**Robot keys to remember:**

<https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#test-data-syntax>

<https://www.edureka.co/blog/robot-framework-tutorial/>

What folders should your repository consists.

**Configs** – Holds basic configurations that you need for your automation (e.g URL, Browser, Libraries, any universal variables that the values would change but important in running your automation)

* Setup.robot

**Objects** – Holds all variables you need for a specific test scenario

* TC1\_Object.robot

**Scenarios** – Holds all your keywords for a specific test case

* Login\_Scenario.robot

**TestCases** – holds all your test cases

* TC1\_Login.robot // naming

**TestData** – Holds the test data that can be used entirely on the testing

* TD\_Login.robot

**TestResult** – Holds the log, output and result of the test run.

* Sanity\_EXT\_01012020\_Log.html
* Sanity\_EXT\_01012020\_Output
* Sanity\_EXT\_01012020\_Report.html

**Handling Elements in Robot Framework**

Input text

* Visible - to determine if the element is visible
  + Element should be visible ${tbx\_name} or //input[@class=’name’]
* Enabled – to determine if the text box enabled
  + Element should be enabled ${tbx\_name} or xpath location of the element
* Input – to input a value on the text box
  + Input Text ${tbx\_name} or //input[@class=’name’] johndoe@yopmail.com
* Clear – to clear the text box
  + Clear element text ${tbx\_name} or //input[@class=’name’]

Title

* Title should be ${title\_name}

Radio buttons

* Select radio button \_name\_of\_the\_button \_value\_of\_radio\_button

Check box

* Select checkbox \_name\_of\_checkbox
* Unselect checkbox \_name\_of\_checkbox

Dropdown

* Select from list by label \_name\_of\_dropdown \_value
* Select from list by index \_name\_of\_dropdwon \_index
* Unselect from list by label/index/value \_name\_of\_dropdown

Waits and Timeouts

* Sleep
  + Sleep 3 (seconds)
* Selenium speed – every statement will have a delay or wait
  + Set selenium speed 3 seconds
* wait until page contains – default is 5 seconds
  + wait until page contains ${element\_name} timeout 10 seconds
* implicit wait – will wait to specific element for some time
  + set selenium implicit wait 10 seconds

Close single and multiple browser

* close browser – close single browser
* close all browser – close multiple browser

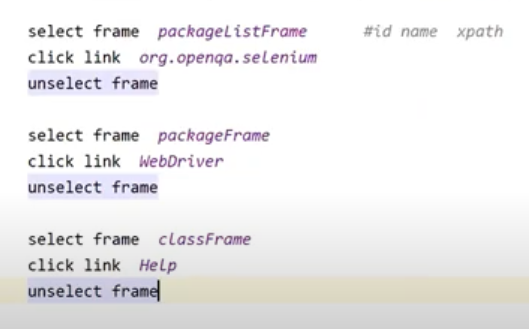
Alerts and Frames

* handle alert accept #to click ok button
* handle alert dismiss #to click cancel button
* handle alert leave #to disregard the alert

Verify the value inside the alert

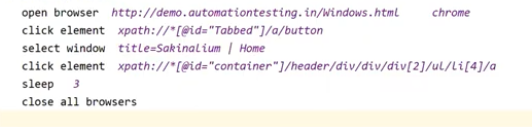
* alert should be present \_value\_on\_the\_alerts

Frames

* select frame ${frame\_location}   
  

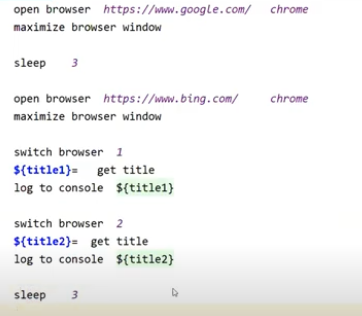
Handle tabbed windows

* select tab on window



Handle multiple browsers

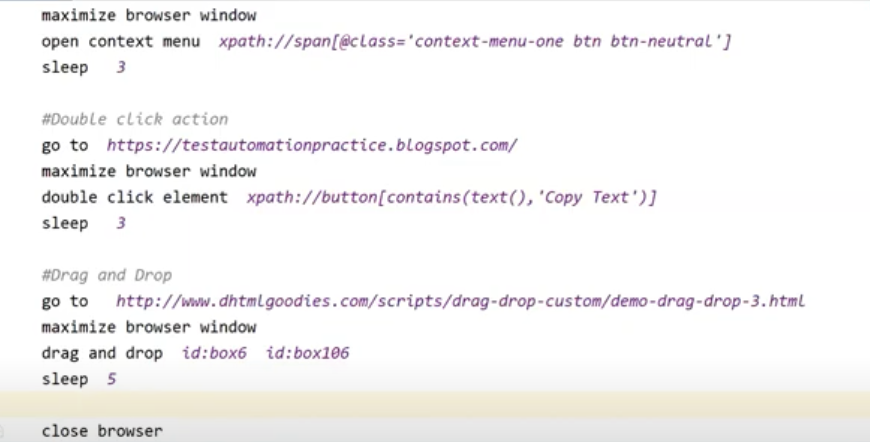
* go to
* go back
* get location
* select browser



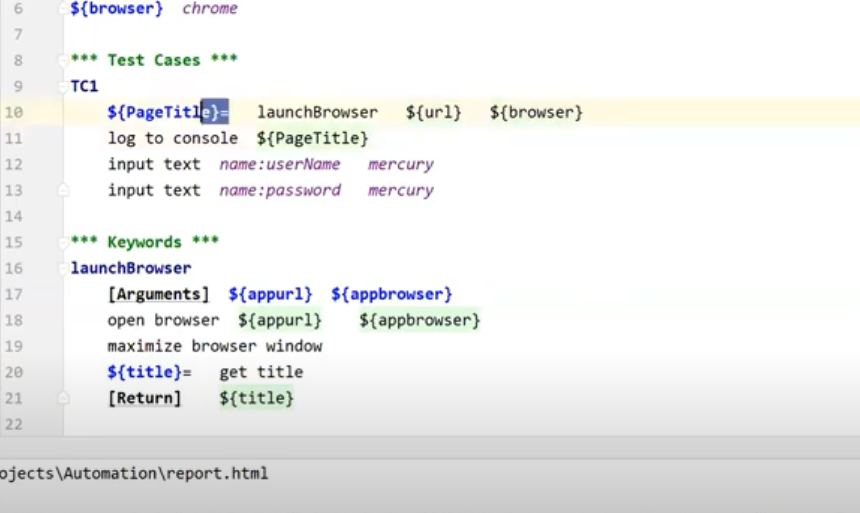
Capture Element

* For specific element
  + Capture element screenshot location\_of\_element directory/where to store the SS
* For full page
  + Capture page screenshot directory/where to store the ss

Mouse operations



Keywords and Resources



Scrolling page using JavaScriptExecutor

* Scrolling page till it reach a pixel number
  + Execute javascript window.scrollTo(0,1500) – (horizontal, vertical)
* Scrolling page till find element on page
  + Scroll element into view location\_of\_the \_element
* Scroll page till the bottom
  + End of the page – execute javascript window.scrollTo(0,document.body.scrollHeight)
  + Till starting of the page - execute javascript window.scrollTo(0, -document.body.scrollHeight)

For loop

#list

ForLoop  
 ${item} create list *emely gerald ronald izzie grey* FOR ${i} IN ${item}  
 log to console ${i}  
 END