======Opening and closing browser ============

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
  
\*\*\* Test Cases \*\*\*  
*#Begin Web Test  
 # open browser https://www.nopcommerce.com/en/demo chrome  
 # open browser https://www.google.co.in/ chrome  
 # close browser*Browser test  
 open browser *https://www.google.co.in/ chrome* close browser  
---------------------------------------------------------------------

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${Browser} *chrome*${URL} *https://www.google.co.in/*\*\*\* Test Cases \*\*\*  
Browser test  
 open browser ${URL} ${Browser}  
 close browser

Working with variable concepts

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${Browser} *chrome*${URL} *https://www.facebook.com/*\*\*\* Test Cases \*\*\*  
Browser test  
 open browser ${URL} ${Browser}  
 input text *id:email hello123* input text *id:pass 12345656* close browser

Different type of locator :- id, name, Class name, Tag name, Link text, Partial Link Text, CSS Selector, Xapth

|  |  |  |
| --- | --- | --- |
| **Strategy** | **Match based on** | **Example** |
| id | Element id. | id:example |
| name | name attribute. | name:example |
| identifier | Either id or name. | identifier:example |
| class | Element class. | class:example |
| tag | Tag name. | tag:div |
| xpath | XPath expression. | xpath://div[@id="example"] |
| css | CSS selector. | css:div#example |
| dom | DOM expression. | dom:document.images[5] |
| link | Exact text a link has. | link:The example |
| partial link | Partial link text. | partial link:he ex |
| sizzle | Sizzle selector deprecated. | sizzle:div.example |
| jquery | jQuery expression. | jquery:div.example |
| default | Keyword specific default behavior. | default:example |

Examples below

**By name:-** *input text name:pass 12345566***By id** :- input text *id:email 123456***By Xpath** *:- input text xpath://input[@name='email'] 1234556*

**By CSS** :- input text *css:input[name='email'] 123456*

====================================================

Entering data in text fields

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${Browser} *chrome*${URL} *https://www.facebook.com/*\*\*\* Test Cases \*\*\*  
Browser test  
 open browser ${URL} ${Browser}  
 **input text *name:email 1234567* input text *name:pass 12345566***click element *xpath://button[@type='submit']* close browser

Maximizing the window and clearing the text box and entering data in text box, clicking ok button   
  
\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${Browser} *chrome*${URL} *https://www.facebook.com/*\*\*\* Test Cases \*\*\*  
Browser test  
 open browser ${URL} ${Browser}  
 **maximize browser window**  
 **clear element text *name:email* clear element text *name:pass***input text *name:email 1234567* input text *name:pass 12345566* **click element *xpath://button[@type='submit']* close browser**  
=======================================================================

Click on link, button, Radio button, check box

Selecting radio button :- **select radio button *ageGroup 15 – 50***

**Ex:- select radio button *< group> <value>***

Select check box :- **select checkbox <name>:<value>**

Select link :- **click link xpath://input[@name=’value’]**

Select button :- **click button xpath://input[@name=’value’]**

**Example to click link and wait for 4 second using sleep**   
\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${Browser} *chrome*${URL} *https://www.facebook.com/*\*\*\* Test Cases \*\*\*  
Browser test  
 Open Browser ${URL} ${Browser}  
 Maximize Browser Window  
 Click Link *link:Forgotten password?  
 #click link xpath://input[@name=’value’]* Sleep *4s* Close Browser

**Locator CSS with syntax (with tag and id, any locator)/ with tag and class with any locator**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *https://facebook.com*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
open\_browser  
 open browser ${URL} ${Browser}  
 maximize browser window  
 **input text *css=input#email[name='email'] hello1111***input text *id=pass 123456* sleep *1s* click button *name=login* close browser  
=============================================================

**Multiple attribute with OR condition**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *https://facebook.com*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
open\_browser  
 open browser ${URL} ${Browser}  
 maximize browser window  
 input **text *css=input#email[name='email'] hello1111***input text *id=pass 123456* sleep *1s* **click button *xpath://button[@value='1' or @name='login']***close browser  
===============================================================

**Radio button code**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *http://demo.guru99.com/test/radio.html*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
open\_browser  
 Open Browser ${URL} ${Browser}  
 Sleep *3s* Maximize Browser Window  
 **Select Radio Button *webform Option 2*** *==========================================================================*

***Check Box code***

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *http://demo.guru99.com/test/radio.html*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
open\_browser  
 Open Browser ${URL} ${Browser}  
 Sleep *3s* Maximize Browser Window  
 **Select Radio Button webform Option 2**Sleep *2s* Select Checkbox *id:vfb-6-1  
#<input type="checkbox" name="webform" id="vfb-6-1" value="checkbox2">*

***Drop down (Select by value/index/label***

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *https://formstone.it/components/dropdown/demo/*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
open\_browser  
 Open Browser ${URL} ${Browser}  
 Sleep *3s* Maximize Browser Window  
 **Select from list by index *name:demo\_basic 1***

**Select from list by value *name:demo\_basic 0***

*-------------------------------------------------------------------*

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *https://formstone.it/components/dropdown/demo/*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 Open Browser ${URL} ${Browser}  
 Maximize Browser Window  
 *#Set Selenuim Speed 2seconds  
 #Select From List by Index name:demo\_basic 1  
 #Select From List by Value name:demo\_basic 0* **Select From List by Label *name:demo\_basic Two*** *=======================================================================*

***Write keywords without argument***

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *https://www.facebook.com/r.php?locale=en\_GB*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 Open Browser ${URL} ${Browser}  
 Maximize Browser Window  
 **Enter Username Password email** sleep *2s* close Browser  
  
\*\*\* Keywords \*\*\*  
**Enter Username Password email**  
 Input Text *xpath://input[@aria-label='First name'] hello* Input Text *xpath://input[@aria-label='Surname'] hello12345* Input Text *name:reg\_email\_\_ 123456789  
=============================================================*

***With arguments***

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *https://www.facebook.com/r.php?locale=en\_GB*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 Open Browser ${URL} ${Browser}  
 Maximize Browser Window  
 **Enter Username Password email *hello12 1234556 h@gmail.com* sleep *2s***close Browser  
  
\*\*\* Keywords \*\*\*  
**Enter Username Password email** **[Arguments] ${username} ${pasword} ${email}**  
 Input Text *xpath://input[@aria-label='First name']* **${username}**  
 Input Text *xpath://input[@aria-label='Surname']* **${pasword}**  
 Input Text *name:reg\_email****\_\_* ${email}**

**How to define veriable inside the test case**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *https://www.facebook.com/r.php?locale=en\_GB*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 **${Var1} set variable *ravikumar nirale* log to console ${var1}**

**To avoid errors we need to install below packages in pycham interpreter**

1. I have installed ONLY Intellibot plugin from jar from the link: https://github.com/lte2000/intellibot/blob/develop/intellibot.jar

2. Then I went to project's python interpreter and add following packages:

**- robotframework**

**- robotframework-pythonlibcore**

**- robotframework-seleniumlibrary**

**- selenium**

**3. Restart PyCharm**

**Fetching the value from list**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Library Collections*\*\*\* Variables \*\*\*  
\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 @{list1} create list *hello 2222 hi 3333 bye 12ed34* **${list\_len} get length ${list1}  
 log to console ${list\_len}  
 ${list\_ele} get from list ${list1} *4* log to console ${list\_ele}**

**FOR LOOP to display values**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Library Collections*\*\*\* Variables \*\*\*  
\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 *# @{list1} create list hello 2222 hi 3333 bye 12ed34* **FOR ${i} IN RANGE *1 10* log to console ${i}  
 END**

**FOR LOOP for List**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Library Collections*\*\*\* Variables \*\*\*  
\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 @{list1} create list *hello 2222 hi 3333 bye 12ed34***FOR ${i} IN ${list1}  
 log to console ${i}  
 END**

**Storing the keyword in to the variable and running the variable***Library SeleniumLibrary*\*\*\* Variables \*\*\*  
\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 ${**key\_var} set variable *log to console* run keyword ${key\_var} *welcome to learning***

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Library Collections*\*\*\* Variables \*\*\*  
\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 **${var} set variable *NO* run keyword if *'*${var}*'=='YES' log to console value found* run keyword if *'*${var}*'=='NO' log to console value not******found***

***Set selenium speed and Sleep :-***

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Library Collections*\*\*\* Variables \*\*\*  
${URL} *https://www.facebook.com/*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 **${speed\_val}= get selenium speed  
 log to console ${speed\_val}**  
 open browser ${URL} ${Browser}  
 maximize browser window  
 set selenium speed *5 seconds* input text *id:email hello123* input text *id:pass 123456* **${speed\_val}= get selenium speed  
 log to console ${speed\_val}** close browser

Sleep 2seconds :- it will wait for only one time.

**Get/Set Timeout**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Library Collections*\*\*\* Variables \*\*\*  
${URL} *https://www.facebook.com/*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 **${speed\_val}= get selenium timeout**  
 **log to console ${speed\_val}**  
 open browser ${URL} ${Browser}  
 maximize browser window  
 **${speed\_val}= set selenium timeout *10 seconds* log to console ${speed\_val}** wait until page contains *Add Account* input text *id:email hello123* input text *id:pass 123456* close browser

**Implicit wait :- set selenium implicit wait *10 seconds (If element found before 10 second it will execute and move to next line else it will wait for 10seconds and fail the test case***

***Take screen shot run time:-*** *can be current director or any path.*

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *https://www.facebook.com/*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 open browser ${URL} ${Browser}  
 maximize browser window  
 input text *id:email hello123* input text *id:pass 123456* **capture page screenshot** *C:/Users/admin/Desktop/screen\_shots\_pyton\_robot/TC001.png* close browser

**Close all browser/close window/close browser**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *https://www.facebook.com/*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 open browser ${URL} ${Browser}  
 maximize browser window  
 input text *id:email hello123* input text *id:pass 123456* capture page screenshot */TC001.png* close browser  
 close all browsers  
 close window

**Go to, Go back and get location:-**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *https://www.facebook.com/*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 open browser ${URL} ${Browser}  
 maximize browser window  
  **${new\_url} get location**  
 log to console ${new\_url}  
 **go to *https://www.google.co.in***sleep *2 seconds* ${new\_url} get location  
 log to console ${new\_url}  
 **go back**  
 sleep *2 seconds* ${new\_url} get location  
 log to console ${new\_url}  
 close window

**How to use Mouse in Robot Framework**

**1.Open Context Menu (Right click)**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${URL} *https://www.facebook.com/*${Browser} *chrome*\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 open browser ${URL} ${Browser}  
 maximize browser window  
 **open context menu *xpath://input[@name='pass']***sleep *5 seconds* close window  
===================================================

**2.Double Click** :- **double click element *xpath://input[@name='pass']***

1. **mouse down *xpath://input[@name='pass']***
2. **mouse up *xpath://input[@name='pass']***
3. **mouse over *xpath://input[@name='pass']***

**Keyboard operations :- 1. Key pressed with alpha/numeric/ special char**

**press keys *name:pass hello***

***Synax:- press keys <xpath> value***  *# Pressing enter from keyboard*

*Ascii value of enter key is 13*

**press keys *name:pass hello***

*example :-* **press keys *xpath://button[@type='submit']*** [***\\13***](file:///\\13)

***Wait commands***

1. ***Wait until page contains***

Example :- **wait until page contains *Videos***

1. ***Wait until page contains elements***

**wait until page contains element *<xpath>***

1. ***Wait until elements contents***

**Example :- wait until element contains *<xpath> <text>***

1. ***Wait until element visible***
2. ***Wait until element enable***

**Multi window handling :-**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${var1} *https://google.co.in*${var2} *https://facebook.com*\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 open browser ${var1} *chrome* maximize browser window  
 sleep *2 seconds* open browser ${var2} *chrome* maximize browser window  
 sleep *2 seconds* **switch browser *1***${url}= get location  
 log to console ${url}  
 **switch browser *2***${url}= get location  
 log to console ${url}  
 close all browsers

**Multiple Tabs :- switch window *<title/url>***

***Or***

***@{list} get window handler***

***: For ${i} IN ${list}***

***Log to console ${i}***

***end***

**Validate checkbox is selected or not**

**select checkbox *name:remember*** *:- clicked on check box*

**checkbox should be selected *name:****remember :- checking clicked or not*

**select checkbox *name:remember*** *:- again clicking to uncheck the check box*

**checkbox should not be selected *name:remember*** *:- checking check box is not selected*

***Validating the text***

1. ***Element text should be <Xpath> <text to search>***
2. ***Element text should not be <Xpath> <text to search>***
3. ***Element text should contain <Xpath> <partial text to search >***
4. ***Element text should not contain <Xpath> < Partial text to search>***

**Validating Title , element should be Visible and element should be Enabled**

**title should be *<title name>*${val}= get title  
log to console ${val}**

**element should be enabled *<xpath>*input text *name:login hello #enter text in text box***

**element should be disabled *<xpath>***

**element should be visible *<xpath>*input text *name:login hello #enter text in text box***

**element should not be visible *<xpath>*input text *name:login hello #enter text in text box***

**Resource file**

**1.Keywords with arguments**

**2. without arguments**

**3. With return value**

**Note :- create one file in resource with .robot and same concept as keyword with and without argument**

**In Setting add Resource ../<folder name>/<filename>**

**[Return] ${Title} #from resource file to test case file**

**Log ${return value} # it will store result in log files report.html**

**Example :- sample\_text.robot <file>**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Resource ../Resource/resource1.robot*\*\*\* Variables \*\*\*  
  
\*\*\* Test Cases \*\*\*  
Open Browser Test case  
 Open webpage and maximize it

**Resource1.robot <file>**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*\*\*\* Variables \*\*\*  
${var1} *https://google.co.in*${var2} *https://facebook.com*\*\*\* Keywords \*\*\*  
Open webpage and maximize it  
 open browser ${var1} *chrome* maximize browser window  
 sleep *2 seconds* open browser ${var2} *chrome* maximize browser window  
 sleep *2 seconds* switch browser *1* ${url}= get location  
 log ${url}  
 log to console ${url}  
 switch browser *2* ${url}= get location  
 log ${url}  
 log to console ${url}  
 close all browsers

**Adding details to the test cases (Documentation and time out )**

**[Documentation] *this test cases is used to verify open the browser and maximize the window***

**[Timeout] *10s***

***[Timeout] 2min 8s***

***Test case need to execute with in the time specified else test case will fail.***

**Adding setup and tear down to test cases :-**

**Test case level**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Resource ../Resource/resource1.robot*Documentation *1st test case in robot framework*\*\*\* Variables \*\*\*  
  
  
\*\*\* Test Cases \*\*\*  
TC001\_Test case  
 [Documentation] *this first test cases in robot framework* **[Setup] Open website  
 [Teardown] Close website**  
 input text *name:email hello12233* input text *name:pass 1234567*

*Resource1.robot file*

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*Documentation *resource file data*\*\*\* Variables \*\*\*  
${URL} *http://www.facebook.com*${Browser} *chrome*\*\*\* Keywords \*\*\*  
Open website  
 open browser ${URL} ${Browser}  
 maximize browser window  
  
Close website  
 ${title}= get title  
 log ${title}  
 log to console ${title}  
 close browser

**Test suite level :- it will be applicable to all test cases with in the file**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Resource ../Resource/resource1.robot*Documentation *1st test case in robot framework here we can try this***Test Setup Open website  
Test Teardown Close website**  
\*\*\* Variables \*\*\*  
  
  
\*\*\* Test Cases \*\*\*  
TC001\_Test case  
 [Documentation] *this first test cases in robot framework* input text *name:email hello12233* input text *name:pass 1234567*

**Test Suites :-**

**If test suites are inside the folder we can execute all the test suites with syntax:- robot folder name**

**Before and after test suites execution**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Resource ../Resource/resource1.robot*Documentation *1st test case in robot framework here we can try this*Suite Setup Before execution of test cases  
Suite Teardown After execution of test cases  
Test Setup Open website  
Test Teardown Close website

**Resource1.robot file data**

Before execution of test cases  
 log *Prerequest test suites*After execution of test cases  
 log *after test execution of test suite*

**Executing folder with multiple test suites inside**

Create **\_\_init\_\_.robot file**   
which contain following data

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*Documentation *1st test case in robot framework here we can try this*Suite Setup Before execution of test cases  
Suite Teardown After execution of test cases

\*\*\* Keywords \*\*\*

Before execution of test cases  
 log *Prerequest test suites*After execution of test cases  
 log *after test execution of test suite*

*Run that folder with* ***robot <folder name>***

**Tags ;-**

1. **Tags at test case level**
2. **Default Tags**
3. **Forced Tags**

**Tags are used for which test case need to execute (ex:- smoke, sanity)**

**Example :- (Tags at test case level)**

TC001\_Test case  
 [Tags] *smoke* [Documentation] *this first test cases in robot framework* input text *name:email hello12233* input text *name:pass 1234567*TC002\_Test case  
 [Tags] *smoke sanity* log *second test case execution*

*Run* ***robot –i smoke <Filename.robot>***

***Robot –i sanity <Filename.robot>***

**Example :- Default Tags**

**If tags are not applied then it is called default tags**

**Sample\_test.robot <file >**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Resource ../Resource/resource1.robot*Documentation *1st test case in robot framework here we can try this*Suite Setup Before execution of test cases  
Suite Teardown After execution of test cases  
Test Setup Open website  
Test Teardown Close website

Default Tags *DETG*

Force Tags *All\_TC*

\*\*\* Variables \*\*\*  
  
  
\*\*\* Test Cases \*\*\*  
TC001\_Test case  
 [Tags] *smoke* [Documentation] *this first test cases in robot framework* input text *name:email hello12233* input text *name:pass 1234567*TC002\_Test case  
 log *second test case execution*TC003\_Test case  
 [Tags] *Regression* log *3ed test case execution*TC004\_Test case  
 [Tags] *smoke* log *4th test case execution  
-----------------------------------------------*

*Sample\_test1.robot <fie>*

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Resource ../Resource/resource1.robot*Documentation *2nd test case in robot framework here we can try this*Suite Setup Before execution of test cases  
Suite Teardown After execution of test cases  
Test Setup Open website  
Test Teardown Close website

Default Tags *DETG*

Force Tags *All\_TC*

\*\*\* Variables \*\*\*  
  
  
\*\*\* Test Cases \*\*\*  
TC005\_Test case  
log *5th test case execution*TC006\_Test case  
 [Tags] *regression* log *6th test case execution*TC007\_Test case  
 [Tags] *smoke* log *7th test case execution*TC008\_Test case  
 log *8th test case execution  
--------------------------------------------*

*Run command :-* ***robot –i* *All\_TC Test\_cases*** *<foldername>   
Force tag execute all test cases without applying tag to test cases.*

***Control execution Tags :-***

***1.Include Tags –i [ use AND |OR |NOT)***

***2. Exclude –e***

***3. Execute particular test case –t***

***4. Execute particular test case –s***

***5. Report location***

***Include tags***

* ***All the test cases are executed command to run :- robot <folder name>***
* ***Regression Tag execution :- robot –i Regression <folder name>***
* ***Executing multiple tags :- robot –i smokeORSantiy <folder name>***
* ***Robot –i smokeANDsanity <folder name>***
* ***Executing test cases having regression but not having smoke tags   
  robot –i RegressionNOTsmoke <Folder name>***

***Exclude the test case use –e option***

* ***Robot –e smoke <folder name> #Apart from smoke every thing will execute***

***Execute particular test case “-t option”***

***Run below command :- robot –t “test case name” <folder name>***

***Example :- robot -t "TC006\_Test case" Test\_cases***

***Executing particular test suite from folder (-s option)***

***Syntax:- robot –s “test suite name with out .robot” <folder name>***

***robot -s 'Sample\_test1' Test\_cases***

***Fetching data from run time   
(Fetch Page title, selenium speed, text, value , from list/dropdown, element attribute, element count, location, source )***

**get text <xpath>, *get value <xpath> , get title, get selenium speed*get list items <xpath> , *get selenium timeout,***

**get source, get** **time**

**get element attribute *name:pass class (class value we will get )***

**get element count *class:field # we get count for how many element have class field***

**User Defined Keywords**

***Sample\_test.robot <file>***

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary  
Resource C:/Users/admin/Desktop/Robot\_dummy\_project/Resource/resource1.robot*

\*\*\* Test Cases \*\*\*  
  
TC012\_Fetch\_Test\_case  
 Create folder at run time *hello123 bye456*

*Userkeywords.py <file>*

import os  
def createfolder(folder\_name):  
 os.mkdir("D:\\"+folder\_name)  
  
def createsubfolder(folder\_name,subfolder\_name):  
 os.mkdir("D:\\"+folder\_name+"\\"+subfolder\_name)

**Resource1.robot <file>**

\*\*\* Settings \*\*\*  
*Library SeleniumLibrary*Documentation *resource file data  
Library ../ExternalKeywords/Userkeywords.py*

\*\*\* Keywords \*\*\*

Create folder at run time  
 [Arguments] ${folder\_name} ${subfolder\_name}  
 createfolder ${folder\_name}  
 createsubfolder ${folder\_name} ${subfolder\_name}  
 log *folder creation successful*

**Work on Excel File.**

**Pip install openpyxl** in order to work with excel sheet

*# import packages*import openpyexcel  
  
***#load workbook***wk=openpyexcel.load\_workbook("C:\\Users\\admin\\Desktop\\Automation\_python\\excel\_auto\_data.xlsx")  
  
***#read sheet names***print(wk.sheetnames)  
  
***#Read active sheet name***print("active sheet is ", wk.active)  
  
**#creating object for sheet and fetching title**sheetname= wk["data"]  
print(sheetname.title)

***#reading data from cell***print(sheetname['A4'].value)  
print(sheetname['A1'].value)

***#Alternative method to read data from cell***c1=sheetname.cell(3,1)  
print(c1.value)

***#Alternative method to read data from cell***c1=sheetname.cell(column=3,row=2)  
print(c1.value)

***#To get row and column name*** print("row name ",c1.row)  
print("column name",c1.column)

***#To get how many rows having data in rows and columns***rows= sheetname.max\_row  
columns=sheetname.max\_column  
print("total row are",rows)  
print("total columns are", columns)

***#to print values of all rows and columns***for i in range(1,rows+1):  
 for j in range(1,columns+1):  
 cc= sheetname.cell(i,j).value  
 print(cc)

**Write data in to the excel sheet :-**

*# import packages*import openpyexcel  
  
***#load workbook***wk=openpyexcel.load\_workbook("C:\\Users\\admin\\Desktop\\Automation\_python\\excel\_auto\_data1.xlsx")  
  
***#read sheet names***print(wk.sheetnames)  
  
***#Read active sheet name***print("active sheet is ", wk.active)  
  
***#to rename the sheet***sh = wk.active  
sh.title="Hello world"  
print(sh.title)  
  
***#Adding data to cell***sh['A5'].value="hello world data"  
  
***#creating new sheet***wk.create\_sheet(title="basvaraj")  
  
***#enter data to particular cell***sh1= wk['basvaraj']  
sh1['A2']='123456'

***#remove sheet***print("deleting sheet basavaraj")  
wk.remove(sh1)

***#saving workbook***wk.save("C:\\Users\\admin\\Desktop\\Automation\_python\\excel\_auto\_data1.xlsx")

**JSON (Java Script Object Notation):-**Simple JSON data below :-

{

"name": "ravikumar",

"age": 23,

"email":"bng.ravi123@gmail.com"

}

**Data with Array :-**

{

"name": "ravikumar",

"age": 23,

"email":"bng.ravi123@gmail.com",

"city":["Bidar","bng","Hubli"]

}

**JSON with object key and value**

{

"name": "ravikumar",

"age": 23,

"email":"bng.ravi123@gmail.com",

"city":["Bidar","bng","Hubli"],

**"Address":**

{

"Street":"ABCDEFG",

"Hno":"fikdjfsdjlkf",

"**contact number":**

{

"landline":"378348324874389",

"Mobile number":"928328238923"

}

}

}

Searching value of they key using website [**www.jsonpath.com**](http://www.jsonpath.com)

**Moving to child via .**

**$.city[1]** output:- bng

**$.Address.Hno** output:-"fikdjfsdjlkf"

**$.Address.contact number.Mobile number** output:- "928328238923"

**Parse Dictionary to JSON, Parse JSON to Dictionary**

import json  
  
odcis='{"hello":"world","hey":"good"}'  
print(type(odcis))  
json\_result = json.loads(odcis)  
  
print(json\_result["hey"])

**output :-**

python .\Parse.py

<class 'str'>

good

**Revisit 139 video to 140 videos**

**BDD framework (Test cases )**

**Given, when, Then (Use keyword concept for this )**

**Recording Test case using robot framework**

**Plug in on chrome :- robot corder**

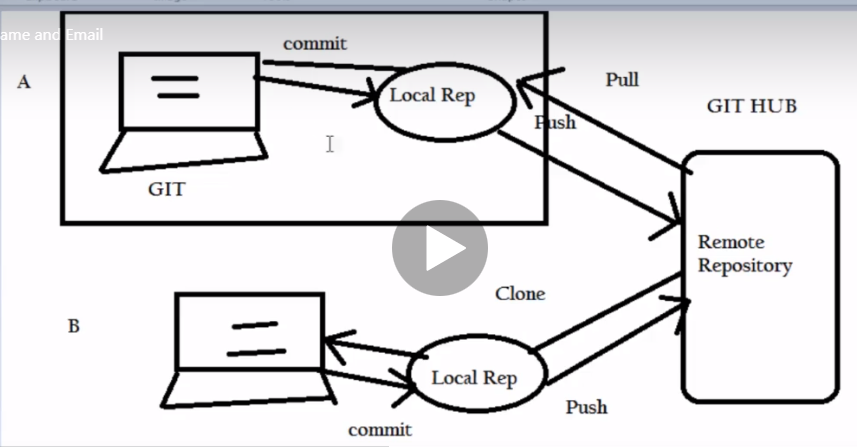
**Add extension to chrome, open robot corder and click on Record , start test case execution and stop recording , save file , copy file and paste on pycham file run it , there is a option scan it will give all the element locator for that page, In setting there is option select locator , you can also add sleep between each step and you can add title before and after execution of test cases.**

**Data Driven in robot Framework**

**done**

GIT :- 1. Set up GIT & GIT HUB

Concept



Create account in git hub (github.com) sign in or sign up and create repository

Created git hub for me :- <https://github.com/ravikumar2609/Automation_testing.git>

Create local GIT :- Download git bash from <https://git-scm.com/download/win>   
and install.

Check from command line **:- git –version**

Set user name and password on local machine in order to know who is pushed the code to server

**Git config –global user.name “username”**

**git config --global user.name "ravikumar2609"**

**Git config –global user.email “usernameid”**

**To check user added successfully**

**git config user.name   
git config user.email**

**git config --list**

**Creating local repository :-**

C:\Users\admin\Desktop\Automation\_python\local\_git\_repo   
mkdir sample\_project

Cd sample\_project (move inside that directory and type **“git init”**

**Working with GIT**

Files will be in 3 stage

1. UnTracked
2. Stage
3. Tracked

Place the files in local repo folder   
**git status** :- it will display Untrack

**Git add <file\_name>**

**Git add . ( all the files in current folder will move to stage state)**

**Git commit -m <msg> (Move to Track stage)**