Installing and getting started

- 1. Selenium is a web browser automation tool used for web testing and scraping.
- 2. Install Selenium (https://www.selenium.dev/)) pip install selenim or conda install Application: Web testing and Web Scraping Tool.
- 3. Download Chrome/Firefox(geckodriver)/Safari/Chromium/ Drivers as of your choice For Chrome: check the browser version and download appropriate one.
 - https://chromedriver.chromium.org/ (https://chromedriver.chromium.org/)
- 4. Select a website
 - http://example.webscraping.com/)
 - https://webscraper.io/test-sites (https://webscraper.io/test-sites)
- 5. Getting started with Selenium

Getting started

```
In [30]: import selenium
from selenium import webdriver as wb
In [31]: webD= wb.Chrome("F:\Selenium\chromedriver.exe")
In [32]: webD.get('https://www.google.com')
```

Navigating throught Page

The page element will have "Name", "Class", "Id", "xpath" or some other kind of identifier

Finding element by name

```
In [24]: ele1=webD.find_element_by_name('q')
In [25]: ele1
Out[25]: <selenium.webdriver.remote.webelement.WebElement (session="3e07b6f888cf372b6d93 e7d64fd54ec1", element="5b336dec-e7c9-407a-ba57-ae018b6887e6")>
In [26]: ele1.send_keys('Web Scraping')
In [27]: btnEle=webD.find_element_by_name('btnK')
```

```
In [28]: btnEle.click()
```

Finding element by xpath

//*[@id="tsf"]/div[2]/div[1]/div[1]/div/div[2]/input

/html/body/div[1]/div[3]/form/div[2]/div[1]/div[1]/div/div[2]/input

```
In [33]: eleXP=webD.find_element_by_xpath('/html/body/div[1]/div[3]/form/div[2]/div[1]/div
In [34]: eleXP.send_keys('Web Scraping')
In [35]: eleXP.send_keys('')

/html/body/div[1]/div[3]/form/div[2]/div[1]/div[2]/div[2]/center/input[1]
In [36]: #btnEXpa=webD.find_element_by_xpath('//*[@id="tsf"]/div[2]/div[1]/div[3]/center/input[1]
In [37]: btnEXpa=webD.find_element_by_xpath('/html/body/div[1]/div[3]/form/div[2]/div[1]/div[3]/center/input[1]
In [37]: btnEXpa.click()
```

Finding element by Class

```
eleCla=webD.find element by class name('SDkEP')
In [13]:
In [14]:
         eleCla2=eleCla.find element by class name('a4bIc')
In [15]:
         eleCla2Div=eleCla2.find element by css selector('input')
         eleCla2Div.send keys('Web Scraping')
In [16]:
In [17]: | webD.find_element_by_class_name('FPdoLc')
Out[17]: <selenium.webdriver.remote.webelement.WebElement (session="9608b0431079a74d60f1
         72cf2b6c94e5", element="13adaf43-f9ef-4d8d-aea9-abd371c0b274")>
In [18]:
         centeEle=webD.find_element_by_css_selector('center')
In [19]:
         searhEle=centeEle.find_element_by_class_name('gNO89b')
In [20]: | searhEle.click()
```

Some other methods like

webD.find_element_by_id, webD.find_element_by_tag_name

Filling forms using selenium

https://forms.gle/LRYwqvGx1Uhq5qYt5 (https://forms.gle/LRYwqvGx1Uhq5qYt5)

```
In [24]:
         import selenium
         from selenium import webdriver as wb
         webD=wb.Chrome('chromedriver.exe')
         webD.get('https://forms.gle/LRYwqvGx1Uhq5qYt5')
In [25]: question2=webD.find element by xpath('//*[@id="mG61Hd"]/div/div/div/2]/div[2]/div
In [26]:
         question2Ele=question2.find_element_by_xpath('//*[@id="mG61Hd"]/div/div/div[2]/di
In [27]: question2Ele.text
Out[27]: 'Scraping\nData Science\nOther:'
In [28]:
         optionEle=question2Ele.find element by class name('freebirdFormviewerViewItemsRad
In [29]: eleOptions=optionEle.find elements by class name('freebirdFormviewerViewItemsRadi
In [30]: for i in eleOptions:
             print(i.text)
         Scraping
         Data Science
         Other:
In [31]: for el in eleOptions:
             if el.text == 'Scraping':
                 el.click()
 In [ ]:
```

Select element in Selenium (Not Required)

```
In [ ]: from selenium.webdriver.support.ui import Select
    import selenium
    from selenium import webdriver as wb
    webD=wb.Chrome('chromedriver.exe')
    webD.get('https://forms.gle/LRYwqvGx1Uhq5qYt5')
```

Action Chain

ActionChains are a way to automate low level interactions such as mouse movements, mouse button actions, key press, and context menu interactions. This is useful for doing more complex actions like hover over and drag and drop.

Drag and Drop

```
In [41]: from selenium.webdriver import ActionChains
         import selenium
         from selenium import webdriver as wb
         webD=wb.Chrome('C:\Program Files (x86)\chromedriver.exe')
         # webD=wb.Firefox('C:/Users/swsh/Desktop/selenium Tutorial/')
         # webD.get('https://www.w3schools.com/html/html5 draganddrop.asp')
         webD.get('http://dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html
In [42]: #Madrid
         element = webD.find element by xpath('//*[@id="box7"]')
In [43]: #Spain
         target = webD.find element by xpath('//*[@id="box107"]')
In [44]: | action chains = ActionChains(webD)
In [45]: #Method 1: Drag and Drop
         action chains.drag and drop(element, target).perform()
         #Method 2: simulating Mouse Actions
In [46]:
         action chains.click and hold(element).move to element(target).release(target).per
```

Taking a screenshot of a window

```
In [47]: scShot=webD.get_screenshot_as_png()
In [48]: with open('msis.png','wb') as f:
    f.write(scShot)
```