

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

```
Requirement already satisfied: selenium in c:\users\rakhi\coding folders\lib\site-packages (4.19.0)
Requirement already satisfied: urllib3[socks]<3,>=1.26 in c:\users\rakhi\coding folders\lib\site-packages (from selenium) (1.26.16)
Requirement already satisfied: trio~=0.17 in c:\users\rakhi\coding folders\lib\site-packages (from selenium) (0.25.0)
Requirement already satisfied: trio-websocket~=0.9 in c:\users\rakhi\coding folders\lib\site-packages (from selenium) (0.11.1)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\rakhi\coding folders\lib\site-packages (from selenium) (2023.7.22)
Requirement already satisfied: typing_extensions>=4.9.0 in c:\users\rakhi\coding folders\lib\site-packages (from selenium) (4.10.0)
Requirement already satisfied: attrs>=23.2.0 in c:\users\rakhi\coding folders\lib\site-packages (from trio~=0.17->selenium) (23.2.0)
Requirement already satisfied: sortedcontainers in c:\users\rakhi\coding folders\lib\site-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: idna in c:\users\rakhi\coding folders\lib\site-packages (from trio~=0.17->selenium) (3.4)
Requirement already satisfied: outcome in c:\users\rakhi\coding folders\lib\site-packages (from trio~=0.17->selenium) (1.3.0.post0)
Requirement already satisfied: sniffio>=1.3.0 in c:\users\rakhi\coding folders\lib\site-packages (from trio~=0.17->selenium) (1.3.1)
Requirement already satisfied: cffi>=1.14 in c:\users\rakhi\coding folders\lib\site-packages (from trio~=0.17->selenium) (1.15.1)
Requirement already satisfied: wsproto>=0.14 in c:\users\rakhi\coding folders\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\rakhi\coding folders\lib\site-packages (from urllib3[socks]<3,>=1.26->selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\users\rakhi\coding folders\lib\site-packages (from cffi>=1.14->trio~=0.17->selenium) (2.21)
Requirement already satisfied: h11<1,>=0.9.0 in c:\users\rakhi\coding folders\lib\site-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0)
```

```
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 [ ]: # Q1: 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
```

```
In [29]: driver=webdriver.Chrome()
driver.get("https://www.naukri.com")
```

```
In [30]: job_title = driver.find_element(By.CLASS_NAME,"suggestor-input")
job_title.send_keys('Data Scientist')

search = driver.find_element(By.CLASS_NAME,"qsbSubmit")
search.click()
```

```
In [31]: job_title=[]
job_loc=[]
comp_name=[]
exp_req=[]
```

```
In [32]: title_tags=driver.find_elements(By.CLASS_NAME,"title")
        for i in title_tags:
            title=i.text
            job_title.append(title)
            job_title[:10]

        loc_tags=driver.find_elements(By.CLASS_NAME,"locwidth")
        for i in loc_tags:
            location=i.text
            job_loc.append(location)
            job_loc[:10]

        comp_tags=driver.find_elements(By.CLASS_NAME,"logoImage")
        for i in comp_tags:
            company=i.text
            comp_name.append(company)
            comp_name[:10]

        exp_tags=driver.find_elements(By.CLASS_NAME,"expwidth")
        for i in exp_tags:
            experience=i.text
            exp_req.append(experience)
            exp_req[:10]
```

```
In [33]: #So lets check th length of ech element.
print(len(job_title[:10]),len(job_loc[:10]),len(comp_name[:10]),len(exp_req[:10]))

10 10 10 10
```

```
In [37]: df=pd.DataFrame({'Title':job_title[:10],'Location':job_loc[:10],'Company Name':comp_name[:10],'Experience-required':exp_req[:10]})
df
```

Out[37]:

	Title	Location	Company Name	Experience-required
0	Senior Data Scientist, Reporting	Kolkata, Mumbai, New Delhi, Hyderabad/Secunder...		6-7 Yrs
1	Data Scientist, Marketing	Kolkata, Mumbai, New Delhi, Hyderabad/Secunder...		4-6 Yrs
2	Data Scientist	Kolkata, Mumbai, New Delhi, Hyderabad, Pune, C...		3-6 Yrs
3	Sr. Data Scientist	Kolkata, Mumbai, New Delhi, Hyderabad, Pune, C...		2-6 Yrs
4	Sr. Data Scientist	Kolkata, Mumbai, New Delhi, Hyderabad, Pune, C...		5-6 Yrs
5		Hyderabad, Telangana, Gurugram, Haryana, Banga...		6-11 Yrs
6		Hyderabad, Gurugram, Bengaluru		8-13 Yrs
7		Greater Noida, Bengaluru		4-6 Yrs
8	AI Data Scientist	Kolkata, Mumbai, New Delhi, Hyderabad/Secunder...		7-8 Yrs
9	GenAI Data Scientist	Kolkata, Mumbai, New Delhi, Hyderabad/Secunder...		3-6 Yrs

```
In [ ]: # Q2:scrap data for Data Science job in Bangalore
```

```
In [39]: driver=webdriver.Chrome()  
driver.get("https://www.shine.com/")
```

```
In [44]: job_title=driver.find_element(By.CLASS_NAME,"form-controle")  
job_title.send_keys('Data Scientist')
```

```
-----  
NoSuchElementException Traceback (most recent call last)
```

```
Cell In[44], line 1
```

```
----> 1 job_title=driver.find_element(By.CLASS_NAME,"form-controle")  
      2 job_title.send_keys('Data Scientist')
```

```
File ~\coding folders\Lib\site-packages\selenium\webdriver\remote\webdriver.py:741, in WebDriver.find_element(self, by, value)  
    738     by = By.CSS_SELECTOR  
    739     value = f'[name="{value}"]'  
--> 741 return self.execute(Command.FIND_ELEMENT, {"using": by, "value": value})["value"]
```

```
File ~\coding folders\Lib\site-packages\selenium\webdriver\remote\webdriver.py:347, in WebDriver.execute(self, driver_command, params)  
    345 response = self.command_executor.execute(driver_command, params)  
    346 if response:  
--> 347     self.error_handler.check_response(response)  
    348     response["value"] = self._unwrap_value(response.get("value", None))  
    349     return response
```

```
File ~\coding folders\Lib\site-packages\selenium\webdriver\remote\errorhandler.py:229, in ErrorHandler.check_response(self, response)  
    227     alert_text = value["alert"].get("text")  
    228     raise exception_class(message, screen, stacktrace, alert_text) # type: ignore[call-arg] # mypy is not smart enough
```

```
In [33]: location=driver.find_element(By.XPATH,"/html/body/div/div[2]/div/div/div[1]/div[1]/div/div[2]/div/div/form/div/div[1]/ul/li[2]/div")
location.send_keys('Bangalore')
```

```
In [31]: search=driver.find_element(By.CLASS_NAME,"")
search.click()
```

ElementClickInterceptedException Traceback (most recent call last)

Cell In[31], line 2

```
1 search=driver.find_element(By.XPATH,"/html/body/div/div[2]/div/div/div[1]/div[1]/div/div[2]/div/div[1]/form/div/div[2]/div")
----> 2 search.click()
```

File ~\coding folders\Lib\site-packages\selenium\webdriver\remote\webelement.py:94, in WebElement.click(self)

```
92 def click(self) -> None:
93     """Clicks the element."""
---> 94     self._execute(Command.CLICK_ELEMENT)
```

File ~\coding folders\Lib\site-packages\selenium\webdriver\remote\webelement.py:395, in WebElement._execute(self, command, params)

```
393     params = {}
394     params["id"] = self._id
--> 395     return self._parent.execute(command, params)
```

File ~\coding folders\Lib\site-packages\selenium\webdriver\remote\webdriver.py:347, in WebDriver.execute(self, driver_command, params)

```
345 response = self.command_executor.execute(driver_command, params)
346 if response:
```

```
In [34]: job_title=[]  
job_loc=[]  
comp_name=[]  
exp_req=[]
```

```
In [35]: title_tags=driver.find_elements(By.XPATH,'//div[@class="cust-job-tuple layout-wrapper lay-2 sjw_tuple"]/div/a')  
for i in title_tags:  
    title=i.text  
    job_title.append(title)  
  
loc_tags=driver.find_elements(By.XPATH,'//span[@class="locwidth"]')  
for i in loc_tags:  
    location=i.text  
    job_loc.append(location)  
  
comp_tags=driver.find_elements(By.XPATH,'//div[@class="row2"]/span/a[1]')  
for i in comp_tags:  
    company=i.text  
    comp_name.append(company)  
  
exp_tags=driver.find_elements(By.XPATH,'//span[@class="expwidth"]')  
for i in exp_tags:  
    experience=i.text  
    exp_req.append(experience)
```

```
In [36]: print(len(job_title),len(job_loc),len(comp_name),len(exp_req))
```

0 0 0 0

```
In [38]: df=pd.DataFrame({'Title':job_title,'Location':job_loc,'Company Name':comp_name,'Experience-required':exp_req})
df
```

Out[38]:


Title	Location	Company Name	Experience-required
-------	----------	--------------	---------------------

```
In [ ]:
```



```
In [ ]: # Q3:scrap 100 review data for iphone11 phone
```

```
In [45]: driver=webdriver.Chrome()  
driver.get("https://www.flipkart.com/apple-iphone-11-black-64-gb/product-reviews/itm4e5041ba101fd?pid=MOBFWQ6BXGJCEYNY&lid=LSTMOR")
```



```
In [46]: Rati_list = []  
Review_summ = []  
Full_rev = []
```

```

In [48]: for i in range(0,10):
        rati_list=driver.find_elements(By.CLASS_NAME,"_3LWZlK _1BLPMq")
        review_summ=driver.find_element(By.CLASS_NAME,"_2-N8zT")
        full_rev=driver.find_element(By.CLASS_NAME,"t-ZTKy")

        for i in rati_list:
            rating=i.text
            Rati_list.append(rating)
            Rati_list[:100]

        for i in review_summ:
            review=i.text
            Review_summ.append(review)
            Review_summ[:100]

        for i in full_rev:
            full_re=i.text
            Full_rev.append(full_re)
            Full_rev[:100]

next_button=driver.find_element(By.XPATH,"/html/body/div/div/div[3]/div/div/div[2]/div[13]/div/div/nav/a[11]")
next_button.click()

```

WebDriverException

Traceback (most recent call last)

Cell In[48], line 2

```

1 for i in range(0,10):
----> 2     rati_list=driver.find_elements(By.CLASS_NAME,"_3LWZlK _1BLPMq")
      3     review_summ=driver.find_element(By.CLASS_NAME,"_2-N8zT")
      4     full_rev=driver.find_element(By.CLASS_NAME,"t-ZTKy")

```

```

3     review_summ=driver.find_element(By.CLASS_NAME,"_2-N8zT")
4     full_rev=driver.find_element(By.CLASS_NAME,"t-ZTKy")

```

File ~\coding folders\Lib\site-packages\selenium\webdriver\remote\webdriver.py:771, in WebDriver.find_elements(self, by, value)

```

767     value = f'[name="{value}"]'
769 # Return empty list if driver returns null
770 # See https://github.com/SeleniumHQ/selenium/issues/4555
--> 771 return self.execute(Command.FIND_ELEMENTS, {"using": by, "value": value})["value"] or []

```

File ~\coding folders\Lib\site-packages\selenium\webdriver\remote\webdriver.py:347, in WebDriver.execute(self, driver_command, params)

```

345 response = self.command_executor.execute(driver_command, params)
346 if response:
--> 347     self.error_handler.check_response(response)
348     response["value"] = self._unwrap_value(response.get("value", None))
349     return response

```

File ~\coding folders\Lib\site-packages\selenium\webdriver\remote\errorhandler.py:229, in ErrorHandler.check_response(self, response)

```

227     alert_text = value["alert"].get("text")
228     raise exception_class(message, screen, stacktrace, alert_text) # type: ignore[call-arg] # mypy is not smart enough here
--> 229 raise exception_class(message, screen, stacktrace)

```

WebDriverException: Message: disconnected: not connected to DevTools
(failed to check if window was closed: disconnected: not connected to DevTools)
(Session info: chrome=123.0.6312.106)

Stacktrace:

```

GetHandleVerifier [0x00007FF61F0A7032+63090]
(No symbol) [0x00007FF61F012C82]
(No symbol) [0x00007FF61EEAEC65]
(No symbol) [0x00007FF61F0034B1]

```

```
(No symbol) [0x00007FF61F012C82]  
(No symbol) [0x00007FF61EEAEC65]  
(No symbol) [0x00007FF61EE9C31B]  
(No symbol) [0x00007FF61EE9BEC0]  
(No symbol) [0x00007FF61EEB0EC1]  
(No symbol) [0x00007FF61EF32AE9]  
(No symbol) [0x00007FF61EF16D83]  
(No symbol) [0x00007FF61EEE83A8]  
(No symbol) [0x00007FF61EEE9441]  
GetHandleVerifier [0x00007FF61F4A25AD+4238317]  
GetHandleVerifier [0x00007FF61F4DF70D+4488525]  
GetHandleVerifier [0x00007FF61F4D79EF+4456495]  
GetHandleVerifier [0x00007FF61F180576+953270]  
(No symbol) [0x00007FF61F01E54F]  
(No symbol) [0x00007FF61F019224]  
(No symbol) [0x00007FF61F01935B]  
(No symbol) [0x00007FF61F009B94]  
BaseThreadInitThunk [0x00007FFAC84D257D+29]  
RtlUserThreadStart [0x00007FFACA60AA58+40]
```

```
In [49]: print(len(Rati_list[:100]),len(Review_summ[:100]),len(Full_rev[:100]))
```

```
0 0 0
```

```
In [51]: df=pd.DataFrame({'Rating':Rati_list[:100],'Review Summary':Review_summ[:100],'Full Review':Full_rev[:100]})
```

```
In [ ]:
```

```
In [ ]: # Q4:Scrape data forfirst 100 sneakers you find when you visit flipkart.com
```

```
In [30]: driver=webdriver.Chrome()  
driver.get("https://www.flipkart.com/")
```

```
In [32]: search_g= driver.find_element(By.XPATH,"//input[@type='text']")  
search_g.send_keys('sneakers')  
  
search_btn=driver.find_element(By.CLASS_NAME,'_1cmsER')  
search_btn.click()
```

```
In [33]: B_name=[]  
Price=[]  
P_desc=[]
```

```
In [34]: for i in range(3):
        b_name=driver.find_elements(By.XPATH,"//div[@class='_2WkVRV']")
        p_desc=driver.find_elements(By.XPATH,"//a[@class='IRpwTa']")
        price =driver.find_elements(By.XPATH,"//div[@class='_25b18c']")

        for j in b_name:
            B_name.append(j.text)
        B_name[:100]

        for k in p_desc:
            P_desc.append(k.text)
        P_desc[:100]

        for l in price:
            Price.append(l.text)
        Price[:100]
```

```
In [35]: B_name[:100]
```

```
Out[35]: ['Deals4you',
          'Krors',
          'Deals4you',
          'URBANBOX',
          'RED TAPE',
          'RED TAPE']
```

'RED TAPE',
'RED TAPE',
'BRUTON',
'Deals4you',
'andD',
'aadi',
'Shozie',
"K' Footlance",
'RapidBox',
'ATOM',
'Cross Finger',
'RED TAPE',
'Krors',
"K' Footlance",
'RED TAPE',
'Hida',
'RED TAPE',
'Layasa',
'aadi',
'RapidBox',
'RapidBox',
'RED TAPE',
'Zixer',
'Deals4you',
'aadi',
'BRUTON',
"K' Footlance",
'Airson',
'asian',
'RED TAPE',
'RED TAPE'

'ORICUM',
'RED TAPE',
'Layasa',
'U.S. POLO ASSN.',
'RED TAPE',
'Deals4you',
'Krors',
'Deals4you',
'URBANBOX',
'RED TAPE',
'RED TAPE',
'BRUTON',
'Deals4you',
'andD',
'aadi',
'Shozie',
"K' Footlance",
'RapidBox',
'ATOM',
'Cross Finger',
'RED TAPE',
'Krors',
"K' Footlance",
'RED TAPE',
'Hida',
'RED TAPE',
'Layasa',
'aadi',
'RapidBox',
'RapidBox',
'RED TAPE',
'Zixer',
'Deals4you'.


```
'RED TAPE',  
'RED TAPE',  
'BRUTON',  
'Deals4you',  
'andD',  
'aadi',  
'Shozie',  
"K' Footlance",  
'RapidBox',  
'ATOM',  
'Cross Finger',  
'RED TAPE',  
'Krors',  
"K' Footlance",  
'RED TAPE',  
'Hida']
```

```
print(len(B_name[:100])),print(len(price[:100])),print(len(Product_desc[:100]))
```

100

45

38

(None, None, None)

```
sneakers=pd.DataFrame({})  
sneakers['Brand_name']=Brand_Name[:100]  
sneakers['Price']=Price[:100]  
sneakers['Product_Description']=Product_Description[:100]  
sneakers
```

```
In [ ]: # Q5: 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:
```

```
In [2]: driver=webdriver.Chrome()
driver.get("https://www.amazon.in/")
```

```
In [3]: search_g= driver.find_element(By.XPATH,"//input[@type='text']")
search_g.send_keys('Laptop')

search_btn=driver.find_element(By.XPATH,"//input[@id='nav-search-submit-button']")
search_btn.click()
```

```
In [4]: Title=[]
Price=[]
Rating=[]
```

```
In [5]: filter_=driver.find_element(By.XPATH,'//i[@class="a-icon a-icon-checkbox"]')
filter_
```

```
Out[5]: <selenium.webdriver.remote.webelement.WebElement (session="d24c496d1d1330d4c6823c0eef81f719", element="f.0D1914980C26DC08A1DE73
D81CC61DBB.d.AFD2F6F89EBB6C45D7B2FDBBE7A67679.e.426")>
```

```
In [15]: title_ = driver.find_elements(By.CLASS_NAME, "a-size-medium a-color-base a-text-normal")
price_ = driver.find_elements(By.CLASS_NAME, "a-price-whole")
rating_ = driver.find_elements(By.CLASS_NAME, "a-row a-size-small")

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

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

for i in rating_:
    rating = i.text
    Rating.append(rating)
    Rating[:10]
```

```
In [16]: print(len(Title[:10]), len(Price[:10]), len(Rating[:10]))

0
```

```
Out[16]: (None, 10, 10)
```

```
In [17]: print(Title)
print(Price)
print(Rating)
```

```
[]
['1,38,990', '69,990', '62,990', '1,14,590', '49,650', '66,290', '99,990', '57,990', '86,490', '79,990', '91,990', '76,990',
'1,04,990', '64,990', '71,990', '31,500', '87,230', '93,990', '39,990', '68,990', '1,00,000', '90,070', '1,60,829', '68,990',
'85,990', '1,38,990', '69,990', '62,990', '1,14,590', '49,650', '66,290', '99,990', '57,990', '86,490', '79,990', '91,990', '7
6,990', '1,04,990', '64,990', '71,990', '31,500', '87,230', '93,990', '39,990', '68,990', '1,00,000', '90,070', '1,60,829', '6
8,990', '85,990']
['', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '']
```

```
In [ ]: df=pd.DataFrame({'Title':Title,'Price':Price,'Rating':Rating})
df
```

```
In [ ]:
```

```
In [ ]: # Q6: Write a python program to scrape data for Top 1000 Quotes of All Time.
```

```
In [74]: driver=webdriver.Chrome()  
driver.get("https://www.azquotes.com/")
```

```
In [75]: top_quotes_button = driver.find_element(By.XPATH, "/html/body/div[1]/div[1]/div[1]/div/div[3]/ul/li[5]/a")  
top_quotes_button.click()
```

```
In [77]: Quotes=[]  
Authors=[]  
Types_Of_Quotes=[]
```

```
In [86]: start=0  
end=10  
for page in range(start,end):  
  
    quotes = driver.find_elements(By.CLASS_NAME,"title")  
    authors = driver.find_elements(By.CLASS_NAME,"author")  
    types_of_quotes = driver.find_elements(By.CLASS_NAME,"tags")  
  
    for i in quotes:  
        quotes_=i.text  
        Quotes.append(quotes_)  
        Quotes[:1000]  
  
    for i in authors:  
        authors_=i.text  
        Authors.append(authors_)  
        Authors[:1000]
```

```
Authors[:1000]

for i in types_of_quotes:
    types_of_quotes_=i.text
    Types_Of_Quotes.append(types_of_quotes_)
    Types_Of_Quotes[:1000]

next_button=driver.find_element(By.XPATH,"/html/body/div[1]/div[2]/div/div/div/div[1]/div/div[4]/li[12]")
next_button.click()
time.sleep(11)
```

```
[87]: print(len(Quotes[:1000])),(len(Authors[:1000])),(len(Types_Of_Quotes[:1000]))
```

312

```
[87]: (None, 300, 300)
```

```
In [71]: print(Quotes)
```

```
['Authors:', 'The essence of strategy is choosing what not to do.', 'One cannot and must not try to erase the past merely because it does not fit the present.', 'Patriotism means to stand by the country. It does not mean to stand by the president.', 'Death is something inevitable. When a man has done what he considers to be his duty to his people and his country, he can rest in peace. I believe I have made that effort and that is, therefore, why I will sleep for the eternity.', 'You have to love a nation that celebrates its independence every July 4, not with a parade of guns, tanks, and soldiers who file by the White House in a show of strength and muscle, but with family picnics where kids throw Frisbees, the potato salad gets iffy, and the flies die from happiness. You may think you have overeaten, but it is patriotism.', 'Be more concerned with your character than your reputation, because your character is what you really are, while your reputation is merely what others think you are.', 'Weak people revenge. Strong people forgive. Intelligent People Ignore.', "A mind is like a parachute. It doesn't work if it is not open.", 'Never be afraid to raise your voice for honesty and truth and compassion against injustice and lying and greed. If people all over the world...would do this, it would change the earth.', 'There are three kinds of men. The one that learns by reading. The few who learn by observation. The rest of them have to pee on the electric fence for themselves.', "A strong nation, like a strong person, can afford to be gentle, firm, thoughtful, and restrained. It can afford to extend a helping hand to others. It's a weak nation, like a weak person, that must behave with bluster and boasting and rashness and other signs of insecurity.", 'The difference between stupidity and genius is that genius has its limits.', 'We the people are the rightful masters of both Congress and the courts, not to overthrow the Constitution but to overthrow the men who pervert the Constitution.', 'With or without religion, you would have good people doing good things and evil people doing evil things. But for good people to do evil things, that takes religion.', 'Human kindness has never weakened the stamina or softened the fiber of a free people. A nation does not have to be cruel to be tough.', 'A person who never made a mistake never tried anything new.', 'The mission in life is not merely to survive, but to thrive; and to do so with some passion, some compassion, some
```

```
In [ ]: df=pd.DataFrame({'Quotes':Quotes,'Authors':Authors,'Types Of Quotes':Types_Of_Quotes})
df
```

```
In [ ]:
```

```
In [ ]: # Q7:Write a python program to display list of respected former Prime Ministers of India
```

```
In [4]: driver=webdriver.Chrome()  
driver.get("https://www.jagranjosh.com/general-knowledge/list-of-all-prime-ministers-of-india-1473165149-1")
```

```
In [5]: name=[]  
born_dead=[]  
term_of_office=[]  
remarks=[]
```



```
In [7]: name_=driver.find_elements(By.XPATH, '/html/body/div[1]/main/div[1]/div[1]/article/div[3]/div[6]/div/table/tbody/tr[2]/td[2]/div/s
born_dead_=driver.find_elements(By.XPATH, '/html/body/div[1]/main/div[1]/div[1]/article/div[3]/div[6]/div/table/tbody/tr[2]/td[3]'
term_of_office_=driver.find_elements(By.XPATH, '/html/body/div[1]/main/div[1]/div[1]/article/div[3]/div[6]/div/table/tbody/tr[2]/t
remarks_=driver.find_elements(By.XPATH, '/html/body/div[1]/main/div[1]/div[1]/article/div[3]/div[6]/div/table/tbody/tr[2]/td[5]')

for i in name_:
    Name=i.text
    name.append(Name)

for i in born_dead_:
    Born_dead=i.text
    born_dead.append(Born_dead)

for i in term_of_office_:
    Term_of_office=i.text
    term_of_office.append(Term_of_office)

for i in remarks_:
    Remarks=i.text
    remarks.append(Remarks)
```

```
In [9]: print(len(name),len(born_dead),len(term_of_office),len(remarks))
```

```
2 2 1 1
```

```
In [10]: df=pd.DataFrame({'Name':name, 'Born-Dead':born_dead, 'Term of Office':term_of_office, 'Remarks':remarks})
df
```

```
In [ ]: # Q8:Write a python program to display list of 50 Most expensive cars in the world
```

```
In [18]: driver=webdriver.Chrome()  
driver.get("https://www.motor1.com/")
```

```
In [26]: title_=driver.find_element(By.CLASS_NAME,"m1-search-panel-input")  
title_.send_keys("50 most expensive cars")  
  
search_=driver.find_element(By.CLASS_NAME,"m1-search-panel-button")  
search_.click()
```

ElementNotInteractableException Traceback (most recent call last)

Cell In[26], line 2

```
1 title_=driver.find_element(By.CLASS_NAME,"m1-search-panel-input")  
----> 2 title_.send_keys("50 most expensive cars")  
4 search_=driver.find_element(By.CLASS_NAME,"m1-search-panel-button")  
5 search_.click()
```

File ~\coding folders\Lib\site-packages\selenium\webdriver\remote\webelement.py:231, in WebElement.send_keys(self, *value)

```
228         remote_files.append(self._upload(file))  
229         value = "\n".join(remote_files)  
--> 231 self._execute(  
232     Command.SEND_KEYS_TO_ELEMENT, {"text": "".join(keys_to_typing(value)), "value": keys_to_typing(value)}  
233 )
```

File ~\coding folders\Lib\site-packages\selenium\webdriver\remote\webelement.py:395, in WebElement._execute(self, command, params)

```
393     params = {}  
394     params["id"] = self._id  
--> 395     return self._parent.execute(command, params)
```

```
GetHandleVerifier [0x00007FF6FEF5A0F+4456463]  
GetHandleVerifier [0x00007FF6FEC005B6+953270]  
(No symbol) [0x00007FF6FEA9E58F]  
(No symbol) [0x00007FF6FEA99264]  
(No symbol) [0x00007FF6FEA9939B]  
(No symbol) [0x00007FF6FEA89BD4]  
BaseThreadInitThunk [0x00007FF9BE48257D+29]  
RtlUserThreadStart [0x00007FF9BF8EAA58+40]
```

```
In [ ]: car_name=[]  
price=[]
```

```
In [ ]: car_name=drive.find_elements(By.XPATH,'//h3[@class="subheader"]')  
price=drive.find_elements(By.XPATH,'')  
  
for i in car_name:  
    car_name=i.text  
    car_name.append(car_name)  
  
for i in price:  
    price=i.text  
    price.append(price)
```

```
In [ ]: print(len(car_name)),(len(price))
```

```
In [ ]: df=pd.DataFrame({'Car Name':car_name,'Price':price})  
df
```