**Selenium Important Functions**

**Scroll up to certain element:**

JavascriptExecutor JJ = (JavascriptExecutor) driver;

WebElement element = driver.findElement(By.*xpath*("//\*[@id='appendixUploadForm"+All\_Sections\_Id\_array.get(0)+"\_browse']/span[2]"));

JJ.executeScript("arguments[0].scrollIntoView(true);",element);

**Display alert using JavaScript:**

JavascriptExecutor JJ = (JavascriptExecutor) driver;

JJ.executeScript("alert('Be patient we are find out section whose not replaced yet...');");

**Browser name, version and OS:**

Capabilities cap = ((RemoteWebDriver) driver).getCapabilities();

String browserName = cap.getBrowserName().toLowerCase();

System.out.println(browserName);

String os = cap.getPlatform().toString();

System.out.println(os);

String v = cap.getVersion().toString();

System.out.println(v);

**Handle multiple window**

String winHandleBefore = driver.getWindowHandle();

**for**(String winHandle : driver.getWindowHandles()){

driver.switchTo().window(winHandle);

}

driver.close();

driver.switchTo().window(winHandleBefore);

**Open new tab**

((JavascriptExecutor)driver).executeScript("window.open();");

**alert handling:**

driver.switchTo().alert().accept();

**Reload the page:**

driver.navigate().refresh();

**Drag and Drop:**

WebElement Sourcelocator = driver.findElement(By.*xpath*("//\*[@id='colchooser\_gridfindreports']/div/div/div[1]/ul/li[9]"));

WebElement Destinationlocator = driver.findElement(By.*xpath*("//\*[@id='colchooser\_gridfindreports']/div/div/div[1]/ul/li[8]"));

*HighlightElement*(driver, Sourcelocator);

*HighlightElement*(driver, Destinationlocator);

Actions drag = **new** Actions(driver);

drag.dragAndDrop(Sourcelocator,Destinationlocator).perform();

**Hover Element:**

Actions action = **new** Actions(driver);

action.moveToElement(ASN).perform();

Thread.*sleep*(6000); **<- For hold that hover effect.**

**Drag and Drop(Slider):**

JavascriptExecutor js = (JavascriptExecutor) driver;

WebElement slider = driver.findElement(By.*xpath*("//\*[@id='Sliderbar']/ul/li[2]/div[3]/div[1]/div[4]"));

Actions builder = **new** Actions(driver);

Action dragAndDrop = builder.clickAndHold(slider).moveByOffset(40,0).release().build();

dragAndDrop.perform();

**Change Attribute of webelement:**

<div class="slider-handle min-slider-handle round" style="top: 28%;" tabindex="0"/>

WebElement Element = driver.findElement(By.*xpath*(".//\*[@id='snrbm']/ul/li[2]/div[3]/div[1]/div[4] "));

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].setAttribute('style', 'top: 50%;')", Element);

**print pdf file from local into console :**

first import pdfbox and fontbox jar files.

PDDocument pd= PDDocument.*load*(**new** File("C:\\Users\\compu\\Downloads\\pdf.pdf"));

System.***out***.println("Total Pages of PDF: "+pd.getNumberOfPages());

PDFTextStripper pdf= **new** PDFTextStripper();

System.***out***.println("Search included PDF into All Appendix sections: \n"+pdf.getText(pd));

**Handle frames:**

driver.switchTo().frame("cq-cf-frame");

**or**

driver.switchTo().defaultContent(); // you are now outside both frames

driver.switchTo().frame("cq-cf-frame");

**xpath for text:**

.//\*[@id='appendixUploadForm118\_browse']/span[text()='Add Files']

**for get last element**

(//div[@class='pull-right']/div/button)[last()]

**For first element**

(//div[@class='pull-right']/div/button)[1]

**Contains function**

//a[text()='upload document' and contains(@onclick,'uploadType=cover')]

**Xpath functions**

<http://www.way2tutorial.com/xml/xpath_functions_with_examples.php>

**xpath for to get “td” element on the basis of it’s “th” element in table.**

//\*[@id='divViewProjectInfo']/table/tbody/tr[th[text()='Project Name:']]/td

**Xpath for to get element globally which present under div/div/div:**

//div[h6[contains(text(),'EDR Radius Map')]]//a[contains(@ng-click,'openFile') or contains(@ng-click,'mergePDF')]

**For AutoIT tool relative path and it’s functions:**

<https://www.autoitscript.com/autoit3/docs/macros.htm>

**Handle dropdown:**

Select state = **new** Select(dropdown xpath);

state.selectByIndex(1);

or

state.selectByValue(“option1”);

**get the selected option:**

WebElement mySelectElement = driver.findElement(By.id("mySelect"));

Select dropdown= new Select(mySelectElement);

dropdown.selectByVisibleText("Italy");

WebElement option = dropdown.getFirstSelectedOption();

System.out.println(option.getText()); //output "Italy

**list of options from a dropdown element:**

WebElement mySelectElement = driver.findElement(By.id("mySelect"));

Select dropdown= new Select(mySelectElement);

List options = dropdown.getOptions();

for (WebElement option : options) {

System.out.println(option.getText()); //output "option1", "option2", "option3"

}

**Firefox handle pop-up and save file directory**

FirefoxProfile firefoxProfile = new FirefoxProfile();

firefoxProfile.setPreference("browser.download.folderList",2);

firefoxProfile.setPreference("browser.download.manager.showWhenStarting",false);

firefoxProfile.setPreference("browser.download.dir","c:\\downloads");

firefoxProfile.setPreference("browser.helperApps.neverAsk.saveToDisk","text/csv");

capture screenshot

**public** **static** **void** Take\_Screenshot(WebDriver driver, String Screenshot\_Name) **throws** Exception {

TakesScreenshot ts= (TakesScreenshot)driver;

File Source= ts.getScreenshotAs(OutputType.***FILE***);

FileUtils.*copyFile*(Source, **new** File("src/Test\_Data/CapturedScreenshots/"+Screenshot\_Name+".png"));

System.***out***.println("Error Screenshot taken please open Captured Screenshots folder");

}

**To check exist element or not:**

**private** **boolean** ExistsElement(WebElement Element) {

**try** {

Element.getText();

} **catch** (NoSuchElementException e) {

**return** **false**;

}

**return** **true**;

}

If element is not present then driver takes too much time to find element. To run operation quickly then we need to add “0” implicit wait before try block.

**private** **boolean** ExistsElement(WebElement Element) {

driver.manage().timeouts().implicitlyWait(0, TimeUnit.***SECONDS***);

**try** {

Element.getText();

} **catch** (NoSuchElementException e) {

**return** **false**;

}

**return** **true**;

}

There are 3 important things going on here. In order:

1.Setting implicity\_wait to 0 so that WebDriver does not implicitly wait.

2.Returning True when the element is found.

3.Catching the NoSuchElementException and returning False when we discover that the element is not present instead of stopping the test with an exception.

If(ExistsElement(Element\_A)){

}

**Wait Element:**

//below is the explicit wait which wait for 10 sec to run further program.

WebDriverWait myWaitVar= **new** WebDriverWait(driver, 10);

myWaitVar.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("/html/body/div/div/div/div/div/div/div/div/form/input[1]")));

//implicit wait

*wd*.manage().timeouts().implicitlyWait(30, TimeUnit.***SECONDS***);

**Scanner:**

**For get entered value from console panel of eclipse**

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter the browser");

String brows=sc.next();

**if**(brows.equalsIgnoreCase("chrome"))

{

System.*setProperty*("webdriver.chrome.driver", "..\\PARCEL\_projectDashboard\\src\\browserDrivers\\chromedriver.exe");

*wd*=**new** ChromeDriver();

}

**Adjust Slider:**

WebElement slider = driver.findElement(By.*xpath*("//\*[@id='snrbm']/ul/li[2]/div[3]/div[1]/div[4]"));

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].setAttribute('style', 'top: 50%;')",slider);

**Sort Array(For compare arrays we need first sort array if we don’t care about their order):**

Collections.*sort*(All\_Popout\_Databases\_array);

Collections.*sort*(All\_Databases\_array);

**Enter text into tinymce editor:**

driver.switchTo().frame(driver.findElement(Tinymce\_iframe));

JavascriptExecutor JJ = (JavascriptExecutor) driver;

JJ.executeScript("arguments[0].innerHTML = '<h1>Set text using innerHTML</h1>'", driver.findElement(Tinymce\_Editor));

**Without switching frame:**

JJ.executeScript("tinyMCE.activeEditor.setContent('<h1>Native API text</h1> TinyMCE')");

**For write PDF file data into console panel:**

Excel\_Data excel= **new** Excel\_Data(2, "src\\Test\_Data\\Appendices.xlsx");

String Downloaded\_PDF="";

**for**(**int** e=1; e<=excel.rowcount3; e++){

Downloaded\_PDF=excel.Open\_Downloaded\_PDF(1, 0);

PDDocument pd= PDDocument.load(**new** File(""+Downloaded\_PDF+""));

System.***out***.println("Total Pages of PDF: "+pd.getNumberOfPages());

PDFTextStripper pdf= **new** PDFTextStripper();

System.***out***.println("Search included PDF into Photograph Appendix section: \n"+pdf.getText(pd));

**How to remove characters from string:**

String s1 = " 8 sometext 7 3";

String arr[] = s1.trim().split("[a-zA-Z ]+"); // Please note a space is there after Z

String sum = "";

**for** (**int** i = 0; i < arr.length; i++){

sum = sum+arr[i];

}

System.***out***.println(sum);

**Break, continue for loop:**

1. If you want to skip a particular iteration, use continue.

2. If you want to break out of the immediate loop use break

3 If there are 2 loop, outer and inner.... and you want to break out of both the loop from the inner loop, use break with label.

for(int i= 0 ; i < 5; i++){

if(i==2){

continue;

}

System.out.print(i);

}

This will print

0134

**break**

for(int i=0 ; i<5 ; i++){

if (i==2){

break;

}

}

**break with label**

lab1: for(int j=0 ; j<5 ; j++){

for(int i=0 ; i<5 ; i++){

if (i==2){

break lab1;

}

}

}

**Drag and Drop for target and destination in different frames:**

//First hold the source element

Actions builder = **new** Actions(driver);

builder.clickAndHold(Library);

Action action = builder.build();

action.perform();

//switch to destination frame where we want to drop data.

WebElement Tinymce\_iframe = driver.findElement(By.*xpath*("//\*[@id='content4\_ifr']"));

driver.switchTo().frame(Tinymce\_iframe);

*HighlightElement*(driver, driver.findElement(Tinymce\_Editor));

builder = **new** Actions(driver);

builder.moveToElement(driver.findElement(Tinymce\_Editor)).perform();

builder.build();

builder.release();

//drop the data into destination.

builder.release(driver.findElement(Tinymce\_Editor)).perform();

*HighlightElement*(driver, driver.findElement(Tinymce\_Editor));

driver.switchTo().defaultContent();

**If element not clickable using “.click()” method:**

Actions builder = **new** Actions(driver);

builder.moveToElement(webelement).click(webelement);

builder.perform();