CSE- 322
Software Engineering Lab

Lab : 05
Testing using Selenium

Fahad Ahmed
Lecturer, Dept. of CSE
E-mail: fahadahmed@uap-bd.edu

WebDriver Navigational Commands

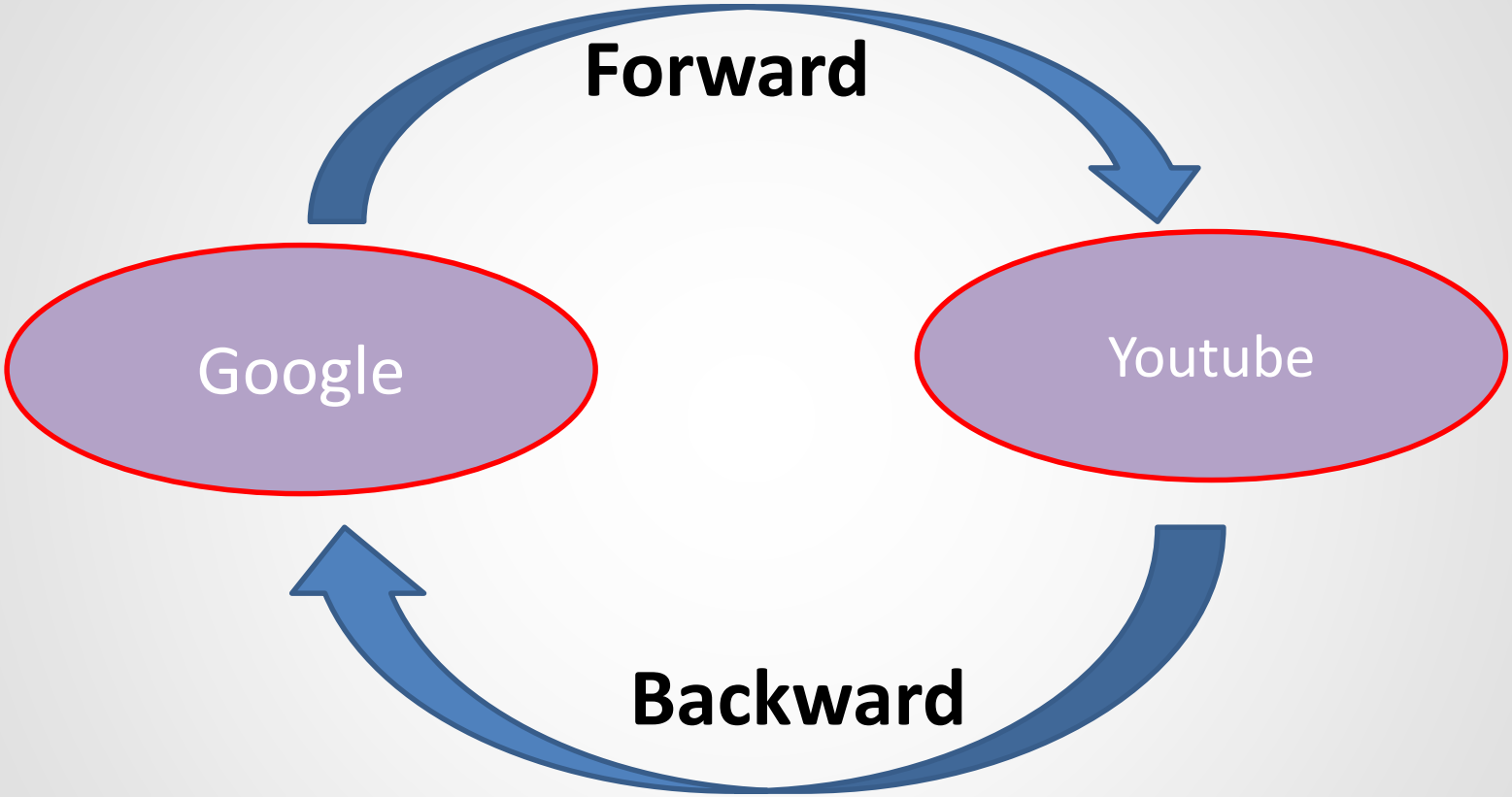
There are four methods defined in the navigation class that is frequently used. They are as follows:

1. **to(String args) [use in Java]**
2. **forward()**
3. **back()**
4. **refresh()**

WebDriver Navigational Commands

driver.get()	driver.navigate().to()
1. The driver.get() method is used to open a specified URL of web application and wait until the whole page gets loaded.	1. The driver.navigate().to() method is used to navigate a specified URL of web application and it does not wait till the whole page gets loaded.
2. It does not maintain the browser history and cookies.	2. It maintains the browser history and cookies.

WebDriver Navigational Commands



WebDriver Navigational Commands

```
from selenium import webdriver
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\drivers\\chromedriver.exe")

driver.get("https://www.google.com/")
print(driver.current_url)

time.sleep(2)
driver.get("https://www.youtube.com/")
print(driver.current_url)
time.sleep(2)

driver.back() # back to google
print("\nBack to : ")
print(driver.title)
time.sleep(2)

driver.forward() # forward to UAP
print("\nforward to :")
print(driver.title)
time.sleep(2)
driver.close()
```

WebDriver Navigational Commands

"C:\Users\Fahad Ahmed\PycharmProjects\SeleniumProject01\venv\Scripts\python.exe"
"C:/Users/Fahad Ahmed/PycharmProjects/SeleniumProject01/MultiBrowser.py"
https://www.google.com/
Google
https://www.youtube.com/
YouTube

Back to :
Google

forward to :
YouTube

Process finished with exit code 0

WebDriver Navigational Commands: Refresh a page

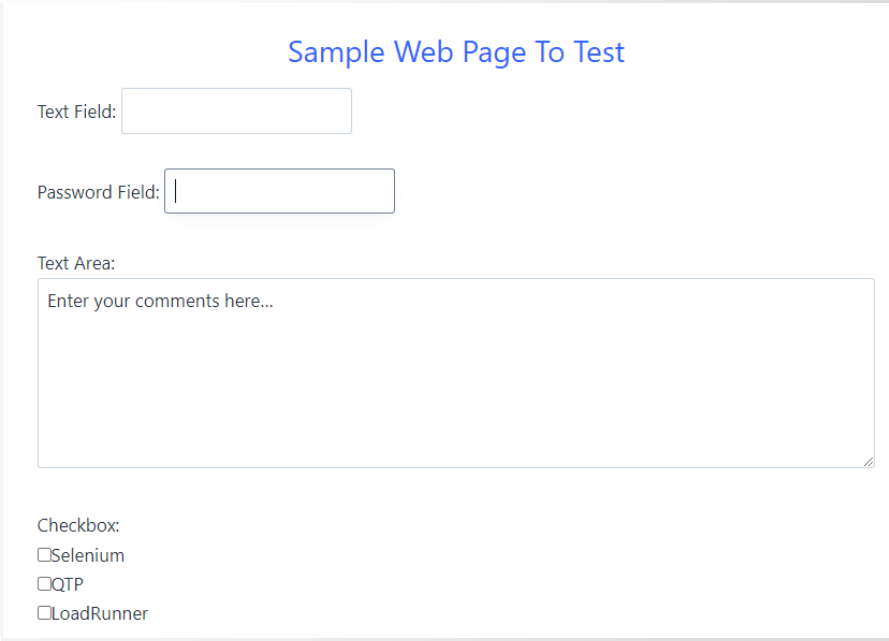
```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\drivers\\chromedriver.exe")

driver.get("https://www.youtube.com/")
print(driver.title)
time.sleep(2)

driver.refresh()
print("\nRefresh the page :")
time.sleep(2)
print(driver.title)
driver.close()
```

Locators in Selenium Python

Locator	Description
tag name	Locates elements whose tag name matches the search value
css selector	Locates elements matching a CSS selector
class name	Locates elements whose class name contains the search value (compound class names are not permitted)
id	Locates elements whose ID attribute matches the search value
name	Locates elements whose NAME attribute matches the search value
link text	Locates anchor elements whose visible text matches the search value
partial link text	Locates anchor elements whose visible text contains the search value. If multiple elements are matching, only the first one will be selected.
xpath	Locates elements matching an XPath expression

Locators in Selenium Python

https://opensource-demo.orangehrmlive.com/web/index.php/auth/login



Locators in Selenium Python: built-in locator strategies

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\drivers\\chromedriver.exe")

    # Navigate to Url
driver.get("https://www.example.com")

    # Get all the elements available with tag name 'p'
elements = driver.find_elements(By.TAG_NAME, 'p')
time.sleep(3)
for e in elements:
    print(e.text)
```

https://www.selenium.dev/documentation/webdriver/elements/finders/

Locators in Selenium Python: built-in locator strategies

"C:\Users\Fahad Ahmed\PycharmProjects\Selenium-01\venv\Scripts\python.exe"
"C:/Users/Fahad Ahmed/PycharmProjects/Selenium-01/main.py"
C:\Users\Fahad Ahmed\PycharmProjects\Selenium-01\main.py:4: DeprecationWarning:
executable_path has been deprecated, please pass in a Service object
  driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\drivers\\chromedriver.exe")

This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.
More information...

https://www.selenium.dev/documentation/webdriver/elements/finders/

Locators in Selenium Python: By.CSS_SELECTOR

```
from selenium.webdriver.common.by import By
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\drivers\\chromedriver.exe")

driver.get("https://www.google.com")
driver.find_element(By.CSS_SELECTOR, '[name="q"]').send_keys("webElement")
time.sleep(3)

# Get attribute of current active element
attr = driver.switch_to.active_element.get_attribute("title")
print(attr)
```

Locators in Selenium Python: By.CSS_SELECTOR

"C:\Users\Fahad Ahmed\PycharmProjects\Selenium-01\venv\Scripts\python.exe"
"C:/Users/Fahad Ahmed/PycharmProjects/Selenium-01/main.py"
C:\Users\Fahad Ahmed\PycharmProjects\Selenium-01\main.py:4: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
  driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad Ahmed\\PycharmProjects\\drivers\\chromedriver.exe")

সার্চ করুন

Interacting with web elements

There are some basic commands that can be executed on an element:

- click (applies to any element)
- send keys (only applies to text fields and content editable elements)
- clear (only applies to text fields and content editable elements)
- submit (only applies to form elements)

Interacting with web elements:perform "ENTER" keyboard action

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\drivers\\chromedriver.exe")

 # Navigate to url
driver.get("http://www.google.com")
time.sleep(4)

    # Enter "webdriver" text and perform "ENTER" keyboard action
driver.find_element(By.NAME, "q").send_keys("webdriver" + Keys.ENTER)
```

Interacting with web elements: Clears the entered text

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\drivers\\chromedriver.exe")

# Navigate to url
driver.get("http://www.google.com")
# Store 'SearchInput' element
time.sleep(2)
SearchInput = driver.find_element(By.NAME, "q")
SearchInput.send_keys("selenium")
time.sleep(2)
# Clears the entered text
SearchInput.clear()
```

Information about web elements

Is Displayed

This method is used to check if the connected Element is displayed on a webpage. Returns a Boolean value, True if the connected element is displayed in the current browsing context else returns false.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\drivers\\chromedriver.exe")

# Navigate to the url
driver.get("https://www.selenium.dev/selenium/web/inputs.html")
time.sleep(2)
# Get boolean value for is element display
is_email_visible = driver.find_element(By.NAME, "email_input").is_displayed()
time.sleep(2)
print(is_email_visible)
```

Information about web elements

Is Displayed

This method is used to check if the connected Element is displayed on a webpage. Returns a Boolean value, True if the connected element is displayed in the current browsing context else returns false.

```
"C:\Users\Fahad Ahmed\PycharmProjects\Selenium-01\venv\Scrip
C:\Users\Fahad Ahmed\PycharmProjects\Selenium-01\main.py:5:
  driver = webdriver.Chrome(executable_path="C:\\Users\\Faha
True

Process finished with exit code 0
```

Information about web elements

Is Enabled

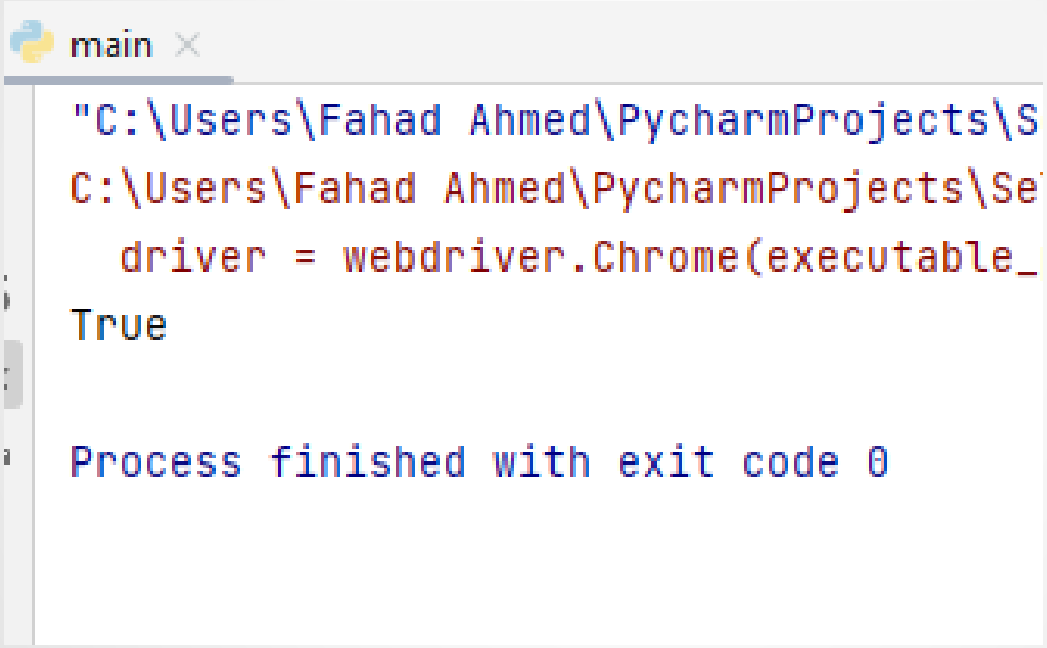
This method is used to check if the connected Element is enabled or disabled on a webpage. Returns a boolean value, True if the connected element is enabled in the current browsing context else returns false.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\drivers\\chromedriver.exe")

    # Navigate to url
driver.get("http://www.google.com")

    # Returns true if element is enabled else returns false
value = driver.find_element(By.NAME, 'btnK').is_enabled()

print(value)
```

Information about web elements

Is Enabled

This method is used to check if the connected Element is enabled or disabled on a webpage. Returns a boolean value, True if the connected element is enabled in the current browsing context else returns false.



Information about web elements

Size and Position

It is used to fetch the dimensions and coordinates of the referenced element.
The fetched data body contain the following details:
X-axis position from the top-left corner of the element
y-axis position from the top-left corner of the element
Height of the element
Width of the element

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\drivers\\chromedriver.exe")

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

    # Returns TagName of the element
res = driver.find_element(By.CSS_SELECTOR, "h1").rect
time.sleep(2)
print(res)
```

Information about web elements

Size and Position

It is used to fetch the dimensions and coordinates of the referenced element.
The fetched data body contain the following details:
X-axis position from the top-left corner of the element
y-axis position from the top-left corner of the element
Height of the element
Width of the element

```
"C:\Users\Fahad Ahmed\PycharmProjects\Selenium-01\venv\Scri
C:\Users\Fahad Ahmed\PycharmProjects\Selenium-01\main.py:5:
  driver = webdriver.Chrome(executable_path="C:\\Users\\Fah
{'height': 43, 'width': 600, 'x': 217, 'y': 133.4375}

Process finished with exit code 0
```

Information about web elements

Get CSS Value

Retrieves the value of specified computed style property of an element in the current browsing context.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\drivers\\chromedriver.exe")

driver.get("https://www.selenium.dev/documentation/webdriver/elements/file_upload/")

# Retrieves the computed style property 'color' of linktext
cssValue = driver.find_element(By.LINK_TEXT, "File Upload").value_of_css_property('color')
time.sleep(2)
print(cssValue)
```

rgba(73, 80, 87, 1)

Web Element In Selenium: XPATH, CSS_SELECTOR

Web Element In Selenium: XPATH, CSS_SELECTOR

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\SeleniumProject01\\drivers\\chromedriver.exe")

driver.get("https://opensource-demo.orangehrmlive.com/")
print(driver.current_url)

driver.find_element(By.XPATH, "//input[@id='txtUsername']").send_keys("Admin")
print("find username field\n")
driver.find_element(By.CSS_SELECTOR,"input[type='password']").send_keys("admin123")
print("find password field\n")
time.sleep(2)
driver.find_element(By.XPATH, "//input[@value='LOGIN']").click()
print("find username field\n")

driver.find_element(By.XPATH,"//a[contains(text(),'Welcome')]").click()
print("find welcome text \n")
time.sleep(2)
driver.find_element(By.XPATH,"//a[contains(text(),'Logout')]").click()
print("After Logout ", driver.current_url)
assert "login" in driver.current_url
driver.close()
```

The `assert` keyword lets you test if a condition in your code returns True, if not, the program will raise an AssertionError.

Web Element In Selenium:LINK_TEXT

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\SeleniumProject01\\drivers\\chromedriver.exe")

driver.get("https://opensource-demo.orangehrmlive.com/")
print(driver.current_url)
time.sleep(2)
driver.find_element(By.LINK_TEXT,"Forgot your password?").click()
print("Click on: Forgot your password?\n")
driver.close()
```

Web Element In Selenium: PARTIAL_LINK_TEXT

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\SeleniumProject01\\drivers\\chromedriver.exe")

driver.get("https://opensource-demo.orangehrmlive.com/")
print(driver.current_url)
time.sleep(2)
driver.find_element(By.PARTIAL_LINK_TEXT,"Forgot your").click()
print("Click on: Forgot your password?\n")
time.sleep(2)
driver.close()
```

Web Element In Selenium: TAG_NAME

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\SeleniumProject01\\drivers\\chromedriver.exe")

driver.get("https://opensource-demo.orangehrmlive.com/")

time.sleep(2)
driver.find_element(By.TAG_NAME,"a").click()
print(driver.current_url)
time.sleep(2)
driver.close()
```

Select class for multiple select dropdown

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.select import Select
import time
driver=webdriver.Chrome(executable_path="C:\\Users\\Fahad Ahmed\\PycharmProjects\\web-drivers\\chromedriver.exe")

# get method to launch the URL
driver.get ("https://www.softwaretestingmaterial.com/sample-webpage-to-automate/")
# select class provide the methods to handle the options in dropdown
s = Select (driver.find_element_by_name ("multipleselect[]"))
# select an option with visible text method
s.select_by_visible_text ("Performance Testing")
time.sleep(2)

# deselect an option with visible text method
s.deselect_by_visible_text ("Performance Testing")
time.sleep (2)
# select an option with index method
s.select_by_index(2)
time.sleep(2)
# deselect an option with index method
s.deselect_by_index(2)
time.sleep(2)
# select an option with value method
s.select_by_value ("msselenium")
time.sleep(2)
# deselect an option with value method
s.deselect_by_value ("msselenium")
time.sleep (2)
# select an option with value method
s.select_by_value ("msagile")
time.sleep (2)
# deselect all options
s.deselect_all ()
time.sleep(2)
```

Web Element In Selenium: screenshot

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
from PIL import Image  ## pip install Pillow

driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\SeleniumProject01\\drivers\\chromedriver.exe")

driver.get("https://opensource-demo.orangehrmlive.com/")

time.sleep(2)
driver.save_screenshot("image.png")

# Loading the image
image = Image.open("image.png")

# Showing the iamge
image.show()
driver.close()
```

Web Element In Selenium: screenshot

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
from PIL import Image ## pip install pillow

driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\SeleniumProject01\\drivers\\chromedriver.exe")

driver.get("https://opensource-demo.orangehrmlive.com/")


element = driver.find_element_by_id("divLogo");
location = element.location;
size = element.size;
driver.save_screenshot("logoImage.png");

# crop image
x = location['x'];
y = location['y'];
width = location['x']+size['width'];
height = location['y']+size['height'];
im = Image.open('logoImage.png')
im = im.crop((int(x), int(y), int(width), int(height)))
im.save('logo-section-Image.png')

driver.quit()
```

Web Element In Selenium: Report Generation

```
from selenium import webdriver
import unittest
import HTMLTestRunner    ## pip install html-testRunner in command line
from selenium.webdriver.common.by import By

class TestReport(unittest.TestCase):

    @classmethod
    def setUpClass(cls):
        cls.driver = webdriver.Chrome(executable_path="C:\\Users\\Fahad
Ahmed\\PycharmProjects\\SeleniumProject01\\drivers\\chromedriver.exe")
        cls.driver.implicitly_wait(2)

    def test_search_one(self):
        self.driver.get("https://opensource-demo.orangehrmlive.com/")
        self.driver.find_element(By.LINK_TEXT,"Forgot your password?").click()

    def test_search_two(self):
        self.driver.get("https://opensource-demo.orangehrmlive.com/")
        self.driver.find_element(By.LINK_TEXT, "Forgot your").click()

    @classmethod
    def tearDownClass(cls):
        cls.driver.close()
        cls.driver.quit()
        print("Test Completed")

if __name__ == '__main__':
    unittest.main(testRunner=HTMLTestRunner.HTMLTestRunner(output='C:/Users/Fahad
Ahmed/PycharmProjects/SeleniumProject01/Reports'))
```

Web Element In Selenium: Report Generation

```
from selenium import webdriver
import unittest
import HTMLTestRunner    ## pip install html-testRunner in command line
from selenium.webdriver.common.by import By

class TestReport(unittest.TestCase):

    @classmethod
    def setUpClass(cls):
```

Unittest Results

Start Time: 2021-10-02 11:46:52

Duration: 11.86 s

Summary: Total: 3, Pass: 2, Error: 1

__main__.TestReport	Status	
test_Case_1	Error	View
test_Case_2	Pass	
test_Case_3	Pass	

Total: 3, Pass: 2, Error: 1 -- Duration: 11.86 s

Thanks to All