = open(r"E:\google\images"+str(i)+".jpg","wb")
```

```
    file.write(response.content)
```

```
Doenloading 0 of 300 images
Doenloading 1 of 300 images
Doenloading 2 of 300 images
Doenloading 3 of 300 images
Doenloading 4 of 300 images
Doenloading 5 of 300 images
Doenloading 6 of 300 images
Doenloading 7 of 300 images
Doenloading 8 of 300 images
Doenloading 9 of 300 images
Doenloading 10 of 300 images
Doenloading 11 of 300 images
Doenloading 12 of 300 images
Doenloading 13 of 300 images
Doenloading 14 of 300 images
Doenloading 15 of 300 images
Doenloading 16 of 300 images
Doenloading 17 of 300 images
Doenloading 18 of 300 images
Doenloading 19 of 300 images
Doenloading 20 of 300 images
Doenloading 21 of 300 images
Doenloading 22 of 300 images
Doenloading 23 of 300 images
Doenloading 24 of 300 images
Doenloading 25 of 300 images
Doenloading 26 of 300 images
Doenloading 27 of 300 images
Doenloading 28 of 300 images
Doenloading 29 of 300 images
Doenloading 30 of 300 images
Doenloading 31 of 300 images
Doenloading 32 of 300 images
Doenloading 33 of 300 images
Doenloading 34 of 300 images
Doenloading 35 of 300 images
Doenloading 36 of 300 images
Doenloading 37 of 300 images
Doenloading 38 of 300 images
Doenloading 39 of 300 images
Doenloading 40 of 300 images
Doenloading 41 of 300 images
Doenloading 42 of 300 images
Doenloading 43 of 300 images
Doenloading 44 of 300 images
Doenloading 45 of 300 images
Doenloading 46 of 300 images
Doenloading 47 of 300 images
Doenloading 48 of 300 images
Doenloading 49 of 300 images
Doenloading 50 of 300 images
Doenloading 51 of 300 images
Doenloading 52 of 300 images
Doenloading 53 of 300 images
Doenloading 54 of 300 images
Doenloading 55 of 300 images
Doenloading 56 of 300 images
Doenloading 57 of 300 images
Doenloading 58 of 300 images
Doenloading 59 of 300 images
Doenloading 60 of 300 images
Doenloading 61 of 300 images
Doenloading 62 of 300 images
Doenloading 63 of 300 images
Doenloading 64 of 300 images
Doenloading 65 of 300 images
Doenloading 66 of 300 images
Doenloading 67 of 300 images
Doenloading 68 of 300 images
Doenloading 69 of 300 images
Doenloading 70 of 300 images
Doenloading 71 of 300 images
Doenloading 72 of 300 images
Doenloading 73 of 300 images
Doenloading 74 of 300 images
Doenloading 75 of 300 images
Doenloading 76 of 300 images
Doenloading 77 of 300 images
Doenloading 78 of 300 images
Doenloading 79 of 300 images
Doenloading 80 of 300 images
Doenloading 81 of 300 images
Doenloading 82 of 300 images
Doenloading 83 of 300 images
Doenloading 84 of 300 images
Doenloading 85 of 300 images
Doenloading 86 of 300 images
Doenloading 87 of 300 images
Doenloading 88 of 300 images
Doenloading 89 of 300 images
Doenloading 90 of 300 images
Doenloading 91 of 300 images
Doenloading 92 of 300 images
Doenloading 93 of 300 images
Doenloading 94 of 300 images
Doenloading 95 of 300 images
Doenloading 96 of 300 images
Doenloading 97 of 300 images
Doenloading 98 of 300 images
Doenloading 99 of 300 images
Doenloading 100 of 300 images
Doenloading 101 of 300 images
Doenloading 102 of 300 images
Doenloading 103 of 300 images
Doenloading 104 of 300 images
Doenloading 105 of 300 images
Doenloading 106 of 300 images
Doenloading 107 of 300 images
```

```
Doenloading 108 of 300 images
Doenloading 109 of 300 images
Doenloading 110 of 300 images
Doenloading 111 of 300 images
Doenloading 112 of 300 images
Doenloading 113 of 300 images
Doenloading 114 of 300 images
Doenloading 115 of 300 images
Doenloading 116 of 300 images
Doenloading 117 of 300 images
Doenloading 118 of 300 images
Doenloading 119 of 300 images
Doenloading 120 of 300 images
Doenloading 121 of 300 images
Doenloading 122 of 300 images
Doenloading 123 of 300 images
Doenloading 124 of 300 images
Doenloading 125 of 300 images
Doenloading 126 of 300 images
Doenloading 127 of 300 images
Doenloading 128 of 300 images
Doenloading 129 of 300 images
Doenloading 130 of 300 images
Doenloading 131 of 300 images
Doenloading 132 of 300 images
Doenloading 133 of 300 images
Doenloading 134 of 300 images
Doenloading 135 of 300 images
Doenloading 136 of 300 images
Doenloading 137 of 300 images
Doenloading 138 of 300 images
Doenloading 139 of 300 images
Doenloading 140 of 300 images
Doenloading 141 of 300 images
Doenloading 142 of 300 images
Doenloading 143 of 300 images
Doenloading 144 of 300 images
Doenloading 145 of 300 images
Doenloading 146 of 300 images
Doenloading 147 of 300 images
Doenloading 148 of 300 images
Doenloading 149 of 300 images
Doenloading 150 of 300 images
Doenloading 151 of 300 images
Doenloading 152 of 300 images
Doenloading 153 of 300 images
Doenloading 154 of 300 images
Doenloading 155 of 300 images
Doenloading 156 of 300 images
Doenloading 157 of 300 images
Doenloading 158 of 300 images
Doenloading 159 of 300 images
Doenloading 160 of 300 images
Doenloading 161 of 300 images
Doenloading 162 of 300 images
Doenloading 163 of 300 images
Doenloading 164 of 300 images
Doenloading 165 of 300 images
Doenloading 166 of 300 images
Doenloading 167 of 300 images
Doenloading 168 of 300 images
Doenloading 169 of 300 images
Doenloading 170 of 300 images
Doenloading 171 of 300 images
Doenloading 172 of 300 images
Doenloading 173 of 300 images
Doenloading 174 of 300 images
Doenloading 175 of 300 images
Doenloading 176 of 300 images
Doenloading 177 of 300 images
Doenloading 178 of 300 images
Doenloading 179 of 300 images
Doenloading 180 of 300 images
Doenloading 181 of 300 images
Doenloading 182 of 300 images
Doenloading 183 of 300 images
Doenloading 184 of 300 images
Doenloading 185 of 300 images
Doenloading 186 of 300 images
Doenloading 187 of 300 images
Doenloading 188 of 300 images
Doenloading 189 of 300 images
Doenloading 190 of 300 images
Doenloading 191 of 300 images
Doenloading 192 of 300 images
Doenloading 193 of 300 images
Doenloading 194 of 300 images
Doenloading 195 of 300 images
Doenloading 196 of 300 images
Doenloading 197 of 300 images
Doenloading 198 of 300 images
Doenloading 199 of 300 images
Doenloading 200 of 300 images
Doenloading 201 of 300 images
Doenloading 202 of 300 images
Doenloading 203 of 300 images
Doenloading 204 of 300 images
Doenloading 205 of 300 images
Doenloading 206 of 300 images
Doenloading 207 of 300 images
Doenloading 208 of 300 images
Doenloading 209 of 300 images
Doenloading 210 of 300 images
Doenloading 211 of 300 images
Doenloading 212 of 300 images
Doenloading 213 of 300 images
Doenloading 214 of 300 images
Doenloading 215 of 300 images
Doenloading 216 of 300 images
Doenloading 217 of 300 images
Doenloading 218 of 300 images
Doenloading 219 of 300 images
```

```
Doenloading 220 of 300 images
Doenloading 221 of 300 images
Doenloading 222 of 300 images
Doenloading 223 of 300 images
Doenloading 224 of 300 images
Doenloading 225 of 300 images
Doenloading 226 of 300 images
Doenloading 227 of 300 images
Doenloading 228 of 300 images
Doenloading 229 of 300 images
Doenloading 230 of 300 images
Doenloading 231 of 300 images
Doenloading 232 of 300 images
Doenloading 233 of 300 images
Doenloading 234 of 300 images
Doenloading 235 of 300 images
Doenloading 236 of 300 images
Doenloading 237 of 300 images
Doenloading 238 of 300 images
Doenloading 239 of 300 images
Doenloading 240 of 300 images
Doenloading 241 of 300 images
Doenloading 242 of 300 images
Doenloading 243 of 300 images
Doenloading 244 of 300 images
Doenloading 245 of 300 images
Doenloading 246 of 300 images
Doenloading 247 of 300 images
Doenloading 248 of 300 images
Doenloading 249 of 300 images
Doenloading 250 of 300 images
Doenloading 251 of 300 images
Doenloading 252 of 300 images
Doenloading 253 of 300 images
Doenloading 254 of 300 images
Doenloading 255 of 300 images
Doenloading 256 of 300 images
Doenloading 257 of 300 images
Doenloading 258 of 300 images
Doenloading 259 of 300 images
Doenloading 260 of 300 images
Doenloading 261 of 300 images
Doenloading 262 of 300 images
Doenloading 263 of 300 images
Doenloading 264 of 300 images
Doenloading 265 of 300 images
Doenloading 266 of 300 images
Doenloading 267 of 300 images
Doenloading 268 of 300 images
Doenloading 269 of 300 images
Doenloading 270 of 300 images
Doenloading 271 of 300 images
Doenloading 272 of 300 images
Doenloading 273 of 300 images
Doenloading 274 of 300 images
Doenloading 275 of 300 images
Doenloading 276 of 300 images
Doenloading 277 of 300 images
Doenloading 278 of 300 images
Doenloading 279 of 300 images
Doenloading 280 of 300 images
Doenloading 281 of 300 images
Doenloading 282 of 300 images
Doenloading 283 of 300 images
Doenloading 284 of 300 images
Doenloading 285 of 300 images
Doenloading 286 of 300 images
Doenloading 287 of 300 images
Doenloading 288 of 300 images
Doenloading 289 of 300 images
Doenloading 290 of 300 images
Doenloading 291 of 300 images
Doenloading 292 of 300 images
Doenloading 293 of 300 images
Doenloading 294 of 300 images
Doenloading 295 of 300 images
Doenloading 296 of 300 images
Doenloading 297 of 300 images
Doenloading 298 of 300 images
Doenloading 299 of 300 images
```

In [18]:
```
driver.close()
```

In [ ]:
```
```

In [ ]:
```
```

Q4 : 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. Details to be scraped: “Brand Name”, “Smartphone name”, “Colour”, “RAM”, “Storage(ROM)”, “Primary Camera”, “Secondary Camera”, “Display Size”, “Display Resolution”, “Processor”, “Processor Cores”, “Battery Capacity”, “Price”, “Product URL”. Incase if any of the details is missing then replace it by “- “. Save your results in a dataframe and CSV.

In [2]:
```
# connecting to the webdriver
driver=webdriver.Chrome(r"C:/Users/HP/Downloads/chromedriver_win32 (1)/chromedriver.exe")
```

In [3]:

```
# getting the webpage of mentioned url
url = "https://www.flipkart.com/"
driver.get(url)
```

In [4]:
```
# closing login popup button
lonin_x_btn = driver.find_element_by_xpath("//div[@class='_2QfC02']//button").click()
```

In [5]:
```
# search for web element
search_bar = driver.find_element_by_xpath("//input[@class='_3704LK']")

# sending keys to search product
search_bar.send_keys("pixel 4A")
```

In [6]:
```
# location the search button using xpath
search_btn = driver.find_element_by_xpath("//button[@class='L0Z3Pu']")

# clicking on search button
search_btn.click()
```

In [7]:
```
# fetching 1st page of URLs of smartphone
page1_url = []
urls = driver.find_elements_by_xpath("//a[@class='_1fQZEK']")
for url in urls:
    page1_url.append(url.get_attribute('href'))
```

In [8]:
```
len(page1_url)
```

Out[8]:
```
24
```

In [9]:
```
# creating empty list
Smartphones = ({})
Smartphones['Brand'] = []
Smartphones['Phone name'] = []
Smartphones['Colour'] = []
Smartphones['RAM'] = []
Smartphones['Storage(ROM)'] = []
Smartphones['Primary Camera'] = []
Smartphones['Secondary Camera'] = []
Smartphones['Display Size'] = []
Smartphones['Display Resolution'] = []
Smartphones['Processor'] = []
Smartphones['Processor Cores'] = []
Smartphones['Battery Capacity'] = []
Smartphones['Price'] = []
Smartphones['URL'] = []
```

In [10]:
```
# scraping data from each url of page 1
for url in page1_url:
    driver.get(url)
    print("Scraping URL = ",url)
    Smartphones['URL'].append(url)
    time.sleep(2)


    #clicking on read more button to get more information
    try:
        read_more = driver.find_element_by_xpath("//button[@class='_2KpZ6l _1FH0tX']")
        read_more.click()
    except NoSuchElementException:
        print("Exception occured while moving to next page")

    #scraping brand name of smartphone
    try:
        brand_tags = driver.find_element_by_xpath("//span[@class='B_NuCI']")
        Smartphones['Brand'].append(brand_tags.text.split()[0])
    except NoSuchElementException:
        Smartphones['Brand'].append('-')


    # scraping name of smartphones
    try:
        name_tags = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][1]/table/tbody/tr[3]/td[2]/ul/li")
        Smartphones['Phone name'].append(name_tags.text)
    except NoSuchElementException:
        Smartphones['Phone name'].append('-')

    #scraping colour of smartphone
    try:
        color_tags = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][1]/table/tbody/tr[4]/td[2]/ul/li")
        Smartphones['Colour'].append(color_tags.text)
    except NoSuchElementException:
        Smartphones['Colour'].append('-')

    # scraping RAM data of smartphone
    try:
        ram_tags = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][4]/table[1]/tbody/tr[2]/td[2]/ul/li")
        Smartphones['RAM'].append(ram_tags.text)
    except NoSuchElementException:
        Smartphones['RAM'].append('-')

    #scraping ROM data of smartphones
    try:
        rom = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][4]/table[1]/tbody/tr[1]/td[2]/ul/li")
        Smartphones['Storage(ROM)'].append(rom.text)
    except NoSuchElementException:
        Smartphones['Storage(ROM)'].append('-')

    # scraping  Primary camera data of smartphone
    try:
        pri =driver.find_element_by_xpath("//div[@class='_3k-BhJ'][5]/table[1]/tbody/tr[2]/td[2]/ul/li")
        Smartphones['Primary Camera'].append(pri.text)
    except NoSuchElementException:
```

```
        Smartphones['Primary Camera'].append('-')

    # scraping secondary camera data of smartphone
    try:
        sec = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][5]/table[1]/tbody/tr[6]/td[1]")
        if sec != 'Secondary Camera' :
            if driver.find_element_by_xpath("//div[@class='_3k-BhJ'][5]/table[1]/tbody/tr[5]/td[1]").text == "Secondary Camera":
                sec_cam =driver.find_element_by_xpath("//div[@class='_3k-BhJ'][5]/table[1]/tbody/tr[5]/td[2]/ul/li")
            else :
                raise NoSuchElementException
        else :
            sec_cam = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][5]/table[1]/tbody/tr[6]/td[2]/ul/li")
        Smartphones['Secondary Camera'].append(sec_cam.text)
    except NoSuchElementException:
        Smartphones['Secondary Camera'].append('-')


    #scraping display size data of smartphone
    try:
        disp = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][2]/div")
        if disp.text != 'Display Features' : raise NoSuchElementException
        disp_size = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][2]/table[1]/tbody/tr[1]/td[2]/ul/li")
        Smartphones['Display Size'].append(disp_size.text)
    except NoSuchElementException:
        Smartphones['Display Size'].append('-')


    #scraping display resolution of smartphone
    try:
        disp = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][2]/div")
        if disp.text != 'Display Features' : raise NoSuchElementException
        disp_reso = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][2]/table[1]/tbody/tr[2]/td[2]/ul/li")
        Smartphones['Display Resolution'].append(disp_reso.text)
    except NoSuchElementException:
        Smartphones['Display Resolution'].append('-')


    #scraping processor of smartphone
    try:
        pro = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][3]/table[1]/tbody/tr[2]/td[1]]")
        if pro.text != 'Processor Type' : raise NoSuchElementException
        processor = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][3]/table[1]/tbody/tr[2]/td[2]/ul/li")
        Smartphones['Processor'].append(processor.text)
    except NoSuchElementException:
        Smartphones['Processor'].append('-')



    # scraping processor core of smartphone
    try:
        core = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][3]/table[1]/tbody/tr[3]/td[1]")
        if core.text != 'Processor Core' :
            core = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][3]/table[1]/tbody/tr[2]/td[1]")
            if core.text != 'Processor Core' :
                raise NoSuchElementException
            else :
                cores = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][3]/table[1]/tbody/tr[2]/td[2]/ul/li")
        else :
            cores = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][3]/table[1]/tbody/tr[3]/td[2]/ul/li")
        Smartphones['Processor Cores'].append(disp_reso.text)
    except NoSuchElementException:
        Smartphones['Processor Cores'].append('-')



    # scraping the battery capacity of smartphone
    try:
        if driver.find_element_by_xpath("//div[@class='_3k-BhJ'][10]/div").text != "Battery & Power Features" :
            if driver.find_element_by_xpath("//div[@class='_3k-BhJ'][9]/div").text == "Battery & Power Features" :
                bat_tags = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][9]/table/tbody/tr/td[1]")
                if bat_tags.text != "Battery Capacity" : raise NoSuchElementException
                bat_capa = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][9]/table/tbody/tr/td[2]/ul/li")
            elif driver.find_element_by_xpath("//div[@class='_3k-BhJ'][8]/div").text == "Battery & Power Features" :
                bat_tags = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][8]/table/tbody/tr/td[1]")
                if bat_tags.text != "Battery Capacity" : raise NoSuchElementException
                bat_capa = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][8]/table/tbody/tr/td[2]/ul/li")
            else:
                raise NoSuchElementException
        else :
            bat_tags = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][10]/table/tbody/tr/td[1]")
            if bat_tags.text != "Battery Capacity" : raise NoSuchElementException
            bat_capa = driver.find_element_by_xpath("//div[@class='_3k-BhJ'][10]/table/tbody/tr/td[2]/ul/li")
        Smartphones['Battery Capacity'].append(bat_capa.text)
    except NoSuchElementException:
        Smartphones['Battery Capacity'].append('-')



    # scraping price of smartphone
    try:
        price_tags = driver.find_element_by_xpath("//div[@class='_30jeq3 _16Jk6d']")
        Smartphones['Price'].append(price_tags.text)
    except NoSuchElementException:
          Smartphones['Price'].append('-')
```

```
Scraping URL =  https://www.flipkart.com/google-pixel-4a-just-black-128-gb/p/itm023b9677aa45d?pid=MOBFUSBNAZGY7HQU&lid=LSTMOBFUSBNAZGY7HQUWHTF0C&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_1&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBFUSBNAZGY7HQU.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze8000000163497327161 4&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/redmi-9-power-fiery-red-64-gb/p/itmcd78b9bc04aa9?pid=MOBFYZF8XVSHPK5M&lid=LSTMOBFYZF8XVSHPK5MMUYMJE&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_2&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBFYZF8XVSHPK5M.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze8000000163497327161 4&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/redmi-9-power-mighty-black-64-gb/p/itm481ab234a6553?pid=MOBFYZ94UWPWQRNF&lid=LSTMOBFYZ94UWPWQRNFAT4EPE&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_3&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBFYZ94UWPWQRNF.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze8000000163497327161 4&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/redmi-9-power-blazing-blue-64-gb/p/itm9533b02ba34ef?pid=MOBFYZ7HHGJUATYD&lid=LSTMOBFYZ7HHGJUATYDOCEFJ&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_4&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBFYZ7HHGJUATYD.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze8000000163497327161 4&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/redmi-9-carbon-black-64-gb/p/itm4fb151383983b?pid=MOBFVHMPGATFZXWR&lid=LSTMOBFVHMPGATFZXWRGISFTZ&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_5&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBFVHMPGATFZXWR.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze8000000163497327161 4&qH=9b26a23b2cff510d
```

```
Scraping URL =  https://www.flipkart.com/redmi-9-sky-blue-64-gb/p/itm4fb151383983b?pid=MOBFV5FPUS8UHG9H&lid=LSTMOBFV5FPUS8UHG9HSKN7BI&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_6&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBFV5FPUS8UHG9H.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/kall-z5-green-16-gb/p/itmd861401a8b939?pid=MOBG7FGHHHHZFHZ9&lid=LSTMOBG7FGHHHHZFHZ9BQANIH&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_7&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBG7FGHHHHZFHZ9.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/oppo-a16-crystal-black-64-gb/p/itmeeb6ad985655f?pid=MOBG7FKZKUR7MNYZ&lid=LSTMOBG7FKZKUR7MNYZLKC2TP&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_8&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBG7FKZKUR7MNYZ.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/oppo-a16-pearl-blue-64-gb/p/itmeeb6ad985655f?pid=MOBG7FNH8GMZGMYT&lid=LSTMOBG7FNH8GMZGMYTIP1FAG&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_9&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBG7FNH8GMZGMYT.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/redmi-10-prime-astral-white-128-gb/p/itm9377b4c936baa?pid=MOBG6RGJGTKTUVD9&lid=LSTMOBG6RGJGTKTUVD9PEZNSP&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_10&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBG6RGJGTKTUVD9.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/tecno-spark-7t-nebula-orange-128-gb/p/itmd78a578fca8d6?pid=MOBG7F6RUGHFTDPS&lid=LSTMOBG7F6RUGHFTDPS6FMXMN&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_11&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBG7F6RUGHFTDPS.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/samsung-m02s-black-32-gb/p/itme9edd93df8425?pid=MOBFZG4CZGKXZBQ2&lid=LSTMOBFZG4CZGKXZBQ2MCOQGI&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_12&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBFZG4CZGKXZBQ2.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/redmi-9-sporty-orange-64-gb/p/itm4fb151383983b?pid=MOBFV5FPCJC9ZKRB&lid=LSTMOBFV5FPCJC9ZKRBZGOLK2&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_13&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBFV5FPCJC9ZKRB.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/samsung-galaxy-m02-black-32-gb/p/itm16e9cd93db916?pid=MOBGY73CSBKNXA5W&lid=LSTMOBGY73CSBKNXA5WNQLDVS&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_14&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBGY73CSBKNXA5W.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/panasonic-eluga-i6-black-16-gb/p/itm876e1eeb85ac1?pid=MOBFS3GJ5CR7PVEW&lid=LSTMOBFS3GJ5CR7PVEWNORSVR&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_15&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBFS3GJ5CR7PVEW.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/redmi-9-power-mighty-black-128-gb/p/itm74c84b4084cc5?pid=MOBGYMD9AVZZH9WX&lid=LSTMOBGYMD9AVZZH9WXIJAHX4&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_16&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBGYMD9AVZZH9WX.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/samsung-galaxy-m31-ocean-blue-128-gb/p/itm1e7ce328b07b6?pid=MOBFUYQTFBAYCFKG&lid=LSTMOBFUYQTFBAYCFKGCDJE4J&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_17&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBFUYQTFBAYCFKG.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/itel-it5026/p/itm2d630b978656b?pid=MOBFXMP7GSETA7RX&lid=LSTMOBFXMP7GSETA7RXEFRNOE&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_18&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBFXMP7GSETA7RX.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/redmi-note-10s-cosmic-purple-128-gb/p/itm3ca65d9e1c1e9?pid=MOBG6WTAV33CMGGY&lid=LSTMOBG6WTAV33CMGGYIUGWAW&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_19&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBG6WTAV33CMGGY.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/redmi-9a-midnight-black-32-gb/p/itmeabd39a0cd669?pid=MOBFVDA2RWTCFBHZ&lid=LSTMOBFVDA2RWTCFBHZRWDIZM&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_20&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBFVDA2RWTCFBHZ.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/oppo-a55-starry-black-64-gb/p/itmfd07b94a9a543?pid=MOBG7FMJGBPWKCQT&lid=LSTMOBG7FMJGBPWKCQTSLN8OB&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_21&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBG7FMJGBPWKCQT.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/oppo-a55-rainbow-blue-128-gb/p/itmfd07b94a9a543?pid=MOBG7PWRERKFXKYX&lid=LSTMOBG7PWRERKFXKYXIHS3WW&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_22&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBG7PWRERKFXKYX.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/oppo-a55-starry-black-128-gb/p/itmfd07b94a9a543?pid=MOBG7PX2FNAZHDJZ&lid=LSTMOBG7PX2FNAZHDJZGMRQ2E&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_23&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBG7PX2FNAZHDJZ.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
Scraping URL =  https://www.flipkart.com/oppo-a55-rainbow-blue-64-gb/p/itmfd07b94a9a543?pid=MOBG7FMQYH2HAM64&lid=LSTMOBG7FMQYH2HAM64NINTAC&marketplace=FLIPKART&q=pixel+4A&store=tyy%2F4io&srno=s_1_24&otracker=search&otracker1=search&fm=SEARCH&iid=51c07edd-7e2d-475f-9453-c60d7a8e19c5.MOBG7FMQYH2HAM64.SEARCH&ppt=hp&ppn=homepage&ssid=x57itk1ze80000001634973271614&qH=9b26a23b2cff510d
```

In [11]:

```
# checking lengths of all scraped data

print(len(Smartphones['Brand']),len(Smartphones['Phone name']), len(Smartphones['Colour']),len(Smartphones['RAM']),len(Smartphones['Storag
```

```
24 24 24 24 24 24 24 24 24 24 24 24 24 24
```

In [12]:

```
# framing the DataFrame

df = pd.DataFrame.from_dict(Smartphones)
df
```

Out[12]:

	Brand	Phone name	Colour	RAM	Storage(ROM)	Primary Camera	Secondary Camera	Display Size	Display Resolution	Processor	Processor Cores	Battery Capacity	Price	URL
0	Google	Pixel 4a	Just Black	6 GB	128 GB	12.2MP Rear Camera	8MP Front Camera	14.76 cm (5.81 inch)	2340 x 1080 Pixels	-	2340 x 1080 Pixels	3140 mAh	₹25,999	https://www.flipkart.com/google-pixel-4a-just-...
1	REDMI	9 Power	Fiery Red	4 GB	64 GB	48MP + 8MP + 2MP + 2MP	8MP Front Camera	16.59 cm (6.53 inch)	2340 x 1080$$pixel	-	2340 x 1080$$pixel	6000 mAh	₹11,499	https://www.flipkart.com/redmi-9-power-fiery-r...
2	REDMI	9 Power	Mighty Black	4 GB	64 GB	48MP + 8MP + 2MP + 2MP	8MP Front Camera	16.59 cm (6.53 inch)	2340 x 1080$$pixel	-	2340 x 1080$$pixel	6000 mAh	₹11,499	https://www.flipkart.com/redmi-9-power-mighty-...
3	REDMI	9 Power	Blazing Blue	4 GB	64 GB	48MP + 8MP + 2MP + 2MP	8MP Front Camera	16.59 cm (6.53 inch)	2340 x 1080$$pixel	-	2340 x 1080$$pixel	6000 mAh	₹12,888	https://www.flipkart.com/redmi-9-power-blazing...
4	Redmi	Redmi 9	Carbon Black	4 GB	64 GB	13MP + 2MP	5MP Front Camera	16.59 cm (6.53 inch)	720 x 1600$$pixel	-	720 x 1600$$pixel	5000 mAh	₹9,348	https://www.flipkart.com/redmi-9-carbon-black-...
5	Redmi	Redmi 9	Sky Blue	4 GB	64 GB	13MP + 2MP	5MP Front Camera	16.59 cm (6.53 inch)	720 x 1600$$pixel	-	-	5000 mAh	₹9,285	https://www.flipkart.com/redmi-9-sky-blue-64-g...
6	I	Z5	Green	3 GB	16 GB	8MP Rear Camera	5MP Front Camera	13.84 cm (5.45 inch)	480 x 960$$pixel	-	480 x 960$$pixel	3000 mAh	₹4,995	https://www.flipkart.com/kall-z5-green-16-gb/p...
7	OPPO	A16	Crystal Black	4 GB	64 GB	Primary Camera	-	16.56 cm (6.52 inch)	720 x 1080$$pixel	-	720 x 1080$$pixel	-	₹12,690	https://www.flipkart.com/oppo-a16-crystal-blac...
8	OPPO	A16	Pearl Blue	4 GB	64 GB	Primary Camera	-	16.56 cm (6.52 inch)	720 x 1080$$pixel	-	720 x 1080$$pixel	-	₹13,490	https://www.flipkart.com/oppo-a16-pearl-blue-6...

9	REDMI	10 Prime	Astral White	6 GB	128 GB	8MP + 2MP + 2MP	8MP Front Camera	16.51 cm (6.5 inch)	2400 x 1080$$Pixel	-	2400 x 1080$$Pixel	6000 mAh	₹15,497	https://www.flipkart.com/redmi-10-prime-astral...
10	Tecno	Spark 7T	Nebula Orange	4 GB	128 GB	48MP Rear Camera	8MP Front Camera	16.56 cm (6.52 inch)	720 x 1600$$pixel	-	720 x 1600$$pixel	6000 mAh	₹10,990	https://www.flipkart.com/tecno-spark-7t-nebula...
11	SAMSUNG	M02s	Black	3 GB	32 GB	13MP + 2MP + 2MP	-	16.51 cm (6.5 inch)	720 x 1600$$pixel	-	720 x 1600$$pixel	5000 mAh	₹9,388	https://www.flipkart.com/samsung-m02s-black-32...
12	Redmi	Redmi 9	Sporty Orange	4 GB	64 GB	13MP + 2MP	5MP Front Camera	16.59 cm (6.53 inch)	720 x 1600$$pixel	-	-	5000 mAh	₹9,469	https://www.flipkart.com/redmi-9-sporty-orange...
13	SAMSUNG	Galaxy M02	Black	2 GB	32 GB	13MP + 2MP	-	16.64 cm (6.55 inch)	720 x 1600 $$Pixel	-	720 x 1600 $$Pixel	5000 mAh	₹8,256	https://www.flipkart.com/samsung-galaxy-m02-bl...
14	Panasonic	ELUGA I6	Black	2 GB	16 GB	8MP Rear Camera	5MP Front Camera	13.84 cm (5.45 inch)	960 x 480 pixel	-	960 x 480 pixel	3000 mAh	₹5,490	https://www.flipkart.com/panasonic-eluga-i6-bl...
15	REDMI	9 Power	Mighty Black	6 GB	128 GB	48MP + 8MP + 2MP + 2MP	8MP Front Camera	16.59 cm (6.53 inch)	2340 x 1080$$pixel	-	2340 x 1080$$pixel	6000 mAh	₹15,884	https://www.flipkart.com/redmi-9-power-mighty-...
16	SAMSUNG	Galaxy M31	Ocean Blue	8 GB	128 GB	64MP + 8MP + 5MP + 5MP	32MP Front Camera	16.26 cm (6.4 inch)	2340 x 1080$$pixel	-	2340 x 1080$$pixel	6000 mAh	₹19,989	https://www.flipkart.com/samsung-galaxy-m31-oc...
17	Itel	it5026	Blue	4 MB	4 MB	-	-	6.1 cm (2.4 inch)	240 x 320$$pixel	-	-	1200 mAh	₹1,099	https://www.flipkart.com/itel-it5026/p/itm2d63...
18	REDMI	Note 10S	Cosmic Purple	6 GB	128 GB	64MP + 8MP + 2MP + 2MP	13MP Front Camera	16.33 cm (6.43 inch)	2400 x 1080$$pixel	-	2400 x 1080$$pixel	5000 mAh	₹16,699	https://www.flipkart.com/redmi-note-10s-cosmic...
19	Redmi	Redmi 9A	Midnight Black	2 GB	32 GB	13MP Rear Camera	-	16.59 cm (6.53 inch)	720 x 1600$$pixel	-	720 x 1600$$pixel	5000 mAh	₹7,899	https://www.flipkart.com/redmi-9a-midnight-bla...
20	OPPO	A55	Starry Black	4 GB	64 GB	Primary Camera	-	16.54 cm (6.51 inch)	1920 x 1080$$pixel	-	1920 x 1080$$pixel	-	₹15,480	https://www.flipkart.com/oppo-a55-starry-black...
21	OPPO	A55	Rainbow Blue	6 GB	128 GB	Primary Camera	-	16.54 cm (6.51 inch)	1920 x 1080$$pixel	-	1920 x 1080$$pixel	-	₹17,490	https://www.flipkart.com/oppo-a55-rainbow-blue...
22	OPPO	A55	Starry Black	6 GB	128 GB	Primary Camera	-	16.54 cm (6.51 inch)	1920 x 1080$$pixel	-	1920 x 1080$$pixel	-	₹17,490	https://www.flipkart.com/oppo-a55-starry-black...
23	OPPO	A55	Rainbow Blue	4 GB	64 GB	Primary Camera	-	16.54 cm (6.51 inch)	1920 x 1080$$pixel	-	1920 x 1080$$pixel	-	₹15,480	https://www.flipkart.com/oppo-a55-rainbow-blue...

In [13]:
```
# saving the data in csv
df.to_csv("smartphones.csv")
```

In [15]:
```
driver.close()
```

In [ ]:

In [ ]:

Q5 : Write a program to scrap geospatial coordinates (latitude, longitude) of a city searched on google maps.

In [16]:
```
# connecting to the webdriver
driver=webdriver.Chrome(r"C:/Users/HP/Downloads/chromedriver_win32 (1)/chromedriver.exe")
```

In [17]:
```
# getting mentioned url and opening google maps web page
url = 'https://www.google.co.in/maps'
driver.get(url)
time.sleep(2)
```

In [18]:
```
# entering the city name in search bar
City = input('Enter City name that has to be searched : ')
search_bar = driver.find_element_by_id('searchboxinput')
search_bar.click()
time.sleep(2)

#sending keys to find cities
search_bar.send_keys(City)

#checking for webelement and clicking on search button
search_btn = driver.find_element_by_id("searchbox-searchbutton")
search_btn.click()
time.sleep(2)

try:
    url_str = driver.current_url
    print("URL Extracted: ", url_str)
    latitude_longitude = re.findall(r'@(.*)data',url_str)
```

```
    if len(latitude_longitude):
        lat_lng_list = latitude_longitude[0].split(",")
        if len(lat_lng_list)>=2:
            latitude = lat_lng_list[0]
            longitude = lat_lng_list[1]
        print("Latitude = {}, Longitude = {}".format(latitude, longitude))
except Exception as e:
        print("Error: ", str(e))
```

```
Enter City name that has to be searched : Jaipur
URL Extracted:  https://www.google.co.in/maps/place/Jaipur,+Rajasthan/@23.9740114,78.422961,7z/data=!4m5!3m4!1s0x396c4adf4c57e281:0xce1c63a0cf22e09!8m2!3d26.9124336!4d75.7872709
Latitude = 23.9740114, Longitude = 78.422961
```

In [19]:
```
driver.close()
```

In [ ]:

In [ ]:

Q6 : Write a program to scrap details of all the funding deals for second quarter (i.e. July 20 – September 20) from trak.in.

In [20]:
```
# connecting to the webdriver
driver=webdriver.Chrome(r"C:/Users/HP/Downloads/chromedriver_win32 (1)/chromedriver.exe")
```

In [21]:
```
# opening the url track.in
url = "https://trak.in/"
driver.get(url)
time.sleep(2)
```

In [29]:
```
# getting xpath for funding deals and clicking on the button
fund_button = driver.find_element_by_xpath("//li[@id='menu-item-51510']/a").get_attribute('href')
driver.get(fund_button)

#Empty Lists
fund_deals = {}
fund_deals['Date'] = []
fund_deals['Startup Name'] = []
fund_deals['Industry/Vertical'] = []
fund_deals['Sub_Vertical'] = []
fund_deals['Location'] = []
fund_deals['Investor'] = []
fund_deals['Investment Type'] = []
fund_deals['Amount(in USD)'] = []


for i in range(48,51):

    # scraping data of data
    date = driver.find_elements_by_xpath("//table[@id='tablepress-{}']/tbody/tr/td[2]".format(i))
    for d in date:
        fund_deals['Date'].append(d.text)

    # scraping data of startup name
    startup_name = driver.find_elements_by_xpath("//table[@id='tablepress-{}']/tbody/tr/td[3]".format(i))
    for name in startup_name:
        fund_deals['Startup Name'].append(name.text)


    #scraping data of industry or vertical
    industry = driver.find_elements_by_xpath("//table[@id='tablepress-{}']/tbody/tr/td[4]".format(i))
    for ind in industry:
        fund_deals['Industry/Vertical'].append(ind.text)


    #scraping data of sub-vertical
    sub_vertical = driver.find_elements_by_xpath("//table[@id='tablepress-{}']/tbody/tr/td[5]".format(i))
    for sv in sub_vertical:
        fund_deals['Sub_Vertical'].append(sv.text)


    # scraping data of location
    location = driver.find_elements_by_xpath("//table[@id='tablepress-{}']/tbody/tr/td[6]".format(i))
    for loc in location:
        fund_deals['Location'].append(loc.text)


    # scraping data of investor
    investor = driver.find_elements_by_xpath("//table[@id='tablepress-{}']/tbody/tr/td[7]".format(i))
    for invest in investor:
        fund_deals['Investor'].append(invest.text)


    # scraping data of investment type
    investment_type = driver.find_elements_by_xpath("//table[@id='tablepress-{}']/tbody/tr/td[8]".format(i))
    for invtype in investment_type:
        fund_deals['Investment Type'].append(invtype.text)


    # scraping data of amount
    amount = driver.find_elements_by_xpath("//table[@id='tablepress-{}']/tbody/tr/td[9]".format(i))
    for amt in amount:
        fund_deals['Amount(in USD)'].append(amt.text)
```

In [30]:
```
# checking lengths of all scraped data
print(len(fund_deals['Date']),
len(fund_deals['Startup Name']),
len(fund_deals['Industry/Vertical']),
```

```
len(fund_deals['Sub_Vertical']),
len(fund_deals['Location']),
len(fund_deals['Investor']),
len(fund_deals['Investment Type']),
len(fund_deals['Amount(in USD)']
))
```

```
30 30 30 30 30 30 30 30
```

In [31]:
```
# creating DataFrame for scraped data
fund_data = pd.DataFrame(fund_deals)
fund_data
```

Out[31]:

	Date	Startup Name	Industry/Vertical	Sub_Vertical	Location	Investor	Investment Type	Amount(in USD)
0	15/07/2020	Flipkart	E-commerce	E-commerce	Bangalore	Walmart Inc	M&A	1,200,000,000
1	16/07/2020	Vedantu	EduTech	Online Tutoring	Bangalore	Coatue Management	Series D	100,000,000
2	16/07/2020	Crio	EduTech	Learning Platform for Developers	Bangalore	021 Capital	pre-Series A	934,160
3	14/07/2020	goDutch	FinTech	Group Payments	Mumbai	Matrix India,Y Combinator, Global Founders Cap...	Seed	1,700,000
4	13/07/2020	Mystifly	Airfare Marketplace	Ticketing, Airline Retailing, and Post-Ticketi...	Singapore and Bangalore	Recruit Co. Ltd.	pre-Series B	3,300,000
5	09/07/2020	JetSynthesys	Gaming and Entertainment	Gaming and Entertainment	Pune	Adar Poonawalla and Kris Gopalakrishnan.	Venture-Series Unknown	400,000
6	10/07/2020	gigIndia	Marketplace	Crowd Sourcing, Freelance	Pune	Incubate Fund India and Beyond Next Ventures	pre-Series A	974,200
7	15/07/2020	PumPumPum	Automotive Rental	Used Car-leasing platform	Gurgaon	Early Adapters Syndicate	Seed	292,800
8	14/07/2020	FLYX	OTT Player	Streaming Social Network	New York and Delhi	Raj Mishra, founder of AIT Global Inc	pre-Seed	200,000
9	13/07/2020	Open Appliances Pvt. Ltd.	Information Technology	Internet-of-Things Security Solutions	Bangalore	Unicorn India Ventures	Venture-Series Unknown	500,000
10	15/08/2020	Practo	HealthTech	Health care and Wellness	Bangalore	A1A Company	Series F	32,000,000
11	13/08/2020	Medlife	E-commerce	Online Pharmacy	Bangalore	Prasid Uno Family Trust and SC Credit Fund		23,000,000
12	13/08/2020	HungerBox	FoodTech	Online Food Delivery Service	Bangalore	One97, Sabre Partners Trust, Pratithi Investme...	Series D1	1,560,000
13	04/08/2020	Dunzo	Hyper-local Logistics	Online Delivery Services	Bangalore	Existing Backers	In Progress	30,000,000
14	11/08/2020	Terra.do	EduTech	Online Climate School, E-learning	Stanford, California,	Stanford Angels and Entrepreneurs (India), BEE...	Seed	1,400,000
15	12/08/2020	Classplus	EduTech	E-learning, Online Tutoring	Noida	Falcon Edge	In Progress	upto 15,000,000
16	14/08/2020	Niyo	FinTech	Financial Services	Bangalore	Niyo Solutions Inc.		6,000,000
17	10/08/2020	ZestMoney	FinTech	Financial Services	Bangalore	Primrose Hills Ventures		10,670,000
18	07/08/2020	FreshToHome	E-commerce	Food Delivery	Bangalore	Ascent Capital	Venture	16,200,000
19	13/08/2020	Eduvanz	FinTech	Financial Services	Mumbai	Sequoia India, Unitus	Series A	5,000,000
20	08/09/2020	Byju's	EduTech	Online Tutoring	Bangalore	Silver Lake, Tiger Global, General Atlantic an...	Private Equity	500,000,000
21	12/09/2020	mCaffeine	Personal Care	Skincare & Haircare	Mumbai	Amicus Capital Private Equity I LLP, Amicus Ca...	Series B	3,000,000
22	09/09/2020	Qshala	EduTech	Online Curiosity Platform for Kids	Bangalore	Rainmatter Capital	Angel	370,000
23	02/09/2020	Winzo	Online Gaming	Online Gaming	New Delhi	Kalaari Capital Partners, IndigoEdge Managemen...	Series B	15,500,000
24	09/09/2020	Hippo Video	Video Customer Experience(CX) Platform	Video Customer Experience(CX) Platform	Newark, Delaware, United States of Amercia	Alpha Wave Incubation, Exfinity Venture Partne...	Series A	4,500,000
25	07/09/2020	Melorra	E-commerce	Online Jewelry Store	Bangalore	Shadow Holdings, Lightbox.	Debt Financing	upto 8,900,000
26	07/09/2020	1mg	E-commerce	Online Pharmacy	Gurgaon	Gaja Capital, Tata Capital, Partners Group	In Progress	100,000,000
27	31/08/2020	mfine	HealthTech	On-Demand Healthcare Services	Bangalore	Caretech Pte Inc	Series B	5,400,000
28	31/08/2020	Apna	Human Resources	Recruitment Platform	Bangalore	Lightspeed India and Sequoia Capital India	Series A	8,000,000
29	03/09/2020	Railofy	Transportation	WL & RAC protection platform	Mumbai	Chiratae Ventures	Seed	950,000

In [32]:
```
# saving data in csv file
fund_data.to_csv("trak_in.csv")
```

In [33]:
```
driver.close()
```

In [ ]:

In [ ]:

Q7 : Write a program to scrap all the available details of best gaming laptops from digit.in.

In [2]:
```
# connecting to the webdriver
driver=webdriver.Chrome(r"C:/Users/HP/Downloads/chromedriver_win32 (1)/chromedriver.exe")
```

In [3]:
```
# opening the url digit.in
```

```
url = "https://www.digit.in/"
driver.get(url)
time.sleep(2)
```

In [4]:
```
# searching for best Laptop
best_gam_laptops = driver.find_element_by_xpath("//div[@class='listing_container']//ul//li[9]").click()
time.sleep(3)
```

In [5]:
```
# creating empty list
Laptop_Name = []
Operating_sys = []
Display = []
Processor = []
Memory = []
Weight = []
Dimensions = []
Graph_proc = []
Price = []
```

In [6]:
```
#scraping the data of laptop names
laptop_name = driver.find_elements_by_xpath("//div[@class='right-container']/div/a/h3")
for name in laptop_name:
    Laptop_Name.append(name.text)

#scraping the data of operating system
try:
    op_sys = driver.find_elements_by_xpath("//div[@class='product-detail']/div/ul/li[1]/div/div")
    for os in op_sys:
        Operating_sys.append(os.text)
except NoSuchElementException:
    pass


#scraping data of display of the Laptop
try:
    display = driver.find_elements_by_xpath("//div[@class='product-detail']/div/ul/li[2]/div/div")
    for disp in display:
        Display.append(disp.text)
except NoSuchElementException:
    pass


# scraping data of processor
try:
    processor = driver.find_elements_by_xpath("//div[@class='Spcs-details'][1]/table/tbody/tr[5]/td[3]")
    for pro in processor:
        Processor.append(pro.text)
except NoSuchElementException:
    pass


# scraping the data of memory
try:
    memory = driver.find_elements_by_xpath("//div[@class='Spcs-details'][1]/table/tbody/tr[6]/td[3]")
    for memo in memory:
        Memory.append(memo.text)
except NoSuchElementException:
    pass


# scraping data of weight
try:
    weight = driver.find_elements_by_xpath("//div[@class='Spcs-details'][1]/table/tbody/tr[7]/td[3]")
    for wgt in weight:
        Weight.append(wgt.text)
except NoSuchElementException:
    pass


# scraping data of dimensions
try:
    dimension = driver.find_elements_by_xpath("//div[@class='Spcs-details'][1]/table/tbody/tr[8]/td[3]")
    for dim in dimension:
        Dimensions.append(dim.text)
except NoSuchElementException:
    pass


# scraping data of graph processor
try:
    graph = driver.find_elements_by_xpath("//div[@class='Spcs-details'][1]/table/tbody/tr[9]/td[3]")
    for gra in graph:
        Graph_proc.append(gra.text)
except NoSuchElementException:
    pass


# scraping the data of price
try:
    price = driver.find_elements_by_xpath("//td[@class='smprice']")
    for pri in price:
        Price.append(pri.text.replace('₹ ','Rs'))
except NoSuchElementException:
    pass
```

In [7]:
```
print(len(Laptop_Name),
len(Operating_sys),
len(Display),
len(Processor),
len(Memory),
len(Weight),
len(Dimensions),
len(Graph_proc),
len(Price))
```

```
10 10 10 10 10 10 10 10 10
```

In [8]:
```
#creating DataFrame for scraped data
Gaming_Laptop=pd.DataFrame({})
Gaming_Laptop['Laptop Name'] = Laptop_Name
Gaming_Laptop['Operating System'] =Operating_sys
Gaming_Laptop['Display'] = Display
Gaming_Laptop['Processor'] = Processor
Gaming_Laptop['Memory'] = Memory
Gaming_Laptop['Weight'] = Weight
Gaming_Laptop['Dimensions'] = Dimensions
Gaming_Laptop['Graphical Processor'] = Graph_proc
Gaming_Laptop['Price'] = Price
Gaming_Laptop
```

Out[8]:

	Laptop Name	Operating System	Display	Processor	Memory	Weight	Dimensions	Graphical Processor	Price
0	ACER NITRO 5 RYZEN 9 (2021)	WINDOWS 10	15.6" (1920 X 1080)	AMD Ryzen 9 Octa Core | 2.4 GHz	1 TB HDD/16 GBGB DDR4	2.4	363.4 x 255 x 23.9	NVIDIA GeForce RTX 3070	Rs129,990
1	MSI STEALTH 15M 11TH GEN CORE I7-11375H (2021	WINDOWS 10	15.6" (1920 X 1080)	Intel Core i7 11th Gen - 11375H | NA	1 TB SSD/16 GBGB DDR4	1.7	358.3 x 248 x 16.15	NVIDIA GeForce RTX 3060	Rs134,990
2	ASUS ROG STRIX SCAR 15 RYZEN 9-5900HX (2021)	WINDOWS 10	15.6" (2560 X 1440)	AMD Ryzen 9 Octa Core - 5900HX | 3.3 GHz	2 TB SSD/32 GBGB DDR4	2.30	354 x 259 x 22.6	NVIDIA GeForce RTX 3080	Rs268,990
3	ALIENWARE AREA 51M R2	WINDOWS 10 HOME	17.3" (1920 X 1080)	10th Gen Intel® Core™ i7-10700 | 2.90 GHz	1 TB SSD/16 GBGB DDR4	4.1	27.65 x 402.6 x 319.14	Intel® UHD Graphics 630	Rs342,989
4	ALIENWARE M15 R3	WINDOWS 10 HOME	15.6" (3840 X 2160)	10th Gen Intel® Core™ i9-10980HK | NA	1 TB SSD/16 GBGB DDR4	NA	NA	NA	Rs319,990
5	ASUS ROG STRIX SCAR 15	WINDOWS 10 HOME	15.6" (1920 X 1080)	AMD Ryzen™ 9 5900HX | 3.3 GHz	1 TB SSD/16 GBGB DDR4	2.30	35.4 x 25.9 x 2.26	NVIDIA® GeForce RTX™ 3070	Rs215,990
6	ASUS ROG ZEPHYRUS G14	WINDOWS 10 HOME	14" (1920 X 1080)	AMD 3rd Gen Ryzen 9 | 3.3 GHz	1 TB SSD/16 GBGB DDR4	1.65	32.5 x 22.1 x 1.8	NVIDIA GeForce RTX 2060	Rs164,990
7	LENOVO LEGION 5I	WINDOWS 10 PRO	15.6" (1920 X 1080)	10th Gen Intel® Core™ i5-10300H | 2.50 GHz	1 TB SSD/16 GBGB DDR4	2.3	363.06 x 259.61 x 23.57	NVIDIA® GeForce® GTX 1650 4GB	Rs76,988
8	ASUS ROG ZEPHYRUS DUO 15	WINDOWS 10	15.6" (3840 X 1100)	Intel Core i7 10th Gen 10875H | NA	512 GB SSD/4 GBGB DDR4	2.4	268.30 x 360.00 x 20.90	NVIDIA GeForce RTX 2070 Max-Q	Rs185,000
9	ACER ASPIRE 7 GAMING	WINDOWS 10 HOME	15.6" (1920 X 1080)	AMD Ryzen™ 5-5500U hexa-core | NA	512 GB SSD/8 GBGB DDR4	2.15	2.29 x 36.3 x 25.4	NVIDIA® GeForce® GTX 1650	Rs62,968

In [9]:
```
# saving the data to csv
Gaming_Laptop.to_csv("Gaming_Laptops.csv")
```

In [10]:
```
driver.close()
```

In [ ]:

In [ ]:

Q8 : 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 [2]:
```
# connecting to the webdriver
driver=webdriver.Chrome(r"C:/Users/HP/Downloads/chromedriver_win32 (1)/chromedriver.exe")
```

In [3]:
```
# getting the specified url
url = "https://www.forbes.com/?sh=41bd46d2254c"
driver.get(url)
```

In [4]:
```
#let's get option button from the page
opt_btn = driver.find_element_by_xpath("//div[@class='header__left']//button")
opt_btn.click()
time.sleep(3)

#select billionaires from options
blns = driver.find_element_by_xpath("/html/body/div[1]/header/nav/div[3]/ul/li[1]")
blns.click()
time.sleep(3)
#select world billionaire
bln_list = driver.find_element_by_xpath("/html/body/div[1]/header/nav/div[3]/ul/li[1]/div[2]/ul/li[2]/a")
bln_list.click()
time.sleep(4)
```

In [5]:
```
# scraping required data from the web page
# creating empty lists
Rank = []
Person_Name = []
Net_worth = []
Age = []
Citizenship = []
Source = []
Industry = []


while(True):

    # scraping the data of rank of the billionaires
    rank_tag = driver.find_elements_by_xpath("//div[@class='rank']")
    for rank in rank_tag:
        Rank.append(rank.text)
    time.sleep(1)
```

```
    # scraping the data  of names of the billionaires
    name_tag = driver.find_elements_by_xpath("//div[@class='personName']/div")
    for name in name_tag:
        Person_Name.append(name.text)
    time.sleep(1)


    # scraping the data of age of the billionaires
    age_tag = driver.find_elements_by_xpath("//div[@class='age']/div")
    for age in age_tag:
        Age.append(age.text)
    time.sleep(1)


    # scraping the data of citizenship of the billionaires
    cit_tag = driver.find_elements_by_xpath("//div[@class='countryOfCitizenship']")
    for cit in cit_tag:
        Citizenship.append(cit.text)
    time.sleep(1)


    # scraping the data of source of income of the billionaires
    sour_tag = driver.find_elements_by_xpath("//div[@class='source']")
    for sour in sour_tag:
        Source.append(sour.text)
    time.sleep(1)


    # scraping data of industry of the billionaires
    ind_tag = driver.find_elements_by_xpath("//div[@class='category']//div")
    for ind in ind_tag:
        Industry.append(ind.text)
    time.sleep(1)


    # scraping data of net_worth of billionaires
    net_tag = driver.find_elements_by_xpath("//div[@class='netWorth']/div")
    for net in net_tag:
        Net_worth.append(net.text)
    time.sleep(1)


    # clicking on next button
    try:
        next_button = driver.find_element_by_xpath("//button[@class='pagination-btn pagination-btn--next ']")
        next_button.click()
    except:
        break
```

In [6]:

```
print(len(Rank),
len(Person_Name),
len(Net_worth),
len(Age),
len(Citizenship),
len(Source),
len(Industry))
```

```
2755 2755 2755 2755 2755 2755 2755
```

In [7]:

```
# framing Data
Billionaires = pd.DataFrame({})
Billionaires['Rank'] = Rank
Billionaires['Name'] = Person_Name
Billionaires['Net Worth'] = Net_worth
Billionaires['Age'] = Age
Billionaires['Citizenship'] = Citizenship
Billionaires['Source'] = Source
Billionaires['Industry'] = Industry
Billionaires
```

Out[7]:

	Rank	Name	Net Worth	Age	Citizenship	Source	Industry
0	1.	Jeff Bezos	$177 B	57	United States	Amazon	Technology
1	2.	Elon Musk	$151 B	49	United States	Tesla, SpaceX	Automotive
2	3.	Bernard Arnault & family	$150 B	72	France	LVMH	Fashion & Retail
3	4.	Bill Gates	$124 B	65	United States	Microsoft	Technology
4	5.	Mark Zuckerberg	$97 B	36	United States	Facebook	Technology
...	...	...	...	...	...	...	...
2750	2674.	Daniel Yong Zhang	$1 B	49	China	e-commerce	Technology
2751	2674.	Zhang Yuqiang	$1 B	65	China	Fiberglass	Manufacturing
2752	2674.	Zhao Meiguang	$1 B	58	China	gold mining	Metals & Mining
2753	2674.	Zhong Naixiong	$1 B	58	China	conglomerate	Diversified
2754	2674.	Zhou Wei family	$1 B	54	China	Software	Technology

2755 rows × 7 columns

In [9]:

```
# saving dataset in csv
Billionaires.to_csv('Forbes_Billionaires.csv')
```

In [10]:

```
driver.close()
```

In [ ]:

In [ ]:

Q9 : Write a program to extract at least 500 Comments, Comment upvote and time when comment was posted from any YouTube Video.

In [11]:
```
# connecting to the webdriver
driver=webdriver.Chrome(r"C:/Users/HP/Downloads/chromedriver_win32 (1)/chromedriver.exe")
```

In [12]:
```
# opening the youtube.com
url = "https://www.youtube.com/"
driver.get(url)
time.sleep(2)
```

In [16]:
```
# finding element for search bar
search_bar = driver.find_element_by_xpath("//div[@class='ytd-searchbox-spt']/input")
search_bar.send_keys("GOT")      # entering video name
time.sleep(2)
```

In [17]:
```
#clicking on search button
search_btn = driver.find_element_by_id("search-icon-legacy")
search_btn.click()
time.sleep(2)
```

In [18]:
```
# clicking on first video
video = driver.find_element_by_xpath("//yt-formatted-string[@class='style-scope ytd-video-renderer']")
video.click()
```

In [19]:
```
# 1000 times we scroll down by 10000 in order to generate more comments
for _ in range(1000):
    driver.execute_script("window.scrollBy(0,10000)")
```

In [22]:
```
# creating empty lists
comments = []
comment_time = []
Time = []
Likes = []
No_of_Likes = []

# scrape comments
cm = driver.find_elements_by_id("content-text")
for i in cm:
    if i.text is None:
        comments.append("--")
    else:
        comments.append(i.text)
time.sleep(4)


# scrape time when comment was posted
tm = driver.find_elements_by_xpath("//a[contains(text(),'ago')]")
for i in tm:
    Time.append(i.text)

for i in range(0,len(Time),2):
    comment_time.append(Time[i])
time.sleep(4)


# scrape the comment likes
like = driver.find_elements_by_xpath("//span[@class='style-scope ytd-comment-action-buttons-renderer']")
for i in like:
    Likes.append(i.text)

for i in range(1,len(Likes),2):
    No_of_Likes.append(Likes[i])
```

In [23]:
```
print(len(comments),len(comment_time),len(No_of_Likes))
```

```
1459 1459 1459
```

In [24]:
```
# creating dataframe for scraped data

Youtube = pd.DataFrame({})
Youtube['Comment'] = comments[:500]
Youtube['Comment Time'] = comment_time[:500]
Youtube['Comment Upvotes'] = No_of_Likes[:500]
Youtube
```

Out[24]:

	Comment	Comment Time	Comment Upvotes
0	Episode 2: I promised to fight for the living....	2 years ago (edited)	3.1K
1	After watching S8 i just wish that hodor shoul...	9 months ago	1.9K
2	After 10,000 years the Night King made it past...	1 year ago	1.1K
3	Jon Snow : Knows Nothing, But Did Everything. ...	4 months ago	777
4	I'm glad I only recently watched Game of Thron...	2 months ago	187
...	...	...	...
495	You know it was serious when I raised my phone...	2 years ago	602
496	Khalese: terrible leader, if even a leader at ...	7 months ago	
497	Season 8 trailer.....the world's greatest Cat...	1 year ago	1
498	The long night it really was just one episode...	1 week ago	
499	I have done so many things in life but .....\n...	10 months ago	

500 rows × 3 columns

300 rows × 3 columns

In [25]:
```
#saving the dataframe to csv
Youtube.to_csv("Youtube GOT Comments.csv")
```

In [26]:
```
driver.close()
```

In [ ]:

In [ ]:

Q10 : Write a python program to scrape a data for all available Hostels from https://www.hostelworld.com/ in “London” location. You have to scrape hostel name, distance from city centre, ratings, total reviews, overall reviews, privates from price, dorms from price, facilities and property description.

In [8]:
```
# connecting to the webdriver
driver=webdriver.Chrome(r"C:/Users/HP/Downloads/chromedriver_win32 (1)/chromedriver.exe")
```

In [9]:
```
# getting the web page of mentioned url
url = "https://www.hostelworld.com/"
driver.get(url)
time.sleep(3)
```

In [10]:
```
# locating the location search bar
search_bar = driver.find_element_by_id("search-input-field")

# entering London in search bar
search_bar.send_keys("London")
```

In [11]:
```
# select London
London = driver.find_element_by_xpath("//ul[@id='predicted-search-results']//li[2]")
#clicking on button
London.click()

# do click on Let's Go button
search_btn = driver.find_element_by_id('search-button')
search_btn.click()
```

In [12]:
```
# creating empty list & find required data
hostel_name = []
distance = []
pvt_prices = []
dorms_price = []
rating = []
reviews = []
over_all = []
facilities = []
description = []
url = []
```

In [13]:
```
# scraping the required informations
for i in driver.find_elements_by_xpath("//div[@class='pagination-item pagination-current' or @class='pagination-item']"):
    i.click()
    time.sleep(3)


    # scraping  hostel name
    try:
        name = driver.find_elements_by_xpath("//h2[@class='title title-6']")
        for i in name:
            hostel_name.append(i.text)
    except NoSuchElementException:
        hostel_name.append('-')


    # scraping distance from city centre
    try:
        dist = driver.find_elements_by_xpath("//div[@class='subtitle body-3']//a//span[1]")
        for i in name:
            distance.append(i.text.replace('Hostel - ',''))
    except NoSuchElementException:
        distance.append('-')


    for i in driver.find_elements_by_xpath("//div[@class='prices-col']"):
    # scraping privates from price
        try:
            pvt_price = driver.find_element_by_xpath("//a[@class='prices']//div[1]//div")
            pvt_prices.append(pvt_price.text)
        except NoSuchElementException:
            pvt_prices.append('-')


    for i in driver.find_elements_by_xpath("//div[@class='prices-col']"):
    # scraping dorms from price
        try:
            dorms = driver.find_element_by_xpath("//a[@class='prices']//div[2]/div")
            dorms_price.append(dorms.text)
        except NoSuchElementException:
            dorms_price.append('-')


    # scraping facilities
```

```
    try:
        fac1 = driver.find_elements_by_xpath("//div[@class='has-wifi']")
        fac2 = driver.find_elements_by_xpath("//div[@class='has-sanitation']")
        for i in fac1:
            for j in fac2:
                facilities.append(i.text +', '+ j.text)
    except NoSuchElementException:
        facilities.append('-')


    #fetching url of each hostel
    p_url = driver.find_elements_by_xpath("//div[@class='prices-col']//a[2]")
    for i in p_url:
        url.append(i.get_attribute("href"))

for i in url:
    driver.get(i)
    time.sleep(3)


    # scraping ratings
    try:
        rat = driver.find_element_by_xpath("//div[@class='score orange big' or @class='score gray big']")
        rating.append(rat.text)
    except NoSuchElementException:
        rating.append('-')


    # scraping total review
    try:
        rws = driver.find_element_by_xpath("//div[@class='reviews']")
        reviews.append(rws.text.replace('Total Reviews',''))
    except NoSuchElementException:
        reviews.append('-')


    # fetching over all review
    try:
        overall = driver.find_element_by_xpath("//div[@class='keyword']//span")
        over_all.append(overall.text)
    except NoSuchElementException:
        over_all.append('-')


    # fetching property description
    try:
        disc = driver.find_element_by_xpath("//div[@class='content']")
        description.append(disc.text)
    except NoSuchElementException:
        over_all.append('-')

    # do click on show more button for description
    try:
        driver.find_element_by_xpath("//a[@class='toggle-content']").click()
        time.sleep(4)
    except NoSuchElementException:
        pass
```

In [14]:

```
print(len(hostel_name),
len(distance),
len(pvt_prices),
len(dorms_price),
len(rating),
len(reviews),
len(over_all),
len(facilities),
len(description),
len(url))
```

```
74 74 74 74 74 74 74 1071 74 74
```

In [15]:

```
# creating DataFrame
Hostel = pd.DataFrame({})
Hostel['Hostel Name'] = hostel_name
Hostel['Distance from City Centre'] = distance
Hostel['Ratings'] = rating
Hostel['Total Reviews'] = reviews
Hostel['Overall Reviews'] = over_all
Hostel['Privates from Price'] = pvt_prices
Hostel['Dorms from Price'] = dorms_price
Hostel['Facilities'] = facilities[:74]
Hostel['Description'] = description
Hostel
```

Out[15]:

	Hostel Name	Distance from City Centre	Ratings	Total Reviews	Overall Reviews	Privates from Price	Dorms from Price	Facilities	Description
0	St Christopher's Inn - Liverpool Street	St Christopher's Inn - Liverpool Street	9.3	362	Superb	Rs6862	Rs2128	Free WiFi, Follows Covid-19 sanitation guidance	52 Wilson Street, Finsbury, London, England
1	Astor Hyde Park	Astor Hyde Park	8.9	11355	Fabulous	Rs6862	Rs2128	Free WiFi, Follows Covid-19 sanitation guidance	191 Queensgate, South Kensington, London, England
2	St Christopher's Village	St Christopher's Village	8.3	10886	Fabulous	Rs6862	Rs2128	Free WiFi, Follows Covid-19 sanitation guidance	165 Borough High Street, London, England
3	Smart Camden Inn Hostel	Smart Camden Inn Hostel	9.0	2699	Superb	Rs6862	Rs2128	Free WiFi, Follows Covid-19 sanitation guidance	55/57 Bayham Street, Camden, London, England
4	The Finnish Church in London	The Finnish Church in London	9.5	194	Superb	Rs6862	Rs2128	Free WiFi, Follows Covid-19 sanitation guidance	33 Albion Street, Rotherhithe, London, England
...	...	...	...	...	...	...	...	...	...
69	Barry House	Barry House	9.0	4	Superb	Rs11379	-	Free WiFi, Follows Covid-19 sanitation guidance	12 Sussex Place, Paddington, London, England
70	Park Hotel Essex	Park Hotel Essex	-	0	No Rating	Rs11379	-	Free WiFi, Follows Covid-19 sanitation guidance	327 Cranbrook Road, Ilford, London, England
71	Cranbrook Hotel	Cranbrook Hotel	-	0	No Rating	Rs11379	-	Free WiFi, Follows Covid-19	22/24 Coventry Road, Ilford,

								sanitation guidance	London, England
72	Aron Guest House	Aron Guest House	-	0	No Rating	Rs11379	-	Free WiFi, Follows Covid-19 sanitation guidance	27 South Ealing, London, England
73	MacDonald Hotel	MacDonald Hotel	9.2	248	Superb	Rs11379	-	Free WiFi, Follows Covid-19 sanitation guidance	45 - 46 Argyle Square, Kings Cross, London, En...

74 rows × 9 columns

In [16]:
```
# saving the dataset to csv
Hostel.to_csv("London_Hostels.csv")
```

In [17]:
```
driver.close()
```

In [ ]:

In [ ]:

In [ ]: