In [1]:
```
#question:. Write a python program which searches all the product under a particula
#product to be searched will be taken as input from user. For e.g. If user input is
#guitars
```

In [2]:
```
import pandas as pd
import selenium
from bs4 import BeautifulSoup
import time
from selenium import webdriver
import requests
import re
import warnings
warnings.filterwarnings('ignore')
from selenium.common.exceptions import NoSuchElementException, StaleElementReferen
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
```

In [3]:
```
driver=webdriver.Chrome()
```

In [4]:
```
url="https://www.amazon.in/"
driver.get(url)
```

In [5]:
```
user_input = input('Enter the product that we want to search : ')
```

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

In [6]:
```
search = driver.find_element(By.ID,"twotabsearchtextbox")
search
```

Out[6]:
```
<selenium.webdriver.remote.webelement.WebElement (session="7af8bd0025c5c70c8fddeaa
27b5492a7", element="DA87715CD35306780EA235598B573DCB_element_23")>
```

In [7]:
```
search.send_keys(user_input)
```

In [8]:
```
search_btn = driver.find_element(By.XPATH,"//div[@class='nav-search-submit nav-spri
```

In [9]:
```
search_btn.click()
```

In [10]:
```
# question 2:In the above question, now scrape the following details of each produ
#results and save it in a data frame and csv. In case if any product has less than
#scrape all the products available under that product name. Details to be scraped
#Name", "Name of the Product", "Price", "Return/Exchange", "Expected Delivery", "A
#“Product URL”. In case, if any of the details are missing for any of the product
```

In [11]:
```
urls = []          # empty list
for i in range(0,60):      # for loop to scrape 3 pages
    page_url = driver.find_elements(By.XPATH,"//a[@class='a-link-normal a-text-norm
    for i in page_url:
        urls.append(i.get_attribute("href"))
        next_btn = driver.find_elements(By.XPATH,"//li[@class='a-last']/a")
```

In [12]:
```
len(urls)
```

Out[12]:
```
0
```

In [13]:
```
# 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
        ratings.append(rating.text)
    except NoSuchElementException:
        ratings.append('-')


    #fetching  no of ratings
    try:
        num_rating = driver.find_element(By.XPATH,"//span[@id='acrCustomerReviewTe
        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
        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
        availability.append(avail.text)
    except NoSuchElementException:
        availability.append('-')

    #other details
    try:
        oth_det = driver.find_element(By.XPATH,"//ul[@class='a-unordered-list a-ver
        other_details.append(oth_det.text)
    except NoSuchElementException:
        other_details.append('-')
```

In [14]:
```
print(len(brand_name),
len(product_name),
len(ratings),
len(num_ratings),
len(prices),
len(exchange),
len(exp_delivery),
len(availability),
len(other_details))
```

```
0 0 0 0 0 0 0 0 0
```

In [15]:
```
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[15]:

Brand Name	Name of the Product	Rating	No. of Ratings	Price	Return/Exchange	Expected Delivery	Availability	Other Details	Produ URL

In [16]:
```
#question3:Write a python program to access the search bar and search button on ima
#images each for keywords ‘fruits’, ‘cars’ and ‘Machine Learning’, ‘Guitar’, ‘Cakes
```

In [17]:
```
import pandas as pd
import selenium
from bs4 import BeautifulSoup
import time
from selenium import webdriver
import requests
import re
import warnings
warnings.filterwarnings('ignore')
from selenium.common.exceptions import NoSuchElementException, StaleElementReferen
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
```

In [18]:
```
driver=webdriver.Chrome()
```

In [19]:
```
url="https://images.google.com/"
driver.get(url)
```

In [20]:
```
urls = []
data = []
```

In [21]:
```
search_item = ["Fruits","Cars","Machine Learning"]
for item in search_item:
    driver.get(url)
    time.sleep(5)
```

In [22]:
```
 # 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']").clicl

    # 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)
```

```
  Input In [22]
    search_bar = driver.find_element(By.TAG_NAME,"input")
    ^
IndentationError: unexpected indent
```

In [23]:
```
driver.close()
```

In [24]:
```
#Write a python program to search for a smartphone(e.g.: Oneplus Nord, pixel 4A, et
#and scrape following details for all the search results displayed on 1st page. Det
#Name", "Smartphone name", "Colour", "RAM", "Storage(ROM)", "Primary Camera",
#"Secondary Camera", "Display Size", "Battery Capacity", "Price", "Product URL". In
#details is missing then replace it by "- ". Save your results in a dataframe and C
#
```

In [25]:
```
import pandas as pd
import selenium
from bs4 import BeautifulSoup
import time
from selenium import webdriver
import requests
import re
import warnings
warnings.filterwarnings('ignore')
from selenium.common.exceptions import NoSuchElementException, StaleElementReferen
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
```

In [26]:
```
driver=webdriver.Chrome()
```

In [27]:
```
url="https://www.flipkart.com/"
driver.get(url)
```

In [28]:
```
lonin_x_btn = driver.find_element(By.XPATH,"//div[@class='_2QfC02']//button").clicl
```

In [29]:
```
search_bar = driver.find_element(By.XPATH,"//input[@class='_3704LK']")



search_bar.send_keys("pixel 4A")
```

In [30]:
```
search_btn = driver.find_element(By.XPATH,"//button[@class='L0Z3Pu']")


search_btn.click()
```

In [31]:
```
page1_url = []
urls = driver.find_elements(By.XPATH,"//a[@class='_1fQZEK']")
for url in urls:
    page1_url.append(url.get_attribute('href'))
```

In [32]:
```
len(page1_url)
```

Out[32]:
```
24
```

In [33]:
```
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 [34]:
```
# scraping data from each url of page 1
for url in page1_url:
    driver.get(url)
```

```
    print("Scraping URL = ",url)
    Smartphones['URL'].append(url)

    #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/
        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
        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
        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]/tb
        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]/tbo
        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]/tb
        if sec != 'Secondary Camera' :
            if driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][5]/table[1]/t
                sec_cam =driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][5]/
            else :
                raise NoSuchElementException
        else :
            sec_cam = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][5]/tab
        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
        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
        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]/tb
        if pro.text != 'Processor Type' : raise NoSuchElementException
        processor = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][3]/table
        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]/t
        if core.text != 'Processor Core' :
            core = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][3]/table[1
            if core.text != 'Processor Core' :
                raise NoSuchElementException
            else :
                cores = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][3]/ta
        else :
            cores = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][3]/table
        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 !=
            if driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][9]/div").text
                bat_tags = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][9
                if bat_tags.text != "Battery Capacity" : raise NoSuchElementExcept
                bat_capa = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][9
            elif driver.find_element_by_xpath("//div[@class='_3k-BhJ'][8]/div").te
                bat_tags = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][8
                if bat_tags.text != "Battery Capacity" : raise NoSuchElementExcept
                bat_capa = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][8
            else:
```

```
                raise NoSuchElementException
        else :
            bat_tags = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][10]/ta
            if bat_tags.text != "Battery Capacity" : raise NoSuchElementException
            bat_capa = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][10]/ta
        Smartphones['Battery Capacity'].append(bat_capa.text)
    except NoSuchElementException:
        Smartphones['Battery Capacity'].append('-')
```

```
Scraping URL =  https://www.flipkart.com/google-pixel-6a-charcoal-128-gb/p/itme5ae
89135d44e?pid=MOBGFKX5YUXD74Z3&lid=LSTMOBGFKX5YUXD74Z3MXA2OB&marketplace=FLIPKART&
q=pixel+4A&store=tyy%2F4io&srno=s_1_1&otracker=search&otracker1=search&fm=organic&
iid=ebd0d591-1039-41f1-994c-bfe764319d72.MOBGFKX5YUXD74Z3.SEARCH&ppt=hp&ppn=homepa
ge&ssid=dkp6e0dm3k0000001686905739071&qH=9b26a23b2cff510d
```

```
---------------------------------------------------------------------------
InvalidSelectorException                  Traceback (most recent call last)
Input In [34], in <cell line: 2>()
     92 #scraping processor of smartphone
     93 try:
---> 94     pro = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][3]/table
[1]/tbody/tr[2]/td[1]]")
     95     if pro.text != 'Processor Type' : raise NoSuchElementException
     96     processor = driver.find_element(By.XPATH,"//div[@class='_3k-BhJ'][3]/t
able[1]/tbody/tr[2]/td[2]/ul/li")

File ~\AppData\Roaming\Python\Python39\site-packages\selenium\webdriver\remote\web
driver.py:831, in WebDriver.find_element(self, by, value)
    828     by = By.CSS_SELECTOR
    829     value = f'[name="{value}"]'
--> 831 return self.execute(Command.FIND_ELEMENT, {"using": by, "value": value})
["value"]

File ~\AppData\Roaming\Python\Python39\site-packages\selenium\webdriver\remote\web
driver.py:440, in WebDriver.execute(self, driver_command, params)
    438 response = self.command_executor.execute(driver_command, params)
    439 if response:
--> 440     self.error_handler.check_response(response)
    441     response["value"] = self._unwrap_value(response.get("value", None))
    442     return response

File ~\AppData\Roaming\Python\Python39\site-packages\selenium\webdriver\remote\err
orhandler.py:245, in ErrorHandler.check_response(self, response)
    243         alert_text = value["alert"].get("text")
    244     raise exception_class(message, screen, stacktrace, alert_text)  # typ
e: ignore[call-arg]  # mypy is not smart enough here
--> 245 raise exception_class(message, screen, stacktrace)

InvalidSelectorException: Message: invalid selector: Unable to locate an element w
ith the xpath expression //div[@class='_3k-BhJ'][3]/table[1]/tbody/tr[2]/td[1]] be
cause of the following error:
SyntaxError: Failed to execute 'evaluate' on 'Document': The string '//div[@class
='_3k-BhJ'][3]/table[1]/tbody/tr[2]/td[1]]' is not a valid XPath expression.
  (Session info: chrome=114.0.5735.91)
Stacktrace:
Backtrace:
        GetHandleVerifier [0x00858893+48451]
        (No symbol) [0x007EB8A1]
        (No symbol) [0x006F5058]
        (No symbol) [0x006F83F1]
        (No symbol) [0x006F9631]
        (No symbol) [0x006F96D0]
        (No symbol) [0x007200C0]
        (No symbol) [0x0072069B]
        (No symbol) [0x0074DD92]
        (No symbol) [0x0073A304]
        (No symbol) [0x0074C482]
        (No symbol) [0x0073A0B6]
        (No symbol) [0x00717E08]
        (No symbol) [0x00718F2D]
        GetHandleVerifier [0x00AB8E3A+2540266]
        GetHandleVerifier [0x00AF8959+2801161]
        GetHandleVerifier [0x00AF295C+2776588]
        GetHandleVerifier [0x008E2280+612144]
        (No symbol) [0x007F4F6C]
        (No symbol) [0x007F11D8]
        (No symbol) [0x007F12BB]
        (No symbol) [0x007E4857]
        BaseThreadInitThunk [0x76D900C9+25]
```

```
        RtlGetAppContainerNamedObjectPath [0x776F7B4E+286]
        RtlGetAppContainerNamedObjectPath [0x776F7B1E+238]
```

In [35]:
```
print(len(Smartphones['Brand']),len(Smartphones['Phone name']), len(Smartphones['C
      len(Smartphones['RAM']),len(Smartphones['Storage(ROM)']),len(Smartphones['Pr
      len(Smartphones['Secondary Camera']),len(Smartphones['Display Size']),len(Sm
      len(Smartphones['Processor']),len(Smartphones['Processor Cores']),len(Smartpl
      len(Smartphones['Price']),len(Smartphones['URL']))
```

```
1 1 1 1 1 1 1 1 1 0 0 0 0 1
```

In [36]:
```
df = pd.DataFrame.from_dict(Smartphones)
df
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [36], in <cell line: 1>()
----> 1 df = pd.DataFrame.from_dict(Smartphones)
      2 df

File C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py:1677, in DataFrame.from_dict(cls, data, orient, dtype, columns)
   1674     raise ValueError("only recognize index or columns for orient")
   1676 if orient != "tight":
-> 1677     return cls(data, index=index, columns=columns, dtype=dtype)
   1678 else:
   1679     realdata = data["data"]

File C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py:636, in DataFrame.__init__(self, data, index, columns, dtype, copy)
    630     mgr = self._init_mgr(
    631         data, axes={"index": index, "columns": columns}, dtype=dtype, copy=copy
    632     )
    634 elif isinstance(data, dict):
    635     # GH#38939 de facto copy defaults to False only in non-dict cases
--> 636     mgr = dict_to_mgr(data, index, columns, dtype=dtype, copy=copy, typ=manager)
    637 elif isinstance(data, ma.MaskedArray):
    638     import numpy.ma.mrecords as mrecords

File C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\internals\construction.py:502, in dict_to_mgr(data, index, columns, dtype, typ, copy)
    494     arrays = [
    495         x
    496         if not hasattr(x, "dtype") or not isinstance(x.dtype, ExtensionDtype)
    497         else x.copy()
    498         for x in arrays
    499     ]
    500     # TODO: can we get rid of the dt64tz special case above?
--> 502 return arrays_to_mgr(arrays, columns, index, dtype=dtype, typ=typ, consolidate=copy)

File C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\internals\construction.py:120, in arrays_to_mgr(arrays, columns, index, dtype, verify_integrity, typ, consolidate)
    117 if verify_integrity:
    118     # figure out the index, if necessary
    119     if index is None:
--> 120         index = _extract_index(arrays)
    121     else:
    122         index = ensure_index(index)

File C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\internals\construction.py:674, in _extract_index(data)
    672 lengths = list(set(raw_lengths))
    673 if len(lengths) > 1:
--> 674     raise ValueError("All arrays must be of the same length")
    676 if have_dicts:
    677     raise ValueError(
    678         "Mixing dicts with non-Series may lead to ambiguous ordering."
    679     )

ValueError: All arrays must be of the same length
```

In [37]:
```
#question:Write a program to scrap geospatial coordinates (latitude, longitude) of
```

In [38]:
```
import pandas as pd
import selenium
from bs4 import BeautifulSoup
import time
from selenium import webdriver
import requests
import re
import warnings
warnings.filterwarnings('ignore')
from selenium.common.exceptions import NoSuchElementException, StaleElementReferen
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
```

In [39]:
```
driver=webdriver.Chrome()
```

In [40]:
```
url = 'https://www.google.co.in/maps'
driver.get(url)
time.sleep(2)
```

In [41]:
```
# 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 : hyderbad
URL Extracted:  https://www.google.co.in/maps/place/Hyderabad,+Telangana/@17.50794
24,78.3187968,12z/data=!4m6!3m5!1s0x3bcb99daeaebd2c7:0xae93b78392bafbc2!8m2!3d17.3
85044!4d78.486671!16zL20vMDljNnc?entry=ttu
Latitude = 17.5079424, Longitude = 78.3187968
```

In [42]:
```
#Write a program to scrap all the available details of best gaming laptops from dig
```

In [43]:
```
import pandas as pd
import selenium
from bs4 import BeautifulSoup
import time
from selenium import webdriver
import requests
import re
```

```
import warnings
warnings.filterwarnings('ignore')
from selenium.common.exceptions import NoSuchElementException, StaleElementReferen
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
```

In [45]:
```
driver=webdriver.Chrome()
```

In [46]:
```
url = "https://www.digit.in/"
driver.get(url)
time.sleep(2)
```

In [47]:
```
best_gam_laptops = driver.find_element(By.XPATH,"//div[@class='listing_container']/
```

In [48]:
```
Laptop_Name = []
Operating_sys = []
Display = []
Processor = []
Memory = []
Weight = []
Dimensions = []
Graph_proc = []
Price = []
```

In [49]:
```
#scraping the data of laptop names
laptop_name = driver.find_elements(By.XPATH,"//div[@class='right-container']/div/a/
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/l
    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/
    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]/tabl
    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/t
    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/t
    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]/tabl
    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/tb
    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 [50]:
```
print(len(Laptop_Name),
len(Operating_sys),
len(Display),
len(Processor),
len(Memory),
len(Weight),
len(Dimensions),
len(Price))
```

```
0 7 7 7 7 7 7 7
```

In [51]:
```
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[51]:

	Laptop Name	Operating System	Display	Processor	Memory	Weight	Dimensions	Price
0	NaN	Windows 11 Home	17.3" (2560 x 1440)	16 GB DDR5GB RAM & 1 TB SSD	12 GB DDR6 NVIDIA GeForce RTX 4080 Graphics card	397.1 x 262 x 27 mm dimension & 2.78 kg weight	₹ 269,777	Rs269,777
1	NaN	Windows 11 Home	17.3" (3840 x 2160)	64 GB DDR5GB RAM & 2 TB SSD	16 GB DDR6 NVIDIA GeForce RTX 3080Ti Graphics ...	397 x 330 x 23 mm dimension & 3.3 kg weight	₹ 499,990	Rs499,990
2	NaN	Windows 11 Home	16" (2560 x 1600)	32 GB DDR5GB RAM & 1 TB SSD	NVIDIA GeForce RTX 3070 Ti Graphics card	359.9 x 264.4 x 19.9 mm dimension & 3.6 kg weight	₹ 213,900	Rs213,900
3	NaN	Windows 11 Home	18" (1920 x 1200)	32 GB DDR5GB RAM & 1 TB SSD	12 GB DDR6 NVIDIA GeForce RTX 4080 Graphics card	294 x 399 x 23 mm dimension & 3.1 kg weight	₹ 279,990	Rs279,990
4	NaN	Windows 11 Home	16" (2560 x 1600)	16 GB DDR5GB RAM & 1 TB SSD	8 GB DDR6 NVIDIA GeForce RTX 4060 Graphics card	360 x 279 x 28 mm dimension & 2.6 kg weight	₹ 149,990	Rs149,990
5	NaN	Windows 11 Home	14" (1920 x 1200)	16 GB DDR5GB RAM & 1 TB SSD	8 GB DDR6 AMD Radeon RX 6700S Graphics card	312 x 227 x 19 mm dimension & 1.65 kg weight	₹ 156,990	Rs156,990
6	NaN	Windows 11 Home	15.6" (1920 x 1080)	16 GB DDR5GB RAM & 1 TB SSD	8 GB DDR6 NVIDIA GeForce RTX 4060 Graphics card	& 1.98 kg weight	₹ 130,990	Rs130,990

In [ ]:
```
#Write a python program to scrape the details for all billionaires from www.forbes
#“Rank”, “Name”, “Net worth”, “Age”, “Citizenship”, “Source”, “Industry”.
```

In [52]:
```
import pandas as pd
import selenium
from bs4 import BeautifulSoup
import time
from selenium import webdriver
import requests
import re
import warnings
```

```
warnings.filterwarnings('ignore')
from selenium.common.exceptions import NoSuchElementException, StaleElementReferen
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
```

In [53]:
```
driver=webdriver.Chrome()
```

In [54]:
```
url = "https://www.forbes.com/?sh=41bd46d2254c"
driver.get(url)
```

In [55]:
```
#let's get option button from the page
opt_btn = driver.find_elements(By.XPATH,"//div[@class='header__left']//button")


#select billionaires from options
blns = driver.find_elements(By.XPATH,"/html/body/div[1]/header/nav/div[3]/ul/li[1]


#select world billionaire
bln_list = driver.find_elements(By.XPATH,"/html/body/div[1]/header/nav/div[3]/ul/l
```

In [56]:
```
# 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)



    # 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)
```

```
    # 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)



    # 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)


    # 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)


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

In [57]:
```
print(len(Rank),
len(Person_Name),
len(Net_worth),
len(Age),
len(Citizenship),
len(Source),
len(Industry))
```

```
0 0 0 0 0 0 0
```

In [58]:
```
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[58]:

Rank	Name	Net Worth	Age	Citizenship	Source	Industry

In [59]:
```
#Write a program to extract at least 500 Comments, Comment upvote and time when com
#from any YouTube Video.
```

In [60]:
```
import pandas as pd
import selenium
from bs4 import BeautifulSoup
import time
from selenium import webdriver
import requests
import re
import warnings
warnings.filterwarnings('ignore')
```

```
from selenium.common.exceptions import NoSuchElementException, StaleElementReferen
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
```

In [61]:
```
driver=webdriver.Chrome()
```

In [62]:
```
url = "https://www.youtube.com/"
driver.get(url)
time.sleep(2)
```

In [63]:
```
search_bar = driver.find_element(By.XPATH,"//div[@class='ytd-searchbox-spt']/input
search_bar.send_keys("GOT")
```

In [64]:
```
search_btn = driver.find_element(By.ID,"search-icon-legacy")
search_btn.click()
```

In [65]:
```
video = driver.find_element(By.XPATH,"//yt-formatted-string[@class='style-scope yt
video.click()
```

In [66]:
```
for _ in range(1000):
    driver.execute_script("window.scrollBy(0,10000)")
```

In [67]:
```
# 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)



# 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])



# scrape the comment likes
like = driver.find_elements(By.XPATH,"//span[@class='style-scope ytd-comment-actio
for i in like:
    Likes.append(i.text)

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

In [68]:
```
print(len(comments),len(No_of_Likes))
```

```
100 100
```

In [69]:
```
Youtube = pd.DataFrame({})
Youtube['Comment'] = comments[:500]
#Youtube['Comment Time'] = comment_time[:500]
Youtube['Comment Upvotes'] = No_of_Likes[:500]
Youtube
```

Out[69]:

	Comment	Comment Upvotes
0	I didn't ask for a happy ending, just a well w...	1.5K
1	This is no trailer, this is a collection of hu...	5.5K
2	When you realize your getting old by knowing t...	1.2K
3	I miss this epic fantasy show... despite all i...	479
4	Game of thrones- An experience of a lifetime. ...	262
...	...	...
95	A melhor série da história com o pior final da...	3
96	Miss this show so much. Unfortunate how it end...	634
97	I still can't believe that walking dead is sti...	57
98	The best ever by far. Thanks god this series w...	4
99	I love the characters of game of thrones, i en...	

100 rows × 2 columns

In [70]:
```
#questionWrite a python program to scrape a data for all available Hostels from htt
#“London” location. You have to scrape hostel name, distance from city centre, rati
#reviews, privates from price, dorms from price, facilities and property descriptio
```

In [71]:
```
import pandas as pd
import selenium
from bs4 import BeautifulSoup
import time
from selenium import webdriver
import requests
import re
import warnings
warnings.filterwarnings('ignore')
from selenium.common.exceptions import NoSuchElementException, StaleElementReferen
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
```

In [72]:
```
driver=webdriver.Chrome()
```

In [73]:
```
url = "https://www.hostelworld.com/"
driver.get(url)
```

In [74]:
```
search_bar = driver.find_element(By.ID, "search-input-field")
```

```
---------------------------------------------------------------------------
NoSuchElementException                    Traceback (most recent call last)
Input In [74], in <cell line: 1>()
----> 1 search_bar = driver.find_element(By.ID, "search-input-field")

File ~\AppData\Roaming\Python\Python39\site-packages\selenium\webdriver\remote\webdriver.py:831, in WebDriver.find_element(self, by, value)
    828     by = By.CSS_SELECTOR
    829     value = f'[name="{value}"]'
--> 831 return self.execute(Command.FIND_ELEMENT, {"using": by, "value": value})["value"]

File ~\AppData\Roaming\Python\Python39\site-packages\selenium\webdriver\remote\webdriver.py:440, in WebDriver.execute(self, driver_command, params)
    438 response = self.command_executor.execute(driver_command, params)
    439 if response:
--> 440     self.error_handler.check_response(response)
    441     response["value"] = self._unwrap_value(response.get("value", None))
    442     return response

File ~\AppData\Roaming\Python\Python39\site-packages\selenium\webdriver\remote\errorhandler.py:245, in ErrorHandler.check_response(self, response)
    243         alert_text = value["alert"].get("text")
    244     raise exception_class(message, screen, stacktrace, alert_text)  # type: ignore[call-arg]  # mypy is not smart enough here
--> 245 raise exception_class(message, screen, stacktrace)

NoSuchElementException: Message: no such element: Unable to locate element: {"method":"css selector","selector":"[id="search-input-field"]"}
  (Session info: chrome=114.0.5735.91)
Stacktrace:
Backtrace:
        GetHandleVerifier [0x00858893+48451]
        (No symbol) [0x007EB8A1]
        (No symbol) [0x006F5058]
        (No symbol) [0x00720467]
        (No symbol) [0x0072069B]
        (No symbol) [0x0074DD92]
        (No symbol) [0x0073A304]
        (No symbol) [0x0074C482]
        (No symbol) [0x0073A0B6]
        (No symbol) [0x00717E08]
        (No symbol) [0x00718F2D]
        GetHandleVerifier [0x00AB8E3A+2540266]
        GetHandleVerifier [0x00AF8959+2801161]
        GetHandleVerifier [0x00AF295C+2776588]
        GetHandleVerifier [0x008E2280+612144]
        (No symbol) [0x007F4F6C]
        (No symbol) [0x007F11D8]
        (No symbol) [0x007F12BB]
        (No symbol) [0x007E4857]
        BaseThreadInitThunk [0x76D900C9+25]
        RtlGetAppContainerNamedObjectPath [0x776F7B4E+286]
        RtlGetAppContainerNamedObjectPath [0x776F7B1E+238]
```

In [75]:
```
search_bar.send_keys("London")
```

```
---------------------------------------------------------------------------
NoSuchWindowException                     Traceback (most recent call last)
Input In [75], in <cell line: 1>()
----> 1 search_bar.send_keys("London")

File ~\AppData\Roaming\Python\Python39\site-packages\selenium\webdriver\remote\web
element.py:231, in WebElement.send_keys(self, *value)
    228             remote_files.append(self._upload(file))
    229         value = "\n".join(remote_files)
--> 231 self._execute(
    232     Command.SEND_KEYS_TO_ELEMENT, {"text": "".join(keys_to_typing(value)),
"value": keys_to_typing(value)}
    233 )

File ~\AppData\Roaming\Python\Python39\site-packages\selenium\webdriver\remote\web
element.py:403, in WebElement._execute(self, command, params)
    401     params = {}
    402 params["id"] = self._id
--> 403 return self._parent.execute(command, params)

File ~\AppData\Roaming\Python\Python39\site-packages\selenium\webdriver\remote\web
driver.py:440, in WebDriver.execute(self, driver_command, params)
    438 response = self.command_executor.execute(driver_command, params)
    439 if response:
--> 440     self.error_handler.check_response(response)
    441     response["value"] = self._unwrap_value(response.get("value", None))
    442     return response

File ~\AppData\Roaming\Python\Python39\site-packages\selenium\webdriver\remote\err
orhandler.py:245, in ErrorHandler.check_response(self, response)
    243         alert_text = value["alert"].get("text")
    244     raise exception_class(message, screen, stacktrace, alert_text)  # typ
e: ignore[call-arg]  # mypy is not smart enough here
--> 245 raise exception_class(message, screen, stacktrace)

NoSuchWindowException: Message: no such window: target window already closed
from unknown error: web view not found
  (Session info: chrome=114.0.5735.91)
Stacktrace:
Backtrace:
        GetHandleVerifier [0x00858893+48451]
        (No symbol) [0x007EB8A1]
        (No symbol) [0x006F5058]
        (No symbol) [0x006DD073]
        (No symbol) [0x0073DEBB]
        (No symbol) [0x0074BFD3]
        (No symbol) [0x0073A0B6]
        (No symbol) [0x00717E08]
        (No symbol) [0x00718F2D]
        GetHandleVerifier [0x00AB8E3A+2540266]
        GetHandleVerifier [0x00AF8959+2801161]
        GetHandleVerifier [0x00AF295C+2776588]
        GetHandleVerifier [0x008E2280+612144]
        (No symbol) [0x007F4F6C]
        (No symbol) [0x007F11D8]
        (No symbol) [0x007F12BB]
        (No symbol) [0x007E4857]
        BaseThreadInitThunk [0x76D900C9+25]
        RtlGetAppContainerNamedObjectPath [0x776F7B4E+286]
        RtlGetAppContainerNamedObjectPath [0x776F7B1E+238]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: