# Selenium With Python

## Selenium WebDriver (Function Testing) :

"To begin using Selenium, the first steps include installing Python and PyCharm. Once these are set up, you can proceed to install the Selenium package by navigating to 'File,' then 'Settings,' and selecting 'Python Interpreter' to install the package."

driver.get("https:// username : password@ websitename.com") try this one .

'We Should download selenium Browser's from "https://www.selenium.dev/downloads/"



OZIH UDDIN AL FUZAYEL

### Run Test

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()
#driver = webdriver.Firefox()
#driver = webdriver.Ie()

driver.get("https://google.com")
driver.maximize_window()

print(driver.title) #title of the page
print(driver.current_url) #current url of the page

#driver.close()
driver.quit()
```

### 1 : Navigation Commands

```
driver.back()
driver.forward()
driver.refresh()
```

## 2 : Conditional Commands

```
element = driver.find_element(By.XPATH,
"//*[@id='HTML1']")
print(element.is_displayed())
print(element.is_enabled())
print(element.is_selected())  # This 3 will return True
or False
```

## 3 : .send_Keys(), .clear() , .Click()

```
import time #This import is for time.sleep(5)
element = driver.find_element(By.ID, "name")
element = driver.clear() # This is for clear previous
Data..
element.send_keys("OZIH UDDIN AL FUZAYEL ") #SEND KEYS
("values")


element2 = driver.find_element(By.ID, "sunday")
element2.click() #click()
time.sleep(5)
```

## 4 :Wait's

1) Implicit wait
2) Explicit wait  (Not explained)

-Implicit wait :

```
from selenium import webdriver

driver = webdriver.Chrome()
driver.get("https://examle.com/")

driver.implicitly_wait(10) # This is to implicitly wait
after driver.Get ....

driver.quit()
```

## 5: Input boxes

1) How many input boxes are there?
2) How to provide value.
3) How to get the status.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
driver = webdriver.Chrome()
driver.get("https://fs2.formsite.com/meherpavan/form2/ind
ex.html?1537702596407")

# How many input boxes are there?
input_boxes = driver.find_elements(By.CLASS_NAME,
"text_field")
print("Total input = ", len(input_boxes))

# How to provide value in the text box :
driver.find_element(By.XPATH,
"//*[@id='RESULT_TextField-1']").send_keys("ozih uddin al
fuzayel")

#get_attribute('value')
print(element.get_attribute('value') #it will provide
value of the input Boxes.which will be placeholder.

# How to get the status of input box :
```

```
status = driver.find_element(By.XPATH,
"//*[@id='RESULT_TextField-1']").is_displayed()
print(status)

driver.quit()
```

## 6: Working with Radio Buttons & Check_boxes

1) Radio Button Select or not .
2) Click or not ..

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
driver = webdriver.Chrome()
driver.get("https://fs2.formsite.com/meherpavan/form2/ind
ex.html?1537702596407")

# Start Radio Button
status = driver.find_element(By.ID,
"RESULT_RadioButton-8_0").is_selected()
print(status)

driver.find_element(By.ID,
"RESULT_RadioButton-8_0").click()
# End Radio Button

# Start Check Box
driver.find_element(By.XPATH,
"//*[@id='RESULT_CheckBox-8_0']").click()
# End Checkbox
```

## 7: Drop_Down

1) Select One Option .
2) Find How many options there are .
3) Capture all Options .

```
from selenium import webdriver
from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import Select #This is
for select

driver = webdriver.Chrome()
driver.get("https://fs2.formsite.com/meherpavan/form2/ind
ex.html?1537702596407")

#########From here
dropdown =
Select(driver.find_element(By.ID,"RESULT_RadioButton-9"))
#dropdown.select_by_visible_text("Evening") #Use Visible
text for select or use by value or use index number.
#or
#dropdown.select_by_value("Radio-1")
#or
dropdown.select_by_index(2)
######## to here important and this is for select;

# Count number of options
print(len(dropdown.options)) # uporer variable dropdown

# Here is Capture all Option from drop down :
all_options = dropdown.options
for i in all_options:
print(i.text)

driver.quit()
```

## 8: Links, .text & get_attribute('value)

1) How many links .
2) Capture all the links.
3) Click on the links .

**4).text (element.text) is for text values like = print(link.text)**

```
from selenium import webdriver
from selenium.webdriver.common.by import By
driver = webdriver.Chrome()
driver.get("https://testautomationpractice.blogspot.com/"
)

Links = driver.find_elements(By.TAG_NAME, "a")
print("Total Links are : ", len(Links)) #len of links

for link in Links: #This is for print link text ..
print(link.text) # .text is for text_values

driver.find_element(By.LINK_TEXT, "Blogger").click()
#click the link.

driver.quit()
```

## 9: Alert's / popups

```
from selenium import webdriver
from selenium.webdriver.common.by import By
driver = webdriver.Chrome()
driver.get("https://testautomationpractice.blogspot.com/"
)

driver.find_element(By.XPATH,
"//*[@id='HTML9']/div[1]/button[2]").click()
#alert start
alert = driver.switch_to.alert #This must need for alert

alert.send_keys("i am fuzayel") #if there any input in
alert.#

alert.accept() # this will click ok button
```

```
alert.dismiss() # this will click cancel button
#end alert


#For authentication alert we need to this :
driver.get("https://username : password @ test.com)


driver.quit()
```

## 10: Frame / iframe

1) driver.switch_to.frame("id or name of the frame or index number of the frame")
2) driver.switch_to.default_content()

```
from selenium import webdriver
from selenium.webdriver.common.by import By
driver = webdriver.Chrome()
driver.get("https://example.com/)")
#frame one start
driver.switch_to.frame("first_frame_name's id or xpath just id name or index number ")#it will switch to the frame.
driver.find_element(By.ID, "id name").click() # it will click the link under the frame

driver.switch_to.default_content() # it will come out from frame
#frame one end

driver.switch_to.frame("Just id name or index number of frame") # it will go another frame
driver.find_element(By.LINK_TEXT, "id name").click() # it will click the link under the frame
driver.switch_to.default_content() # again it will come out from frame
driver.close()
```

## 11: Handle Browser windows

```
from selenium import webdriver
from selenium.webdriver.common.by import By
driver = webdriver.Chrome()
driver.get("https://testautomationpractice.blogspot.com/)
")

driver.find_element(By.XPATH,
"//*[@id='HTML4']/div[1]/button").click()
print(driver.current_window_handle) # Current window
handle

all_values = driver.window_handles # it will return all
the handles values of open browser windows .
print(all_values)

for value in all_values:
driver.switch_to.window(value)
print(value.title())

if driver.title == "title name ":
driver.close() # close specific window

driver.quit()
```

## 12: Table

1) How many row's and col's are there?
2) Read Table data rows and cols.
3) Read Data based on condition

```
from selenium import webdriver
from selenium.webdriver.common.by import By
driver = webdriver.Chrome()
driver.get("https://testautomationpractice.blogspot.com/)
")
```

```
'''
element = driver.find_elements(By.XPATH,
"//*[@id='HTML1']/div[1]/table/tbody/tr")
rows = len(element)
print("Total Rows = ", rows)

element2 = driver.find_elements(By.XPATH,
"//*[@id='HTML1']/div[1]/table/tbody/tr[1]/th")
cols = len(element2)
print("Total Columns is = ", cols)
'''

# How to read table data :

table = driver.find_element(By.NAME, "BookTable")
rows = table.find_elements(By.TAG_NAME, "tr")

for row in rows:
cells = row.find_elements(By.TAG_NAME, "td")

for cell in cells:
print(cell.text)
# End How to read table data..



driver.quit()
```

## 13: Mouse Action & .perform()

**1)Mouse Hover.**
**2)Double click ().**
**3)Right Click.**
**4)Drag and Drop().**

**-Mouse Hover :**

```
from selenium import webdriver
from selenium.webdriver.common.by import By
```

```
from selenium.webdriver import ActionChains #This import
is for mouse hover

driver = webdriver.Chrome()
driver.get("https://wordpress.org/")

#here are selector :=

menu = driver.find_element(By.XPATH,
"//*[@id='modal-1-content']/ul/li[2]/a")
sub1 = driver.find_element(By.XPATH,
"//*[@id='modal-1-content']/ul/li[2]/ul/li[1]/a/span")
sub2 = driver.find_element(By.XPATH,
"//*[@id='modal-1-content']/ul/li[2]/ul/li[2]/a/span")
sub3 = driver.find_element(By.XPATH,
"//*[@id='modal-1-content']/ul/li[2]/ul/li[3]/a/span")

#here important things for mouse hover :

action = ActionChains(driver)
action.move_to_element(menu).move_to_element(sub1).move_t
o_element(sub2).move_to_element(sub3).click().perform()

# in the last .perform() must need

driver.quit()
```

**-Double Click :**

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver import ActionChains #Import this
one

driver = webdriver.Chrome()
driver.get("https://testautomationpractice.blogspot.com/"
)
driver.maximize_window()
```

```
element = driver.find_element(By.XPATH,
"//*[@id='HTML10']/div[1]/button")

action = ActionChains(driver)
#Double click()
action.double_click(element).perform()

driver.close()
```

**-Right Click.**

**For right click use =**
**action.context_click(element).perform()**

**-Drag and Drop.**

```
from selenium import webdriver
from selenium.webdriver.common.by import By

from selenium.webdriver import ActionChains #Import this
one

driver = webdriver.Chrome()
driver.get("https://testautomationpractice.blogspot.com/"
)
driver.maximize_window()

source_element = driver.find_element(By.XPATH,
"//*[@id='draggable']")

target_element = driver.find_element(By.XPATH,
"//*[@id='droppable']")

action = ActionChains(driver)
action.drag_and_drop(source_element,target_element).perfo
rm() # perform need

driver.close()
```

## 14: Upload the file

**-Upload image or file:**

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
driver = webdriver.Edge()
driver.get("https://www.foundit.in/")
driver.maximize_window()

element = driver.find_element(By.XPATH,
"//div[@class='heroSection-buttonContainer_secondaryBtn
__text']")
element.click()

element2 = driver.find_element(By.XPATH,
"//*[@id='file-upload']").send_keys("C:/Users/Fuzayel/D
ownloads/fuzayel.pdf")

time.sleep(5)
```

## 15: Cookies

1) Length of cookies
2) Create a new cookies
3) Delete a / all cookies

```
from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()
driver.get("https://www.amazon.in/")
driver.implicitly_wait(10)
driver.maximize_window()

#capture all the cookies :
cookies = driver.get_cookies()
print("Total cookies : ", len(cookies))
```

```
# Add new cookie :
cookie = {'name': 'Mycookie', 'value': '123456'}
driver.add_cookie(cookie)

cookies = driver.get_cookies()
print("after create a new cookie total :", len(cookies))
print(cookies)

# Delete cookie :
driver.delete_cookie("Mycookie")

cookies = driver.get_cookies()
print("after delete a new cookies total : ",
len(cookies))

#delete all cookies :
driver.delete_all_cookies()
cookies = driver.get_cookies()
print("after delete all the cookies : ", len(cookies))
```

## 16: Capture Screenshot's

**-top of the page**

```
from selenium import webdriver
driver = webdriver.Chrome()
driver.get("https://www.amazon.in/")

#screen short in your own location
driver.save_screenshot("C://Users/OZIHUDDIN
ALFUZAYEL/Pictures/Screenshots/home.png")

driver.get_screenshot_as_file("C://Users/OZIHUDDIN
ALFUZAYEL/Pictures/Screenshots/main.png")
```

```
# Get Screenshot in this directory use this code (must
be //name.png)
#import os
#driver.save_screenshot(os.getcwd() + "//fun.png")
```

**-after scroll down 3000px we can use :**

`driver.execute_script("window.scrollBy(0,3000)", " ")`

```
from selenium import webdriver
driver = webdriver.Chrome()
driver.get("https://www.amazon.in/")
driver.maximize_window()

driver.execute_script("window.scrollBy(0,3000)", " ")
#this command is for scrolling down ..

#screenshot in your own location
driver.save_screenshot("C://Users/OZIHUDDIN
ALFUZAYEL/Pictures/Screenshots/again.png")

#driver.get_screenshot_as_file("C://Users/OZIHUDDIN
ALFUZAYEL/Pictures/Screenshots/main.png")
```

## 17: Scrolling Down

```
from selenium import webdriver
from selenium.webdriver.common.by import By
driver = webdriver.Chrome()
driver.get("https://www.amazon.in/")
driver.maximize_window()

#scroll Down page 3000 px
driver.execute_script("window.scrollBy(0, 1000);")
#this command is for scroll down ..
```

```
# scroll down page till find the element :
element = driver.find_element(By.LINK_TEXT, "See all
offers")
driver.execute_script("arguments[0].scrollIntoView();",
element)

# scroll till the last :
driver.execute_script("window.scrollBy(0,
document.body.scrollHeight)")
```

## 18: Disable Opening Notification

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
import time

# Start Set up Chrome options
ops = Options()
ops.add_argument("--disable-notifications")
driver = webdriver.Chrome(options=ops)
# End Set up Chrome options

driver.get("https://whatmylocation.com/")
driver.maximize_window()
time.sleep(2)
driver.quit()
```

## 19: switch to new window or tab

```
from selenium import webdriver
driver = webdriver.Chrome()
driver.get("https://google.com")

#driver.switch_to.new_window("tab")
```

```
driver.switch_to.new_window("window")

driver.get("another website.")

#emni=>

Try:
    print("pass")

except Exception as e:
    print("An error occurred:", str(e))
```

## 20: CheckBox

1) How to click specific checkbox
2) How to click all checkbox

```
from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()
driver.get("https://testautomationpractice.blogspot.com
/")

# For clicking specific checkbox :
driver.find_element(By.XPATH,
"//*[@id='post-body-1307673142697428135']/div[4]/div[4]
/label").click()

#For clicking all checkbox we need to get all Xpath:
element = driver.find_elements(By.XPATH,
"//*[@id='post-body-1307673142697428135']/div[4]/div")

for ele in element:
print(ele.text)
ele.click()
```

```
#This is for UnCheck The check Boxes
for ele in element:
if ele.is_selected():
ele.click()

driver.quit()
```

## 21: Slider

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import
ActionChains
driver = webdriver.Chrome()
driver.get("https://testautomationpractice.blogspot.com
/")
driver.maximize_window()

#from here :

element = driver.find_element(By.XPATH,
"//*[@id='slider']/span")
act = ActionChains(driver)

print("Before sliding ", element.location) #{'x': 805,
'y': 1095}

act.drag_and_drop_by_offset(element, 100, 0).perform()

print("after sliding ", element.location) #{'x': 905,
'y': 1095}
driver.close()
```

# 22: KeyBoard Action.Perform()

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import
ActionChains
from selenium.webdriver.common.keys import Keys

driver = webdriver.Chrome()
driver.implicitly_wait(10)
driver.get("http://textcompare.com/")
driver.maximize_window()

element = driver.find_element(By.NAME, "frm_compare_1")
element2 = driver.find_element(By.NAME,
"frm_compare_2")

action = ActionChains(driver)

element.send_keys("OZIH UDDIN AL FUZAYEL ")

action.key_down(Keys.CONTROL).send_keys("a").key_up(Key
s.CONTROL).perform() #Ctrl+a

action.key_down(Keys.CONTROL).send_keys("c").key_up(Key
s.CONTROL).perform() #Ctrl+c

action.move_to_element(element2).click().perform()
#move next element

action.key_down(Keys.CONTROL).send_keys("v").key_up(Key
s.CONTROL).perform() #Ctrl+v

driver.quit()
```

## 23: Download File .exe file and PDF

**- For Chrome and Edge in desire Location .exe:**

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
import os
location = os.getcwd()
try:
def chrome_setup():
preferences = {"download.default_directory": location}
ops = webdriver.ChromeOptions()
ops.add_experimental_option("prefs", preferences)
driver = webdriver.Chrome(options=ops)
return driver


driver = chrome_setup()

driver.get("https://file-examples.com/index.php/sample-
documents-download/sample-doc-download/")
driver.maximize_window()
driver.find_element(By.XPATH,
"//*[@id='table-files']/tbody/tr[2]/td[5]/a").click()
time.sleep(5)
print("Test Pass")
except:
print("Test Fail")
```

**-DownLoad File For Mozilla FireFox and ignore notification:**

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
try:
```

```
def fireFox_setup():
ops = webdriver.FirefoxOptions()
#this 2 line for ignor notification
ops.set_preference("browser.helperApps.neverAsk.saveToDisk", "application/msword") #application/pdf , application/txt
ops.set_preference("browser.download.manager.showWhenStarting", False)

#ops.set_preference("pdfjs.disabled", True) #This is for PDF Only

driver = webdriver.Firefox(options=ops)
return driver


driver = fireFox_setup()

driver.get("https://file-examples.com/index.php/sample-documents-download/sample-doc-download/")
driver.maximize_window()
driver.find_element(By.XPATH, "//*[@id='table-files']/tbody/tr[2]/td[5]/a").click()
time.sleep(5)
print("Test Pass")
except:
print("Test Fail")
```

- **DownLoad PDF File :**

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
try:
def chrome_setup():
ops = webdriver.ChromeOptions()
```

```
# Set download preferences for PDF files
prefs = {"plugins.always_open_pdf_externally": True}
ops.add_experimental_option("prefs", prefs)

driver = webdriver.Chrome(options=ops)
return driver

driver = chrome_setup()

driver.get("https://file-examples.com/index.php/sample-
documents-download/sample-pdf-download/")
driver.maximize_window()
driver.find_element(By.XPATH,
"//tbody/tr[1]/td[5]/a[1]").click()
time.sleep(5)
print("Test Pass")
except:
print("Test Fail")
```

## 24: HeadLess Test(without Browser)

```
from selenium import webdriver
def headless_chrome():
ops = webdriver.ChromeOptions()
ops.headless = True
driver = webdriver.Chrome(options=ops)
return driver

driver = headless_chrome()

driver.get("https://fuzayel.blogspot.com/")
print(driver.title)
driver.quit()
```

*Ozih Uddin Al Fuzayel*

*B.S.S Hons in Economics*

# Pytest

Goal of Pytest is :

1) Re Usability
2) Maintainability.

Pytest is a Framework : Framework is an organised way of maintaining automation files .

We need to install packages: { and file name test_name.py}

- Selenium
- Pytest
- pytest-html

001 ) For Selenium with Pytest we first create a File the name of { test_fileNamee.py} like this and in this file we create an Automation Test.

1)—----------Automation login test Code Like this :----------

```
Import pytest
from selenium import webdriver

@pytest.fixture()
def driver():
        browser = webdriver.Chrome()
        yield driver

def test_one(driver):
        code for test

def test_two(driver):
        code for test

#####For generate html report : #####
if __name__==”__main__”:
        pytest.main([“-v”, “--html=testresult.html”])
```

—----------Code Like this—----------

-Automation Simple Test :

```
import pytest
from selenium import webdriver
import time
@pytest.fixture()
def browser():
    driver = webdriver.Chrome()
    driver.get("https://google.com")
    driver.implicitly_wait(10)
    driver.maximize_window()
    yield driver
    driver.quit()

def test_one(browser):
    assert "Google" in browser.title

def test_two(browser):
    element = browser.find_element("name", "q")
    element.send_keys("Automation")
    element.submit()
    time.sleep(5)
    browser.back()

if __name__== "__main__" :
    pytest.main(["-v", "--html=report.html"])
```

E-Commerce web :

```
import pytest
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
import os
@pytest.fixture()
def browser():
    driver = webdriver.Chrome()
    driver.get("https://admin-demo.nopcommerce.com/")
    driver.implicitly_wait(10)
    driver.maximize_window()
    yield driver
    driver.quit()
def test_login(browser):
    user_email = browser.find_element(By.ID, "Email")
    user_pass = browser.find_element(By.ID, "Password")
    button = browser.find_element(By.CLASS_NAME, "button-1")
    user_email.clear()
```

```
    user_pass.clear()
    user_email.send_keys("admin@yourstore.com")
    user_pass.send_keys("admin")
    time.sleep(2)
    button.click()
    act_title = browser.title
    if act_title == "Dashboard / nopCommerce administration":
        browser.save_screenshot(os.getcwd() + "//fun.png")
        assert True
    else:
        assert False
    time.sleep(5)
    Log_out = browser.find_element(By.LINK_TEXT, "Logout")
    Log_out.click()
    time.sleep(2)

if __name__ == "__main__":
    pytest.main(["-v", "--html=reporttest.html"])
```

2 )—---------- Capture Screenshot on failure —---------—----------

Import os

if actual_title ==”main title”:

assert True

else:

driver.save_screanshort(`os.getcwd() + "//fun.png"`)

assert False

---------- Capture Screenshot on failure —----------

## Login Test with Different Value by using Pytest

3)----------parameterized testing with pytest / Data Driven Testing—----------

- Parametrization is a technique that can be used to run the same test with different values.

- Testing a login page with different values is often referred to as "parameterized testing" or "data-driven testing."

----------Code : ----------

@pytest.mark.parametrize(“username , password”,[

(“mainesername”, “mainpassword”),

(“fakeUsername” , “fake password”),
])

def test_login(driver, username, password):
Then code ----------

–Login Test with Different value :

```
import pytest
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time

@pytest.fixture()
def driver():
   driver = webdriver.Chrome()
   driver.implicitly_wait(10)
   driver.maximize_window()
   yield driver

#THIS IS FOR DIFFERENT VALUE : {user and pass is =test}
@pytest.mark.parametrize("username, password",[
   ("test", "test"),
   ("fakeUser", "fakePassword"),
   ("test1", "test1"),
])
#End Parametrized

# We Must provide Username and password in Parameter :
def test_Login_page(driver, username, password):
   driver.get("https://trytestingthis.netlify.app/")
   user_name = driver.find_element(By.ID, "uname")
   user_pass = driver.find_element(By.ID, "pwd")
   login_button = driver.find_element(By.XPATH,
"//input[@value='Login']")
   user_name.send_keys(username)
   user_pass.send_keys(password)
   login_button.click()
   time.sleep(4)
   assert "Login Successful" in driver.page_source
   print("Login Passed ")
```

https://seleniumloginpytest.netlify.app/