


```
In [1]: !pip install selenium
```

Collecting selenium

Obtaining dependency information for selenium from <https://files.pythonhosted.org/packages/97/e3/fd7272d6d2c49fd49a79a603cb28c8b5a71f8911861b4a0409b3c006a241/selenium-4.17.2-py3-none-any.whl.metadata> (<https://files.pythonhosted.org/packages/97/e3/fd7272d6d2c49fd49a79a603cb28c8b5a71f8911861b4a0409b3c006a241/selenium-4.17.2-py3-none-any.whl.metadata>)

Downloading selenium-4.17.2-py3-none-any.whl.metadata (6.9 kB)

Requirement already satisfied: urllib3[socks]<3,>=1.26 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from selenium) (1.26.16)

Collecting trio~0.17 (from selenium)

Obtaining dependency information for trio~0.17 from <https://files.pythonhosted.org/packages/14/fb/9299cf74953f473a15accfbe2c15218e766bae8c796f2567c83bae03e98/trio-0.24.0-py3-none-any.whl.metadata> (<https://files.pythonhosted.org/packages/14/fb/9299cf74953f473a15accfbe2c15218e766bae8c796f2567c83bae03e98/trio-0.24.0-py3-none-any.whl.metadata>)

Downloading trio-0.24.0-py3-none-any.whl.metadata (4.9 kB)

Collecting trio-websocket~0.9 (from selenium)

Obtaining dependency information for trio-websocket~0.9 from https://files.pythonhosted.org/packages/48/be/a9ae5f50cad5b6f85bd2574c2c923730098530096e170c1ce7452394d7aa/trio_websocket-0.11.1-py3-none-any.whl.metadata (https://files.pythonhosted.org/packages/48/be/a9ae5f50cad5b6f85bd2574c2c923730098530096e170c1ce7452394d7aa/trio_websocket-0.11.1-py3-none-any.whl.metadata)

Downloading trio_websocket-0.11.1-py3-none-any.whl.metadata (4.7 kB)

Requirement already satisfied: certifi>=2021.10.8 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from selenium) (2023.7.22)

Collecting typing_extensions>=4.9.0 (from selenium)

Obtaining dependency information for typing_extensions>=4.9.0 from https://files.pythonhosted.org/packages/b7/f4/6a90020cd2d93349b442bfcb657d0dc91eee65491600b2cb1d388bc98e6b/typing_extensions-4.9.0-py3-none-any.whl.metadata (https://files.pythonhosted.org/packages/b7/f4/6a90020cd2d93349b442bfcb657d0dc91eee65491600b2cb1d388bc98e6b/typing_extensions-4.9.0-py3-none-any.whl.metadata)

Downloading typing_extensions-4.9.0-py3-none-any.whl.metadata (3.0 kB)

Requirement already satisfied: attrs>=20.1.0 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from trio~0.17->selenium) (22.1.0)

Requirement already satisfied: sortedcontainers in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from trio~0.17->selenium) (2.4.0)

Requirement already satisfied: idna in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from trio~0.17->selenium) (3.4)

Collecting outcome (from trio~0.17->selenium)

Obtaining dependency information for outcome from <https://files.pythonhosted.org/packages/55/8b/5ab7257531a5d830fc8000c476e63c935488d74609b50f9384a643ec0a62/outcome-1.3.0.post0-py2.py3-none-any.whl.metadata> (<https://files.pythonhosted.org/packages/55/8b/5ab7257531a5d830fc8000c476e63c935488d74609b50f9384a643ec0a62/outcome-1.3.0.post0-py2.py3-none-any.whl.metadata>)

Downloading outcome-1.3.0.post0-py2.py3-none-any.whl.metadata (2.6 kB)

Collecting sniffio>=1.3.0 (from trio~0.17->selenium)

Downloading sniffio-1.3.0-py3-none-any.whl (10 kB)

Requirement already satisfied: cffi>=1.14 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from trio~0.17->selenium) (1.15.0)

```
=0.17->selenium) (1.15.1)
Collecting wsproto>=0.14 (from trio-websocket~=0.9->selenium)
    Downloading wsproto-1.2.0-py3-none-any.whl (24 kB)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from urllib3[socks]<3,>=1.26->selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from cffi==1.14->trio~0.17->selenium) (2.21)
Collecting h11<1,>=0.9.0 (from wsproto>=0.14->trio-websocket~=0.9->selenium)
    Downloading h11-0.14.0-py3-none-any.whl (58 kB)
       0.0/58.3 kB ? eta -----.
       58.3/58.3 kB 3.0 MB/s eta 0:00:00

Downloading selenium-4.17.2-py3-none-any.whl (9.9 MB)
       0.0/9.9 MB ? eta -----.
       0.3/9.9 MB 6.7 MB/s eta 0:00:02
       0.6/9.9 MB 5.4 MB/s eta 0:00:02
       0.9/9.9 MB 5.5 MB/s eta 0:00:02
       1.1/9.9 MB 5.2 MB/s eta 0:00:02
       1.4/9.9 MB 5.2 MB/s eta 0:00:02
       1.7/9.9 MB 5.4 MB/s eta 0:00:02
       1.9/9.9 MB 5.1 MB/s eta 0:00:02
       2.2/9.9 MB 5.2 MB/s eta 0:00:02
       2.4/9.9 MB 5.2 MB/s eta 0:00:02
       2.8/9.9 MB 5.2 MB/s eta 0:00:02
       2.9/9.9 MB 5.2 MB/s eta 0:00:02
       3.1/9.9 MB 5.1 MB/s eta 0:00:02
       3.4/9.9 MB 5.0 MB/s eta 0:00:02
       3.6/9.9 MB 5.0 MB/s eta 0:00:02
       3.9/9.9 MB 4.9 MB/s eta 0:00:02
       4.2/9.9 MB 5.0 MB/s eta 0:00:02
       4.4/9.9 MB 5.0 MB/s eta 0:00:02
       4.7/9.9 MB 5.0 MB/s eta 0:00:02
       5.0/9.9 MB 4.9 MB/s eta 0:00:02
       5.2/9.9 MB 5.0 MB/s eta 0:00:01
       5.5/9.9 MB 4.9 MB/s eta 0:00:01
       5.7/9.9 MB 4.9 MB/s eta 0:00:01
       5.9/9.9 MB 4.8 MB/s eta 0:00:01
       6.0/9.9 MB 4.8 MB/s eta 0:00:01
       6.3/9.9 MB 4.8 MB/s eta 0:00:01
       6.3/9.9 MB 4.7 MB/s eta 0:00:01
       6.6/9.9 MB 4.6 MB/s eta 0:00:01
       7.0/9.9 MB 4.7 MB/s eta 0:00:01
       7.2/9.9 MB 4.7 MB/s eta 0:00:01
       7.6/9.9 MB 4.7 MB/s eta 0:00:01
```

```
----- 7.8/9.9 MB 4.7 MB/s eta 0:00:01
----- 8.0/9.9 MB 4.8 MB/s eta 0:00:01
----- 8.3/9.9 MB 4.7 MB/s eta 0:00:01
----- 8.4/9.9 MB 4.7 MB/s eta 0:00:01
----- 8.7/9.9 MB 4.7 MB/s eta 0:00:01
----- 9.0/9.9 MB 4.8 MB/s eta 0:00:01
----- 9.3/9.9 MB 4.8 MB/s eta 0:00:01
----- 9.6/9.9 MB 4.8 MB/s eta 0:00:01
----- 9.8/9.9 MB 4.8 MB/s eta 0:00:01
----- 9.9/9.9 MB 4.8 MB/s eta 0:00:01
----- 9.9/9.9 MB 4.8 MB/s eta 0:00:01
----- 9.9/9.9 MB 4.4 MB/s eta 0:00:00
Downloading trio-0.24.0-py3-none-any.whl (460 kB)
----- 0.0/460.2 kB ? eta -----
----- 358.4/460.2 kB 7.4 MB/s eta 0:00:01
----- 460.2/460.2 kB 4.8 MB/s eta 0:00:00
Downloading trio_websocket-0.11.1-py3-none-any.whl (17 kB)
Downloading typing_extensions-4.9.0-py3-none-any.whl (32 kB)
Downloading outcome-1.3.0.post0-py2.py3-none-any.whl (10 kB)
Installing collected packages: typing_extensions, sniffio, outcome, h11, wsproto, trio, trio-websocket, selenium
Attempting uninstall: typing_extensions
  Found existing installation: typing_extensions 4.7.1
  Uninstalling typing_extensions-4.7.1:
    Successfully uninstalled typing_extensions-4.7.1
Attempting uninstall: sniffio
  Found existing installation: sniffio 1.2.0
  Uninstalling sniffio-1.2.0:
    Successfully uninstalled sniffio-1.2.0
Successfully installed h11-0.14.0 outcome-1.3.0.post0 selenium-4.17.2 sniffio-1.3.0 trio-0.24.0 trio-websocket-0.11.1 typing_extensions-4.9.0 wsproto-1.2.0
```

```
In [124]: import selenium
import pandas as pd
from selenium import webdriver
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
```

```
In [ ]: # 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.
# This task will be done in following steps:
# 1. First get the webpage https://www.shine.com/
# 2. Enter "Data Analyst" in "Job title, Skills" field and enter "Bangalore" in "enter the location" field.
# 3. Then click the searchbutton.
# 4. Then scrape the data for the first 10 jobs results you get.
# 5. Finally create a dataframe of the scraped data.
# Note: All of the above steps have to be done in code. No step is to be done manually
```

```
In [125]: #first we need to connect selenium tool to the google chrome
driver = webdriver.Chrome() #after running this code , a blank window will appear automatically

driver.get("https://www.shine.com/") #by this we can access shine.com page automatically
```

```
In [126]: # designation = driver.find_element(By.CLASS_NAME , 'form-control')

#scraping for data analyst by using XPATH
designation = driver.find_element(By.XPATH , '/html/body/div/div[4]/div/div[2]/div[2]/div/form/div/div[1]/ul[1]
designation.send_keys('Data Analyst') #by running this line in automation window in search bar "Data Analyst"
```

```
In [127]: # now fetching location by using Xpath
location = driver.find_element(By.XPATH , '/html/body/div/div[4]/div/div[2]/div[2]/div/form/div/div[1]/ul/li[2]
location.send_keys("Banglore") #Banglore automated in chrome window
```

In []:

```
In [128]: #for searching we use click
#search = driver.find_element(By.CLASS_NAME , 'searchForm_btnWrap_advance__VYBHN') scrap by class name

search = driver.find_element(By.XPATH , '/html/body/div/div[4]/div/div[2]/div[2]/div/form/div/div[2]/div/button')
search.click() #by running this it will search for the given data
```

```
In [129]: #you need to create following details from given page
job_title = []
job_location = []
company_name = []
experience_required = []
```

```
In [131]: title_tags = driver.find_elements(By.XPATH , '//div[@class]/div/div/h2/a') #'//div[@class="jobCard_jobCard__jjU
#title_tags
for i in title_tags :
    #title = i.text
    job_title.append(i.text)
skill = job_title[0:10]
skill
```

```
Out[131]: ['Lead Data Analyst',
'Data Analyst',
'Vacancy For Data Analyst',
'Clinical Data Analyst',
'Data Management',
'Data Modeler data',
'Data Modeller',
'Data Modeler Bangalore',
'Data Modeler',
'Clinical Data Analyst']
```

```
In [132]: location_tags = driver.find_elements(By.XPATH , '//div[@class=" jobCard_jobCard_lists_item__YxRkV jobCard_loca  
# location_tags  
for i in location_tags :  
    job_location.append(i.text)  
loc = job_location[0:10]  
loc
```

Out[132]: ['Bangalore',
'Bangalore\\n+3',
'Canada\\n+14',
'Gurugram\\n+6',
'China\\n+18',
'Bangalore',
'Bangalore',
'Bangalore',
'Bangalore',
'Mumbai City\\n+4']

```
In [94]: #company_tags = driver.find_elements(By.XPATH , '//div[@class="jobCard_jobCard_cName__mYnow"]/span')  
company_tags = driver.find_elements(By.CLASS_NAME , 'jobCard_jobCard_cName__mYnow')  
#company_tags  
for i in company_tags :  
    company_name.append(i.text)  
company = company_name[0:10]  
company
```

Out[94]: ['ara resources private limited',
'diraa hr services hiring for mncs',
'yogita staffing solution',
'techno endura',
'future solution centre',
'boyen haddin consulting and technol...',
'v-tech data outsourcing']

```
In [95]: experience = driver.find_elements(By.XPATH , '//div[@class=" jobCard_jobCard_lists_item__YxRkV jobCard_jobIcon #experience"
for i in experience :
    exp = i.text
    experience_required.append(exp)
expr = experience_required[0:10]
expr
```

```
Out[95]: ['4 to 9 Yrs',
          '0 to 1 Yr',
          '0 to 3 Yrs',
          '0 to 1 Yr',
          '15 to >25 Yrs',
          '3 to 6 Yrs',
          '3 to 6 Yrs',
          '3 to 6 Yrs',
          '3 to 6 Yrs',
          '0 to 1 Yr']
```

```
In [96]: print(len(skill) , len(loc) , len(company) , len(expr))
```

```
10 10 10 10
```

```
In [97]: # import pandas as pd
df = pd.DataFrame({"Title" : skill , "Location" : loc , "Company_Name" : company , "Experience" : expr})
df
# # job_title = []
# # job_location = []
# # company_name = []
# # experience_required = []
```

Out[97]:

	Title	Location	Company_Name	Experience
0	Data Analyst	Bangalore	ara resources private limited	4 to 9 Yrs
1	Vacancy For Data Analyst	Bangalore\n+3	diraa hr services hiring for mncs	0 to 1 Yr
2	Clinical Data Analyst	Canada\n+14	yogita staffing solution	0 to 3 Yrs
3	Data Management	Gurugram\n+6	techno endura	0 to 1 Yr
4	Data Modeler data	China\n+18	future solution centre	15 to >25 Yrs
5	Data Modeller	Bangalore	boyen haddin consulting and technol...	3 to 6 Yrs
6	Data Modeler Bangalore	Bangalore	boyen haddin consulting and technol...	3 to 6 Yrs
7	Data Modeler	Bangalore	boyen haddin consulting and technol...	3 to 6 Yrs
8	Data Engineer II	Bangalore	boyen haddin consulting and technol...	3 to 6 Yrs
9	Full time Opportunity-Networking Advisor-Cisco...	Bangalore\n+9	v-tech data outsourcing	0 to 1 Yr

```
In [ ]: # 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.
# This task will be done in following steps:
# 1. First get the webpage https://www.shine.com/
# 2. Enter “Data Scientist” in “Job title, Skills” field and enter “Bangalore” in “enter thelocation” field.
# 3. Then click the search button.
# 4. Then scrape the data for the first 10 jobs results you get.
# 5. Finally create a dataframe of the scraped data.
# # Note: All of the above steps have to be done in code. No step is to be done manually.
```

```
In [133]: driver = webdriver.Chrome() #for connecting selenium to chrome
driver.get('https://www.shine.com/') #for reaching the shine.com
```

```
In [109]: designation = driver.find_element(By.XPATH , '/html/body/div/div[4]/div/div[2]/div[2]/div/form/div/div[1]/ul/li[1]')  
designation.send_keys("Data Scientist") #after this Data Scientist automatically filled in job title section
```

```
In [110]: location = driver.find_element(By.XPATH , '/html/body/div/div[4]/div/div[2]/div[2]/div/form/div/div[1]/ul/li[2]')  
location.send_keys("Banglore") #after this Bangalore automatically filled in location section
```

```
In [111]: search = driver.find_element(By.XPATH , '/html/body/div/div[4]/div/div[2]/div[2]/div/form/div/div[2]/div/button')  
search.click() #by this window will searching for filled keyword and reflects to the output
```



```
In [123]: job_title = []
job_location = []
company_name = []
experience_required = []

#Now go for scraping job title
job_tags = driver.find_elements(By.XPATH , '//div[@class]/div/div/h2/a')
#job_tags
for i in job_tags :
    job = i.text
    job_title.append(job)

#scraping details for location
location_tags = driver.find_elements(By.XPATH , '//div[@class=" jobCard_jobCard_lists_item__YxRkV jobCard_locat')
for j in location_tags :
    loc = j.text
    job_location.append(loc)

#scraping details for company name
company_tags = driver.find_elements(By.XPATH , '//div[@class="jobCard_jobCard_cName__mYnow"]')
for x in company_tags :
    comp = x.text
    company_name.append(comp)

# scraping details for required experience for the post
experience = driver.find_elements(By.XPATH , '//div[@class=" jobCard_jobCard_lists_item__YxRkV jobCard_jobIcon')
for z in experience :
    exp = z.text
    experience_required.append(exp)

#now creating a dataframe for scraping data
df = pd.DataFrame({
    'Title' : job_title,
    'Location' : job_location,
    'Company_Name' : company_tags,
    'Experience' : experience_required
})
df
```

Out[123]:

	Title	Location	Company_Name	Experience
0	ML Data Scientist	Bangalore\n+3	<selenium.webdriver.remote.webelement.WebElement...>	5 to 8 Yrs
1	Data Scientist Urgent Vacancy	Canada\n+15	<selenium.webdriver.remote.webelement.WebElement...>	0 to 4 Yrs
2	Data Scientist Recruitment	Canada\n+15	<selenium.webdriver.remote.webelement.WebElement...>	0 to 4 Yrs
3	Data Scientist Recruitment	Canada\n+15	<selenium.webdriver.remote.webelement.WebElement...>	0 to 4 Yrs
4	Data Scientist	Gurugram\n+4	<selenium.webdriver.remote.webelement.WebElement...>	3 to 5 Yrs
5	Data Scientist	Bangalore	<selenium.webdriver.remote.webelement.WebElement...>	6 to 8 Yrs
6	Lead Data Scientist/ Principal Data Scientist	Bangalore\n+1	<selenium.webdriver.remote.webelement.WebElement...>	5 to 9 Yrs
7	Vacancy For Data Scientist Fresher and Experience	Canada\n+14	<selenium.webdriver.remote.webelement.WebElement...>	0 to 3 Yrs
8	Senior Data Scientist	Bangalore\n+1	<selenium.webdriver.remote.webelement.WebElement...>	7 to 11 Yrs
9	Lead Data Scientist	Bangalore\n+1	<selenium.webdriver.remote.webelement.WebElement...>	7 to 10 Yrs
10	Data Scientist	Gurugram\n+7	<selenium.webdriver.remote.webelement.WebElement...>	4 to 9 Yrs
11	Data Scientist Urgent Vacancy	Canada\n+14	<selenium.webdriver.remote.webelement.WebElement...>	0 to 3 Yrs
12	Data Scientist	Bangalore\n+2	<selenium.webdriver.remote.webelement.WebElement...>	0 to 4 Yrs
13	Pharma Data Scientist	Mumbai City\n+4	<selenium.webdriver.remote.webelement.WebElement...>	0 to 1 Yr
14	Hiring For Data Scientist	Canada\n+13	<selenium.webdriver.remote.webelement.WebElement...>	0 to 3 Yrs
15	Hiring For Data Scientist	Canada\n+14	<selenium.webdriver.remote.webelement.WebElement...>	0 to 3 Yrs
16	Data Scientist Urgent Vacancy	Oman\n+13	<selenium.webdriver.remote.webelement.WebElement...>	0 to 4 Yrs
17	Data Scientist Recruitment	Oman\n+13	<selenium.webdriver.remote.webelement.WebElement...>	0 to 4 Yrs
18	Data Scientist	Bangalore	<selenium.webdriver.remote.webelement.WebElement...>	1 to 2 Yrs
19	Data Engineer II	Bangalore\n+9	<selenium.webdriver.remote.webelement.WebElement...>	0 to 1 Yr

In [124]: df[0:10]

Out[124]:

	Title	Location	Company_Name	Experience
0	ML Data Scientist	Bangalore\n+3 <selenium.webdriver.remote.webelement.WebElement...>		5 to 8 Yrs
1	Data Scientist Urgent Vacancy	Canada\n+15 <selenium.webdriver.remote.webelement.WebElement...>		0 to 4 Yrs
2	Data Scientist Recruitment	Canada\n+15 <selenium.webdriver.remote.webelement.WebElement...>		0 to 4 Yrs
3	Data Scientist Recruitment	Canada\n+15 <selenium.webdriver.remote.webelement.WebElement...>		0 to 4 Yrs
4	Data Scientist	Gurugram\n+4 <selenium.webdriver.remote.webelement.WebElement...>		3 to 5 Yrs
5	Data Scientist	Bangalore	<selenium.webdriver.remote.webelement.WebElement...>	6 to 8 Yrs
6	Lead Data Scientist/ Principal Data Scientist	Bangalore\n+1 <selenium.webdriver.remote.webelement.WebElement...>		5 to 9 Yrs
7	Vacancy For Data Scientist Fresher and Experience	Canada\n+14 <selenium.webdriver.remote.webelement.WebElement...>		0 to 3 Yrs
8	Senior Data Scientist	Bangalore\n+1 <selenium.webdriver.remote.webelement.WebElement...>		7 to 11 Yrs
9	Lead Data Scientist	Bangalore\n+1 <selenium.webdriver.remote.webelement.WebElement...>		7 to 10 Yrs

In []: # Q3: In this question you have to scrape data using the filters available on the webpage
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 the web page <https://www.shine.com/>
2. Enter "Data Scientist" in "Skill, Designations, and Companies" field.
3. Then click the search button.
4. Then apply the Location filter and salary filter by checking the respective boxes
5. Then scrape the data for the first 10 jobs results you get.
6. Finally create a dataframe of the scraped data.
Note: All of the above steps have to be done in code. No step is to be done manually.

In [159]: driver = webdriver.Chrome()
driver.get('https://www.shine.com/')

```
In [160]: skills = driver.find_element(By.XPATH , '/html/body/div/div[4]/div/div[2]/div/div[2]/div/form/div/div[1]/ul[1]/li[1]')
skills.send_keys("Data Scientist")
```

```
In [162]: location = driver.find_element(By.XPATH , '/html/body/div/div[4]/div/div[2]/div/div[2]/div/form/div/div[1]/ul[1]/li[2]')
location.send_keys("Delhi-NCR")
```

```
In [163]: salary = driver.find_element(By.XPATH , '/html/body/div/div[4]/div/div[2]/div/div[2]/div/form/div/div[1]/ul[2]/li[1]')
salary.send_keys("3-6")
```

```
In [164]: search = driver.find_element(By.XPATH , '/html/body/div/div[4]/div/div[2]/div/div[2]/div/form/div/div[2]/div/button')
search.click()
```

```
In [170]: job_title = []
job_location = []
company_name = []
experience_required = []

job_tags = driver.find_elements(By.XPATH , '//div[@class]/div/h2/a')
#job_tags
for i in job_tags :
    job = i.text
    job_title.append(job)

location = driver.find_elements(By.XPATH , '//div[@class=" jobCard_jobCard_lists_item__YxRkV jobCard_locationI')
#location
for j in location :
    loc = j.text
    job_location.append(loc)

name = driver.find_elements(By.XPATH , '//div[@class="jobCard_jobCard_cName__mYnow"]')
#name
for k in name :
    comp_name = k.text
    company_name.append(comp_name)

experience = driver.find_elements(By.XPATH , '//div[@class=" jobCard_jobCard_lists_item__YxRkV jobCard_jobIcon')
#experience
for x in experience :
    exp = x.text
    experience_required.append(exp)

df = pd.DataFrame({"Title" : job_title , "Location" : job_location , "Company_Name" : company_name , "Experience" : experience_required})
df[0:10]
```

Out[170]:

	Title	Location	Company_Name	Experience
0	Data Scientist	Gurugram\n+1	right advisors private limited	2 to 7 Yrs
1	Data Scientist	Gurugram\n+4	acme services private limited	3 to 5 Yrs
2	Data Scientist	Noida\n+1	thescholarhat	1 to 3 Yrs
3	Data Scientist - Japanese language	Delhi	right advisors private limited	5 to 8 Yrs
4	DataNeuron - Senior Data Scientist - NLP	Delhi	dataneuron	3 to 4 Yrs
5	Data Scientist	Delhi	adept global	3 to 7 Yrs
6	Senior Data Scientist	Delhi	roadzen	5 to 9 Yrs
7	Python Data Scientist	Noida\n+1	buddy4study	4 to 8 Yrs
8	Data Scientist	Delhi	scrolltab	4 to 8 Yrs
9	Senior/Lead Data Scientist- Noida/Bangalore/Hy...	Noida\n+1	techneplus	3 to 8 Yrs

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

```
# 6. Brand
# 7. Product and Description
# 8. Price
# 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 as usual
# 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 [186]: driver = webdriver.Chrome()

driver.get("https://www.flipkart.com/")

```
In [190]: #product = driver.find_element(By.XPATH , '/html/body/div[1]/div/div[1]/div/div/div/div[1]/div/div[1]/div/  
#product = driver.find_element(By.CLASS_NAME , '_2SmNnR')  
product = driver.find_element(By.CLASS_NAME , 'Pke_EE')  
product.send_keys("sunglasses") #sunglasses will automatically filled in search bar , after running this code
```

```
In [192]: search= driver.find_element(By.CLASS_NAME , '_2iLD_')  
search.click()
```

```
In [198]: brand = []
```

```
In [249]: start = 0  
end = 3  
for page in range(start , end) :  
    brand_tags = driver.find_elements(By.XPATH , '//div[@class=" _2WkVRV"] ')  
    for i in brand_tags :  
        title = i.text  
        brand.append(title)  
    next_button = driver.find_element(By.XPATH , '/html/body/div/div/div[3]/div[1]/div[2]/div[12]/div/div/nav/  
next_button.click()  
    #time.sleep(3)  
len(brand)  
  
Title = brand[0:100]  
#Title
```

```
In [ ]: start = 0  
end = 3
```

```
In [251]: description = []
for page in range(start , end):
    des = driver.find_elements(By.XPATH , '//div[@class=" _2B099V"]/a')
    for i in des :
        desc = i.text
        description.append(desc)
next_button = driver.find_element(By.XPATH , '/html/body/div/div/div[3]/div[1]/div[2]/div[12]/div/div/nav/a')
next_button.click()
#len(description)
Description = description[0:100]
```

```
In [252]: price = []
for page in range(start , end) :
    pr = driver.find_elements(By.XPATH , '//div[@class=" _30jeq3"]')
    for i in pr :
        keemat = i.text
        price.append(keemat)
next_button = driver.find_element(By.XPATH , '/html/body/div/div/div[3]/div[1]/div[2]/div[12]/div/div/nav/a')
next_button.click()
#len(price)
Price = price[0:100]
```

```
In [253]: discount = []
for page in range(start , end) :
    dis = driver.find_elements(By.XPATH , '//div[@class=" _3Ay6Sb"]')
    for i in dis:
        disct = i.text
        discount.append(disct)
next_button = driver.find_element(By.XPATH , '/html/body/div/div/div[3]/div[1]/div[2]/div[12]/div/div/nav/a')
next_button.click()
#len(discount)
Discount = discount[0:100]
```

```
In [234]: print(len(brand) , len(description) , len(price) , len(discount))

1040 240 120 120
```

```
In [255]: import pandas as pd
df = pd.DataFrame({
    "Brand_Name" : Title ,
    "Description" : Description ,
    "Price" : Price ,
    "Discount" : Discount
})
df
```

Out[255]:

	Brand_Name	Description	Price	Discount
0	Lenskart STUDIO	Toughened Glass Lens, UV Protection Aviator, W...	₹775	79% off
1	Lenskart STUDIO		₹775₹3,699 79% off	₹1,029 51% off
2	SRPM	UV Protection Aviator Sunglasses (Free Size)	₹187	68% off
3	ROADWAY		₹1,029₹2,111 51% off	₹299 85% off
4	PC STAR	Polarized, UV Protection, Riding Glasses Wayfa...	₹449	43% off
...
95	Elligator		₹795₹3,699 78% off	₹418 72% off
96	kingsunglasses	UV Protection, Gradient Round Sunglasses (51)	₹375	78% off
97	ROYAL SON		₹309₹599 48% off	₹398 73% off
98	Eclizz	Mirrored Aviator Sunglasses (Free Size)	₹759	62% off
99	Fastrack		₹376₹1,499 74% off	₹289 73% off

100 rows × 4 columns

```
In [ ]: # 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/productreviews/item4e5041ba101fd?pid=MOBFWQ6BXGJCEYNY&Li  
# place=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 [24]: driver = webdriver.Chrome()
```

```
In [25]: driver.get('https://www.flipkart.com/apple-iphone-11-black-64-gb/product-reviews/item4e5041ba101fd?pid=MOBFWQ6B
```

```
In [14]: start = 0  
end = 11  
rating = []  
review_sum = []  
full_review = []
```

```
In [44]: for page in range(start , end) :  
    ratings = driver.find_elements(By.XPATH , '//div[@class=" _3LWZlK _1BLPMq"]')  
    for i in ratings :  
        rat = i.text  
        rating.append(rat)  
    #next_button = driver.find_elements(By.XPATH , '//div[@class=" _2MImiq _1Qnn1K"]/nav/a[11]/span')  
    #next_button = driver.find_elements(By.XPATH , '//a[@class=" _1LKTO3"]/span')  
    #next_button = driver.find_elements(By.XPATH , '//div[@class=" _2MImiq _1Qnn1K"]/nav/a[11]/span')  
    #next_button = driver.find_element(By.XPATH , '/html/body/div/div/div[3]/div/div/div[2]/div/div/nav/  
    next_button = driver.find_element(By.XPATH , '/html/body/div/div/div[3]/div/div/div[2]/div/div[13]/div/div/nav/  
    next_button  
  
    #      next_button.click()  
    rat = rating[0:110]  
    len(rat)
```

```
Out[44]: 110
```

In [40]:

Out[40]: 1368

```
In [36]: for page in range(start , end) :
    rev = driver.find_elements(By.XPATH , '//p[@class="_2-N8zT"]')
    for i in rev :
        review_sum.append(i.text)
#review_sum
    next_button = driver.find_element(By.XPATH , '/html/body/div/div/div[3]/div/div/div[2]/div[13]/div/div/nav')
#    next_button = driver.find_elements(By.XPATH , '//div[@class="_2MImiq _1Qnn1K"]/nav/a[11]/span')
    next_button
#len(review_sum)
```

Out[36]: 110

```
In [37]: for page in range(start , end) :
    full_rev = driver.find_elements(By.XPATH , '//div[@class="col _2wzgFH K0kLPL"]/div[2]/div/div/div')
    for i in full_rev :
        full_review.append(i.text)
#        reviews = i.text
#        full_review.append(reviews)
    next_button = driver.find_element(By.XPATH , '/html/body/div/div/div[3]/div/div/div[2]/div[13]/div/div/nav')
    next_button
#len(full_review)
```

Out[37]: 110

```
In [46]: df = pd.DataFrame({
    "RATING" : rat ,
    "REVIEW_SUMMARY" : review_sum ,
    "FULL REVIEW" : full_review
})
df[0:100]
```

Out[46]:

	RATING	REVIEW_SUMMARY	FULL REVIEW
0	5	Perfect product!	V Good all
1	5	Awesome	iPhone 11 is a good phone. Not a very big diff...
2	5	Wonderful	This is amazing at all
3	5	Classy product	Camera is awesome\nBest battery backup\nA perf...
4	5	Terrific	Very very good
...
95	5	Best in the market!	Good Camera
96	5	Worth every penny	Feeling awesome after getting the delivery of ...
97	5	Perfect product!	Photos super
98	5	Terrific purchase	Value for money 😊
99	5	Just wow!	Perfect Product!!

100 rows × 3 columns

In []:

```
In [269]: # 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
# As shown in the below image, you have to scrape the above attributes.
```

```
In [73]: driver = webdriver.Chrome()
driver.get('https://www.flipkart.com/')
```

```
In [74]: product = driver.find_element(By.XPATH , '/html/body/div[1]/div/div[1]/div/div/div/div/div[1]/div/div[1]/div/d  
product.send_keys('sneakers')
```

```
In [76]: brand_name = []
          description = []
          price = []
          start = 0
          end = 3
```

```
In [90]: for page in range(start , end) :
    brand = driver.find_elements(By.XPATH , '//div[@class="_2WkVRV"]')
    for i in brand :
        title = i.text
        brand_name.append(title)
    next_button = driver.find_element(By.XPATH , '/html/body/div/div/div[3]/div[1]/div[2]/div[12]/div/div/nav/'
    next_button
Brand = brand_name[0:100]
len(Brand)
```

Out[90]: 100

```
In [89]: for page in range(start , end) :
    vivran = driver.find_elements(By.XPATH , '//a[@class="IRpwTa"]')
    for i in vivran :
        description.append(i.text)
    next_button = driver.find_element(By.XPATH , '/html/body/div/div/div[3]/div[1]/div[2]/div[12]/div/div/nav/'
    next_button
Description = description[0:100]
len>Description)
```

Out[89]: 100

```
In [91]: for page in range(start , end):
    kimat = driver.find_elements(By.XPATH , '//div[@class=" _30jeq3"]')
    for i in kimat :
        price.append(i.text)
    next_button = driver.find_element(By.XPATH , '/html/body/div/div/div[3]/div[1]/div[2]/div[12]/div/div/nav/')
    next_button
Price = price[0:100]
len(Price)
```

Out[91]: 100

```
In [93]: df = pd.DataFrame({
    "BRAND_NAME" : Brand ,
    "DESCRIPTION" : Description ,
    "PRICE" : Price
})
df
```

	BRAND_NAME	DESCRIPTION	PRICE
0	BRUTON	Combo Pack Of 2 Casual Shoes Sneakers For Men	₹549
1	aadi	Lightweight,Comfort,Summer,Trendy,Walking,Outd...	₹299
2	Deals4you	Sneakers For Women	₹299
3	URBANBOX	Trending Stylish Casual Outdoor Sneakers Shoes...	₹299
4	RED TAPE	Casual Sneaker Shoes for Men Soft Cushioned ...	₹1,737
...
95	RED TAPE	Casual Sneaker Shoes For Men Enhanced Comfor...	₹1,079
96	KILLER	Combo Pack Of 2 Casual Shoes Sneakers For Men	₹799
97	BRUTON	Lightweight,Comfort,Summer,Trendy,Walking,Outd...	₹375
98	Layasa	Sneakers For Women	₹469
99	Shoe Lab	Trending Stylish Casual Outdoor Sneakers Shoes...	₹399

100 rows × 3 columns

In []:

```
In [ ]: # Q7: Go to webpage https://www.amazon.in/ Enter "Laptop" in the search field and then click the search icon.  
# 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 []:

```
In [149]: driver = webdriver.Chrome()  
driver.get('https://www.amazon.in/')
```

In []:

```
In [102]: product = driver.find_element(By.XPATH , '/html/body/div[1]/header/div/div[1]/div[2]/div/form/div[2]/div[1]/input')  
product.send_keys("Laptop")
```

```
In [103]: search = driver.find_element(By.XPATH , '/html/body/div[1]/header/div/div[1]/div[2]/div/form/div[3]/div/span/input')  
search.click()
```

```
In [105]: # processor = driver.find_element(By.XPATH , '/html/body/div[1]/div[1]/div[2]/div/div[3]/span/div[1]/div[1]')  
processor = driver.find_element(By.XPATH , '/html/body/div[1]/div[1]/div[1]/div[2]/div/div[3]/span/div[1]/div[1]')  
processor.click()
```

```
In [132]: title = []  
rating = []  
price = []  
image = []
```

In [139]: !pip install selenium Pillow

```
Requirement already satisfied: selenium in c:\users\dravin mishra\anaconda3 2\lib\site-packages (4.17.2)
Requirement already satisfied: Pillow in c:\users\dravin mishra\anaconda3 2\lib\site-packages (9.4.0)
Requirement already satisfied: urllib3[socks]<3,>=1.26 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from selenium) (1.26.16)
Requirement already satisfied: trio~0.17 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from selenium) (0.24.0)
Requirement already satisfied: trio-websocket~0.9 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from selenium) (0.11.1)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from selenium) (2023.7.22)
Requirement already satisfied: typing_extensions>=4.9.0 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from selenium) (4.9.0)
Requirement already satisfied: attrs>=20.1.0 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from trio~0.17->selenium) (22.1.0)
Requirement already satisfied: sortedcontainers in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from trio~0.17->selenium) (2.4.0)
Requirement already satisfied: idna in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from trio~0.17->selenium) (3.4)
Requirement already satisfied: outcome in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from trio~0.17->selenium) (1.3.0.post0)
Requirement already satisfied: sniffio>=1.3.0 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from trio~0.17->selenium) (1.3.0)
Requirement already satisfied: cffi>=1.14 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from trio~0.17->selenium) (1.15.1)
Requirement already satisfied: wsproto>=0.14 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from trio-websocket~0.9->selenium) (1.2.0)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from urllib3[socks]<3,>=1.26->selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from cffi>=1.14->trio~0.17->selenium) (2.21)
Requirement already satisfied: h11<1,>=0.9.0 in c:\users\dravin mishra\anaconda3 2\lib\site-packages (from wsproto>=0.14->trio-websocket~0.9->selenium) (0.14.0)
```

In [140]: from PIL import Image

```
In [148]: img = driver.find_elements(By.TAG_NAME , 'img')
for i in img :
    Image = i.get_attribute('src')
    image.append(Image)
```

```
In [ ]:
```

```
In [113]: about_laptop = driver.find_elements(By.XPATH , '//div[@class="a-section a-spacing-none puis-padding-right-smal
for i in about_laptop:
    Title = i.text
    title.append(Title)
description = title[0:10]
len(description)
```

```
Out[113]: 10
```

```
In [125]: ratings = driver.find_elements(By.XPATH , '//span[@class="a-icon-alt"]')
for i in ratings :
    rev = i.text
    rating.append(rev)
ratng = rating[0:10]
len(ratng)
```

```
Out[125]: 10
```

```
In [128]: keemat = driver.find_elements(By.XPATH , '//span[@class="a-price-whole"]')
for i in keemat :
    pr = i.text
    price.append(pr)
Price = price[0:10]
len(Price)
```

```
Out[128]: 10
```

```
In [129]: df = pd.DataFrame({  
    "TITLE" : description ,  
    "RATING" : ratng ,  
    "PRICE" : Price  
})  
df
```

Out[129]:

	TITLE	RATING	PRICE
0	ZEBRONICS NBC 5S Intel Core i7 12th Gen 1255 L...	46,990	
1	Samsung Galaxy Book3 Core i7 13th Gen 1355U - ...	82,990	
2	Acer Travelmate Business Laptop Intel Core i7-...	49,990	
3	ASUS TUF Gaming F15, 15.6"(39.62 cms) FHD 144H...	73,990	
4	MSI Modern 14, Intel 12th Gen. i7-1255U, 36CM ...	49,990	
5	Lenovo IdeaPad Slim 3 Intel Core i7 12th Gen 1...	62,032	
6	HP Victus Gaming Laptop, 12th Gen Intel Core i...	81,990	
7	Samsung Galaxy Book3 Core i7 13th Gen 1355U - ...	82,990	
8	ASUS Vivobook 15, Intel Core i7-12650H 12th Ge...	59,990	
9	MSI Thin GF63, Intel 12th Gen. i7-12650H, 40CM...	84,990	

In []:

```
In [ ]: # 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 webpagehttps://www.azquotes.com/  
# 2. Click on TopQuotes  
# 3. Than scrap a) Quote b) Author c) Type Of Quotes
```

In [111]: driver = webdriver.Chrome()

In [11]: driver.get('https://www.azquotes.com/')

```
In [12]: 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 [13]: quote = []
author = []
Quote_type = []
start_page = 0
end_page = 10
```

```
In [14]: for page in range(0 , 10):
    thaughts = driver.find_elements(By.XPATH , '//a[@class="title"]')
#thaughts
    for i in thaughts :
        quotes = i.text
        quote.append(quotes)
Quote = quote[0:1000]
len(Quote)
```

```
Out[14]: 1000
```

```
In [15]: for page in range(start_page ,end_page):
    auth = driver.find_elements(By.XPATH , '//div[@class="author"]/a')
    for i in auth :
        authr = i.text
        author.append(authr)
Author = author[0:1000]
len(Author)
```

```
Out[15]: 1000
```

```
In [16]: for page in range (0 , 10):
    quote_type = driver.find_elements(By.XPATH , '//div[@class="tags"]')
    for i in quote_type :
        type_quote = i.text
        Quote_type.append(type_quote)
Quote_Type = Quote_type[0:1000]
len(Quote_Type)
```

```
Out[16]: 1000
```

```
In [17]: df = pd.DataFrame({
    "QUOTE" : Quote ,
    "AUTHOR" : Author ,
    "QUOTE_TYPE" : Quote_type
})
df
```

Out[17]:

	QUOTE	AUTHOR	QUOTE_TYPE
0	The essence of strategy is choosing what not t...	Michael Porter	Essence, Deep Thought, Transcendentalism
1	One cannot and must not try to erase the past ...	Golda Meir	Inspiration, Past, Trying
2	Patriotism means to stand by the country. It d...	Theodore Roosevelt	Country, Peace, War
3	Death is something inevitable. When a man has ...	Nelson Mandela	Inspirational, Motivational, Death
4	You have to love a nation that celebrates its ...	Erma Bombeck	4th Of July, Food, Patriotic
...
995	When the going gets weird, the weird turn pro.	Hunter S. Thompson	Music, Sports, Hunting
996	When a train goes through a tunnel and it gets...	Corrie Ten Boom	Trust, Encouraging, Uplifting
997	If you think you are too small to make a diffe...	Dalai Lama	Inspirational, Funny, Change
998	God doesn't require us to succeed, he only req...	Mother Teresa	Success, God, Mother
999	Change your thoughts and you change your world.	Norman Vincent Peale	Inspirational, Motivational, Change

1000 rows × 3 columns

In []:

```
In [ ]: # 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 [112]: driver.get('https://www.jagranjosh.com/')
```

```
In [113]: gk = driver.find_element(By.XPATH , '/html/body/div/header/nav/div/div[3]/ul/li[3]/a') #for accessing gk tag
gk.click() #this automatically hits the gk tag and redirect the page inside it
```

```
In [115]: pm_list = driver.find_element(By.XPATH , '/html/body/div[1]/div[8]/section[17]/div/div[2]/li[11]/a') #for a href
pm_list.click()
```

```
In [116]: name = []
born_dead = []
term_of_office = []
remarks = []
```

```
In [120]: pm_name = driver.find_elements(By.XPATH , '//div[@class="TableData"]/table/tbody/tr/td[2]/div')
for i in pm_name :
    name.append(i.text)
Name = name[0:19]
Name
```

```
Out[120]: ['Jawahar Lal Nehru',
 'Gulzarilal Nanda (Acting)',
 'Lal Bahadur Shastri',
 'Gulzari Lal Nanda (Acting)',
 'Indira Gandhi',
 'Morarji Desai',
 'Charan Singh',
 'Indira Gandhi',
 'Rajiv Gandhi',
 'V. P. Singh',
 'Chandra Shekhar',
 'P. V. Narasimha Rao',
 'Atal Bihari Vajpayee',
 'H. D. Deve Gowda',
 'Inder Kumar Gujral',
 'Atal Bihari Vajpayee',
 'Manmohan Singh',
 'Narendra Modi',
 'Narendra Modi']
```

```
In [121]: b_d = driver.find_elements(By.XPATH , '//div[@class="TableData"]/table/tbody/tr/td[3]')
for i in b_d :
    BD = i.text
    born_dead.append(BD)
bd = born_dead[0:19]
bd
```

```
Out[121]: [(1889-1964),
'(1898-1998)',
'(1904-1966)',
'(1898-1998)',
'(1917-1984)',
'(1896-1995)',
'(1902-1987)',
'(1917-1984)',
'(1944-1991)',
'(1931-2008)',
'(1927-2007)',
'(1921-2004)',
'(1924- 2018)',
'(born 1933)',
'(1919-2012)',
'(1924-2018)',
'(born 1932)',
'(born 1950)',
'(born 1950)']
```

```
In [102]: term = driver.find_elements(By.XPATH , '//div[@class="TableData"]/table/tbody/tr/td[4]')
for i in term :
    term_of_office.append(i.text)
#term_of_office
Term_of_office = term_of_office[0:19]
Term_of_office
```

```
Out[102]: ['15 August 1947 to 27 May 1964\n16 years, 286 days',
'27 May 1964 to 9 June 1964,\n13 days',
'9 June 1964 to 11 January 1966\n1 year, 216 days',
'11 January 1966 to 24 January 1966\n13 days',
'24 January 1966 to 24 March 1977\n11 years, 59 days',
'24 March 1977 to 28 July 1979 \n2 year, 126 days',
'28 July 1979 to 14 January 1980\n170 days',
'14 January 1980 to 31 October 1984\n4 years, 291 days',
'31 October 1984 to 2 December 1989\n5 years, 32 days',
'2 December 1989 to 10 November 1990\n343 days',
'10 November 1990 to 21 June 1991\n223 days',
'21 June 1991 to 16 May 1996\n4 years, 330 days',
'16 May 1996 to 1 June 1996\n16 days',
'1 June 1996 to 21 April 1997\n324 days',
'21 April 1997 to 19 March 1998 \n332 days',
'19 March 1998 to 22 May 2004 \n6 years, 64 days',
'22 May 2004 to 26 May 2014 \n10 years, 4 days',
'26 May 2014 - 2019',
'30 May 2019- Incumbent']
```

```
In [122]: remark = driver.find_elements(By.XPATH , '//div[@class="TableData"]/table/tbody/tr/td[5]/div')
for i in remark :
    remarks.append(i.text)
Remarks = remarks[0:19]
Remarks
```

```
Out[122]: ['The first prime minister of India and the longest-serving PM of India, the first to die in office.',
 'First acting PM of India',
 "He has given the slogan of 'Jai Jawan Jai Kisan' during the Indo-Pak war of 1965",
 '-',
 'First female Prime Minister of India',
 'Oldest to become PM (81 years old) and first to resign from office',
 'Only PM who did not face the Parliament',
 'The first lady who served as PM for the second term',
 'Youngest to become PM (40 years old)',
 'First PM to step down after a vote of no confidence',
 'He belongs to Samajwadi Janata Party',
 'First PM from South India',
 'PM for shortest tenure',
 'He belongs to Janata Dal',
 '----',
 ' The first non-congress PM who completed a full term as PM',
 ' First Sikh PM',
 '4th Prime Minister of India who served two consecutive tenures',
 'First non-congress PM with two consecutive tenures']
```

```
In [123]: df = pd.DataFrame({"PM_NAME" : Name , "Born_and_Dead" : bd , "REMARKS" : Remarks})  
df
```

Out[123]:

	PM_NAME	Born_and_Dead	REMARKS
0	Jawahar Lal Nehru	(1889–1964)	The first prime minister of India and the long...
1	Gulzarilal Nanda (Acting)	(1898–1998)	First acting PM of India
2	Lal Bahadur Shastri	(1904–1966)	He has given the slogan of 'Jai Jawan Jai Kisa...
3	Gulzari Lal Nanda (Acting)	(1898–1998)	-
4	Indira Gandhi	(1917–1984)	First female Prime Minister of India
5	Morarji Desai	(1896–1995)	Oldest to become PM (81 years old) and first t...
6	Charan Singh	(1902–1987)	Only PM who did not face the Parliament
7	Indira Gandhi	(1917–1984)	The first lady who served as PM for the second...
8	Rajiv Gandhi	(1944–1991)	Youngest to become PM (40 years old)
9	V. P. Singh	(1931–2008)	First PM to step down after a vote of no confi...
10	Chandra Shekhar	(1927–2007)	He belongs to Samajwadi Janata Party
11	P. V. Narasimha Rao	(1921–2004)	First PM from South India
12	Atal Bihari Vajpayee	(1924- 2018)	PM for shortest tenure
13	H. D. Deve Gowda	(born 1933)	He belongs to Janata Dal
14	Inder Kumar Gujral	(1919–2012)	-----
15	Atal Bihari Vajpayee	(1924-2018)	The first non-congress PM who completed a ful...
16	Manmohan Singh	(born 1932)	First Sikh PM
17	Narendra Modi	(born 1950)	4th Prime Minister of India who served two con...
18	Narendra Modi	(born 1950)	First non-congress PM with two consecutive ten...

```
In [ ]:
```

```
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 carsin the world..  
# 4. Then scrap the mentioned data and make the dataframe.
```

In []:

```
In [47]: driver.get('https://www.motor1.com/')
```

```
In [50]: search = driver.find_element(By.XPATH , '/html/body/div[10]/div[2]/div/div/div[3]/div/div/div/form/input')
search.send_keys("50 most expensive cars in the world.")
```

```
In [59]: search_button = driver.find_element(By.XPATH , '/html/body/div[10]/div[2]/div/div/div[3]/div/div/div/form/button')
search_button.click()
```

```
In [60]: search_for_50_cars = driver.find_element(By.XPATH , '/html/body/div[10]/div[9]/div/div[1]/div/div/div[1]/div/d  
search for 50 cars.click()
```

```
In [62]: car_name = []
          car_price = []
```

```
In [65]: name = driver.find_elements(By.XPATH , '//h3[@class="subheader"]')
for name in name :
    car_name.append(name.text)
car = car_name[0:50]
```

```
In [69]: price = driver.find_elements(By.XPATH , '//div[@class="postBody description e-content"]/p/strong')  
for price in price :  
    car_price.append(price.text)  
price = car_price[0:50]
```

```
In [70]: df = pd.DataFrame({"CAR_NAME" : car , "CAR_PRICE" : price})  
df
```

Out[70]:

	CAR_NAME	CAR_PRICE
0	Aston Martin Valour	Price: \$1.5 Million
1	McLaren Elva	Price: \$1.7 Million
2	Czinger 21C	Price: \$1.7 Million
3	Ferrari Monza	Price: \$1.7 Million
4	Gordon Murray T.33	Price: \$1.7 Million
5	Koenigsegg Gemera	Price: \$1.7 Million
6	Zenvo TSR-S	Price: \$1.7 Million
7	Hennessey Venom F5	Price: \$1.8 Million
8	Bentley Bacalar	Price: \$1.9 Million
9	Hispano Suiza Carmen Boulogne	Price: \$1.9 Million
10	Bentley Mulliner Batur	Price: \$2.0 Million
11	Deus Vayanne	Price: \$2.0 Million
12	SSC Tuatara	Price: \$2.0 Million
13	Lotus Evija	Price: \$2.1 Million
14	Aston Martin Vulcan	Price: \$2.3 Million
15	Delage D12	Price: \$2.3 Million
16	Ferrari Daytona SP3	Price: \$2.3 Million
17	McLaren Speedtail	Price: \$2.3 Million
18	Rimac Nevera	Price: \$2.4 Million
19	Pagani Utopia	Price: \$2.5 Million
20	Pininfarina Battista	Price: \$2.5 Million
21	Gordon Murray T.50	Price: \$2.6 Million
22	Lamborghini Countach	Price: \$2.6 Million
23	Mercedes-AMG Project One	Price: \$2.7 Million
24	Zenvo Aurora	Price: \$2.8 Million
25	Aston Martin Victor	Price: \$3.0 Million

	CAR_NAME	CAR_PRICE
26	Hennessey Venom F5 Roadster	\$3.0 Million
27	Koenigsegg Jesko	Price: \$3.0 Million
28	Aston Martin Valkyrie	Price: \$3.2 Million
29	W Motors Lykan Hypersport	Price: \$3.4 Million
30	McLaren Solus	\$3.5 Million
31	Lamborghini Sian	Price: \$3.6 million
32	Koenigsegg CC850	Price: \$3.7 Million
33	Bugatti Chiron Super Sport 300+	Price: \$3.9 Million
34	Lamborghini Veneno	Price: \$4.5 Million
35	Bugatti Bolide	Price: \$4.7 Million
36	Pininfarina B95 Speedster	Price: \$4.8 Million
37	Bugatti Mistral	Price: \$5.0 Million
38	Pagani Huayra Imola	Price: \$5.4 Million
39	Bugatti Divo	Price: \$5.8 Million
40	SP Automotive Chaos	Price: \$6.4 Million
41	Pagani Codalunga	Price: \$7.4 Million
42	777 Hypercar	Price: \$7.5 Million
43	Mercedes-Maybach Exelero	Price: \$8.0 Million
44	Bugatti Centodieci	Price: \$9.0 Million
45	Bugatti Chiron Profilée	Price: \$10.8 Million
46	Rolls-Royce Sweptail	Price: \$12.8 Million
47	Bugatti La Voiture Noire	Price: \$13.4 Million
48	Rolls-Royce Boat Tail*	Price: \$28.0 Million (est.)
49	Rolls-Royce La Rose Noire Droptail	Price: \$30 Million (est.)

