**Soft Assertion**

@Test

**public** **void** softAssertion() {

SoftAssert assertion = **new** SoftAssert();

System.***out***.println("Soft assertion started");

assertion.assertEquals(13, 12);

System.***out***.println("Soft assertion ends");

assertion.assertAll(); **一定要加在最後一行**

**WebDriverManager**

1. Add “webdrivermanager”dependency from mvn

**public** **class** TitleCheck

{

WebDriver driver;

@BeforeTest

**public** **void** setupBrowser()

{

// System.setProperty("webdriver.chrome.driver", "chromedriver\_80\_2.exe");

WebDriverManager.*chromedriver*().setup();

driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.get("http://www.google.com");

}

**Facebook Screenshot**

**public** **class** FacebookScreenshot {

**static** WebDriver *driver*;

@Test

**public** **void** captureScreenshot() **throws** Exception

{

System.*setProperty*("webdriver.chrome.driver","chromedriver\_80\_2.exe");

// WebDriverManager.chromedriver().setup();

*driver* = **new** ChromeDriver();

*driver*.manage().window().maximize();

*driver*.get("http://www.facebook.com");

WebElement username=*driver*.findElement(By.*xpath*("//input[@id='email']"));

Utility.*highLightElementcaptureScreenshot*(*driver*, username,"Username Highlighted"); **有個Utility.java檔案在lib裡**

username.sendKeys("dheeru.nmishra@gmail.com");

Utility.*captureScreenshot*(*driver*, "Username Enetered");

WebElement password=*driver*.findElement(By.*xpath*("//input[@id='pass']"));

Utility.*highLightElementcaptureScreenshot*(*driver*, password, "Password Highlighted");

password.sendKeys("1234");

Utility.*captureScreenshot*(*driver*, "Password Enetered");

}

}

**Facebook Screenshot > Utility.java**

**public** **class** FacebookScreenshot {

**static** WebDriver *driver*;

@Test

**public** **void** captureScreenshot() **throws** Exception

{

System.*setProperty*("webdriver.chrome.driver", "chromedriver\_80\_2.exe");

// WebDriverManager.chromedriver().setup();

*driver* = **new** ChromeDriver();

*driver*.manage().window().maximize();

*driver*.get("http://www.facebook.com");

WebElement username=*driver*.findElement(By.*xpath*("//input[@id='email']"));

//highlight Username box

Utility.*highLightElementcaptureScreenshot*(*driver*, username,"Username Highlighted");

username.sendKeys("dheeru.nmishra@gmail.com");

//Capture Username being correctly entered

Utility.*captureScreenshot*(*driver*, "Username Enetered");

WebElement password=*driver*.findElement(By.*xpath*("//input[@id='pass']"));

//highlight Password box

Utility.*highLightElementcaptureScreenshot*(*driver*, password, "Password Highlighted");

password.sendKeys("1234");

//Capture Password being correctly entered

Utility.*captureScreenshot*(*driver*, "Password Enetered");

}

}

**About Return**

We use return whenever we want the returned thing to be used in future in our project code.

For example captureScreenshot method we returned screenshotNme just to use the physical screenshot to attach in our Automation Test Report

**Regular Expression**

^[0-9.]\*$

^ means accept

. means with decimal

0-9 means always in a range of numbers

star means All

$ means reject discarded

-> which means accept from [ ] the rest of things like char, string will be all rejected

**Read&Write Files**

.xlxs > XSSF format

(must with MS office installed\_

.xls > HSSF format

(frequently being used without MS office)

**Scanner Class**

Scanner s=new Scanner(System.in);

int x=s.nextInt();

Float x=s.nextFloat();

Double x=s.nextDouble();

String x=s.nextLine();

**DropDown and Select Tag**

1. normal DropDown (we can select values by using Select class)

2. bootstrap DropDown (not include Select tag)

**Self-Review Note**

Initiates Actions class

Doubleclick/Rightclick

* Actions actions= **new** Actions(driver);
* Use . to call doubleClick and contextClick actions.doubleClick(elementToClick).build().perform();

actions.contextClick(elementToClick).build().perform(); (contextClick is Rightclick)

Mousehover

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

actions.moveToElement(elementToHover).build().perform();

or actions.moveToElement(elementToHover).click(elementToClick).build().perform();

Deal with Alert

* Alert alert = driver.switchTo().alert();

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

* Use . to get AlertText

System.***out***.println("Alert Text\n" + alert.getText());

Scroll Up and Down

* Typecasting “JavascriptExecutor”

JavascriptExecutor je = (JavascriptExecutor) *driver*;

* je.executeScript("arguments[0].scrollIntoView(true);",element); System.***out***.println(element.getText());
* je.executeScript("scroll(0,1400)");

Call another Java file’s method

(that java file.method name(driver, WebElement object, “result”)

Utility.*highLightElementcaptureScreenshot*(*driver*, username,"Username Highlighted");

* **public** **class** RighClick **extends** MouseAction { //if ClassName extends Lib -> you can directly use methodname(argument) ex. *rightClick*(driver, WebElement object);

another Java file’s