In [27]:
```
# !pip install selenium
```

In [28]:
```
# !pip install webdriver-manager
```

In [ ]:
```
# pip install webdriver_manager --upgrade
```

In [4]:
```
# !pip install undetected-chromedriver
```

In [1]:
```
import selenium
from selenium import webdriver
from selenium.webdriver.common.by import By
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.service import Service
import undetected_chromedriver as uc
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
import time
```

In [5]:
```
driver = uc.Chrome()
driver.get(url="https://www.naukri.com/")
```

In [6]:
```
designation= driver.find_element(By.CLASS_NAME, "suggestor-input")
designation.send_keys("Data Analyst")
location = driver.find_element(By.XPATH, "/html/body/div/div[7]/div/div/div[5]/div/div/div/div[1]/div/input")
location.send_keys("Bangalore")
search = driver.find_element(By.CLASS_NAME, "qsbSubmit")
search.click()
```

In [7]:
```
job_title = []
job_location = []
company_name = []
experienced_required = []
```

In [8]:
```
# scrapinng job title from the given page
title_tags = driver.find_elements(By.XPATH, '//a[@class="title ellipsis"]')

for i in title_tags[0:10]:
    title = i.text
    job_title.append(title)
```

```
job_title
```

Out[8]:
```
['Data Analyst',
 'Data Analyst',
 'Data Analyst',
 'Data Analyst',
 'Data Analyst I',
 'Data Analyst',
 'Data Analyst',
 'Data Analyst',
 'Data Analyst',
 'Data Analyst']
```

In [9]:
```
# scrapinng job location from the given page
location_tags = driver.find_elements(By.XPATH, '//span[@class="ellipsis fleft locWdth"]')

for i in location_tags[0:10]:
    location = i.text
    job_location.append(i.text)

job_location
```

Out[9]:
```
['Bangalore/ Bengaluru, Karnataka, Gurgaon/ Gurugram, Haryana',
 'Bangalore/Bengaluru',
 'Bangalore/Bengaluru',
 'Bangalore/Bengaluru, Hyderabad/Secunderabad, Pune',
 'Bangalore/Bengaluru',
 'Bangalore/Bengaluru',
 'Bangalore/Bengaluru',
 'Bangalore/Bengaluru, Chennai',
 'Bangalore/Bengaluru',
 'Bangalore/ Bengaluru, Karnataka(Hebbal Kempapura)']
```

In [10]:
```
# scrapinng company names from the given page
company_tags = driver.find_elements(By.XPATH, '//a[@class="subTitle ellipsis fleft"]')

for i in company_tags[0:10]:
    company = i.text
    company_name.append(i.text)

company_name
```

Out[10]:
```
['Delhivery',
 'Eastvantage',
 'Everest Vacuum',
 'Synchron Infotech',
 'Cerner',
 'Yulu Bikes',
 'Persolkelly India',
 'Shell',
 'Paychex It Solutions',
 'Futurlytic']
```

In [11]:
```
# scrapinng experienced required from the given page
experienced_tags = driver.find_elements(By.XPATH, '//div[@class="ellipsis job-description"]')

for i in experienced_tags[0:10]:
    experience = i.text
    experienced_required.append(i.text)

experienced_required
```

Out[11]:
```
['Knowledge of SQL and Python is a must',
 'Below are the Qualifications and requirements for this position: . Bachelors degree in ...',
 'Mandatory Skills: . You have in-depth knowledge and a good overview of data analytics, ...',
 '. Qualification: . Bachelors degree in Computer Science, Information Technology, Data S...',
 '. At least 4 years additional work experience directly related to the duties of the job...',
 '. Proficiency in Data warehousing concepts would be given preference . 4- 6 years of ex...',
 'Bachelors degree with a focus on Computer science, business systems or IT related degre...',
 'Preferred experience in Retail, CPG or E-commerce. A practical common-sense approach to...',
 'Candidates should be able to communicate effectively with business users, technical per...',
 'Responsibilities for Data analyst.The candidate must be very good in English, both in v...']
```

In [12]:
```
print(len(job_location), len(job_title), len(company_name), len(experienced_required))
```

```
10 10 10 10
```

In [13]:
```
df = pd.DataFrame({"Tite": job_title, "Location": job_location, "Company": company_name, "Experienced": experienced_req
df
```

Out[13]:

	Tite	Location	Company	Experienced
0	Data Analyst	Bangalore/ Bengaluru, Karnataka, Gurgaon/ Guru...	Delhivery	Knowledge of SQL and Python is a must
1	Data Analyst	Bangalore/Bengaluru	Eastvantage	Below are the Qualifications and requirements ...
2	Data Analyst	Bangalore/Bengaluru	Everest Vacuum	Mandatory Skills: . You have in-depth knowledg...
3	Data Analyst	Bangalore/Bengaluru, Hyderabad/Secunderabad, Pune	Synchron Infotech	. Qualification: . Bachelors degree in Compute...
4	Data Analyst I	Bangalore/Bengaluru	Cerner	. At least 4 years additional work experience ...
5	Data Analyst	Bangalore/Bengaluru	Yulu Bikes	. Proficiency in Data warehousing concepts wou...
6	Data Analyst	Bangalore/Bengaluru	Persolkelly India	Bachelors degree with a focus on Computer scie...
7	Data Analyst	Bangalore/Bengaluru, Chennai	Shell	Preferred experience in Retail, CPG or E-comme...
8	Data Analyst	Bangalore/Bengaluru	Paychex It Solutions	Candidates should be able to communicate effec...
9	Data Analyst	Bangalore/ Bengaluru, Karnataka(Hebbal Kempapura)	Futurlytic	Responsibilities for Data analyst.The candidat...

Second Part

In [14]:

```
# scrapinng job title from the given page
url= driver.find_elements(By.XPATH, '//a[@class="title ellipsis"]')
url[0:10]
```

Out[14]:
```
[<undetected_chromedriver.webelement.WebElement (session="0cde17e48f5b160f144915050093b143", element="6ECA047BDAD39185BA15F600FB7E4CFC_element_1826")>,
 <undetected_chromedriver.webelement.WebElement (session="0cde17e48f5b160f144915050093b143", element="6ECA047BDAD39185BA15F600FB7E4CFC_element_1827")>,
 <undetected_chromedriver.webelement.WebElement (session="0cde17e48f5b160f144915050093b143", element="6ECA047BDAD39185BA15F600FB7E4CFC_element_1828")>,
 <undetected_chromedriver.webelement.WebElement (session="0cde17e48f5b160f144915050093b143", element="6ECA047BDAD39185BA15F600FB7E4CFC_element_1829")>,
 <undetected_chromedriver.webelement.WebElement (session="0cde17e48f5b160f144915050093b143", element="6ECA047BDAD39185BA15F600FB7E4CFC_element_1830")>,
 <undetected_chromedriver.webelement.WebElement (session="0cde17e48f5b160f144915050093b143", element="6ECA047BDAD39185BA15F600FB7E4CFC_element_1831")>,
 <undetected_chromedriver.webelement.WebElement (session="0cde17e48f5b160f144915050093b143", element="6ECA047BDAD39185BA15F600FB7E4CFC_element_1832")>,
 <undetected_chromedriver.webelement.WebElement (session="0cde17e48f5b160f144915050093b143", element="6ECA047BDAD39185BA15F600FB7E4CFC_element_1833")>,
 <undetected_chromedriver.webelement.WebElement (session="0cde17e48f5b160f144915050093b143", element="6ECA047BDAD39185BA15F600FB7E4CFC_element_1834")>,
 <undetected_chromedriver.webelement.WebElement (session="0cde17e48f5b160f144915050093b143", element="6ECA047BDAD39185BA15F600FB7E4CFC_element_1835")>]
```

In [15]:
```
for i in url[0:10]:
    print(i.get_attribute('href'))
```

```
https://www.naukri.com/job-listings-data-analyst-delhivery-limited-gurgaon-gurugram-haryana-bangalore-bengaluru-karnataka-1-to-3-years-180723003988
https://www.naukri.com/job-listings-data-analyst-eastvantage-bangalore-bengaluru-4-to-6-years-200723006220
https://www.naukri.com/job-listings-data-analyst-everest-vacuum-bangalore-bengaluru-2-to-5-years-150723500323
https://www.naukri.com/job-listings-data-analyst-synchron-infotech-hyderabad-secunderabad-pune-bangalore-bengaluru-5-to-8-years-190723501894
https://www.naukri.com/job-listings-data-analyst-i-cerner-corporation-bangalore-bengaluru-5-to-10-years-190723502864
https://www.naukri.com/job-listings-data-analyst-yulu-bangalore-bengaluru-4-to-6-years-180723502637
https://www.naukri.com/job-listings-data-analyst-persolkelly-bangalore-bengaluru-0-to-2-years-170723501977
https://www.naukri.com/job-listings-data-analyst-shell-india-markets-private-limited-chennai-bangalore-bengaluru-1-to-2-years-120723909197
https://www.naukri.com/job-listings-data-analyst-paychex-it-solutions-inida-pvt-ltd-bangalore-bengaluru-3-to-8-years-060723500479
https://www.naukri.com/job-listings-data-analyst-futurlytic-bangalore-bengaluru-karnataka-1-to-2-years-180723007069
```

In [16]:
```
job_titles = []
```

In [17]:
```
start = 0
end = 2
for page in range(start,end):
    title = driver.find_elements(By.XPATH, '//a[@class="title ellipsis"]')
```

```
    for i in title:
        job_titles.append(i.text)
    next_bottom = driver.find_element(By.XPATH, '//a[@class="fright fs14 btn-secondary br2"]')
    next_bottom.click()
    time.sleep(3)
```

In [ ]:
```
len(job_titles)
```

In [ ]:
```
job_titles[0:10]
```

Q3: In this question you have to scrape data using the filters available on the webpage as shown below:

You have to use the location and salary filter. You have to scrape data for “Data Scientist” designation for first 10 job results. You have to scrape the job-title, job-location, company name, experience required. The location filter to be used is “Delhi/NCR”. The salary filter to be used is “3-6” lakhs The task will be done as shown in the below steps:

1. first get thewebpage https://www.naukri.com/
2. Enter “Data Scientist” in “Skill, Designations, and Companies” field.
3. Then click the searchbutton.
4. Then apply the location filter and salary filter by checking the respectiveboxes
5. Then scrape the data for the first 10 jobs results youget.
6. Finally create a dataframe of the scrapeddata. Note: All of the above steps have to be done in code. No step is to be done manually.

In [18]:
```
driver = uc.Chrome()
driver.get(url="https://www.naukri.com/")
```

In [21]:
```
designation= driver.find_element(By.CLASS_NAME, "suggestor-input")
designation.send_keys("Data Scientist")
location = driver.find_element(By.XPATH, "/html/body/div/div[7]/div/div/div[5]/div/div/div/div[1]/div/input")
location.send_keys("Delhi/NCR")
search = driver.find_element(By.CLASS_NAME, "qsbSubmit")
search.click()
```

In [23]:
```
job_location = []
```

```
# scrapinng job location from the given page
location_tags = driver.find_elements(By.XPATH, '//span[@class="ellipsis fleft locWdth"]')

for i in location_tags:
    location = i.text
    job_location.append(i.text)

print(len(job_location))
```

```
20
```

In [34]:
```
# scrapinng job salary from the given page
salaries = []
filter_salary_box = driver.find_element(By.XPATH, '/html/body/div[1]/div[4]/div/div/section[1]/div[2]/div[5]/div[2]/div
filter_salary_box.click()
```

In [35]:
```
salary = driver.find_elements(By.XPATH, '//li[@class="fleft br2 placeHolderLi salary"]')

for i in salary:
    salaries.append(i.text)
```

In [36]:
```
salaries
```

Out[36]:
```
['Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed',
 'Not disclosed']
```

In [37]:
```
print(len(salaries))
```

```
20
```

In [38]:
```
# scrapinng job title from the given page
job_title = []
title_tags = driver.find_elements(By.XPATH, '//a[@class="title ellipsis"]')

for i in title_tags:
    title = i.text
    job_title.append(title)
```

In [ ]:
```
len(job_title)
```

In [39]:
```
ds = pd.DataFrame({"Tite": job_title, "Location": job_location, "Salary": salaries})
ds
```

Out[39]:

	Tite	Location	Salary
0	Data Science Specialist	Delhi / NCR, Kolkata, Mumbai, Nagpur, Hyderaba...	Not disclosed
1	Data Scientist	Noida, Kolkata, Mumbai, Hyderabad/Secunderabad...	Not disclosed
2	Data Science consultant	Delhi / NCR, Noida, Kolkata, Mumbai, Hyderabad...	Not disclosed
3	Applied Scientist	Delhi / NCR, Noida, Kolkata, Mumbai, Hyderabad...	Not disclosed
4	Data Scientist	Noida, Kolkata, Mumbai, Hyderabad/Secunderabad...	Not disclosed
5	Junior Data Scientist	Delhi / NCR, Noida, Mumbai, Hyderabad/Secunder...	Not disclosed
6	Deputy Manager - B2C Underwriting	Hyderabad/Secunderabad, Pune, Chennai, Gurgaon...	Not disclosed
7	Data Scientist	Noida, Gurgaon/Gurugram, Bangalore/Bengaluru	Not disclosed
8	Sr. Data Scientist	Noida, Gurgaon/Gurugram, Bangalore/Bengaluru	Not disclosed
9	Senior Specialist	Noida, Hyderabad/Secunderabad, Pune, Gurgaon/G...	Not disclosed
10	Data Scientist II, Tech	Noida, Hyderabad/Secunderabad, Pune, Gurgaon/G...	Not disclosed
11	MACHINE LEARNING / DATA SCIENCE TRAINER	Noida, New Delhi, Bangalore/Bengaluru	Not disclosed
12	Senior Data Scientist	Noida, Uttar Pradesh, Chennai, Tamil Nadu, Ban...	Not disclosed
13	Data Scientist	Kolkata, Mumbai, New Delhi, Hyderabad/Secunder...	Not disclosed
14	Data Scientist/ Senior Data Scientist/ Manager...	Kolkata, Mumbai, New Delhi, Hyderabad/Secunder...	Not disclosed
15	Data Scientist	Kolkata, Mumbai, New Delhi, Hyderabad/Secunder...	Not disclosed
16	Deputy Area Manager - B2C Underwriting Rural	Kolkata, Mumbai, New Delhi, Hyderabad/Secunder...	Not disclosed
17	Data Scientist	Kolkata, Mumbai, New Delhi, Hyderabad/Secunder...	Not disclosed
18	Deputy Manager - B2C Underwriting Rural	Kolkata, Mumbai, New Delhi, Hyderabad/Secunder...	Not disclosed
19	Data Scientist	Kolkata, Mumbai, New Delhi, Hyderabad/Secunder...	Not disclosed

Q4: Scrape data of first 100 sunglasses listings on flipkart.com. You have to scrape four attributes:

1. Brand
2. ProductDescription

3. Price The attributes which you have to scrape is ticked marked in the below imag

To scrape the data you have to go through following steps:

1. Go to Flipkart webpage by url :https://www.flipkart.com/
2. Enter “sunglasses” in the search field where “search for products, brands and more” is written and click the search icon
3. After that you will reach to the page having a lot of sunglasses. From this page you can scrap the required data as usual.
4. After scraping data from the first page, go to the “Next” Button at the bottom other page , then click on it.
5. Now scrape data from this page asusual
6. Repeat this until you get data for 100sunglasses. Note: That all of the above steps have to be done by coding only and not manually.

In [2]:
```
driver = uc.Chrome()
driver.get(url="https://www.flipkart.com/")
```

In [3]:
```
designation= driver.find_element(By.CLASS_NAME, "_3704LK")
designation.send_keys("sunglasses")
search = driver.find_element(By.CLASS_NAME, "L0Z3Pu")
search.click()
```

In [4]:
```
Brand = []
ProductDescription = []
Price = []
```

In [5]:
```
Brand_tag = driver.find_elements(By.XPATH,'//div[@class="_2WkVRV"]')

for i in Brand_tag:
    Brand.append(i.text)

print(len(Brand))
print(Brand)
```

```
40
['john jacobs', 'OAKLEY', 'Elligator', 'SRPM', 'Styloze', 'Fastrack', 'eyedens', 'john jacobs', 'iCopertina', 'SRPM',
'john jacobs', 'john jacobs', 'ROADWAY', 'Elligator', 'PIRASO', 'RMKK', 'john jacobs', 'john jacobs', 'Fastrack', 'New
Specs', 'VINCENT CHASE', 'john jacobs', 'Fastrack', 'Fastrack', 'kingsunglasses', 'RMKK', 'john jacobs', 'john jacobs',
'kingsunglasses', 'BKGE', 'john jacobs', 'Resist', 'Fastrack', 'Sunnies', 'VINCENT CHASE', 'Elligator', 'hayden haiza',
'john jacobs', 'Fastrack', 'Elligator']
```

In [6]:
```
ProductDescription_tag = driver.find_elements(By.XPATH,'//a[@class="IRpwTa"]')
```

```
for i in ProductDescription_tag:
    ProductDescription.append(i.text)

print(len(ProductDescription))
print(ProductDescription)
```

```
39
['Polarized, UV Protection Round Sunglasses (48)', 'HOLBROOK Wayfarer Sunglass', 'UV Protection Cat-eye, Retro Square,
Oval, Round Sungla...', 'UV Protection Wayfarer Sunglasses (50)', 'Riding Glasses, Night Vision Wayfarer Sunglasses (Fr
ee ...', 'UV Protection Rectangular Sunglasses (Free Size)', 'UV Protection Retro Square Sunglasses (52)', 'UV Protecti
on Retro Square Sunglasses (Free Size)', 'UV Protection Wayfarer Sunglasses (50)', 'UV Protection Aviator Sunglasses (5
9)', 'Polarized, UV Protection Retro Square Sunglasses (56)', 'UV Protection Retro Square Sunglasses (Free Size)', 'UV
Protection Round Sunglasses (53)', 'UV Protection Aviator Sunglasses (Free Size)', 'UV Protection Aviator, Round Sungla
sses (55)', 'UV Protection Round Sunglasses (53)', 'Polarized, UV Protection Clubmaster Sunglasses (50)', 'UV Protectio
n Wayfarer Sunglasses (58)', 'UV Protection Rectangular Sunglasses (Free Size)', 'by Lenskart Polarized, UV Protection
Aviator Sunglasses...', 'UV Protection Wayfarer Sunglasses (52)', 'UV Protection Wayfarer Sunglasses (56)', 'Gradient,
UV Protection Wayfarer Sunglasses (Free Size)', 'Mirrored, UV Protection Wayfarer, Rectangular Sunglasse...', 'UV Prote
ction Aviator, Round Sunglasses (55)', 'Polarized, UV Protection Wayfarer Sunglasses (54)', 'UV Protection Wayfarer Sun
glasses (46)', 'Mirrored, UV Protection Wayfarer Sunglasses (Free Size)', 'Polarized, UV Protection Retro Square Sungla
sses (55)', 'UV Protection Retro Square Sunglasses (51)', 'Toughened Glass Lens, UV Protection, Riding Glasses, Ph...',
'UV Protection Wayfarer Sunglasses (Free Size)', 'Gradient, UV Protection, Riding Glasses Aviator Sunglas...', 'UV Prot
ection Wayfarer Sunglasses (59)', 'UV Protection, Mirrored Wayfarer Sunglasses (54)', 'Polarized, UV Protection Aviator
Sunglasses (55)', 'Polarized, UV Protection Aviator Sunglasses (59)', 'UV Protection Aviator Sunglasses (Free Size)',
'UV Protection Round Sunglasses (53)']
```

In [7]:
```
Price_tag = driver.find_elements(By.XPATH, '//div[@class="_30jeq3"]')
for i in Price_tag:
    Price.append(i.text)

print(len(Price))
print(Price)
```

```
40
['₹6,000', '₹5,179', '₹168', '₹194', '₹183', '₹539', '₹599', '₹6,000', '₹199', '₹194', '₹2,499', '₹6,000', '₹372', '₹19
9', '₹309', '₹249', '₹3,599', '₹2,850', '₹729', '₹269', '₹899', '₹2,375', '₹649', '₹649', '₹299', '₹249', '₹1,999',
'₹1,979', '₹274', '₹199', '₹3,134', '₹1,403', '₹679', '₹501', '₹569', '₹229', '₹391', '₹2,374', '₹589', '₹199']
```

I want to create a function to avoid repetition and make the code more organized.

In [9]:
```
def scrape_page(xpath_expr, target_list):
    Brand = []
    ProductDescription = []
    Price = []
```

```
    start = 0
    end = 3
    for page in range(start,end):
        scrape = driver.find_elements(By.XPATH, xpath_expr)
        for i in scrape:
            target_list.append(i.text)

        next_bottom = driver.find_element(By.XPATH, '//a[@class="_1LKTO3"]')
        next_bottom.click()
        time.sleep(3)



# Call the function for each data type
scrape_page('//div[@class="_2WkVRV"]', Brand)
scrape_page('//a[@class="IRpwTa"]', ProductDescription)
scrape_page('//div[@class="_30jeq3"]', Price)
```

In [11]:

```
Brand_100 = Brand[0:100]
ProductDescription_100 = ProductDescription[0:100]
Price_100 = Price[0:100]

data_glasses = pd.DataFrame({"Brand":Brand_100, "ProductDescription":ProductDescription_100, "Price":Price_100})
data_glasses
```

Out[11]:

	Brand	ProductDescription	Price
0	john jacobs	Polarized, UV Protection Round Sunglasses (48)	₹6,000
1	OAKLEY	HOLBROOK Wayfarer Sunglass	₹5,179
2	Elligator	UV Protection Cat-eye, Retro Square, Oval, Rou...	₹168
3	SRPM	UV Protection Wayfarer Sunglasses (50)	₹194
4	Styloze	Riding Glasses, Night Vision Wayfarer Sunglass...	₹183
...	...	...	...
95	RMKK	UV Protection Wayfarer Sunglasses (58)	₹249
96	john jacobs	UV Protection Rectangular Sunglasses (Free Size)	₹3,599
97	john jacobs	by Lenskart Polarized, UV Protection Aviator S...	₹2,850
98	Fastrack	UV Protection Wayfarer Sunglasses (52)	₹729
99	New Specs	UV Protection Wayfarer Sunglasses (56)	₹269

100 rows × 3 columns

Q5: Scrape 100 reviews data from flipkart.com for iphone11 phone. You have to go the link:

url = https://www.flipkart.com/apple-iphone-11-black-128-gb/product-reviews/itm8244e8d955aba?pid=MOBFWQ6BKRYBP5X8&lid=LSTMOBFWQ6BKRYBP5X8IBG6BS&marketplace=FLIPKART

As shown in the above page you have to scrape the tick marked attributes. These are:

1. Rating
2. Review summary
3. Full review
4. You have to scrape this data for first 100reviews. Note: All the steps required during scraping should be done through code only and not manually.

In [67]:
```
driver = uc.Chrome()
driver.get(url="https://www.flipkart.com/apple-iphone-11-black-128-gb/product-reviews/itm8244e8d955aba?pid=MOBFWQ6BKRYB
```

In [14]:
```
Rating = []
Review_summary = []
Full_review = []

rating = driver.find_elements(By.XPATH, '//div[@class="_3LWZlK _1BLPMq"]')

for i in rating:
    Rating.append(i.text)

reviews_summary = driver.find_elements(By.XPATH, '//p[@class="_2-N8zT"]')
for i in reviews_summary:
    Review_summary.append(i.text)

full_review = driver.find_elements(By.XPATH, '//div[@class="t-ZTKy"]')
for i in full_review:
    Full_review.append(i.text)
```

I want to create a function to avoid repetition and make the code more organized.

In [72]:
```
def scrape_page(xpath_expr, target_list):
    Rating = []
    Review_summary = []
    Full_review = []

    start = 0
    end = 5
    for page in range(start,end):
        scrape = driver.find_elements(By.XPATH, xpath_expr)
        for i in scrape:
            target_list.append(i.text)

        next_bottom = driver.find_element(By.XPATH, '//a[@class="ge-49M"]')
        next_bottom.click()
        time.sleep(3)



# Call the function for each data type
```

```
scrape_page( '//div[@class="_3LWZlK _1BLPMq"]', Rating)
scrape_page('//p[@class="_2-N8zT"]', Review_summary)
scrape_page('//div[@class="t-ZTKy"]', Full_review)
```

In [73]:
```
print(len(Rating), len(Review_summary), len(Full_review))
```

```
110 110 110
```

In [92]:
```
d_reviews = pd.DataFrame({"Rating":Rating, "Review_summary":Review_summary, "Full_review":Full_review})
d_reviews.iloc[0:100, :]
```

Out[92]:

	Rating	Review_summary	Full_review
0	5	Terrific	Very very good
1	5	Classy product	Camera is awesome\nBest battery backup\nA perf...
2	5	Wonderful	This is amazing at all
3	5	Brilliant	Excellent Phone.
4	5	Must buy!	It's really awesome
...	...	...	...
95	5	Excellent	Perfect Product!!
96	5	Terrific purchase	V Good all
97	5	Classy product	Good Camera
98	5	Simply awesome	Value for money 😍
99	5	Super!	Go for iPhone 11 , if confused between iPhone ...

100 rows × 3 columns

Q6: Scrape data for first 100 sneakers you find when you visit flipkart.com and search for "sneakers" in the

search field. You have to scrape 3 attributes of each sneaker:

1. Brand
2. Product Description

3. Price As shown in the below image, you have to scrape the above attributes.

In [74]:
```
designation = driver.find_element(By.CLASS_NAME, "_3704LK")
designation.send_keys("sneakers")
click_button = driver.find_element(By.XPATH, '/html/body/div/div/div[1]/div[1]/div[2]/div[2]/form/div/button')
click_button.click()
```

In [49]:
```
Brand = []
Product_Description = []
Price = []



brand = driver.find_elements(By.XPATH, '//div[@class="_2WkVRV"]')

for i in brand:
    Brand.append(i.text)

product_description = driver.find_elements(By.XPATH, '//a[@class="IRpwTa"]')
for i in product_description:
    Product_Description.append(i.text)

price = driver.find_elements(By.XPATH, '//div[@class="_30jeq3"]')
for i in price:
    Price.append(i.text)
```

I want to create a function to avoid repetition and make the code more organized.

In [52]:
```
# I want to create a function to avoid repetition and make the code more organized.

def scrape_page(xpath_expr, target_list):
    scrape = driver.find_elements(By.XPATH, xpath_expr)
    for i in scrape:
        target_list.append(i.text)

# Define the lists to store the extracted data
Brand = []
Product_Description = []
Price = []
```

```
# Call the function for each data type
scrape_page('//div[@class="_2WkVRV"]', Brand)
scrape_page('//a[@class="IRpwTa"]', Product_Description)
scrape_page('//div[@class="_30jeq3"]', Price)
```

In [75]:
```
# I want to create a function to avoid repetition and make the code more organized.

def scrape_page(xpath_expr, target_list):
    Brand = []
    Product_Description = []
    Price = []

    start = 0
    end = 2
    for page in range(start,end):
        scrape = driver.find_elements(By.XPATH, xpath_expr)
        for i in scrape:
            target_list.append(i.text)

        next_bottom = driver.find_element(By.XPATH, '//a[@class="ge-49M"]')
        next_bottom.click()
        time.sleep(3)



# Call the function for each data type
scrape_page('//div[@class="_2WkVRV"]', Brand)
scrape_page('//a[@class="IRpwTa"]', Product_Description)
scrape_page('//div[@class="_30jeq3"]', Price)
```

In [89]:
```
Brand_100= Brand[0:100]
Product_Description_100 = Product_Description[0:100]
Price_100 = Price[0:100]

data = pd.DataFrame({"Brand":Brand_100, "Product_Description":Product_Description_100, "Price":Price_100})
data
```

Out[89]:

	Brand	Product_Description	Price
0	Airson	Junior Zero1 Sports shoes for Men | Gym Traini...	₹989
1	Sparx	Stylish and Comfirtable Shoes For Mens Sneaker...	₹764
2	aadi	Lightweight,Comfort,Summer,Trendy,Walking,Outd...	₹449
3	SFR	Sneakers For Men	₹348
4	Kraasa	Sneakers For Women	₹399
...	...	...	...
95	BRUTON	Modern Trendy Sneakers Shoes Sneakers For Men	₹299
96	ATOM	Sneakers For Men	₹1,389
97	New Balance	XC72 Sneakers For Men	₹6,759
98	Plaeto	Plaeto Womens Starblast Multiplay Sports Shoes...	₹1,499
99	aadi	Lightweight,Comfort,Summer,Trendy,Walking,Outd...	₹499

100 rows × 3 columns

Q8: Write a python program to scrape data for Top 1000 Quotes of All Time.

The above task will be done in following steps:

1. First get the webpage https://www.azquotes.com/
2. Click on Top Quotes
3. Than scrap a) Quote b) Author c) Type Of Quotes

In [93]:
```
driver = uc.Chrome()
driver.get(url = 'https://www.azquotes.com/')
```

In [94]:
```
top_quotes = driver.find_element(By.XPATH, '/html/body/div[1]/div[2]/div[1]/div/div[3]/ul/li[5]/a')
top_quotes.click()
```

In [99]:
```
Quote = []
Author = []
TypeOfQuotes = []
```

```
quote = driver.find_elements(By.XPATH, '//a[@class="title"]')

for i in quote:
    Quote.append(i.text)

author = driver.find_elements(By.XPATH, '//div[@class="author"]')
for i in author:
    Author.append(i.text)

typeofquotes = driver.find_elements(By.XPATH, '//div[@class="tags"]')
for i in typeofquotes:
    TypeOfQuotes.append(i.text)
```

In [100…

```
print(len(Quote), len(Author), len(TypeOfQuotes))
```

```
100 100 100
```

In [106…

```
# I want to create a function to avoid repetition and make the code more organized.

def scrape_page(xpath_expr, target_list):
    Quote = []
    Author = []
    TypeOfQuotes = []

    start = 0
    end = 5
    for page in range(start,end):
        scrape = driver.find_elements(By.XPATH, xpath_expr)
        for i in scrape:
            target_list.append(i.text)

        next_bottom = driver.find_element(By.XPATH,'//li[@class="next"]' )
        next_bottom.click()
        time.sleep(3)



# Call the function for each data type
scrape_page('//a[@class="title"]', Quote)
scrape_page('//div[@class="author"]', Author)
scrape_page('//div[@class="tags"]', TypeOfQuotes)
```

In [108…
```
Quote_1000 = Quote[0:1000]
Author_1000 = Author[0:1000]
TypeOfQuotes_1000 = TypeOfQuotes[0:1000]
```

In [109…
```
dataframe = pd.DataFrame({"Quote":Quote_1000, "Author":Author_1000, "TypeOfQuotes":TypeOfQuotes_1000})
dataframe
```

Out[109]:

	Quote	Author	TypeOfQuotes
0	Can words describe the fragrance of the very b...	Neltje Blanchan	Spring, April, Fragrance
1	Faith is to believe what you do not see; the r...	Saint Augustine	Inspirational, Faith, Spiritual
2	When everything seems to be going against you,...	Henry Ford	Inspirational, Motivational, Positive
3	I have found that if you love life, life will ...	Arthur Rubinstein	Love, Inspirational, Life
4	To disarm the people... was the best and most ...	George Mason	Strength, Peace, Gun
...	...	...	...
995	Don't watch the clock; do what it does. Keep g...	Sam Levenson	Motivational, Moving On, Success
996	Don't wait for extraordinary opportunities. Se...	Orison Swett Marden	Inspirational, Strength, Strong
997	I celebrated Thanksgiving in an old-fashioned ...	Jon Stewart	Funny, Thanksgiving, Holiday
998	If you're not careful, the newspapers will hav...	Malcolm X	Hate, Lying, Media
999	A dress should be tight enough to show you're ...	Edith Head	Fashion, Eyebrows, Icons

1000 rows × 3 columns

Q9: Write a python program to display list of respected former Prime Ministers of India(i.e. Name, Born-Dead,

Term of office, Remarks) from https://www.jagranjosh.com/. This task will be done in following steps:

1. First get the webpagehttps://www.jagranjosh.com/
2. Then You have to click on the GK option
3. Then click on the List of all Prime Ministers of India
4. Then scrap the mentioned data and make theDataFrame.

In [63]:
```
driver = uc.Chrome()
driver.get('https://www.jagranjosh.com/')
```

In [64]:
```
GK_option = driver.find_element(By.XPATH, '/html/body/div[1]/div[1]/div/div[1]/div/div[5]/div/div[1]/header/div[3]/ul/l
GK_option.click()
```

In [71]:
```
List_PM = driver.find_element(By.XPATH,'/html/body/div[1]/div/div/div[2]/div/div[10]/div/div/ul/li[2]/a' )
List_PM.click()
```

In [147…
```
DataFrame = driver.find_element(By.XPATH, '//*[@id="itemdiv"]/div[4]/span/div[3]/table/tbody/tr[1]' )
DataFrame.text.split("\n")
```

Out[147]:
```
['S.N.', 'PM Name', 'Born-Dead', 'Term of office', 'Remark']
```

In [156…
```
table = []


DataFrame = driver.find_elements(By.TAG_NAME, "tr")

for i in DataFrame:
    table.append(i.text.split("\n"))
table
```

Out[156]:
```
[['S.N.', 'PM Name', 'Born-Dead', 'Term of office', 'Remark'],
 ['1.',
  'Jawahar Lal Nehru',
  '(1889–1964)',
  '15 August 1947 to 27 May 1964',
  '16 years, 286 days',
  'The first prime minister of India and the longest-serving PM of India, the first to die in office.'],
 ['2.',
  'Gulzarilal Nanda (Acting)',
  '(1898-1998)',
  '27 May 1964 to 9 June 1964,',
  '13 days',
  'First acting PM of India'],
 ['3.',
  'Lal Bahadur Shastri',
  '(1904–1966)',
  '9 June 1964 to 11 January 1966',
  '1 year, 216 days',
  "He has given the slogan of 'Jai Jawan Jai Kisan' during the Indo-Pak war of 1965"],
 ['4. ',
  'Gulzari Lal Nanda  (Acting)',
  '(1898-1998)',
  '11 January 1966 to 24 January 1966',
  '13 days',
  '-'],
 ['5.',
  'Indira Gandhi',
  '(1917–1984)',
  '24 January 1966 to 24 March 1977',
  '11 years, 59 days',
  'First female Prime Minister of India'],
 ['6.',
  'Morarji Desai',
  '(1896–1995)',
  '24 March 1977 to  28 July 1979 ',
  '2 year, 126 days',
  'Oldest to become PM (81 years old) and first to resign from office'],
 ['7.',
  'Charan Singh',
  '(1902–1987)',
  '28 July 1979 to 14 January 1980',
  '170 days',
  'Only PM who did not face the Parliament'],
 ['8.',
  'Indira Gandhi',
```

```
 '(1917–1984)',
 '14 January 1980 to 31 October 1984',
 '4 years, 291 days',
 'The first lady who served as PM for the second term'],
['9.',
 'Rajiv Gandhi',
 '(1944–1991)',
 '31 October 1984 to 2 December 1989',
 '5 years, 32 days',
 'Youngest to become PM (40 years old)'],
['10.',
 'V. P. Singh',
 '(1931–2008)',
 '2 December 1989 to 10 November 1990',
 '343 days',
 'First PM to step down after a vote of no confidence'],
['11.',
 'Chandra Shekhar',
 '(1927–2007)',
 '10 November 1990 to 21 June 1991',
 '223 days',
 'He belongs to  Samajwadi Janata Party'],
['12.',
 'P. V. Narasimha Rao',
 '(1921–2004)',
 '21 June 1991 to 16 May 1996',
 '4 years, 330 days',
 'First PM from South India'],
['13.',
 'Atal Bihari Vajpayee',
 '(1924- 2018)',
 '16 May 1996 to 1 June 1996',
 '16 days',
 'PM for shortest tenure'],
['14.',
 'H. D. Deve Gowda',
 '(born 1933)',
 '1 June 1996 to 21 April 1997',
 '324 days',
 'He belongs to  Janata Dal'],
['15.',
 'Inder Kumar Gujral',
 '(1919–2012)',
 '21 April 1997 to 19 March 1998 ',
 '332 days',
```

```
 '------'],
['16.',
 'Atal Bihari Vajpayee',
 '(1924-2018)',
 '19 March 1998 to 22 May 2004 ',
 '6 years, 64 days',
 ' The first non-congress PM who completed a full term as PM'],
['17.',
 'Manmohan Singh',
 '(born 1932)',
 '22 May 2004 to 26 May 2014   ',
 '10 years, 4 days',
 ' First Sikh PM'],
['18.',
 'Narendra Modi',
 '(born 1950)',
 '26 May 2014 - 2019',
 '4th Prime Minister of India who served two consecutive tenures'],
['19.',
 'Narendra Modi',
 '(born 1950)',
 '30 May 2019- Incumbent',
 'First non-congress PM with two consecutive tenures'],
['List of all Presidents of India List of Nicknames of Indian Prime Ministers']]
```

In [172…

```
df = pd.DataFrame(table, columns=["PM", "Name", "Born-Dead", "Term of office",  "Span", "Remark"])
df = df.iloc[1:20, :]
df.head()
```

Out[172]:

	PM	Name	Born-Dead	Term of office	Span	Remark
1	1.	Jawahar Lal Nehru	(1889–1964)	15 August 1947 to 27 May 1964	16 years, 286 days	The first prime minister of India and the long...
2	2.	Gulzarilal Nanda (Acting)	(1898-1998)	27 May 1964 to 9 June 1964,	13 days	First acting PM of India
3	3.	Lal Bahadur Shastri	(1904–1966)	9 June 1964 to 11 January 1966	1 year, 216 days	He has given the slogan of 'Jai Jawan Jai Kisa...
4	4.	Gulzari Lal Nanda (Acting)	(1898-1998)	11 January 1966 to 24 January 1966	13 days	-
5	5.	Indira Gandhi	(1917–1984)	24 January 1966 to 24 March 1977	11 years, 59 days	First female Prime Minister of India

In [ ]:

Q10: Write a python program to display list of 50 Most expensive cars in the world (i.e.

Car name and Price) from https://www.motor1.com/ This task will be done in following steps:

1. First get the webpage https://www.motor1.com/
2. Then You have to type in the search bar '50 most expensive cars'
3. Then click on 50 most expensive cars in the world..
4. Then scrap the mentioned data and make the dataframe.

In [173…
```
driver = uc.Chrome()
driver.get('https://www.motor1.com/')
```

In [179…
```
designation= driver.find_element(By.XPATH, '/html/body/div[13]/div[2]/div/div/div[3]/div/div/div/form/input')
designation.send_keys("50 most expensive cars")
```

In [180…
```
search = driver.find_element(By.XPATH, '/html/body/div[13]/div[2]/div/div/div[3]/div/div/div/form/button[1]')
search.click()
```

In [181…
```
most_expensive = driver.find_element(By.XPATH, '/html/body/div[10]/div[9]/div/div[1]/div/div/div[2]/div/div[1]/h3/a')
most_expensive.click()
```

In [183…
```
Car_name  = []
Price = []

cars = driver.find_elements(By.XPATH, '//h3[@class="subheader"]')
for i in cars:
    Car_name.append(i.text)

prices = driver.find_elements(By.TAG_NAME, "strong" )
for i in prices:
    Price.append(i.text)
```

In [184…
```
len(Car_name)
```

Out[184]: 51

In [185…
```
len(Price)
```

Out[185]: 51

In [186…
```
pd.DataFrame({"Car":Car_name, "Price":Price})
```

Out[186]:

	Car	Price
0	De Tomaso P72	Price: $1.3 Million
1	Ferrari LaFerrari	Price: $1.4 Million
2	Pagani Huayra	Price: $1.4 Million
3	McLaren Elva	
4	Czinger 21C	Price: $1.7 Million
5	Ferrari Monza	Price: $1.7 Million
6	Gordon Murray T.33	Price: $1.7 Million
7	Koenigsegg Gemera	Price: $1.7 Million
8	Zenvo TSR-S	Price: $1.7 Million
9	Hennessey Venom F5	Price: $1.7 Million
10	Bentley Bacalar	Price: $1.8 Million
11	Hispano Suiza Carmen Boulogne	Price: $1.9 Million
12	Bentley Mulliner Batur	Price: $1.9 Million
13	Deus Vayanne	Price: $2.0 Million
14	SSC Tuatara	Price: $2.0 Million
15	Lotus Evija	Price: $2.0 Million*
16	Aston Martin Vulcan	Price: $2.1 Million
17	Delage D12	Price: $2.3 Million
18	McLaren Speedtail	Price: $2.3 Million
19	Rimac Nevera	Price: $2.3 Million
20	Pagani Utopia	Price: $2.4 Million
21	Pininfarina Battista	Price: $2.5 Million
22	Ferrari FXX K Evo	Price: $2.5 Million
23	Gordon Murray T.50	Price: $2.6 Million
24	Lamborghini Countach	Price: $2.6 Million

	Car	Price
25	Mercedes-AMG Project One	Price: $2.6 Million
26	Aston Martin Victor	Price: $2.7 Million
27	Hennessey Venom F5 Roadster	Price: $3.0 Million
28	Koenigsegg Jesko	$3.0 Million
29	Aston Martin Valkyrie	Price: $3.0 Million
30	W Motors Lykan Hypersport	Price: $3.2 Million
31	McLaren Solus	Price: $3.4 Million
32	Pagani Huayra Roadster BC	$3.5 Million
33	Bugatti Chiron Pur Sport	Price: $3.5 Million
34	Lamborghini Sian	Price: $3.6 Million
35	Koenigsegg CC850	Price: $3.6 million
36	Bugatti Chiron Super Sport 300+	Price: $3.7 Million
37	Lamborghini Veneno	Price: $3.9 Million
38	Bugatti Bolide	Price: $4.5 Million
39	Bugatti Mistral	Price: $4.7 Million
40	Pagani Huayra Imola	Price: $5.0 Million
41	Bugatti Divo	Price: $5.4 Million
42	SP Automotive Chaos	Price: $5.8 Million
43	Pagani Codalunga	Price: $6.4 Million
44	Mercedes-Maybach Exelero	Price: $7.4 Million
45	Bugatti Centodieci	Price: $8.0 Million
46	Bugatti Chiron Profilée	Price: $9.0 Million
47	Rolls-Royce Sweptail	Price: $10.8 Million
48	Bugatti La Voiture Noire	Price: $12.8 Million
49	Rolls-Royce Boat Tail*	Price: $13.4 Million

	Car	Price
50	Most Expensive Cars In The World	Price: $28.0 Million (est.)

In [ ]: