SELENIUM Cheat Sheet

Creating Driver:

Action	Selenium code	Description
Firefox driver	WebDriver Driver = new FirefoxDriver ();	Driver is an object
IE driver	System.*setProperty*("webdriver.ie. driver", PATH); WebDriver driver = **new** InternetExplorerDriver ();	PATH = path of IEDriver exe file;
chrome Driver	System.*setProperty*("webdriver.ie. driver",PATH); WebDriver driver = **new** ChromeDriver ();	PATH = path of chrome exe file;

Identify Elements:

Action	Code	Description
Find single Element	driver.findElement(locator)	Locator is a location of element
Find multiple Elements	driver.findElements(locator)	Locator is a location of element

Locating UI Elements:

Action	Code	Description
By ID	driver. findElement (By.id(str));	Str is id of element
By Name	driver.findElement(By.name(str));	Str is name of element
By class name	driver. findElement (By. className(str));	Str is class value of element
By CSS selector	driver.findElement(By.cssSelector(str));	Str is cssSelector of element
By link text	driver.findElement(By.linkText(str));	Str is link text of element
By partial link text	driver.findElement(By.partialLinkText(str));	Str is partial text of element
By tag name	driver.findElement(By.tagName(str));	Str is tag name of element
By XPath	driver.findElement(By.xpath(xpath));	Str is xpath of element

Handling Java Script Alerts:

To handle alert first we need to switch to alert.

Alert al=driver.switchTo().alert();
The Actions list.

Action	code
Click on ok in alert	al.accept();
Click on cancel.	al.dismiss()
Type in alert box.	al.sendKeys("text");
Get text from alert box.	al.getText();

Capture Screen Shot of Browser:

Action	Code	Description
Capture screen	File scrFile1 = ((TakesScreenshot)*driver*).getScreenshotAs(OutputType.*FILE*);	It captures screen shot of particular page and stores it in variable
Save to disk	FileUtils.*copyFile*(scrFile1, **new** File("c:\\tmp\\k2.png"));	Save screen shot as k2.png

User Actions:

Action	Code	Description
Write in text fields	driver.findElement(locator).sendKeys(text);	Text: what u want to write locator is a location element
Click button or click radio button or check box	driver. findElement(locator). click();	locator is a location element
Clear text in text field	driver. findElement(locator). clear();	locator is a location element
Navigate back and forward in browser	driver.navigate(). back(); driver.navigate().forward();	
Navigate to frame	driver. switchTo().frame(frame);	frame can be integer value represents position of frame or string represents id of frame or WebElement represents frame of frame.
Navigate to next window or pop up window	driver.switchTo().window(hashCode);	hashCode is hash code of window
Get inner text of element or inner text of table	driver.findElement(locator).getText();	locator is a location element
Working on auto complete/suggestions Or Calendar pop up	driver. findElement(locator). click();	Get the locator of hidden division or element and perform required operation.

select drop down list:

Using Select class we can work on select drop down. Create select object for specific select drop down.

//creating webelement for select dropdown

WebElement usrs=driver.findElement(By.name("users"));

Select usr=new Select(usrs);

We can select options of drop down in 3 different ways as explained below:

Action	code	description
Select by using id of option tag	usr.selectById(ID);	ID is string, value of id attribute of option.
Select by using index of option tag	usr.selectByIndex(i);	i is the position of option
Select by using visible text in option tag	usr.selectByVisibleText(str)	str is the text inside option tag.

Working on excel sheet:

Before working on excel first we need to read excel in input stream using file io stream.

FileInputStream fis=new FileInputStream("Path of .xlsx file");

Action	Code	Description
Convert file io into workbook	Workbook wb = WorkbookFactory.*create*(fis);	Create function creates work book.
Get into specified sheet	Sheet s = wb.getSheet(sheetName); Or Sheet s = wb.getSheetAt(sheetNum);	sheetName is name of the sheet sheetNum is index of sheet
Get into specified row	Row r = s.getRow(rowNum);	
Get into specified column	Cell c = r.getCell(colNum);	

Get cell value	String cellVal = c.getStringCellValue();
	Or
	boolean b = c.getBooleanCellValue();
	or
	Date d = c.getDateCellValue();
	Or
	int I = c.getNumericCellValue();
		Get cell value based on value in excel cell

Get row count	int I = s.getLastRowNum();	
Get Column count	int j = r. getLastCellNum ();	

Write back to excel	c.setCellValue("PASS1");
	FileOutputStream fos = **new** FileOutputStream("Path of .xlsx file ");
	wb.write(fos);
	fos.close();	

Drag, Drop and Mouse Over, Mouse Events:

We use Actions Class for drag and drop
Create an object to action class
Actions a=new Actions(driver);

Action	code	description
Drag and Drop using source and destination	a.dragAndDrop(src, des).build().perform();	Src and dest is the WebElement object of source and destination of drag and drop element.
Drag and drop to specific position	a.dragAndDrop(src, x,y).build().perform();	x and y are integer values for specific position.
Mouse over on specific element.	a.moveToElement(element).build().perform();	Element is an object of WebElement which points to required element.
Mouse right click	a.contextClick(element).build().perform();	Element is an object of WebElement which points to required element.
Mouse movement after right click	a.sendKeys(Keys.<keyboardstrokes>).build().perform();	Keys is a class contains all key strokes such as right left, enter, back button.