WEB SCRAPING-ASSIGNMENT3

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

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

print("Enter the product to be searched: ")
product = input()

url = "https://www.amazon.in/s?k=" + product

page = requests.get(url)

soup = BeautifulSoup(page.content, 'html.parser')

products = soup.find_all(class_='a-link-normal a-text-normal')

for i in range(len(products)):
    print(products[i].text)
```

2.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 has less than 3 pages in search results then scrape all the products available under that product name. Details to be scraped are: "Brand Name", "Name of the Product", "Price", "Return/Exchange", "Expected Delivery", "Availability" and “Product URL”. In case, if any of the details are missing for any of the product then replace it by “-“.

In [ ]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd

product = input("Enter the product you want to search: ")
url = "https://www.amazon.in/s?k="+product
page = requests.get(url)
soup = BeautifulSoup(page.content, 'html.parser')

# Get the brand name
brand_name = soup.find_all('span', class_="a-size-medium a-color-base a-
brand_name_list = []
for i in brand_name:
    brand_name_list.append(i.text)

# Get the product name
title = soup.find_all('span', class_="a-size-base-plus a-color-base a-te
title_list = []
for i in title:
    title_list.append(i.text)

# Get the product price
price = soup.find_all('span', class_="a-price-whole")
price_list = []
for i in price:
    price_list.append(i.text)

# Get the product return and exchange
return_exchange = soup.find_all('span', class_="a-size-base a-color-seco
return_exchange_list = []
for i in return_exchange:
    return_exchange_list.append(i.text)

# Get the product expected delivery
expected_delivery = soup.find_all('span', class_="a-size-base a-color-se
expected_delivery_list = []
for i in expected_delivery:
    expected_delivery_list.append(i.text)

# Get the product availability
availability = soup.find_all('span', class_="a-size-base a-color-base")
availability_list = []
for i in availability:
    availability_list.append(i.text)

# Get the product URL
url_list = []
for link in soup.find_all('a', class_="a-link-normal a-text-normal"):
    url_list.append("https://www.amazon.in" + link.get('href'))

# Create a Dataframe of the details
data = {'Brand Name': brand_name_list, 'Name of the Product': title_list
df = pd.DataFrame(data)

# Save the dataframe as a csv file
df.to_csv('amazon_products.csv', index=False)

print("Data Scraped Successfully")
```

In [ ]:
```
3. Write a python program to access the search bar and search button on 
images each for keywords ‘fruits’, ‘cars’ and ‘Machine Learning’, ‘Guita
```

In [ ]:
```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time
import os
```

In [ ]:
```
# Path to chromedriver
chrome_path = r"C:\Users\User\Downloads\chromedriver_win32\chromedriver.

# Path where you want to save the images
save_path = r"C:\Users\User\Desktop\Python Projects\Web Scraping\Images"

# keywords for which images need to be downloaded
keywords = ['fruits', 'cars', 'Machine Learning', 'Guitar', 'Cakes']

# instantiating chrome options
chrome_options = Options()
chrome_options.add_argument("--disable-infobars")
chrome_options.add_argument("start-maximized")
chrome_options.add_argument("--disable-extensions")

# instantiating the driver
driver = webdriver.Chrome(chrome_options = chrome_options, executable_pa

# accessing google images
driver.get("https://www.google.com/imghp?hl=en")

for keyword in keywords:
        driver.find_element_by_name('q').clear()

        # entering the search keyword and submitting
        driver.find_element_by_name('q').send_keys(keyword)
        driver.find_element_by_name('btnG').click()

        # waiting for the page to load completely
        WebDriverWait(driver, 10).until(EC.visibility_of_element_located

        # get all image thumbnail results
        image_thumb = driver.find_elements_by_css_selector('img.Q4LuWd')

        # downloading the images
        for img in image_thumb[:10]:
                img.click()
                time.sleep(2)

                # click on the expanded image
                driver.find_element_by_css_selector('img.n3VNCb').click(

                # get the image url
                image_url = driver.find_element_by_css_selector('img.n3V

                # download the image using url
                if 'http' in image_url:
                        image_object = requests.get(image_url)
                        try:
                                # save the image
                                image = Image.open(BytesIO(image_object.
                                image_name = keyword + '_' + str(count) 
                                image.save(os.path.join(save_path, image_

                                # incrementing the counter
                                count += 1
                        except OSError:
                                print('could not save image')

                # closing the expanded image
                driver.find_element_by_css_selector('a.bzIqaf.zdktat.Kjw

                time.sleep(2)

# closing the driver
driver.close()
```

In [ ]:
```
4.Write a python program to search for a smartphone(e.g.: Oneplus Nord, 
and scrape following details for all the search results displayed on 1st
Name”, “Smartphone name”, “Colour”, “RAM”, “Storage(ROM)”, “Primary Came
“Secondary Camera”, “Display Size”, “Battery Capacity”, “Price”, “Produc
details is missing then replace it by “- “. Save your results in a dataf
```

In [ ]:
```
import requests
import pandas as pd
from bs4 import BeautifulSoup
```

In [ ]:
```
#url
url = 'https://www.flipkart.com/search?q=OnePlus+Nord&otracker=search&ot
```

In [ ]:
```
#sending get request
page = requests.get(url)

#parse the page
soup = BeautifulSoup(page.text, 'html.parser')

#find the details
container=soup.find_all('div',class_='_3O0U0u')

#create empty lists
brand=[]
name=[]
color=[]
ram=[]
rom=[]
primary_camera=[]
secondary_camera=[]
display_size=[]
battery=[]
price=[]
url=[]
```

In [ ]:
```
#loop over the results
for container in container:
    #Brand
    brand.append(container.find('div',class_='_2B_pmu').text)
    #Name
    name.append(container.find('div',class_='_3wU53n').text)
    #Color
    try:
        color.append(container.find('div',class_='_3ycxrs').text)
    except:
        color.append('-')
    #Ram
    try:
        ram.append(container.find('div',class_='_2RngUh').text.replace('
    except:
        ram.append('-')
    #ROM
    try:
        rom.append(container.find('div',class_='_3ULzGw').text.replace('
    except:
        rom.append('-')
    #Primary Camera
    try:
        primary_camera.append(container.find('div',class_='_2k4JXJ').tex
    except:
        primary_camera.append('-')
    #Secondary Camera
    try:
        secondary_camera.append(container.find('div',class_='_3_yGjZ').t
    except:
        secondary_camera.append('-')
    #Display Size
    try:
        display_size.append(container.find('div',class_='_2_KrJI').text.
    except:
        display_size.append('-')
    #Battery
    try:
        battery.append(container.find('div',class_='_3WHvuP').text.repla
    except:
        battery.append('-')
    #Price
    try:
        price.append(container.find('div',class_='_1vC4OE').text)
    except:
        price.append('-')
    #Product URL
    try:
        url.append('https://www.flipkart.com'+container.find('a', class_
    except:
        url.append('-')

#create dataframe
df=pd.DataFrame({'Brand':brand,'Name':name,'Color':color,'RAM':ram,'ROM'
                 'Secondary Camera':secondary_camera,'Display Size':display_

#export to csv
df.to_csv('OnePlus_Nord_details.csv',index=False)
```

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

In [ ]:
```
import requests
import json
```

In [ ]:
```
# enter the city name
city = 'New York'

# url for the google maps api
url = 'https://maps.googleapis.com/maps/api/geocode/json?address=' + cit

# send the request
r = requests.get(url)
```

In [ ]:
```
# convert the response to a json
response = r.json()

# extract the geospatial coordinates
# lat, lng
lat = response['results'][0]['geometry']['location']['lat']
lng = response['results'][0]['geometry']['location']['lng']

# print the coordinates
print('The coordinates of ' + city + ' are ' + str(lat) + ' ' + str(lng)
```

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

In [17]:
```
import requests
from bs4 import BeautifulSoup

page = requests.get("https://www.digit.in/top-products/best-gaming-lapto

soup = BeautifulSoup(page.content,'html.parser')

laptop_divs = soup.find_all('div', class_='product_box')

for laptop in laptop_divs:

    name = laptop.find('div', class_='product_title').text
    price = laptop.find('span', class_='pricenew').text
    specs = laptop.find('div', class_='product_specs').text

    print('Laptop Name: ', name)
    print('Price: ', price)
    print('Specs: ', specs)
    print()
```

7.Write a python program to scrape the details for all billionaires from www.forbes.com. Details to be scrapped: “Rank”, “Name”, “Net worth”, “Age”, “Citizenship”, “Source”, “Industry”.

In [ ]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

In [ ]:
```
#Get the webpage
url = "https://www.forbes.com/billionaires/#1a2c715d8aa2"
page = requests.get(url)
```

In [ ]:
```
#Parsing the webpage
soup = BeautifulSoup(page.content, 'html.parser')

#Find the table with billionaires details
table = soup.find(class_="table-container")

#Find all rows in the table
table_rows = table.find_all('tr')

#Extract the details
rank = []
name = []
net_worth = []
age = []
citizenship = []
source = []
industry = []

for row in table_rows:
    td = row.find_all('td')
    if len(td) == 7:
        rank.append(td[0].text.strip())
        name.append(td[1].text.strip())
        net_worth.append(td[2].text.strip())
        age.append(td[3].text.strip())
        citizenship.append(td[4].text.strip())
        source.append(td[5].text.strip())
        industry.append(td[6].text.strip())

#Create a dataframe
billionaires_df = pd.DataFrame(
    {'Rank': rank,
     'Name': name,
     'Net Worth': net_worth,
     'Age': age,
     'Citizenship': citizenship,
     'Source': source,
     'Industry': industry})

print(billionaires_df)
```

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

In [ ]:
```
import requests
import json

video_id = input("Please enter a YouTube video id: ")

api_key = 'your_api_key'
url = 'https://www.googleapis.com/youtube/v3/commentThreads?key={}&textF

response = requests.get(url)
data = json.loads(response.text)

comments = []

for item in data['items']:
    comment = item['snippet']['topLevelComment']['snippet']['textDisplay
    comments.append(comment)
    upvotes = item['snippet']['topLevelComment']['snippet']['likeCount']
    time = item['snippet']['topLevelComment']['snippet']['publishedAt']
    print('Comment:', comment)
    print('Upvotes:', upvotes)
    print('Time:', time)
    print()

print('Total comments extracted:', len(comments))
```

9.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 [26]:
```
import requests
from bs4 import BeautifulSoup

#Enter the URL
url = 'https://www.hostelworld.com/hostels/London'

# Fetch the HTML
r = requests.get(url)

# Parse the HTML
soup = BeautifulSoup(r.content, 'html.parser')

# Extract the data
data = soup.find_all('div', attrs={'class': 'propertyCard__details'})
```

In [27]:
```
# Print the data
for item in data:
    name = item.find('h3', attrs={'class': 'propertyCard__name'}).text.s
    distance = item.find('span', attrs={'class': 'propertyCard__distance
    ratings = item.find('span', attrs={'class': 'propertyCard__ratings'}
    total_reviews = item.find('span', attrs={'class': 'propertyCard__tota
    overall_reviews = item.find('span', attrs={'class': 'propertyCard__o
    privates_from_price = item.find('span', attrs={'class': 'propertyCar
    dorms_from_price = item.find('span', attrs={'class': 'propertyCard__
    facilities = item.find('ul', attrs={'class': 'propertyCard__faciliti
    property_description = item.find('p', attrs={'class': 'propertyCard_
    print('Name: ', name)
    print('Distance from city centre: ', distance)
    print('Ratings: ', ratings)
    print('Total Reviews: ', total_reviews)
    print('Overall Reviews: ', overall_reviews)
    print('Privates from Price: ', privates_from_price)
    print('Dorms from Price: ', dorms_from_price)
    print('Facilities: ', facilities)
    print('Property Description: ', property_description)
    print('-------------------------------------------------------')
```

In [ ]: