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

In [1]:
```
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 [2]:
```
#1.1

driver=webdriver.Chrome()
```

In [ ]:
```
driver.get("https://www.shine.com/")
```

In [ ]:
```
#1.2
designation= driver.find_element(By.CLASS_NAME,"form-control  ")
```

In [ ]:
```
designation.send_keys('Data Analyst')
```

In [ ]:
```
location= driver.find_element(By.NAME,"id_loc")
```

In [ ]:
```
location.send_keys('Bangalore')
```

In [ ]:
```
#1.3
search= driver.find_element(By.XPATH,"/html/body/div/div[4]/div/div[2]/div[2]/div[1]/for
search.click()
```

In [ ]:
```
#1.4
#Creating empty list
job_title=[]
job_location=[]
company_name=[]
experience_required=[]
```

In [ ]:
```
#Scraping job titles
title_tags=driver.find_elements(By.XPATH,'//div[@class="parentClass position-relative"]/

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

In [ ]:
```
#Scraping job location
location_tags=driver.find_elements(By.XPATH,'//div[@class=" jobCard_jobCard_lists_item__

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

In [ ]:
```
#Scraping company name
company_tags=driver.find_elements(By.XPATH,'//div[@class="jobCard_jobCard_cName__mYnow"]

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

In [ ]:
```
#Scraping Job experience
experience_tags=driver.find_elements(By.XPATH,'//div[@class=" jobCard_jobCard_lists_item

for i in experience_tags[0:10]:
    exp=i.text
    experience_required.append(exp)
```

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

In [ ]:
```
#1.5 creating dataframe
df=pd.DataFrame({'Title':job_title,'Location':job_location,'Company Name':company_name,'
```

In [ ]:
```
df
```

In [ ]:
```
#2.1

driver.get("https://www.shine.com/")
```

In [ ]:
```
#2.2
designation= driver.find_element(By.CLASS_NAME,"form-control  ")
```

In [ ]:
```
designation.send_keys('Data Scientist')
```

In [ ]:
```
location= driver.find_element(By.NAME,"id_loc")
```

In [ ]:
```
location.send_keys('Bangalore')
```

In [ ]:
```
#2.3
search= driver.find_element(By.XPATH,"/html/body/div/div[4]/div/div[2]/div[2]/div/form/d
search.click()
```

In [ ]:
```
#2.4
#Creating empty list
job_title=[]
job_location=[]
company_name=[]
experience_required=[]
```

In [ ]:
```
#Scraping job title
title_tags=driver.find_elements(By.XPATH,'//div[@class="parentClass position-relative"]/

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

In [ ]:
```
#Scraping job location
location_tags=driver.find_elements(By.XPATH,'//div[@class=" jobCard_jobCard_lists_item__

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

In [ ]:
```
#Scraping company name
company_tags=driver.find_elements(By.XPATH,'//div[@class="jobCard_jobCard_cName__mYnow"]

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

In [ ]:
```
#Scraping Job experience
```

```
experience_tags=driver.find_elements(By.XPATH,'//div[@class=" jobCard_jobCard_lists_item

for i in experience_tags[0:10]:
    exp=i.text
    experience_required.append(exp)
```

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

In [ ]:
```
#2.5 Creating dataframe
df=pd.DataFrame({'Title':job_title,'Location':job_location,'Company Name':company_name,'
```

In [ ]:
```
df
```

In [ ]:
```
#3.1
driver.get("https://www.shine.com/")
```

In [ ]:
```
#3.2
designation= driver.find_element(By.CLASS_NAME,"form-control  ")
```

In [ ]:
```
designation.send_keys('Data Scientist')
```

In [ ]:
```
#3.3
search= driver.find_element(By.XPATH,"/html/body/div/div[4]/div/div[2]/div[2]/div[1]/for
search.click()
```

In [ ]:
```
#3.4
location_dropdown = driver.find_element(By.CLASS_NAME,"filter_filter_lists_items__wlFfo"
```

In [ ]:
```
location_dropdown.click()
```

In [ ]:
```
delhi_option = driver.find_element(By.XPATH, "/html/body/div[1]/div[1]/div[3]/div/div[1]
```

In [ ]:
```
delhi_option.click()
```

In [ ]:
```
ncr_option=driver.find_element(By.XPATH,"/html/body/div[1]/div[1]/div[3]/div/div[1]/div/
```

In [ ]:
```
ncr_option.click()
```

In [ ]:
```
location_result= driver.find_element(By.XPATH,"/html/body/div[1]/div[1]/div[3]/div/div[1
location_result.click()
```

In [ ]:
```
salary_dropdown = driver.find_element(By.XPATH,"/html/body/div[1]/div[1]/div[3]/div/div[
```

In [ ]:
```
salary_dropdown.click()
```

In [ ]:
```
salary_result=driver.find_element(By.XPATH,"/html/body/div[1]/div[1]/div[3]/div/div[1]/d
```

In [ ]:
```
salary_result.click()
```

In [ ]:
```
salary_final=driver.find_element(By.XPATH,"/html/body/div[1]/div[1]/div[3]/div/div[1]/di
```

In [ ]:
```
salary_final.click()
```

In [ ]:
```
#3.5
job_title=[]
job_location=[]
```

```
company_name=[]
experience_required=[]
```

In [ ]:
```
#Scraping job title
title_tags=driver.find_elements(By.XPATH,'//div[@class="parentClass position-relative"]/

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

In [ ]:
```
#Scraping job location
location_tags=driver.find_elements(By.XPATH,'//div[@class=" jobCard_jobCard_lists_item__

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

In [ ]:
```
#Scraping company name
company_tags=driver.find_elements(By.XPATH,'//div[@class="jobCard_jobCard_cName__mYnow"]

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

In [ ]:
```
#Scraping Job experience
experience_tags=driver.find_elements(By.XPATH,'//div[@class=" jobCard_jobCard_lists_item

for i in experience_tags[0:10]:
    exp=i.text
    experience_required.append(exp)
```

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

In [ ]:
```
#2.6 Creating dataframe
df=pd.DataFrame({'Title':job_title,'Location':job_location,'Company Name':company_name,'
```

In [ ]:
```
df
```

In [ ]:
```
#4.1
driver.get("https://www.flipkart.com/")
```

In [ ]:
```
#4.2
product= driver.find_element(By.CLASS_NAME,"Pke_EE")
```

In [ ]:
```
product.send_keys('Sunglasses')
```

In [ ]:
```
search= driver.find_element(By.CLASS_NAME,"_2iLD__")
search.click()
```

In [ ]:
```
#4.3
#Creating empty list
product_brand=[]
product_price=[]
product_description=[]
```

In [ ]:
```
#4.4 Extracting data
```

In [ ]:
```
#Scraping brand
brand=driver.find_elements(By.CLASS_NAME,"_2WkVRV")
```

```
for i in brand:
    product_brand.append(i.text)

next_button=driver.find_element(By.XPATH,'//a[@class="_1LKTO3"]')
next_button.click()
```

In [ ]:
```
print(len(product_brand))
```

In [ ]:
```
brand_b=driver.find_elements(By.CLASS_NAME,"_2WkVRV")

for i in brand_b:
    product_brand.append(i.text)

next_button=driver.find_element(By.XPATH,'//a[@class="_1LKTO3"][2]')
next_button.click()
```

In [ ]:
```
brand_c=driver.find_elements(By.CLASS_NAME,"_2WkVRV")

for i in brand_c[0:20]:
    product_brand.append(i.text)
```

In [ ]:
```
#Scraping price
price=driver.find_elements(By.CLASS_NAME,"_30jeq3")
for i in price[0:40]:
    product_price.append(i.text)

next_button=driver.find_element(By.XPATH,'//a[@class="_1LKTO3"]')
next_button.click()
```

In [ ]:
```
price_b=driver.find_elements(By.CLASS_NAME,"_30jeq3")
for i in price_b[0:40]:
    product_price.append(i.text)

next_button=driver.find_element(By.XPATH,'//a[@class="_1LKTO3"][2]')
next_button.click()
```

In [ ]:
```
price_c=driver.find_elements(By.CLASS_NAME,"_30jeq3")
for i in price_c[0:20]:
    product_price.append(i.text)
```

In [ ]:
```
print(len(product_price))
```

In [ ]:
```
#Scraping description

description=driver.find_elements(By.XPATH,'//div[@class="_2B099V"]/a[1]')
for i in description[0:40]:
    product_description.append(i.text)

next_button=driver.find_element(By.XPATH,'//a[@class="_1LKTO3"]')
next_button.click()
```

In [ ]:
```
description_b=driver.find_elements(By.XPATH,'//div[@class="_2B099V"]/a[1]')
for i in description_b[0:40]:
    product_description.append(i.text)

next_button=driver.find_element(By.XPATH,'//a[@class="_1LKTO3"][2]')
next_button.click()
```

In [ ]:
```
description_c=driver.find_elements(By.XPATH,'//div[@class="_2B099V"]/a[1]')
for i in description_c[0:20]:
```

```
    product_description.append(i.text)
```

In [ ]:
```
print(len(product_description))
```

In [ ]:
```
print(len(product_price),len(product_price),len(product_description))
```

In [ ]:
```
#Creating dataframe
df= pd.DataFrame({'Brand':product_brand,'Price':product_price,'Description':product_desc
```

In [ ]:
```
df
```

In [ ]:
```
#5.
driver.get("https://www.flipkart.com/apple-iphone-11-black-64-gb/product-reviews/itm4e50
```

In [ ]:
```
#Creating empty list
Rating=[]
Review_summary=[]
Full_review=[]
```

In [ ]:
```
#Extracting data
start=0
end=10

for page in range(start,end):

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

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

    r_v =driver.find_elements(By.XPATH,'//p[@class="_2-N8zT"]')

    for i in r_v:
        Review_summary.append(i.text)

    f_r= driver.find_elements(By.XPATH,'//div[@class="t-ZTKy"]')

    for i in f_r:
        Full_review.append(i.text)

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

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

In [ ]:
```
#Creating data frame
df=pd.DataFrame({'Rating':Rating,'Review':Review_summary,'Full Review':Full_review})
```

In [ ]:
```
df
```

In [ ]:
```
#6
driver.get("https://www.flipkart.com/")
```

In [ ]:
```
product_req= driver.find_element(By.CLASS_NAME,"Pke_EE")
```

In [ ]:
```
product_req.send_keys('Sneakers')
```

In [ ]:
```
search= driver.find_element(By.CLASS_NAME,"_2iLD__")
```

```
search.click()
```

In [ ]:
```
#Creating empty list
product_brand=[]
product_price=[]
product_description=[]
```

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

for page in range(start,end):

    brand=driver.find_elements(By.XPATH,'//div[@class="_2WkVRV"]')
    for i in brand:
        product_brand.append(i.text)

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

    description=driver.find_elements(By.XPATH,'//div[@class="_2B099V"]/a[1]')
    for i in description:
        product_description.append(i.text)

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

In [ ]:
```
print(len(product_brand),len(product_price),len(product_description))
```

In [ ]:
```
product_description=product_description[0:100]
product_price=product_price[0:100]
product_brand=product_brand[0:100]
```

In [ ]:
```
print(len(product_brand),len(product_price),len(product_description))
```

In [ ]:
```
#Creating dataframe
df= pd.DataFrame({'Brand':product_brand,'Price':product_price,'Description':product_desc
```

In [ ]:
```
df
```

In [ ]:
```
#7
driver.get("https://www.amazon.in/")
```

In [ ]:
```
product= driver.find_element(By.XPATH,"/html/body/div[1]/header/div/div[1]/div[2]/div/fo
```

In [ ]:
```
product.click()
```

In [ ]:
```
product_class= driver.find_element(By.XPATH,'//div[@class="nav-search-field "]/input')
```

In [ ]:
```
product_class.send_keys('Laptop')
```

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

In [ ]:
```
selection= driver.find_element(By.XPATH,"/html/body/div[1]/div[1]/div[1]/div[2]/div/div[
```

In [ ]:
```
selection.click()
```

In [ ]:
```
#Creating empty list
title=[]
ratings=[]
price=[]
```

In [ ]:
```
#Extracting data
title_= driver.find_elements(By.XPATH,'//h2[@class="a-size-mini a-spacing-none a-color-b

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

In [ ]:
```
price_=driver.find_elements(By.XPATH,'//span[@class="a-price"]/span/span[2]')

for i in price_[0:10]:
    price_=i.text
    price.append(price_)
```

In [ ]:
```
ratings_= driver.find_elements(By.XPATH,'//i[@class="a-icon a-icon-star-small a-star-sma

for i in ratings_[0:10]:
    ratings_=i.text
    ratings.append(ratings_)
```

In [ ]:
```
print(len(title),len(ratings),len(price))
```

In [ ]:
```
#Creating dataframe
df=pd.DataFrame({'Title':title,'Rating':ratings,'Price':price})
```

In [ ]:
```
df
```

In [43]:
```
#8.1
driver.get('https://www.azquotes.com/')
```

In [44]:
```
#8.2
top_quotes= driver.find_element(By.XPATH,'//div[@class="mainmenu"]/ul/li[5]/a')
top_quotes.click()
```

In [ ]:
```
#8.3 Creating empty list
quote=[]
author=[]
type_quote=[]
```

In [ ]:
```
#Extracting data

start=0
end=10


for page in range(start,end):

    quote_=driver.find_elements(By.XPATH,'//ul[@class="list-quotes"]/li/div/p/a[2]')
    for i in quote_:
        quote.append(i.text)

    author_=driver.find_elements(By.XPATH,'//div[@class="author"]/a')
```

```
    for i in author_:
        author.append(i.text)

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

    next_button=driver.find_element(By.XPATH,"/html/body/div[1]/div[2]/div/div/div/div[1
    next_button.click()
    time.sleep(1)
```

In [47]:
```
print(len(quote),len(author),len(type_quote))
```

```
700 700 700
```

In [48]:
```
#Creating data frame

df=pd.DataFrame({'Quote':quote,'Author':author,'Type of Quote':type_quote})
```

In [49]:
```
df
```

Out[49]:

	Quote	Author	Type of Quote
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
...	...	...	...
695	Regret for the things we did can be tempered b...	Sydney J. Harris	Love, Inspirational, Motivational
696	America... just a nation of two hundred millio...	Hunter S. Thompson	Gun, Two, Qualms About
697	For every disciplined effort there is a multip...	Jim Rohn	Inspirational, Greatness, Best Effort
698	The spiritual journey is individual, highly pe...	Ram Dass	Spiritual, Truth, Yoga
699	The mind is not a vessel to be filled but a fi...	Plutarch	Inspirational, Leadership, Education

700 rows × 3 columns

In [3]:
```
#10.1
driver.get('https://www.motor1.com/')
```

In [14]:
```
#10.2
search= driver.find_element(By.XPATH,"/html/body/div[10]/div[2]/div/div/div[3]/div/div/b
search.click()
```

In [7]:
```
top_cars= driver.find_element(By.XPATH,"/html/body/div[10]/div[2]/div/div/div[3]/div/div
```

In [17]:
```
top_cars.send_keys('50 most expensive cars')
```

In [15]:
```
search= driver.find_element(By.XPATH,'//button[@class="m1-search-panel-button m1-search-
search.click()
```

In [18]:
```
Exp_cars= driver.find_element(By.XPATH,"/html/body/div[10]/div[9]/div/div[1]/div/div/div
Exp_cars.click()
```

In [23]:
```
#Creating empty list
Car_name=[]
Car_price=[]
```

In [24]:
```
#Scraping data(name)

Car= driver.find_elements(By.XPATH,'//h3[@class="subheader"]')

for i in Car[0:50]:
    name=i.text
    Car_name.append(name)
```

In [25]:
```
#Scraping data(price)
Price= driver.find_elements(By.XPATH,'//div[@class="postBody description e-content"]/p/s

for i in Price[0:50]:
    price=i.text
    Car_price.append(price)
```

In [26]:
```
# creating dataframe
df=pd.DataFrame({'Name':Car_name,'Price':Car_price})
```

In [27]:
```
df
```

Out[27]:

	Name	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
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.)

In [28]:
```
#9.1
driver.get('https://www.jagranjosh.com/')
```

In [29]:
```
#9.2
Gkoption= driver.find_element(By.XPATH,'//ul[@class="Header_navLink__8eXbJ"]/li[7]/a[1]'
```

In [30]:
```
Gkoption.click()
```

In [33]:
```
#9.2
listofPM= driver.find_element(By.XPATH,"/html/body/div[1]/div[8]/section[17]/div/div/ul[
```

In [ ]:
```
listofPM.click()
```

In [35]:
```
#9.3 Extracting data
#Creating empty list
Name_PM=[]
Born_Dead=[]
Term_of_office=[]
Remarks=[]
```

In [36]:
```
#Scraping Name
Name = driver.find_elements(By.XPATH, '//div[@class="Details_StoryBody__85cfI"]/div[9]/p
for i in Name:
    PM = i.text
    Name_PM.append(PM)
```

In [37]:
```
#Scraping Born-dead
b_d = driver.find_elements(By.XPATH,'//div[@class="Details_StoryBody__85cfI"]/div[9]/p[5
for i in b_d:
    Bo_dead = i.text
    Born_Dead.append(Bo_dead)
```

In [38]:
```
#Scraping Term of office
t_o_o = driver.find_elements(By.XPATH,'//div[@class="Details_StoryBody__85cfI"]/div[9]/p
for i in t_o_o:
    terms = i.text
    Term_of_office.append(terms)
```

In [42]:
```
#Scraping Remarks
r_r = driver.find_elements(By.XPATH,'//div[@class="Details_StoryBody__85cfI"]/div[9]/p[7
for i in r_r:
    Re = i.text
    Remarks.append(Re)
```

In [43]:
```
#Creating dataframe
df=pd.DataFrame({'Name':Name_PM,'Born-Dead':Born_Dead,'Term of Office':Term_of_office,'R
```

In [44]:
```
df
```

Out[44]:

	Name	Born-Dead	Term of Office	Remarks
0	Jawahar Lal Nehru	(1889–1964)	15 August 1947 to 27 May 1964	16 years, 286 days

In [39]:

In [40]:

In [ ]:

In [ ]: