**Python Commands**

1. **Application Commands**

get () – Opening the application URL

title - To capture the title of the current webpage

current\_url – To capture the current URL of the webpage

page\_source – To capture source code of the page

1. **Conditional Commands**

is\_displayed ()

is\_enabled ()

is\_selected ()

1. **Browser Commands**

close () – will close single browser window (where driver focused)

quit () – will close multiple browser windows (This will kill all the process)

1. **Navigational Commands**

back ()

forward ()

refresh ()

1. **Wait Commands**

time. sleep (time in millisecond)

Advantage

1.Simple to use

Disadvantage

1.Performance of the script is very poor

2.If the element is not available within the mentioned time, still there is a chance of getting exception

Implicit wait (time)

Advantage

1.Single statement

2.Performance will not be reduced (If the element is available within the time it will proceed to further statement)

Disadvantage

1. If the element is not available within the mentioned time, still there is a chance of getting exception

Explicit wait -Works based on conditions

my\_wait = WebDriverWait (driver, 10, poll\_frequency=2,

ignored\_exceptions=[NoSuchElementException,ElementNotVisibleException,  
 ElementNotSelectableException,  
 Exception])

searchLink = my\_wait. until (EC.presence\_of\_element\_located((By.XPATH, "//h3[text()='Google Flights']")))  
searchLink.click()

Advantage

1.More efficient than the previous commands (time. sleep () and implicit\_wait () )

Disadvantage

1. We have to insert the command in multiple places so users will feel some difficulty.

find\_element () – Will return single web element

find\_elements () – Will return the web elements in the form of lists

text & get\_attribute ()

text – Returns the inner text of the element

get \_attribute () – Returns any values of the web element

**Handling Alerts and popups**

myalerts = driver. swich\_to. alert

myalert. accept ()

myalert. dismiss()

**Basic Authentication pop up**

We have to inject the username and password to the URL and bypass the URL by adding the username and password in front of the URL

Example:

Actual URL = <https://the-internet.herokuapp.com/basic_auth>

Username = admin Password = admin

Bypassed URL = <https://admin:admin@the-internet.herokuapp.com/basic_auth>

**frame/ iframe/ form**

switch\_to. frame (“name of the frame”)

swich\_to. frame (“id of the frame”)

swich\_to. frame (web element)

swich\_to. frame (index) – index will only work if it is only one frame otherwise it won’t work

switch\_to.default\_content () => This will switch driver back to the main frame

**Handling browser windows**

driver. switch\_to. window (window ID)

current\_window\_handle => This will return the window id of single browser window

window\_handles => This will return the window id’s of multiple browser windows