In [ ]:

answer 1

In [1]:
```
pip install requests beautifulsoup4 pandas
```

```
Note: you may need to restart the kernel to use updated packages.Looking in indexes: htt
ps://pypi.tuna.tsinghua.edu.cn/simple
Requirement already satisfied: requests in c:\users\dell\.conda\lib\site-packages (2.31.
0)
Requirement already satisfied: beautifulsoup4 in c:\users\dell\.conda\lib\site-packages
(4.12.2)
Requirement already satisfied: pandas in c:\users\dell\.conda\lib\site-packages (2.0.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\dell\.conda\lib\site
-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dell\.conda\lib\site-packages (f
rom requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\dell\.conda\lib\site-packa
ges (from requests) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\.conda\lib\site-packa
ges (from requests) (2023.7.22)
Requirement already satisfied: soupsieve>1.2 in c:\users\dell\.conda\lib\site-packages
(from beautifulsoup4) (2.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\dell\.conda\lib\site-p
ackages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\.conda\lib\site-packages (f
rom pandas) (2022.7)
Requirement already satisfied: tzdata>=2022.1 in c:\users\dell\.conda\lib\site-packages
(from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in c:\users\dell\.conda\lib\site-packages
(from pandas) (1.25.2)
Requirement already satisfied: six>=1.5 in c:\users\dell\.conda\lib\site-packages (from
python-dateutil>=2.8.2->pandas) (1.16.0)
```

Answer 1

In [10]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd


url = "https://www.shine.com/job-search/data-analyst-jobs-in-bangalore"
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
}

response = requests.get(url, headers=headers)

if response.status_code == 200:
    soup = BeautifulSoup(response.text, "html.parser")
    job_listings = soup.find_all("li", class_="srlLi")


    for listing in job_listings[:10]:
        job_title=[]
        job_title = listing.find("h2", class_="srlHeading").text.strip()
        job_titles.append(job_title)

        job_location=[]
```

```
        job_location = listing.find("span", class_="srlCompLoc").text.strip()
        job_locations.append(job_location)

        company_name=[]
        company_name = listing.find("span", class_="srlCName").text.strip()
        company_names.append(company_name)

        experirnce_required=[]
        experience = listing.find("span", class_="srlExp").text.strip()
        experience_required.append(experience)


    data = {
        "Job Title": job_titles,
        "Job Location": job_locations,
        "Company Name": company_names,
        "Experience Required": experience_required}

    df = pd.DataFrame(data)

    print(df)

else:
    print("Failed to retrieve the webpage.")
```

```
Empty DataFrame
Columns: [Job Title, Job Location, Company Name, Experience Required]
Index: []
```

In [ ]:

Answer 2

In [8]:

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

url = "https://www.shine.com/job-search/data-scientist-jobs-in-bangalore"
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
}


response = requests.get(url, headers=headers)



if response.status_code == 200:
    soup = BeautifulSoup(response.text, "html.parser")
    job_listings = soup.find_all("li", class_="srlLi")


    for listing in job_listings[:10]:
        job_title=[]
        job_title = listing.find("h2", class_="srlHeading").text.strip()
        job_titles.append(job_title)

        job_location=[]
        job_location = listing.find("span", class_="srlCompLoc").text.strip()
        job_locations.append(job_location)

        company_name=[]
        company_name = listing.find("span", class_="srlCName").text.strip()
        company_names.append(company_name)
```

```
    data = {
        "Job Title": job_titles,
        "Job Location": job_locations,
        "Company Name": company_names}

    df = pd.DataFrame(data)


    print(df)

else:
    print("Failed to retrieve the webpage.")
```

```
Empty DataFrame
Columns: [Job Title, Job Location, Company Name]
Index: []
```

In [ ]:

Answer 3

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


url = "https://www.shine.com/job-search/data-scientist-jobs"
url

payload = {
    "query": "Data Scientist",
    "loc": "Delhi/NCR",
    "minexp": "3",
    "maxexp": "6"}


response = requests.get(url, data=payload)


if response.status_code == 200:
    soup = BeautifulSoup(response.text, "html.parser")


    job_listings = soup.find_all("li", class_="srlLi")


    job_titles = []
    job_locations = []
    company_names = []
    experience_required = []


    for listing in job_listings[:10]:

        job_title = listing.find("h2", class_="srlHeading").text.strip()
        job_titles.append(job_title)


        job_location = listing.find("span", class_="srlCompLoc").text.strip()
        job_locations.append(job_location)


        company_name = listing.find("span", class_="srlCName").text.strip()
```

```
        company_names.append(company_name)


        experience = listing.find("span", class_="srlExp").text.strip()
        experience_required.append(experience)


    data = {
        "Job Title": job_titles,
        "Job Location": job_locations,
        "Company Name": company_names,
        "Experience Required": experience_required,
    }

    df = pd.DataFrame(data)


    print(df)
```

In [ ]:

Answer 4

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

brands = []
descriptions = []
prices = []

base_url = "https://www.flipkart.com/"
base_url


page_number = 1

max_sunglasses = 100

while len(brands) < max_sunglasses:

    search_url = f"{base_url}/search?q=sunglasses&page={page_number}"


    response = requests.get(search_url)


    if response.status_code == 200:
        soup = BeautifulSoup(response.text, "html.parser")


        sunglasses_listings = soup.find_all("div", class_="_1AtVbE")


        for listing in sunglasses_listings:

            brand = listing.find("div", class_="_2WkVRV").text.strip()
            brands.append(brand)


            description = listing.find("a", class_="IRpwTa").text.strip()
            descriptions.append(description)
```

```
            price = listing.find("div", class_="_30jeq3").text.strip()
            prices.append(price)


            if len(brands) >= max_sunglasses:
                break


        page_number += 1
    else:
        print("Failed to retrieve the webpage.")
        break


data = {
    "Brand": brands,
    "Product Description": descriptions,
    "Price": prices,
}

df = pd.DataFrame(data)


print(df.head(100))
import requests
from bs4 import BeautifulSoup
import pandas as pd


brands = []
descriptions = []
prices = []


base_url = "https://www.flipkart.com/"
base_url


page_number = 1

max_sunglasses = 100


while len(brands) < max_sunglasses:

    search_url = f"{base_url}/search?q=sunglasses&page={page_number}"


    response = requests.get(search_url)


    if response.status_code == 200:
        soup = BeautifulSoup(response.text, "html.parser")


        sunglasses_listings = soup.find_all("div", class_="_1AtVbE")


        for listing in sunglasses_listings:

            brand = listing.find("div", class_="_2WkVRV").text.strip()
            brands.append(brand)


            description = listing.find("a", class_="IRpwTa").text.strip()
```

```
            descriptions.append(description)


            price = listing.find("div", class_="_30jeq3").text.strip()
            prices.append(price)


            if len(brands) >= max_sunglasses:
                break


        page_number += 1
    else:
        print("Failed to retrieve the webpage.")
        break


data = {
    "Brand": brands,
    "Product Description": descriptions,
    "Price": prices,
}

df = pd.DataFrame(data)
print(df.head(100))
```

In [ ]:

Answer 5

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


review_ratings = []
review_summaries = []
full_reviews = []

url = "https://www.flipkart.com/apple-iphone-11-black-64-gb/productreviews/itm4e5041ba10
url

max_reviews = 100

reviews_scraped = 0
while reviews_scraped < max_reviews:

    response = requests.get(url)

    if response.status_code == 200:
        soup = BeautifulSoup(response.text, "html.parser")


        review_containers = soup.find_all("div", class_="_27M-vq")
        for container in review_containers:

            rating = container.find("div", class_="_3LWZlK").text.strip()
            review_ratings.append(rating)


            summary = container.find("p", class_="_2-N8zT").text.strip()
            review_summaries.append(summary)
```

```
            review_text = container.find("div", class_="t-ZTKy").text.strip()
            full_reviews.append(review_text)


            reviews_scraped += 1


            if reviews_scraped >= max_reviews:
                break


        next_page = soup.find("a", class_="_1LKTO3")


        if next_page:
            url = "https://www.flipkart.com" + next_page["href"]
        else:
            break

data = {
    "Rating": review_ratings,
    "Review Summary": review_summaries,
    "Full Review": full_reviews,
}

df = pd.DataFrame(data)
print(df)
```

In [ ]:

Answer 6

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

sneaker_brands = []
sneaker_descriptions = []
sneaker_prices = []


base_url = "https://www.flipkart.com/"
base_url

page_number = 1

max_sneakers = 100


while len(sneaker_brands) < max_sneakers:

    search_url = f"{base_url}/search?q=sneakers&page={page_number}"


    response = requests.get(search_url)


    if response.status_code == 200:
        soup = BeautifulSoup(response.text, "html.parser")


        sneaker_listings = soup.find_all("div", class_="_1AtVbE")
```

```
        for listing in sneaker_listings:

            brand = listing.find("div", class_="_2WkVRV").text.strip()
            sneaker_brands.append(brand)


            description = listing.find("a", class_="IRpwTa").text.strip()
            sneaker_descriptions.append(description)


            price = listing.find("div", class_="_30jeq3").text.strip()
            sneaker_prices.append(price)


            if len(sneaker_brands) >= max_sneakers:
                break


        page_number += 1
    else:
        print("Failed to retrieve the webpage.")
        break


data = {
    "Brand": sneaker_brands,
    "Product Description": sneaker_descriptions,
    "Price": sneaker_prices,
}

df = pd.DataFrame(data)
print(df)
```

In [ ]:

Answer 7

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


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

params = {"field-keywords": "Laptop", "rh": "n:1375425031"}


response = requests.get(url, params=params)


if response.status_code == 200:
    soup = BeautifulSoup(response.text, "html.parser")


    laptop_listings = soup.find_all("div", class_="s-result-item")


    for listing in laptop_listings[:10]:
        laptop_tittle=[]
        title = listing.find("span", class_="a-text-normal").text.strip()
```

```
        laptop_titles.append(title)

        laptop_rating=[]
        rating = listing.find("span", class_="a-icon-alt")
        if rating:
            laptop_ratings.append(rating.text.strip())
        else:
            laptop_ratings.append("Not available")

        laptop_price=[]
        price = listing.find("span", class_="a-price-whole")
        if price:
            laptop_prices.append(price.text.strip())
        else:
            laptop_prices.append("Not available")


for i in range(10):
    print("Title:", laptop_titles[i])
    print("Ratings:", laptop_ratings[i])
    print("Price:", laptop_prices[i])
    print()
```

In [ ]:

Answer 8

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



url = "https://www.azquotes.com/"
url

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.text, "html.parser")


    top_quotes_link = None
    for a in soup.find_all("a", href=True):
        if "Top Quotes" in a.text:
            top_quotes_link = a["href"]
            break

    if top_quotes_link:

        top_quotes_url = url + top_quotes_link
        response = requests.get(top_quotes_url)

        if response.status_code == 200:
            soup = BeautifulSoup(response.text, "html.parser")


            quotes = []
            for quote_elem in soup.find_all("div", class_="wrap-block"):
                quote_text = quote_elem.find("a", class_="title").text.strip()
                quotes.append(quote_text)
```

```
                authors = []
                author_elem = quote_elem.find("div", class_="author")
                author_text = author_elem.find("a").text.strip()
                authors.append(author_text)


                types = []
                type_elem = quote_elem.find("div", class_="kw-item")
                type_text = type_elem.find("a").text.strip()
                types.append(type_text)


            for i in range(1000):
                print(f"{i + 1}. Quote: {quotes[i]}")
                print(f"   Author: {authors[i]}")
                print(f"   Type: {types[i]}")
                print()
        else:
            print("Failed to retrieve the 'Top Quotes' page.")
    else:
        print("No link to 'Top Quotes' found on the main page.")
else:
    print("Failed to retrieve the main page.")
```

In [ ]:

Answer 9

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


url = "https://www.jagranjosh.com/"
url


response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.text, "html.parser")


    gk_option = soup.find("a", text="GK")
    if gk_option:
        gk_url = gk_option.get("href")

        response = requests.get(gk_url)


        if response.status_code == 200:
            soup = BeautifulSoup(response.text, "html.parser")


            prime_ministers_link = soup.find("a", text="List of all Prime Ministers of I
            if prime_ministers_link:
                prime_ministers_url = prime_ministers_link.get("href")


                response = requests.get(prime_ministers_url)


                if response.status_code == 200:
```

```
                    soup = BeautifulSoup(response.text, "html.parser")


                    prime_ministers_table = soup.find("table", class_="tablestyle2")


                    names = []
                    born_dead = []
                    term_of_office = []
                    remarks = []


                    for row in prime_ministers_table.find_all("tr")[1:]:
                        columns = row.find_all("td")


                        name = columns[0].text.strip()
                        birth_death = columns[1].text.strip()
                        term = columns[2].text.strip()
                        remark = columns[3].text.strip()


                        names.append(name)
                        born_dead.append(birth_death)
                        term_of_office.append(term)
                        remarks.append(remark)


                    data = {
                        "Name": names,
                        "Born-Dead": born_dead,
                        "Term of Office": term_of_office,
                        "Remarks": remarks,
                    }

                    df = pd.DataFrame(data)


                    print(df)

                else:
                    print("Failed to retrieve the page with the list of Prime Ministers.
            else:
                print("Link to 'List of all Prime Ministers of India' not found.")
        else:
            print("Failed to retrieve the GK page.")
    else:
        print("GK option not found on the main page.")
else:
    print("Failed to retrieve the main page.")
```

In [ ]:

Answer 10

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


url = "https://www.motor1.com/"
url
```

```
response = requests.get(url)


if response.status_code == 200:
    soup = BeautifulSoup(response.text, "html.parser")


    search_bar = soup.find("input", id="searchbox")
    if search_bar:
        search_bar["value"] = "50 most expensive cars"


        search_form = soup.find("form", id="searchform")
        if search_form:
            response = requests.post(url, data={"search": "50 most expensive cars"})


            if response.status_code == 200:
                soup = BeautifulSoup(response.text, "html.parser")


                link = soup.find("a", text="50 Most Expensive Cars in the World")
                if link:
                    link_url = link.get("href")


                    response = requests.get(link_url)


                    if response.status_code == 200:
                        soup = BeautifulSoup(response.text, "html.parser")


                        car_table = soup.find("table", class_="table")


                        car_names = []
                        car_prices = []


                        for row in car_table.find_all("tr")[1:]:
                            columns = row.find_all("td")


                            name = columns[0].text.strip()
                            price = columns[1].text.strip()


                            car_names.append(name)
                            car_prices.append(price)


                        data = {
                            "Car Name": car_names,
                            "Price": car_prices,
                        }

                        df = pd.DataFrame(data)


                        print(df)

                    else:
                        print("Failed to retrieve the page with the list of cars.")
                else:
                    print("Link to '50 Most Expensive Cars in the World' not found.")
```

```
            else:
                print("Search request failed.")
        else:
            print("Search form not found.")
    else:
        print("Search bar not found.")
else:
    print("Failed to retrieve the main page.")
```