

SELENIUM REPORT

```
In [2]: import time
from selenium import webdriver as wd
# to send keys
from selenium.webdriver.common.keys import Keys
```

```
driver = wd.Chrome(ChromeDriverManager().install())
```

CODE TO OPEN GOOGLE AND ENTER SELENIUM AUTOMATICALLY

```
In [2]: #the instance of Chrome WebDriver is created

driver = wd.Chrome(executable_path='C:/Users/anany/.wdm/drivers/chromedriver/win32/91.0.4472.101/chromedriver.exe')

# driver.get method will navigate to the page given by the url

driver.get("http://www.google.com")

# this is to find the element there are a number of ways to do that

element = driver.find_element_by_xpath("/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input")
```

Ways to find/locate Elements

If we want to find/locate this

```
input type="text" name="passwd" id="passwd-id"
```

This can be typed

1. element = driver.find_element_by_id("passwd-id")
2. element = driver.find_element_by_name("passwd")
3. element = driver.find_element_by_xpath("//input[@id='passwd-id']")
4. element = driver.find_element_by_css_selector("input#passwd-id") """"

```
In [3]: #we are sending keys, this is similar to entering keys from the keyboard

element.send_keys("Selenium", Keys.ENTER)

# Web diver waits for this much time irrespective of the conditions

time.sleep(2)

# to close the browser window we can use driver.quit() as well

driver.close()
```

IMPORTANT NOTE

Difference between **driver.close()** and **driver.quit()** is that quit will exit entire browser whereas close will close one tab, but if just one tab was open, by default most browser will exit entirely.

COOKIES

```
In [3]: # The instance of Chrome WebDriver is created

driver=wd.Chrome(executable_path='C:/Users/anany/.wdm/drivers/chromedriver/win32/91.0.4472.101/chromedriver.exe')

# Go to the correct domain
driver.get("http://www.google.com")

# Now set the cookie. This one's valid for the entire domain
# A dictionary object is used , with required keys - "name" and "value"; Adds a cookie to the current session

cookie = {'name' : 'foo', 'value' : 'bar'}
driver.add_cookie(cookie)

# And now output all the available cookies for the current URL

driver.get_cookies()

Out[3]: [{'domain': 'www.google.com',
'httpOnly': False,
'name': 'foo',
'path': '/',
'secure': True,
'value': 'bar'},
{'domain': '.google.com',
'expiry': 1639746540,
'httpOnly': True,
'name': 'NID',
'path': '/',
'sameSite': 'None',
'secure': True,
'value': '217=nIupbHm4gNQp5gcyErX20CBHlthz18etbjcZAVXt4CtXDs7e7VTxp5YCs-T3FdprxTM11CYPH8Br80dV_26VSHR8Hhbg9_z19nBLg9HGigdJKtZ7AuHFpu7w8Pg2gQT0S52TfjPATpRPj8aWEJMLLNxREUJQEodSnPpmijftd34'},
{'domain': '.google.com',
'expiry': 1626527341,
'httpOnly': False,
'name': '1P_JAR',
'path': '/',
'sameSite': 'None',
'secure': True,
'value': '2021-06-17-13'}]
```

Explicit Waits

An explicit wait is a code you define to wait for a certain condition to occur before proceeding further in the code.The extreme case of this is time.sleep(), which sets the condition to an exact time period to wait.

Expected Conditions

By default, WebDriverWait calls the ExpectedCondition every 500 milliseconds until it returns success. ExpectedCondition will return true (Boolean) in case of success or not null if it fails to locate an element.

1. title_is
2. title_contains
3. presence_of_element_located
4. visibility_of_element_located
5. presence_of_all_elements_located
6. text_to_be_present_in_element
7. text_to_be_present_in_element_value
8. frame_to_be_available_and_switch_to_it
9. invisibility_of_element_located
10. element_to_be_clickable
11. staleness_of
12. element_to_be_selected
13. element_located_to_be_selected
14. element_selection_state_to_be
15. element_located_selection_state_to_be
16. alert_is_present

```
In [7]: from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By

#the instance of Chrome WebDriver is created

driver=wd.Chrome(executable_path='C:/Users/anany/.wdm/drivers/chromedriver/win32/91.0.4472.101/chromedriver.exe')

# Go to the correct domain

driver.get("http://www.instagram.com")
try:

# Selenium will wait for a maximum of 10 seconds for an element matching the given criteria to be found.

    element = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.XPATH, "/html/body/div[1]/section/main/article/div[2]/div[1]/div/form/div/div[1]/div/label/input"))
    )
# If no element is found in that time, a TimeoutException is thrown

finally:
    # Web diver waits for this much time

    time.sleep(2)

    # to close the browser window

    driver.close()
```

Implicit Waits

An implicit wait tells WebDriver to poll the DOM for a certain amount of time when trying to find any element (or elements) not immediately available. The default setting is 0 (zero). Once set, the implicit wait is set for the life of the WebDriver object.

```
In [9]: # the instance of Chrome WebDriver is created

driver=wd.Chrome(executable_path='C:/Users/anany/.wdm/drivers/chromedriver/win32/91.0.4472.101/chromedriver.exe')

# this works best on urls that take time loading

driver.implicitly_wait(10) # seconds

driver.get("http://www.instagram.com")

# Web diver waits for this much time

time.sleep(2)

# to close the browser window

driver.close()
```

USING SELENIUM TO ENTER USERNAME AND PASSWORD IN A WEBSITE

```
In [3]: # the instance of Chrome WebDriver is created

driver = wd.Chrome(executable_path='C:/Users/anany/.wdm/drivers/chromedriver/win32/91.0.4472.101/chromedriver.exe')

# driver.get method will navigate to the page given by the url

driver.get("https://sdsclub.com/")

# this will find the element using xpath and .click() method will click in the element

driver.find_element_by_xpath('/html/body/header/div/div/div[2]/a').click()

# this finds the username input feild

username=driver.find_element_by_xpath("/html/body/div[2]/div[2]/div[4]/div/div/div/form/div[1]/input")

# this will perform the action and automatically enter the username we wish to enter

username.send_keys("username")

# this finds the password input feild

passwords=driver.find_element_by_xpath("/html/body/div[2]/div[2]/div[4]/div/div/div/form/div[2]/input")

# this will perform the action and automatically enter the username we wish to enter

passwords.send_keys("password")

# Web diver waits for this much time

time.sleep(3)

# this is to close the driver window

driver.quit()
```