Q1: Write a python program to scrape data for “Data Analyst” Job position in “Bangalore” location. You have to scrape the job-title, job-location, company_name, experience_required. You have to scrape first 10 jobs data.

In [42]:
```
import selenium
import pandas as pd
from selenium import webdriver
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
import time
```

In [43]:
```
#let's first connect to the driver
driver=webdriver.Chrome(r"chromedriver.exe")
```

In [44]:
```
#opening the naukri page on automated chrome browser
driver.get("https://www.naukri.com/")
```

In [45]:
```
driver.maximize_window()
```

In [13]:
```
#entering designation and location as required in the question
designation=driver.find_element(By.CLASS_NAME,"suggestor-input")
designation.send_keys('Data Analyst')
```

In [14]:
```
location=driver.find_element(By.XPATH,"/html/body/div[1]/div[7]/div/div/div[5]/div/div/div/div[1]/div/input")
location.send_keys('Bangalore')
```

In [15]:
```
search=driver.find_element(By.CLASS_NAME,"qsbSubmit")
search.click()
```

In [16]:
```
job_title=[]
job_location=[]
company_name=[]
experience_required=[]
```

In [21]:
```
#scrapping 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)

#scrapping job location  from the given page
location_tags=driver.find_elements(By.XPATH,'//span[@class="ellipsis fleft locWdth"]')
for i in title_tags[0:10]:
    location=i.text
    job_location.append(location)

#scrapping company name from the given page
company_tags=driver.find_elements(By.XPATH,'//a[@class="subTitle ellipsis fleft"]')
for i in title_tags[0:10]:
    company=i.text
    company_name.append(company)
#scrapping job experience from the given page
experience_tags=driver.find_elements(By.XPATH,'//span[@class="ellipsis fleft expwdth"]')
for i in experience_tags[0:10]:
    exp=i.text
    experience_required.append(exp)
```

In [18]:
```
print(len(job_title),len(job_location),len(company_name),len(experience_required))
```

```
10 10 10 10
```

In [22]:
```
import pandas as pd
df=pd.DataFrame({'Title':job_title,'Location':job_location,' company_name': company_name,'Experience': experien
```

In [23]:
```
df
```

Out[23]:

	Title	Location	company_name	Experience
0	Data Analyst	Data Analyst	Data Analyst	2-4 Yrs
1	Tech Data Analyst	Tech Data Analyst	Tech Data Analyst	3-6 Yrs
2	Data Analyst	Data Analyst	Data Analyst	5-8 Yrs
3	Data Analyst	Data Analyst	Data Analyst	4-6 Yrs
4	Celonis & Salesforce Data Analyst	Celonis & Salesforce Data Analyst	Celonis & Salesforce Data Analyst	3-6 Yrs
5	Celonis & Salesforce Data Analyst	Celonis & Salesforce Data Analyst	Celonis & Salesforce Data Analyst	2-7 Yrs
6	Data Analyst	Data Analyst	Data Analyst	3-5 Yrs
7	Data Analyst	Data Analyst	Data Analyst	6-9 Yrs
8	Data Analyst	Data Analyst	Data Analyst	6-9 Yrs
9	Data Analyst	Data Analyst	Data Analyst	5-10 Yrs
10	Data Analyst	Data Analyst	Data Analyst	2-4 Yrs
11	Tech Data Analyst	Tech Data Analyst	Tech Data Analyst	3-6 Yrs
12	Data Analyst	Data Analyst	Data Analyst	5-8 Yrs
13	Data Analyst	Data Analyst	Data Analyst	4-6 Yrs
14	Celonis & Salesforce Data Analyst	Celonis & Salesforce Data Analyst	Celonis & Salesforce Data Analyst	3-6 Yrs
15	Celonis & Salesforce Data Analyst	Celonis & Salesforce Data Analyst	Celonis & Salesforce Data Analyst	2-7 Yrs
16	Data Analyst	Data Analyst	Data Analyst	3-5 Yrs
17	Data Analyst	Data Analyst	Data Analyst	6-9 Yrs
18	Data Analyst	Data Analyst	Data Analyst	6-9 Yrs
19	Data Analyst	Data Analyst	Data Analyst	5-10 Yrs

Q2:Write a python program to scrape data for “Data Scientist” Job position in “Bangalore” location. You have to scrape the job-title, job-location, company_name. You have to scrape first 10 jobs data.

In [29]:
```
#opening the naukri page on automated chrome browser
driver.get("https://www.naukri.com/")
```

In [30]:
```
driver.maximize_window()
```

In [46]:
```
#entering designation and location as required in the question
designation=driver.find_element(By.CLASS_NAME,"suggestor-input")
designation.send_keys('Data Scientist')
```

In [47]:
```
location=driver.find_element(By.XPATH,"/html/body/div[1]/div[7]/div/div/div[5]/div/div/div/div[1]/div/input")
location.send_keys('Bangalore')
```

In [48]:
```
search=driver.find_element(By.CLASS_NAME,"qsbSubmit")
search.click()
```

In [34]:
```
job_title=[]
job_location=[]
company_name=[]
```

In [35]:
```
#scrapping 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)

#scrapping job location  from the given page
location_tags=driver.find_elements(By.XPATH,'//span[@class="ellipsis fleft locWdth"]')
for i in title_tags[0:10]:
    location=i.text
    job_location.append(location)

#scrapping company name from the given page
company_tags=driver.find_elements(By.XPATH,'//a[@class="subTitle ellipsis fleft"]')
for i in title_tags[0:10]:
    company=i.text
    company_name.append(company)
```

In [36]:
```
print(len(job_title),len(job_location),len(company_name))
```

```
10 10 10
```

In [37]:
```
import pandas as pd
df=pd.DataFrame({'Title':job_title,'Location':job_location,' company_name': company_name})
df
```

Out[37]:

	Title	Location	company_name
0	Permanent Opportunity - Data Scientist(Snaplog...	Permanent Opportunity - Data Scientist(Snaplog...	Permanent Opportunity - Data Scientist(Snaplog...
1	Analystics & Modeling Specialist	Analystics & Modeling Specialist	Analystics & Modeling Specialist
2	Machine Learning (AI) Architect	Machine Learning (AI) Architect	Machine Learning (AI) Architect
3	Staff Data Scientist	Staff Data Scientist	Staff Data Scientist
4	Data Scientist	Data Scientist	Data Scientist
5	Hiring For Data Scientist	Hiring For Data Scientist	Hiring For Data Scientist
6	Data Scientist	Data Scientist	Data Scientist
7	Director/Senior Director - Data Science	Director/Senior Director - Data Science	Director/Senior Director - Data Science
8	Manager/Senior Manager - Data Science	Manager/Senior Manager - Data Science	Manager/Senior Manager - Data Science
9	Data Scientist	Data Scientist	Data Scientist

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

In [50]:
```
filter1_location=driver.find_element(By.XPATH,"/html/body/div[1]/div[4]/div/div/section[1]/div[2]/div[13]/div[2
filter1_location.click()
```

In [52]:
```
filter2_salary=driver.find_element(By.XPATH,"/html/body/div[1]/div[4]/div/div/section[1]/div[2]/div[6]/div[2]/d
filter2_salary.click()
```

In [53]:
```
job_title=[]
job_location=[]
company_name=[]
experience_required=[]
```

In [58]:
```
#scrapping 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)

#scrapping job location  from the given page
location_tags=driver.find_elements(By.XPATH,'//span[@class="ellipsis fleft locWdth"]')
for i in title_tags[0:10]:
    location=i.text
    job_location.append(location)

#scrapping company name from the given page
company_tags=driver.find_elements(By.XPATH,'//a[@class="subTitle ellipsis fleft"]')
for i in title_tags[0:10]:
    company=i.text
    company_name.append(company)
#scrapping job experience from the given page
experience_tags=driver.find_elements(By.XPATH,'//span[@class="ellipsis fleft expwdth"]')
for i in experience_tags[0:10]:
    exp=i.text
    experience_required.append(exp)
```

In [29]:
```
print(len(job_title),len(job_location),len(company_name),len(experience_required))
```

```
10 10 10 10
```

In [59]:
```
import pandas as pd
df=pd.DataFrame({'Title':job_title,'Location':job_location,' company_name': company_name,'Experience': experien
df
```

Out[59]:

	Title	Location	company_name	Experience
0	Junior Data Scientist	Junior Data Scientist	Junior Data Scientist	0-2 Yrs
1	Data Scientist	Data Scientist	Data Scientist	3-7 Yrs
2	Data Scientist	Data Scientist	Data Scientist	3-8 Yrs
3	Data Scientist	Data Scientist	Data Scientist	2-4 Yrs
4	Python and ML Trainer	Python and ML Trainer	Python and ML Trainer	3-8 Yrs
5	Data Scientist	Data Scientist	Data Scientist	7-12 Yrs
6	Intern	Intern	Intern	0-1 Yrs
7	Lead Assistant Manager	Lead Assistant Manager	Lead Assistant Manager	2-6 Yrs
8	Data Scientist	Data Scientist	Data Scientist	2-4 Yrs
9	Junior Data Scientist	Junior Data Scientist	Junior Data Scientist	1-6 Yrs
10	Junior Data Scientist	Junior Data Scientist	Junior Data Scientist	0-2 Yrs
11	Data Scientist	Data Scientist	Data Scientist	3-7 Yrs
12	Data Scientist	Data Scientist	Data Scientist	3-8 Yrs
13	Data Scientist	Data Scientist	Data Scientist	2-4 Yrs
14	Python and ML Trainer	Python and ML Trainer	Python and ML Trainer	3-8 Yrs
15	Data Scientist	Data Scientist	Data Scientist	7-12 Yrs
16	Intern	Intern	Intern	0-1 Yrs
17	Lead Assistant Manager	Lead Assistant Manager	Lead Assistant Manager	2-6 Yrs
18	Data Scientist	Data Scientist	Data Scientist	2-4 Yrs
19	Junior Data Scientist	Junior Data Scientist	Junior Data Scientist	1-6 Yrs
20	Junior Data Scientist	Junior Data Scientist	Junior Data Scientist	0-2 Yrs
21	Data Scientist	Data Scientist	Data Scientist	3-7 Yrs
22	Data Scientist	Data Scientist	Data Scientist	3-8 Yrs
23	Data Scientist	Data Scientist	Data Scientist	2-4 Yrs
24	Python and ML Trainer	Python and ML Trainer	Python and ML Trainer	3-8 Yrs
25	Data Scientist	Data Scientist	Data Scientist	7-12 Yrs
26	Intern	Intern	Intern	0-1 Yrs
27	Lead Assistant Manager	Lead Assistant Manager	Lead Assistant Manager	2-6 Yrs
28	Data Scientist	Data Scientist	Data Scientist	2-4 Yrs
29	Junior Data Scientist	Junior Data Scientist	Junior Data Scientist	1-6 Yrs

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

1. Brand
2. ProductDescription
3. Price

In [47]:
```
#let's first connect to the driver
driver=webdriver.Chrome(r"chromedriver.exe")
```

In [48]:
```
#opening the flipkart page on automated chrome browser
driver.get("https://www.flipkart.com/")
```

In [49]:
```
#closing the Login popup
login=driver.find_element(By.XPATH,"/html/body/div[2]/div/div/button")
login.click()
```

In [50]:
```
#searching for sunglasses
search=driver.find_element(By.CLASS_NAME,"_3704LK")
search.send_keys('sunglasses')
```

In [51]:
```
#clicking on search button
search_btn=driver.find_element(By.CLASS_NAME,"_34RNph")
search_btn.click()
```

In [58]:
```
import time
from selenium.webdriver.common.by import By
```

In [52]:
```
#creating functions to scrape the similar attributes from different pages
```

In [66]:
```
def scrape_brand_name(url):
    brand=[]
    driver.get(url)
```

```
    try:
        brand_el=driver.find_elements(By.XPATH,"//div[@class='_2WkVRV']")
        time.sleep(3)

        for i in brand_el:

            brand.append(i.text)
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")

        return brand
    except StaleElementReferenceException as e:
        print("exception raised",e)
```

In [67]:
```
def scrape_product_des(url):
    product_des=[]
    driver.get(url)
    product_el=driver.find_elements(By.XPATH,"//a[@class='IRpwTa']")
    for i in product_el:
        try:
            product_des.append(i.text)
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")
        except:
            product_des.append("--")

    return product_des
```

In [68]:
```
def scrape_price(url):
    price=[]
    driver.get(url)
    price_el=driver.find_elements(By.XPATH,"//div[@class='_30jeq3']")

    for i in price_el:
        try:
            price.append(i.text)
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")

        except:
            price.append("--")

    return price
```

In [69]:
```
def scrape_discount(url):
    discount=[]
    driver.get(url)

    discount_el=driver.find_elements(By.XPATH,"//div[@class='_3Ay6Sb']/span")

    for i in discount_el:

        try:

            discount.append(i.text)
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")
        except:
            discount.append("--")

    return discount
```

In [70]:
```
import time
```

In [71]:
```
from selenium.common.exceptions import StaleElementReferenceException
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
```

In [72]:
```
brand_list=[]
product_list=[]
discount_list=[]
price_list=[]
urls='https://www.flipkart.com/search?q=sunglasses&otracker=search&otracker1=search&marketplace=FLIPKART&as-sho
for i in ['1','2','3']:

   url= urls+i
   time.sleep(5)
   brand=scrape_brand_name(url)
   brand_list.extend(brand)
   product=scrape_product_des(url)
   product_list.extend(product)
   discount=scrape_discount(url)
   discount_list.extend(discount)
   price=scrape_price(url)
   price_list.extend(price)
```

In [73]:
```
length=[len(brand_list),len(product_list),len(discount_list),len(price_list)]
length
```

Out[73]:
```
[120, 111, 120, 120]
```

In [74]:
```
#creating DataFrame for the scraped Data
import pandas as pd
df5=pd.DataFrame()
df5["BRAND"]=brand_list[:100]
df5["PRODUCT_DESCRIPTION"]=product_list[:100]
df5["DISCOUNT"]=discount_list[:100]
df5["PRICE"]=price_list[:100]
df5
```

Out[74]:

	BRAND	PRODUCT_DESCRIPTION	DISCOUNT	PRICE
0	NuVew	UV Protection Sports Sunglasses (65)	86% off	₹129
1	NuVew	UV Protection Wayfarer Sunglasses (50)	88% off	₹195
2	SRPM	UV Protection Wayfarer Sunglasses (50)	88% off	₹149
3	SRPM	UV Protection Cat-eye, Retro Square, Oval, Rou...	86% off	₹179
4	Elligator	UV Protection Clubmaster Sunglasses (54)	75% off	₹149
...	...	...	...	...
95	ROYAL SON	UV Protection Aviator Sunglasses (58)	20% off	₹319
96	GANSTA	Polarized, UV Protection Sports Sunglasses (68)	56% off	₹873
97	PIRASO	Others Cat-eye, Retro Square, Wayfarer Sunglas...	81% off	₹244
98	METRONAUT	by Lenskart Polarized, UV Protection Aviator S...	65% off	₹349
99	OCHILA	Others Aviator Sunglasses (54)	73% off	₹398

100 rows × 4 columns

Q5: Scrape 100 reviews data from flipkart.com for iphone11 phone. You have to go the link: https://www.flipkart.com/apple-iphone-11-black-64-gb/product-reviews/itm4e5041ba101fd?pid=MOBFWQ6BXGJCEYNY&lid=LSTMOBFWQ6BXGJCEYNYZXSHRJ&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.

In [114…
```
#let's first connect to the driver
driver=webdriver.Chrome(r"chromedriver.exe")
```

In [115…
```
#opening the flipkart url on automated chrome browser
driver.get("https://www.flipkart.com/apple-iphone-11-black-64-gb/product-reviews/itm4e5041ba101fd?pid=MOBFWQ6BX
```

In [116…
```
driver.maximize_window()
```

In [ ]:
```
#defining functions to scrape data
```

In [108…
```
def scrape_rating():
    rating=[]
    driver.refresh()
    rating_el=driver.find_elements(By.XPATH,"//div[@class='_3LWZlK _1BLPMq']")
    time.sleep(3)
    for i in rating_el:
        try:
            rating.append(i.text)
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")
        except:
            rating.append("--")
    return rating
```

In [117…
```
def scrape_review_summary():
    review_sum=[]
    driver.refresh()
    rev_sum=driver.find_elements(By.XPATH,"//p[@class='_2-N8zT']")
    time.sleep(3)
    for i in rev_sum:
        try:
            review_sum.append(i.text)
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")
        except:
            review_sum.append("--")
    return review_sum
```

In [118…
```
def scrape_full_review():
    full_review=[]
    driver.refresh()
```

```
    rev_el=driver.find_elements(By.XPATH,"//div[@class='t-ZTKy']")
    time.sleep(3)
    for i in rev_el:
        try:
            full_review.append(i.text.replace("\n","  New Line: "))
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")
        except:
            full_review.append("--")
    return full_review
```

In [119…
```
import time
```

In [121…
```
rating=[]
review_sum=[]
full_review=[]
length=len(rating)
while(length<=100):
    driver.refresh()
    rating.extend(scrape_rating())
    review_sum.extend(scrape_review_summary())
    full_review.extend(scrape_full_review())
    time.sleep(5)
    length=len(rating)
    next_btn=driver.find_element(By.XPATH,"//a[@class='_1LKTO3']")
    next_btn.click()

len(review_sum)
```

Out[121]:
```
110
```

In [122…
```
#checking if the data was scraped properly
```

In [123…
```
length_list=[len(full_review),len(rating),len(review_sum)]
length_list
```

Out[123]:
```
[110, 110, 110]
```

In [124…
```
review_sum
```

Out[124]:
```
['Fabulous!',
 'Classy product',
 'Worth every penny',
 'Perfect product!',
 'Good choice',
 'Perfect product!',
 'Highly recommended',
 'Highly recommended',
 'Value-for-money',
 'Perfect product!',
 'Simply awesome',
 'Perfect product!',
 'Best in the market!',
 'Highly recommended',
 'Value-for-money',
 'Worth every penny',
 'Pretty good',
 'Perfect product!',
 'Highly recommended',
 'Great product',
 'Fabulous!',
 'Classy product',
 'Worth every penny',
 'Perfect product!',
 'Good choice',
 'Perfect product!',
 'Highly recommended',
 'Highly recommended',
 'Value-for-money',
 'Perfect product!',
 'Simply awesome',
 'Perfect product!',
 'Best in the market!',
 'Highly recommended',
 'Value-for-money',
 'Worth every penny',
 'Pretty good',
 'Perfect product!',
 'Highly recommended',
 'Great product',
 'Fabulous!',
 'Classy product',
 'Worth every penny',
 'Perfect product!',
 'Good choice',
 'Perfect product!',
 'Highly recommended',
 'Highly recommended',
```

```
 'Value-for-money',
 'Perfect product!',
 'Simply awesome',
 'Perfect product!',
 'Best in the market!',
 'Highly recommended',
 'Value-for-money',
 'Worth every penny',
 'Pretty good',
 'Perfect product!',
 'Highly recommended',
 'Great product',
 'Fabulous!',
 'Classy product',
 'Worth every penny',
 'Perfect product!',
 'Good choice',
 'Perfect product!',
 'Highly recommended',
 'Highly recommended',
 'Value-for-money',
 'Perfect product!',
 'Simply awesome',
 'Perfect product!',
 'Best in the market!',
 'Highly recommended',
 'Value-for-money',
 'Worth every penny',
 'Pretty good',
 'Perfect product!',
 'Highly recommended',
 'Great product',
 'Fabulous!',
 'Classy product',
 'Worth every penny',
 'Perfect product!',
 'Good choice',
 'Perfect product!',
 'Highly recommended',
 'Highly recommended',
 'Value-for-money',
 'Perfect product!',
 'Simply awesome',
 'Perfect product!',
 'Best in the market!',
 'Highly recommended',
 'Value-for-money',
 'Worth every penny',
 'Pretty good',
 'Perfect product!',
 'Highly recommended',
 'Great product',
 'Fabulous!',
 'Classy product',
 'Worth every penny',
 'Perfect product!',
 'Good choice',
 'Perfect product!',
 'Highly recommended',
 'Highly recommended',
 'Value-for-money',
 'Perfect product!']
```

In [125…

```
#creating a DataFrame for the scraped data
import pandas as pd
df6=pd.DataFrame()
df6["INDEX"]=range(1,101)
df6["RATING"]=rating[:100]
df6["REVIEW_SUMMARY"]=review_sum[:100]
df6["FULL_REVIEW"]=full_review[:100]
df6.set_index("INDEX",inplace=True)
df6
```

Out[125]:

INDEX	RATING	REVIEW_SUMMARY	FULL_REVIEW
1	5	Fabulous!	This is my first iOS phone. I am very happy wi...
2	5	Classy product	Best and amazing product.....phone looks so pr...
3	5	Worth every penny	i11 is worthy to buy, too much happy with the ...
4	5	Perfect product!	It's a must buy who is looking for an upgrade ...
5	4	Good choice	So far it's been an AMAZING experience coming ...
...	...	...	...
96	5	Worth every penny	Previously I was using one plus 3t it was a gr...
97	4	Pretty good	I was using Iphone 6s and also Oneplus 6t. Bot...
98	5	Perfect product!	Value for money New Line: 5 star rating New ...
99	5	Highly recommended	What a camera .....just awesome ..you can feel...
100	5	Great product	Amazing Powerful and Durable Gadget. New Line...

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. ProductDescription
3. Price

In [141…
```
#let's first connect to the driver
driver=webdriver.Chrome(r"chromedriver.exe")
```

In [142…
```
#opening the flipkart page on automated chrome browser
driver.get("https://www.flipkart.com/")
```

In [145…
```
#closing the Login popup
login=driver.find_element(By.XPATH,"/html/body/div[2]/div/div/button")
login.click()
```

In [143…
```
driver.maximize_window()
```

In [144…
```
#searching for sneakers
search=driver.find_element(By.CLASS_NAME,"_3704LK")
search.send_keys('sneakers')
```

In [146…
```
#clicking on search button
search_btn=driver.find_element(By.CLASS_NAME,"_34RNph")
search_btn.click()
```

In [147…
```
#creating functions to scrape data
import time
```

In [148…
```
def scrape_brand():
    brand=[]
    brand_el=driver.find_elements(By.XPATH,"//div[@class='_2WkVRV']")
    time.sleep(3)
    for i in brand_el:
        try:
            brand.append(i.text)
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")
        except:
            brand.append("--")
    return brand
```

In [149…
```
def scrape_product_des():
    product_des=[]
    des_el=driver.find_elements(By.XPATH,"//a[@class='IRpwTa' or @class='IRpwTa _2-ICcC']")
    print(len(des_el))
    time.sleep(3)
    for i in des_el:
        try:
            product_des.append(i.text)
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")
        except:
            product_des.append("--")
    return product_des
```

In [150…
```
def scrape_price():
    price=[]
    price_el=driver.find_elements(By.XPATH,"//div[@class='_30jeq3']")
```

```
    time.sleep(3)
    for i in price_el:
        try:
            price.append(i.text)
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")
        except:
            price.append("--")

    return price
```

In [138…
```
#scraping data using functions
```

In [153…
```
brand=[]
product_des=[]
price=[]
length=len(brand)
while(length<=100):
    driver.refresh()
    brand.extend(scrape_brand())
    product_des.extend(scrape_product_des())
    price.extend(scrape_price())

    time.sleep(3)
    next_btn=driver.find_element(By.XPATH,"//a[@class='_1LKTO3']")
    next_btn.click()
    length=len(brand)
len(price)
```

```
40
40
40
 120
```

Out[153]:

In [154…
```
#inspecting the length of scraped attributes
length_list=[len(brand),len(product_des),len(price)]
length_list
```

Out[154]:
```
[120, 120, 120]
```

In [155…
```
#creating DataFrame for the scraped data
import pandas as pd
sneakers=pd.DataFrame()
sneakers["INDEX"]=range(1,101)
sneakers["BRAND"]=brand[:100]
sneakers["PRODUCT_DESCRIPTION"]=product_des[:100]
sneakers["PRICE"]=price[:100]
sneakers.set_index("INDEX",inplace=True)
sneakers
```

Out[155]:

INDEX	BRAND	PRODUCT_DESCRIPTION	PRICE
1	Layasa	Sneakers For Men	₹299
2	WHITE WALKERS	Stylish & Trending Outdoor Walking Comfortable...	₹429
3	aadi	Synthetic Leather |Lightweight|Comfort|Summer|...	₹249
4	aadi	Synthetic Leather |Lightweight|Comfort|Summer|...	₹249
5	World Wear Footwear	Latest Exclusive Affordable Collection of Tren...	₹249
...	...	...	...
96	World Wear Footwear	Exclusive Affordable Collection of Trendy & St...	₹249
97	SFR	Casual Shoes For Men Sneakers For Men	₹499
98	Shoe Island	Icon-X White Gold Leather High Top Casual Danc...	₹585
99	World Wear Footwear	Latest Exclusive Affordable Collection of Tren...	₹299
100	PUMA	Player Sneakers For Men	₹1,159

100 rows × 3 columns

Q7: Go to webpage https://www.amazon.in/ Enter “Laptop” in the search field and then click the search icon. Then set CPU Type filter to “Intel Core i7” as shown in the below image: After setting the filters scrape first 10 laptops data. You have to scrape 3 attributes for each laptop:

1. Title
2. Ratings
3. Price

In [186…
```
#let's first connect to the driver
driver=webdriver.Chrome(r"chromedriver.exe")
```

In [187…
```
#opening the amazon page on automated chrome browser
driver.get("https://www.amazon.in/")
```

In [188…
```
driver.maximize_window()
```

In [189…
```
search_col=driver.find_element(By.XPATH,"//input[@class='nav-input nav-progressive-attribute']")
search_col.send_keys('laptop')
```

In [190…
```
search_button=driver.find_element(By.XPATH,"//div[@class='nav-search-submit nav-sprite']")
search_button.click()
```

In [191…
```
title_el=driver.find_elements(By.XPATH,"//h2[@class='a-size-mini a-spacing-none a-color-base s-line-clamp-2']")
title=[]
for i in title_el[:10]:
    title.append(i.text)
title
```

Out[191]:
```
['Lenovo IdeaPad Slim 3 Intel Core i5 12th Gen 15.6" (39.62cm) FHD Thin & Light Laptop (8GB/512GB SSD/Windows
11/Office 2021/Backlit/2Yr Warranty/3months Game Pass/Arctic Grey/1.63Kg), 82RK0062IN',
 'Dell Vostro 14-inch Laptop (35.56 cms) | Windows 11 and MS Office 2021 | Intel i3-1115G4 | 8GB DDR4 SDRAM an
d 512GB SSD | FHD Screen | Laptop for Work | 3420 - Carbon Black (D552276WIN9BE)',
 'Lenovo V15 Intel Celeron N4500 15.6" (39.62 cm) FHD (1920x1080) Antiglare 250 Nits Thin and Light Laptop (8G
B RAM/256GB SSD/Windows 11 Home/Black/1Y Onsite/1.7 kg), 82QYA00MIN',
 'Lenovo IdeaPad Slim 3 Intel Core i3-1115G4 11th Gen 15.6" (39.62cm) FHD Laptop (8GB/256GB SSD/Win 11/Office
2021/2 Year Warranty/3 Month Game Pass/Platinum Grey/1.7Kg), 81X800LCIN',
 'HP 15s, Ryzen 5-5500U, 16GB RAM/512GB SSD 15.6-inches(39.6 cm) FHD, Micro-Edge, Anti-Glare Display/Alexa Bui
lt-in/Windows 11 /AMD Radeon Graphics/Dual Speakers/MS Office 2021/1.69 Kg, 15s-eq2182AU',
 'Dell Vostro 3420 Laptop, Intel Core i5-1135G7, 16GB/512GB SSD/14.0" (35.54Cms) FHD WVA AG/Win 11 + MSO\'21,
15 Month McAfee, Carbon Black, 1.48Kgs',
 'ASUS VivoBook 15 (2021), 15.6-inch (39.62 cm) HD, Dual Core Intel Celeron N4020, Thin and Light Laptop (4GB
RAM/256GB SSD/Integrated Graphics/Windows 11 Home/Transparent Silver/1.8 Kg), X515MA-BR011W',
 'HP 15s,11th Gen Intel Core i3-1115G4 8GB RAM/512GB SSD 15.6-inch(39.6 cm) Micro-Edge Anti-Glare FHD Laptop/A
lexa Built-in/Win 11/Intel UHD Graphics/Dual Speakers/MS Office 2021/1.69 Kg, 15s-fq2673TU',
 'Acer Aspire 3 Intel Core i5 12th Generation (16GB/512 GB SSD/Windows 11 Home/MS Office/1.7 Kg/Silver) A315-5
9 with 15.6-inch (39.6 cms) Full HD Laptop',
 'Dell Vostro 14-inch Laptop (35.56 cms) | Windows 11 and MS Office 2021 | Intel i3-1115G4 | 8GB DDR4 SDRAM an
d 512GB SSD | FHD Screen | Laptop for Work | 3420 - Carbon Black (D552276WIN9BE)']
```

In [192…
```
price_el=driver.find_elements(By.XPATH,"//span[@class='a-price-whole']")
price=[]
for i in price_el[:10]:
    price.append(i.text)
price
```

Out[192]:
```
['55,400',
 '39,990',
 '23,490',
 '33,990',
 '49,990',
 '69,990',
 '54,990',
 '25,990',
 '40,990',
 '50,990']
```

In [193…
```
url2=driver.find_elements(By.XPATH,"//a[@class='a-link-normal a-text-normal']")

rating_url=[]
for i in url2[:10]:
    rating_url.append(i.get_attribute('href'))
rating_url
```

Out[193]:
```
['https://aax-eu.amazon.in/x/c/RFlg8ncOlno6_zP46J9Dqw0AAAGIcSwM4wMAAAH2AQBvbm9fdHhuX2JpZDEgICBvbm9fdHhuX2ltcDE
gICCjLeXP/https://www.amazon.in/dp/B0BDL2DPSS?pd_rd_i=B0BDL2DPSS&pf_rd_p=ea7f11f8-e26c-4b10-84ad-15aa33ea30ac&
pf_rd_r=WBKE7PM84FHDQF9EA0K3&pd_rd_wg=zpNDi&pd_rd_w=AWZEr&pd_rd_r=e6fb0bca-a18d-490e-bc2a-f12a8f2214ca',
 'https://aax-eu.amazon.in/x/c/RLAQps0OZgmQL496htK6sA8AAAGIcSwM3gMAAAH2AQBvbm9fdHhuX2JpZDEgICBvbm9fdHhuX2ltcDE
gICDc6PbC/https://www.amazon.in/dp/B0B3BLY13H?pd_rd_i=B0B3BLY13H&pd_rd_plhdr=t&pd_rd_plhdr=t',
 'https://aax-eu.amazon.in/x/c/RLAQps0OZgmQL496htK6sA8AAAGIcSwM3gMAAAH2AQBvbm9fdHhuX2JpZDEgICBvbm9fdHhuX2ltcDE
gICDc6PbC/https://www.amazon.in/dp/B0B3BLY13H?pd_rd_i=B0B3BLY13H&pd_rd_plhdr=t&pd_rd_plhdr=t',
 'https://aax-eu.amazon.in/x/c/RNyPvw5aQE6BHV9UY0UjLgoAAAGIcSwM3gMAAAH2AQBvbm9fdHhuX2JpZDEgICBvbm9fdHhuX2ltcDE
gICB4ifKx/https://www.amazon.in/dp/B0B2RBP83P?pd_rd_i=B0B2RBP83P&pd_rd_plhdr=t&pd_rd_plhdr=t',
 'https://aax-eu.amazon.in/x/c/RNyPvw5aQE6BHV9UY0UjLgoAAAGIcSwM3gMAAAH2AQBvbm9fdHhuX2JpZDEgICBvbm9fdHhuX2ltcDE
gICB4ifKx/https://www.amazon.in/dp/B0B2RBP83P?pd_rd_i=B0B2RBP83P&pd_rd_plhdr=t&pd_rd_plhdr=t',
 'https://aax-eu.amazon.in/x/c/RPqMUhgrxsQyTKAnBvSWNzoAAAGIcSwM3gMAAAH2AQBvbm9fdHhuX2JpZDEgICBvbm9fdHhuX2ltcDE
gICCZaC5e/https://www.amazon.in/dp/B0BDL387YQ?pd_rd_i=B0BDL387YQ&pd_rd_plhdr=t&pd_rd_plhdr=t',
 'https://aax-eu.amazon.in/x/c/RPqMUhgrxsQyTKAnBvSWNzoAAAGIcSwM3gMAAAH2AQBvbm9fdHhuX2JpZDEgICBvbm9fdHhuX2ltcDE
gICCZaC5e/https://www.amazon.in/dp/B0BDL387YQ?pd_rd_i=B0BDL387YQ&pd_rd_plhdr=t&pd_rd_plhdr=t',
 'https://aax-eu.amazon.in/x/c/RISm2B8NBzQwQxCcM1FaWvQAAAGIcSwM3gMAAAH2AQBvbm9fdHhuX2JpZDEgICBvbm9fdHhuX2ltcDE
gICC8ds8-/https://www.amazon.in/dp/B079J5SSLL?pd_rd_i=B079J5SSLL&pd_rd_plhdr=t&pd_rd_plhdr=t',
 'https://aax-eu.amazon.in/x/c/RISm2B8NBzQwQxCcM1FaWvQAAAGIcSwM3gMAAAH2AQBvbm9fdHhuX2JpZDEgICBvbm9fdHhuX2ltcDE
gICC8ds8-/https://www.amazon.in/dp/B079J5SSLL?pd_rd_i=B079J5SSLL&pd_rd_plhdr=t&pd_rd_plhdr=t',
 'https://aax-eu.amazon.in/x/c/RPF3HizHiAMCawBbDdpJZloAAAGIcSwM3gMAAAH2AQBvbm9fdHhuX2JpZDEgICBvbm9fdHhuX2ltcDE
gICAXxDqN/https://www.amazon.in/dp/B082PKX18C?pd_rd_i=B082PKX18C&pd_rd_plhdr=t&pd_rd_plhdr=t']
```

In [194…
```
import time
rating=[]
for i in rating_url[:10]:
    driver.get(i)
```

```
    time.sleep(3)
    try:

        ratings = driver.find_element(By.XPATH,".//span[@class='a-icon-alt']/..")
        ratings = ratings.get_attribute('innerHTML').split(">")[1].split(" ")[0]
        rating.append(ratings)
    except:


        rating.append("00")
rating
```

Out[194]:
```
['4.3', '4.5', '4.3', '4.3', '4.3', '4.3', '4.3', '4.4', '4.4', '4.3']
```

In [195…
```
len(price)
```

Out[195]:
```
10
```

In [196…
```
len(title)
```

Out[196]:
```
10
```

In [197…
```
#creating DataFrame for the scraped data
import pandas as pd
laptop_df=pd.DataFrame()
laptop_df['INDEX']=range(1,11)
laptop_df['TITLE']=title[:10]
laptop_df['RATING']=rating[:10]
laptop_df['PRICE']=price[:10]
laptop_df.set_index('INDEX',inplace=True)
laptop_df
```

Out[197]:

INDEX	TITLE	RATING	PRICE
1	Lenovo IdeaPad Slim 3 Intel Core i5 12th Gen 1...	4.3	55,400
2	Dell Vostro 14-inch Laptop (35.56 cms) | Windo...	4.5	39,990
3	Lenovo V15 Intel Celeron N4500 15.6" (39.62 cm...	4.3	23,490
4	Lenovo IdeaPad Slim 3 Intel Core i3-1115G4 11t...	4.3	33,990
5	HP 15s, Ryzen 5-5500U, 16GB RAM/512GB SSD 15.6...	4.3	49,990
6	Dell Vostro 3420 Laptop, Intel Core i5-1135G7,...	4.3	69,990
7	ASUS VivoBook 15 (2021), 15.6-inch (39.62 cm) ...	4.3	54,990
8	HP 15s,11th Gen Intel Core i3-1115G4 8GB RAM/5...	4.4	25,990
9	Acer Aspire 3 Intel Core i5 12th Generation (1...	4.4	40,990
10	Dell Vostro 14-inch Laptop (35.56 cms) | Windo...	4.3	50,990

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 TopQuotes
3. Than scrap a) Quote b) Author c) Type Of Quotes

In [249…
```
#let's first connect to the driver
driver=webdriver.Chrome(r"chromedriver.exe")
```

In [250…
```
#opening the A-Z quotes page on automated chrome browser
driver.get("https://www.azquotes.com/")
```

In [251…
```
driver.maximize_window()
```

In [253…
```
top_quotes=driver.find_element(By.XPATH,"/html/body/div[1]/div[1]/div[1]/div/div[3]/ul/li[5]/a")
top_quotes.click()
```

In [254…
```
#creating functions to scrape data
import time
```

In [255…
```
def scrape_quote():
    quote=[]
    quote_el=driver.find_elements(By.XPATH,"//a[@class='title']")
    time.sleep(3)
    for i in quote_el:
        try:
            quote.append(i.text)
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")
        except:
            quote.append("--")
```

```
    return quote
```

In [256…
```
def scrape_author():
    author=[]
    author_el=driver.find_elements(By.XPATH,"//div[@class='author']")
    print(len(author_el))
    time.sleep(3)
    for i in author_el:
        try:
            author.append(i.text)
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")
        except:
            author.append("--")
    return author
```

In [257…
```
def scrape_typeofquote():
    quotetype=[]
    quotetype_el=driver.find_elements(By.XPATH,"//div[@class='tags']")
    time.sleep(3)
    for i in  quotetype_el:
        try:
            quotetype.append(i.text)
            driver.execute_script("window.scrollBy(0,document.body.scrollHeight0)")
        except:
            price.append("--")

    return quotetype
```

In [258…
```
import time
```

In [259…
```
quote=[]
author=[]
quotetype=[]
length=len(author)
while(length<=1000):
    driver.refresh()
    quote.extend(scrape_quote())
    author.extend(scrape_author())
    quotetype.extend(scrape_typeofquote())
    time.sleep(3)
    length=len(author)
    next_btn=driver.find_element(By.XPATH,"//li[@class='next']")
    next_btn.click()
len(quote)
```

```
100
100
100
100
100
100
100
100
100
100
```

```
---------------------------------------------------------------------------
NoSuchElementException                    Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_6340\703435462.py in <module>
     10     time.sleep(3)
     11     length=len(author)
---> 12     next_btn=driver.find_element(By.XPATH,"//li[@class='next']")
     13     next_btn.click()
     14 len(quote)

~\anaconda3\lib\site-packages\selenium\webdriver\remote\webdriver.py in find_element(self, by, value)
    829             value = f'[name="{value}"]'
    830
--> 831         return self.execute(Command.FIND_ELEMENT, {"using": by, "value": value})["value"]
    832
    833     def find_elements(self, by=By.ID, value: Optional[str] = None) -> List[WebElement]:

~\anaconda3\lib\site-packages\selenium\webdriver\remote\webdriver.py in 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

~\anaconda3\lib\site-packages\selenium\webdriver\remote\errorhandler.py in 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":"xpath","selector":"//li[
@class='next']"}
  (Session info: chrome=113.0.5672.127)

Stacktrace:
Backtrace:
        GetHandleVerifier [0x004D8893+48451]
        (No symbol) [0x0046B8A1]
        (No symbol) [0x00375058]
        (No symbol) [0x003A0467]
        (No symbol) [0x003A069B]
        (No symbol) [0x003CDD92]
        (No symbol) [0x003BA304]
        (No symbol) [0x003CC482]
        (No symbol) [0x003BA0B6]
        (No symbol) [0x00397E08]
        (No symbol) [0x00398F2D]
        GetHandleVerifier [0x00738E3A+2540266]
        GetHandleVerifier [0x00778959+2801161]
        GetHandleVerifier [0x0077295C+2776588]
        GetHandleVerifier [0x00562280+612144]
        (No symbol) [0x00474F6C]
        (No symbol) [0x004711D8]
        (No symbol) [0x004712BB]
        (No symbol) [0x00464857]
        BaseThreadInitThunk [0x76A400C9+25]
        RtlGetAppContainerNamedObjectPath [0x775E7B4E+286]
        RtlGetAppContainerNamedObjectPath [0x775E7B1E+238]
```

In [260…

```
length_list=[len(quote),len(author),len(quotetype)]
length_list
```

Out[260]: [1000, 1000, 1000]

In [261…

```
#creating DataFrame for the scraped data
import pandas as pd
top_1000_authors=pd.DataFrame()
top_1000_authors["INDEX"]=range(1,1001)
top_1000_authors["QUOTE"]=quote[:1000]
top_1000_authors["AUTHOR"]=author[:1000]
top_1000_authors["QUOTE_TYPE"]=quotetype[:1000]
top_1000_authors.set_index("INDEX",inplace=True)
top_1000_authors
```

Out[261]:

INDEX	QUOTE	AUTHOR	QUOTE_TYPE
1	The essence of strategy is choosing what not t...	Michael Porter	Essence, Deep Thought, Transcendentalism
2	One cannot and must not try to erase the past ...	Golda Meir	Inspiration, Past, Trying
3	Patriotism means to stand by the country. It d...	Theodore Roosevelt	Country, Peace, War
4	Death is something inevitable. When a man has ...	Nelson Mandela	Inspirational, Motivational, Death
5	You have to love a nation that celebrates its ...	Erma Bombeck	4th Of July, Food, Patriotic
...	...	...	...
996	Regret for the things we did can be tempered b...	Sydney J. Harris	Love, Inspirational, Motivational
997	America... just a nation of two hundred millio...	Hunter S. Thompson	Gun, Two, Qualms About
998	For every disciplined effort there is a multip...	Jim Rohn	Inspirational, Greatness, Best Effort
999	The spiritual journey is individual, highly pe...	Ram Dass	Spiritual, Truth, Yoga
1000	The mind is not a vessel to be filled but a fi...	Plutarch	Inspirational, Leadership, Education

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/.

In [14]:
```
#let's first connect to the driver
driver=webdriver.Chrome(r"chromedriver.exe")
```

In [15]:
```
#opening the jagranjosh page on automated chrome browser
driver.get("https://www.jagranjosh.com/")
```

In [16]:
```
driver.maximize_window()
```

In [17]:
```
#clicking on GK button
GK=driver.find_element(By.XPATH,"/html/body/div[1]/div[1]/div/div[1]/div/div[5]/div/div[1]/header/div[3]/ul/li[
GK.click()
```

In [18]:
```
#clicking the list of all prime misnisters in India link
PM_link=driver.find_element(By.XPATH,"/html/body/div[1]/div/div/div[2]/div/div[10]/div/div/ul/li[2]/a")
PM_link.click()
```

In [30]:
```
# name_el=driver.find_element(By.XPATH,"//div[@class='table-box']/table/tbody/tr")
# for i in name_el[0:21]:
#     name_el.text
name_el=driver.find_elements(By.XPATH,"//div[@class='table-box'][1]/table/tbody/tr")
name=[]
for i in name_el[:21]:
    name.append(i.text.replace('\n','  '))
name
```

Out[30]:
```
['S.N.  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 minist
er 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 slog
an 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 Minis
ter of India',
 '6.  Morarji Desai  (1896–1995)  24 March 1977 to  28 July 1979  2 year, 126 days  Oldest to become PM (81 ye
ars 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 Parlia
ment',
 '8.  Indira Gandhi  (1917–1984)  14 January 1980 to 31 October 1984  4 years, 291 days  The first lady who ser
ved 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 (4
0 years old)',
 '10.  V. P. Singh  (1931–2008)  2 December 1989 to 10 November 1990  343 days  First PM to step down after a v
ote of no confidence',
 '11.  Chandra Shekhar  (1927–2007)  10 November 1990 to 21 June 1991  223 days  He belongs to  Samajwadi Janat
a Party',
 '12.  P. V. Narasimha Rao  (1921–2004)  21 June 1991 to 16 May 1996  4 years, 330 days  First PM from South In
dia',
 '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-congr
ess 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 t
enures',
 '19.  Narendra Modi  (born 1950)  30 May 2019- Incumbent  First non-congress PM with two consecutive tenures']
```

In [35]:
```
import pandas as pd
```

In [35]:
```
import pandas as pd
top_PM=pd.DataFrame()
top_PM["INDEX"]=range(0,20)
top_PM["PM_LIST"]=name[:21]
top_PM.set_index("INDEX",inplace=True)
top_PM
```

Out[35]:

INDEX	PM_LIST
0	S.N. Name Born-Dead Term of office Remark
1	1. Jawahar Lal Nehru (1889–1964) 15 August ...
2	2. Gulzarilal Nanda (Acting) (1898-1998) 27...
3	3. Lal Bahadur Shastri (1904–1966) 9 June 1...
4	4. Gulzari Lal Nanda (Acting) (1898-1998) ...
5	5. Indira Gandhi (1917–1984) 24 January 196...
6	6. Morarji Desai (1896–1995) 24 March 1977 ...
7	7. Charan Singh (1902–1987) 28 July 1979 to...
8	8. Indira Gandhi (1917–1984) 14 January 198...
9	9. Rajiv Gandhi (1944–1991) 31 October 1984...
10	10. V. P. Singh (1931–2008) 2 December 1989...
11	11. Chandra Shekhar (1927–2007) 10 November...
12	12. P. V. Narasimha Rao (1921–2004) 21 June...
13	13. Atal Bihari Vajpayee (1924- 2018) 16 Ma...
14	14. H. D. Deve Gowda (born 1933) 1 June 199...
15	15. Inder Kumar Gujral (1919–2012) 21 April...
16	16. Atal Bihari Vajpayee (1924-2018) 19 Mar...
17	17. Manmohan Singh (born 1932) 22 May 2004 ...
18	18. Narendra Modi (born 1950) 26 May 2014 -...
19	19. Narendra Modi (born 1950) 30 May 2019- ...

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/

In [2]:
```
#let's first connect to the driver
driver=webdriver.Chrome(r"chromedriver.exe")
```

In [3]:
```
#opening the moor1.com page on automated chrome browser
driver.get("https://www.motor1.com/")
```

In [4]:
```
driver.maximize_window()
```

In [5]:
```
search=driver.find_element(By.XPATH,"//input[@class='m1-search-panel-input m1-search-form-text']")
search.send_keys('50 most expensive cars')
```

In [6]:
```
search_btn=driver.find_element(By.XPATH,"//button[@class='m1-search-panel-button m1-search-form-button-animate
search_btn.click()
```

In [7]:
```
#clicking on the link 50 most expensive cars
link=driver.find_element(By.XPATH,"/html/body/div[3]/div[9]/div/div[1]/div/div/div[2]/div/div[1]/h3/a").click()
```

In [9]:
```
price_el=driver.find_elements(By.XPATH,"/html/body/div[3]/div[7]/div[2]/div[1]/div[2]/div[1]/p/strong")
price=[]
for i in price_el[:50]:
    price.append(i.text)
price
```

Out[9]:
```
['Price: $1.3 Million',
 'Price: $1.4 Million',
 'Price: $1.4 Million',
 '',
 'Price: $1.7 Million',
 'Price: $1.7 Million',
 'Price: $1.7 Million',
 'Price: $1.7 Million',
 'Price: $1.7 Million',
 'Price: $1.7 Million',
 'Price: $1.8 Million',
 'Price: $1.9 Million',
 'Price: $1.9 Million',
 'Price: $2.0 Million',
 'Price: $2.0 Million',
 'Price: $2.0 Million*',
 'Price: $2.1 Million',
 'Price: $2.3 Million',
 'Price: $2.3 Million',
 'Price: $2.3 Million',
 'Price: $2.4 Million',
 'Price: $2.5 Million',
 'Price: $2.5 Million',
 'Price: $2.6 Million',
 'Price: $2.6 Million',
 'Price: $2.6 Million',
 'Price: $2.7 Million',
 'Price: $3.0 Million',
 '$3.0 Million',
 'Price: $3.0 Million',
 'Price: $3.2 Million',
 'Price: $3.4 Million',
 '$3.5 Million',
 'Price: $3.5 Million',
 'Price: $3.6 Million',
 'Price: $3.6 million',
 'Price: $3.7 Million',
 'Price: $3.9 Million',
 'Price: $4.5 Million',
 'Price: $4.7 Million',
 'Price: $5.0 Million',
 'Price: $5.4 Million',
 'Price: $5.8 Million',
 'Price: $6.4 Million',
 'Price: $7.4 Million',
 'Price: $8.0 Million',
 'Price: $9.0 Million',
 'Price: $10.8 Million',
 'Price: $12.8 Million',
 'Price: $13.4 Million']
```

In [11]:
```
cars_el=driver.find_elements(By.XPATH,"//h3[@class='subheader']")
cars=[]
for i in cars_el[:50]:
    cars.append(i.text)
cars
```

Out[11]:
```
['De Tomaso P72',
 'Ferrari LaFerrari',
 'Pagani Huayra',
 'McLaren Elva',
 'Czinger 21C',
 'Ferrari Monza',
 'Gordon Murray T.33',
 'Koenigsegg Gemera',
 'Zenvo TSR-S',
 'Hennessey Venom F5',
 'Bentley Bacalar',
 'Hispano Suiza Carmen Boulogne',
 'Bentley Mulliner Batur',
 'Deus Vayanne',
 'SSC Tuatara',
 'Lotus Evija',
 'Aston Martin Vulcan',
 'Delage D12',
 'McLaren Speedtail',
 'Rimac Nevera',
 'Pagani Utopia',
 'Pininfarina Battista',
 'Ferrari FXX K Evo',
 'Gordon Murray T.50',
 'Lamborghini Countach',
 'Mercedes-AMG Project One',
 'Aston Martin Victor',
 'Hennessey Venom F5 Roadster',
 'Koenigsegg Jesko',
 'Aston Martin Valkyrie',
 'W Motors Lykan Hypersport',
 'McLaren Solus',
 'Pagani Huayra Roadster BC',
 'Bugatti Chiron Pur Sport',
 'Lamborghini Sian',
 'Koenigsegg CC850',
 'Bugatti Chiron Super Sport 300+',
 'Lamborghini Veneno',
 'Bugatti Bolide',
 'Bugatti Mistral',
 'Pagani Huayra Imola',
 'Bugatti Divo',
 'SP Automotive Chaos',
 'Pagani Codalunga',
 'Mercedes-Maybach Exelero',
 'Bugatti Centodieci',
 'Bugatti Chiron Profilée',
 'Rolls-Royce Sweptail',
 'Bugatti La Voiture Noire',
 'Rolls-Royce Boat Tail*']
```

In [12]:
```python
#creating DataFrame for the scraped data
import pandas as pd
top_50_expensive_cars=pd.DataFrame()
top_50_expensive_cars["INDEX"]=range(1,51)
top_50_expensive_cars["CARS"]=cars[:51]
top_50_expensive_cars["PRICE"]=price[:51]
top_50_expensive_cars.set_index("INDEX",inplace=True)
top_50_expensive_cars
```

Out[12]:

INDEX	CARS	PRICE
1	De Tomaso P72	Price: $1.3 Million
2	Ferrari LaFerrari	Price: $1.4 Million
3	Pagani Huayra	Price: $1.4 Million
4	McLaren Elva	
5	Czinger 21C	Price: $1.7 Million
6	Ferrari Monza	Price: $1.7 Million
7	Gordon Murray T.33	Price: $1.7 Million
8	Koenigsegg Gemera	Price: $1.7 Million
9	Zenvo TSR-S	Price: $1.7 Million
10	Hennessey Venom F5	Price: $1.7 Million
11	Bentley Bacalar	Price: $1.8 Million
12	Hispano Suiza Carmen Boulogne	Price: $1.9 Million
13	Bentley Mulliner Batur	Price: $1.9 Million
14	Deus Vayanne	Price: $2.0 Million
15	SSC Tuatara	Price: $2.0 Million
16	Lotus Evija	Price: $2.0 Million*
17	Aston Martin Vulcan	Price: $2.1 Million
18	Delage D12	Price: $2.3 Million
19	McLaren Speedtail	Price: $2.3 Million
20	Rimac Nevera	Price: $2.3 Million
21	Pagani Utopia	Price: $2.4 Million
22	Pininfarina Battista	Price: $2.5 Million
23	Ferrari FXX K Evo	Price: $2.5 Million
24	Gordon Murray T.50	Price: $2.6 Million
25	Lamborghini Countach	Price: $2.6 Million
26	Mercedes-AMG Project One	Price: $2.6 Million
27	Aston Martin Victor	Price: $2.7 Million
28	Hennessey Venom F5 Roadster	Price: $3.0 Million
29	Koenigsegg Jesko	$3.0 Million
30	Aston Martin Valkyrie	Price: $3.0 Million
31	W Motors Lykan Hypersport	Price: $3.2 Million
32	McLaren Solus	Price: $3.4 Million
33	Pagani Huayra Roadster BC	$3.5 Million
34	Bugatti Chiron Pur Sport	Price: $3.5 Million
35	Lamborghini Sian	Price: $3.6 Million
36	Koenigsegg CC850	Price: $3.6 million
37	Bugatti Chiron Super Sport 300+	Price: $3.7 Million
38	Lamborghini Veneno	Price: $3.9 Million
39	Bugatti Bolide	Price: $4.5 Million
40	Bugatti Mistral	Price: $4.7 Million
41	Pagani Huayra Imola	Price: $5.0 Million
42	Bugatti Divo	Price: $5.4 Million
43	SP Automotive Chaos	Price: $5.8 Million
44	Pagani Codalunga	Price: $6.4 Million
45	Mercedes-Maybach Exelero	Price: $7.4 Million
46	Bugatti Centodieci	Price: $8.0 Million
47	Bugatti Chiron Profilée	Price: $9.0 Million
48	Rolls-Royce Sweptail	Price: $10.8 Million
49	Bugatti La Voiture Noire	Price: $12.8 Million
50	Rolls-Royce Boat Tail*	Price: $13.4 Million

In [ ]: