**Robot Keyword Library**

|  |  |  |  |
| --- | --- | --- | --- |
| Keyword | Significance | Arguments that it accepts | Example |
| open browser | To open the browser | 2 – url & browser | open browser <https://www.nopcommerce.com/> chrome |
| click element | Clicks on specified element | 1 – element identifier | click element *xpath://span[text()='Product']* |
| close browser | Closes the browser | 0 | close browser |
| maximize browser window | To maximize the window | 0 | maximize browser window |
| title should be | To verify title of the page | 1 | title should be *Facebook – log in or sign up*  [note – title should not be provided in quotes] |
| page should contain | To verify some text on the webpage | 1 (text that should be verified on webpage) | page should contain *login is successful* |
| set selenium speed | It delays between two commands | 1 | set selenium speed *2seconds* |
|  |  |  |  |
| Input Text - "C:\Users\VJ\Desktop\Switch\Robot\Input\_Box.txt" | | | |
| element should be visible | To check element is visible or not | 1 | element should be visible ${"email"} |
| element should be enabled | To check element is enable or not | 1 | element should be enabled ${"email"} |
| input text | To enter some value in box | 2 | input text ${"email"} *vj@rocks.com* |
| clear element text | To clear the value in the box | 1 | clear element text ${"email"} |
|  |  |  |  |
| Radio & Check Box - "C:\Users\VJ\Desktop\Switch\Robot\RadioBox\_CheckBox.txt" | | | |
| select radio button | To select the radio button from multiple options | 2 (value & name from html code) | select radio button *sex Female* |
| select checkbox | To select the check box | 1 (name from html code) | select checkbox *BlackTea* |
| unselect checkbox | To unselect the check box | 1 (name from html code) | unselect checkbox *BlackTea* |
|  |  |  |  |
| Dropdown option - "C:\Users\VJ\Desktop\Switch\Robot\Dropdown\_ListBox.txt" | | | |
| select from list by label | To select the value from dropdown using visible text | 2 (name attribute of dropdown component from html code & visible text of the dropdown option) | select from list by label *continents Asia* |
| select from list by index | To select the value from dropdown using index number of option from the list | 2 (name attribute of dropdown component from html code & index number of the option from list. Index number starts with 0) | select from list by index *continents 6* |
| select from list by value | To select the value from dropdown using value attribute of the option from html code | 2 (name attribute of dropdown component from html code & value attribute of dropdown option to be selected from the html code) | select from list by value *continents con3* |
|  |  |  |  |
| List Box - "C:\Users\VJ\Desktop\Switch\Robot\Dropdown\_ListBox.txt" | | | |
| select from list by label | To select option from the list. | 2 (name attribute of complete list component from html code & visible text of the option to be selected) | select from list by label *selenium\_commands Switch Commands* |
| unselect from list by label | To unselect option from the list. | 2 (name attribute of complete list component from html code & visible text of the option to be unselected) | unselect from list by label *selenium\_commands Wait Commands* |
|  |  |  |  |
| Waits - "C:\Users\VJ\Desktop\Switch\Robot\Waits.txt" | | | |
| Sleep | To pause the specific step | 1 (time in seconds) | sleep *2* |
| set selenium speed | To delay all the commands below it | 1 (time) | set selenium speed *3 seconds* |
| get selenium speed | To know the current selenium speed. By default it is 0 seconds. | Output of this commands needs to be stored in some variable | ${speed} = get selenium speed |
| set selenium timeout | To change the default selenium timeout value | 1 (time) | set selenium timeout *10 seconds* |
| wait until page contains | To verify certain text on the webpage | 1 (text to be displayed on the webpage) | wait until page contains *Register* |
| get selenium timeout | To know the current selenium timeout. By default is 5 seconds. | Output of this commands needs to be stored in some variable | ${time}= get selenium timeout |
| set selenium implicit wait | To change the value of implicit wait. | 1 (time) | set selenium implicit wait *10 seconds* |
| get selenium implicit time | To know the current value of implicit time. By default is 0 seconds. | Output of this commands needs to be stored in some variable | ${implicittime} = get selenium implicit time |
|  |  |  |  |
| Alert - "C:\Users\VJ\Desktop\Switch\Robot\Alert.txt" | | | |
| handle alert | To accept / dismiss the alert | 1 (accept / dismiss) | handle alert *accept*  handle alert *dismiss* |
| alert should be present *Press a button!* | To verify the text present in the alert window | 1 (text on the alert window) | alert should be present *Press a button!* |
|  |  |  |  |
| Frame - "C:\Users\VJ\Desktop\Switch\Robot\Frames.txt" | | | |
| select frame | To switch to other frame | 1 (name / id / xpath of the frame) | select frame *packageListFrame* |
| unselect frame | like coming to default content in selenium java | 0 | unselect frame |
|  |  |  |  |
| Browser closing - "C:\Users\VJ\Desktop\Switch\Robot\closeBrowsers.txt" | | | |
| close browser | To recently opened browser (only one browser | 0 | close browser |
| close all browsers | To close all the opened browsers | 0 | close all browsers |
|  |  |  |  |
| Browser Navigation - "C:\Users\VJ\Desktop\Switch\Robot\browserNavigation.txt" | | | |
| select window | To switch to anather window on same browser | 1 (title / name / url) | select window *title=SeleniumHQ Browser Automation* |
| switch browser | In case of multiple browsers open, to switch to another browser | 1 (index of browser) | switch browser *2* |
| go to | To move to specified url on same page | 1 (url) | go to *https://www.bing.com* |
| go back | To go back to the previous url | 0 | go back |
| get location | To know the url of current webpage | Output of this commands needs to be stored in some variable | ${loc}= get location |
|  |  |  |  |
| Screenshot - "C:\Users\VJ\Desktop\Switch\Robot\screenshot.txt" | | | |
| capture element screenshot | To capture screenshot of specific element / image | 2 (image locator, address & name of generated image) | capture element screenshot *xpath://img[@src="/webres/ images/logo.png"] logo.png* |
| capture page screenshot | To take screenshot of complete page | 1 (address & name of generated image) | capture page screenshot *C:/Users/VJ/page.png* |
|  |  |  |  |
| Mouse Actions - "C:\Users\VJ\Desktop\Switch\Robot\MouseActions.txt" | | | |
| open context menu | To perform the right click operation (means to open context menu) | 1 (element location) | open context menu *xpath://span[text()="right click me"]* |
| double click element | To perfrom double click on the element | 1 (element location) | double click element *xpath://button[text()="Double-Click Me To See Alert"]* |
| drag and drop | To perform drag & drop | 2 (source element & destination element) | drag and drop *xpath://span[text()="Draggable 3"] id:mydropzone* |
|  |  |  |  |
| Scrolling - "C:\Users\VJ\Desktop\Switch\Robot\ScrollJavascriptExecutor.txt" | | | |
| execute javascript *window.scrollTo()* | To scroll the webpage by specific pixcels | 2 (x & y pixels by which page to be scrolled) | execute javascript *window.scrollTo(0,1500)* |
| scroll element into view | To scroll the page till we find specific element | 1 (element locator) | scroll element into view *xpath://img[@src="flags-normal/flag-of-India.png"]* |
| execute javascript *window.scrollTo()* | To scroll till bottom / top of the page | 1 | execute javascript *window.scrollTo (0,document.body.scrollHeight)* |
|  |  |  |  |
| Links - "C:\Users\VJ\Desktop\Switch\Robot\LinkCountAndPrint.txt" | | | |
| get element count | To find total number of elements matching the provided xpath | Output of this commands needs to be stored in some variable | ${LinksCount}= get element count *xpath://a* |
|  |  |  |  |
| Table - "C:\Users\VJ\Desktop\Switch\Robot\Table.txt" | | | |
| get element count | To count total number of rows | Output of this commands needs to be stored in some variable | ${columns}=get element count *xpath://table[@name="Table"]/ tbody/tr* |
| get element count | To count total number of columns | Output of this commands needs to be stored in some variable | ${columns}=get element count *xpath://table[@name="Table"]/ tbody/tr[2]/td* |
| get text | To get the data from the table | Output of this commands needs to be stored in some variable | ${data}=get text *xpath://table[@name="Table"] /tbody/tr[5]/td[1]* |
| table column should contain | To validate wether text is present in specific column (at any row) or not | 3 (table locator, column number, text to be verified) | table column should contain *xpath://table[@name="Book"] 2 Author* |
| table row should contain | To validate wether text is present in specific row (at any column) or not | 3 (table locator, row number, text to be verified) | table column should contain *xpath://table[@name="Book"] 4 Javascript* |
| table cell should contain | To validate wether text is present in specified cell or not | 4 (table locator, row number, column number, text to be verified) | table column should contain *xpath://table[@name="Book"] 5 2 Mukesh* |
| table header should contain | To validate wether provided table header is correct or not | 2 (table locator, header) | table column should contain *xpath://table[@name="Book"] BookName* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |