**Select Class:**

We use ‘**Select**’ class to handle single/multi select **list box (Drop-downs)**. It takes ‘***WebElement***’ as argument.

In order to select the required option present in the list box, we use selectBy methods of select class. First we should find the list box element using findElement() method then we should create instance of select class by specifying the list element as the argument for select class constructor and then we can call any one the following methods

**For single select list box we can use any one of the above methods**

1.selectByIndex(int);

2.selectByValue(string)

3.selectByVisibleText(string)

**For multi select list box we can use any one of the above methods**

1.selectByIndex(int);

2.selectByValue(string)

3.selectByVisibleText(string)

4.deSlectByIndex(int);

5.deSelectByValue(string)

6.deSelectByVisibleText(string)

7.deSelectAll()

For multi select list box, the above select methods will keep already selected options as it is and they will also select the newly specified options.

* We use ‘**getOption()**’ method to get all the options present in the listbox.

**Note:** We can use methods of select class to handle list box only if the list box is developed using “select” html tag. If it is developed using any other tag and if you try to use select class’ methods, it will throw **“unExpectedTagNameException”**. These type of list boxes are called as customized list boxes. To handle customized list boxes, we use “sendKeys()” method only where we pass the required option to select as argument for sendKeys method.

**Actions Class:**

It is used to handle keyboard and mouse related operation

**Dropdown Menus:** When we move mouse pointer on the menu, it will display the sub-menus, these are called as dropdown menus. In order to handle it we use Actions class.

Actions class constructor is overloaded which takes WebDriver type as argument i.e. in order to perform actions on the WebElement, we should specify WebDriver reference as argument for Actions class constructor.

Whenever we call any method of ‘Actions’ class we must call **‘perform()’** method at the end.

To place the mouse pointer on the element we use **‘moveToElement()’** method of Actions class.

1.**How do you handle ‘Drop-down’ menu? Or how do you mouse hover on the element?**

🡪 Using ‘moveToElement()’ method of ‘Actions’ class.—see program

**2.How do you perform ‘Drag & Drop’ in Selenium?**

🡪 Using ‘dragAndDrop()’ method of ‘Actions’ class.—see program

**3.How do you ‘Double click’ on the element?**

🡪 Using ‘doubleClick()’ method of ‘Actions’ class

**4.How do you perform keyboard & mouse actions without using Selenium & JavaScript?**

🡪 Using ‘Robot’ class.

Open ‘notepad’ & type ‘Hi’ using ‘Robot’ class:

|  |
| --- |
| public class KeyboardActionUsing\_RobotClass  {  public static void main(String[] args) throws IOException, AWTException  {  **Runtime.getRuntime().exec("notepad");**  **Robot r=new Robot();**  **r*.keyPress(KeyEvent.VK\_H);***  **r*.keyRelease(KeyEvent.VK\_H);***  **r.*keyPress(KeyEvent.VK\_I);***  **r.*keyRelease(KeyEvent.VK\_I);***  }  } |

**5. When we get ‘AWTException’?**

🡪 When we create Object of Robot class

AWT stands for abstract window toolkit

**6. What is ‘Context Menu’? How do you handle this in Selenium?**

**🡪** When we right click on any element we get list of options which is called as ‘Context Menu’.

**🡪** To ‘Right click’ on the element we use ‘**contextClick()**’ method of ‘Actions’ class. We cannot inspect the context menu hence to select the options we use shortcuts such as ‘t’ for ‘New Tab’, ‘w’ for ‘New Window’ etc… To type or press these shortcuts we use ‘**Robot’** class.

|  |
| --- |
| driver.get("http://localhost:8080/login.do");  *//Find element & right click*  ***WebElement*** link=**driver.findElement*(By.linkText("actiTIME Inc.")*);**  ***Actions*** action=**new Actions(driver);**  action.**contextClick(link).perform();**  *//Press 'W' in context menu to open in new browser*  **Robot** r=**new Robot();**  r.**keyPress*(KeyEvent.VK\_W)*;**  r.**keyRelease(*KeyEvent.VK\_W*);**  //Minimize child browser(Press 'Alt+Space' then press 'N'  r.keyPress(KeyEvent.VK\_ALT);  r.keyPress(KeyEvent.VK\_SPACE);  r.keyRelease(KeyEvent.VK\_ALT);  r.keyPress(KeyEvent.VK\_N);  driver.close(); |