Select Class

- A web page consists of multiple elements such as text fields, buttons, menus, drop-down options. As every tester is aware, Selenium is the most widely used tool for end-to-end automation testing of a website.
    - Introduction to Select Class in Selenium
    - Different Select Methods
    - Selecting Multiple items with select command

Introduction to Select Class in Selenium

In Selenium, the Select class provides the implementation of the HTML SELECT tag. A Select tag provides the helper methods with select and deselect options. As Select is an ordinary class, its object is created by the keyword New and also specifies the location of the web element.

Syntax : **Select refObj = new Select();**

In the syntax above, it clearly states that Select is asking for an element type object for its constructor, i.e it will create an object of the select class.

Different Select Methods

The following are the most commonly used methods to deal with a drop-down list:

- **selectByVisibleText: selectByVisibleText(String arg0): void**

This method is used to select one of the options in a drop-down box or an option among multiple selection boxes. It takes a parameter of String which is one of the values of Select element and it returns nothing.

Syntax : obj.selectByVisibleText(“text”);

Example:

```
Select refObj =new Select(driver.findElement(By.id("search-box")));
    refObj.selectByVisibleText("Automation");
```

- **selectByIndex: selectByIndex(int arg0) : void**

This method is similar to ‘selectByVisibleText’, but the difference here is that the user has to provide the index number for the option rather than text. It takes the integer parameter which is the index value of Select element and it returns nothing.

Syntax : refObj.selectByIndex(int);

Example:

```
Select refObj= new Select(driver.findElement(By.id("Seacrch-box")));
    refObj.selectByIndex(4);
```

- **selectByValue: selectByValue(String arg0) : void**

This method asks for the value of the desired option rather than the option text or an index. It takes a String parameter which is one of the values of Select element and it does not return anything.

Syntax : refObj.selectByValue(“text”);

Example:

```
Select refObj = new Select(driver.findElement(By.id("Search-box")));
  refObj.selectByValue("Automation Testing");
```

- **getOptions: getOptions( ) : List<WebElement>**

This method gets all the options belonging to the Select tag. It takes no parameter and returns List<WebElements>.

Syntax : refObj.getOptions();

Example:

```
Select refObj = new Select(driver.findElement(By.id("Search-box")));
List <WebElement> elementCount = refObj.getOptions();
System.out.println(elementCount.size());
```

- **deselectAll()**

This method clears all the selected entries. This is only valid when the drop-down element supports multiple selections.

Syntax : **refObj.deselectAll();**

Now let's delve deeper to understand how to select multiple items with the Select command.

How to select multiple items with the Select command?

- The multiple select attribute is a boolean expression. This method specifies that multiple options can be selected at once. These options vary for different operating systems and browsers.
- **For Windows**: Press the control (ctrl) button to select multiple options.
- **For Mac**: Hold down the command button to select multiple options.
- Use the ***isMultiple()*** method to select multiple commands.
- ***isMultiple()***: boolean – This method informs whether the Select element supports multiple selection options at the same time or not. This method accepts nothing and returns a boolean value (true/false).

Syntax : **refObj.isMultiple();**

Example:

```
Select refObj= new Select(driver.findElement(By.id(Element_ID)));
```

```
refObj.selectByIndex(index);
// Or can be used as
refObj.selectByVisibleText(text);
// Or can be used as
refObj.selectByValue(value);
```

Now let’s look at a real-time example of the select command.

selectByIndex(int)

```
public class SelectClassSelectByIndex {
    @Test
    public void singleSelect() throws Exception{
        WebDriverManager.chromedriver().setup();
        WebDriver driver = new ChromeDriver();
        driver.get("https://demoqa.com/select-menu");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        WebElement oldStyle = driver.findElement(By.id("oldSelectMenu"));
        Select refObj=new Select(oldStyle);
        refObj.selectByIndex(2);
        driver.close();
    }
}
```

SelectByValue(“value”)

```
public class SelectClassSelectByValue {
    @Test
    public void singleSelect() throws Exception{
        WebDriverManager.chromedriver().setup();
        WebDriver driver = new ChromeDriver();
        driver.get("https://demoqa.com/select-menu");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        WebElement oldStyle = driver.findElement(By.id("oldSelectMenu"));
        Select refObj=new Select(oldStyle);
        refObj.selectByValue("4");
        driver.close();
    }
}
```

SelectByVisibleText(“text”)

```
public class SelectClassSelectByVisibleText {
        @Test
        public void singleSelect() throws Exception{
                WebDriverManager.chromedriver().setup();
                WebDriver driver = new ChromeDriver();
                driver.get("https://demoqa.com/select-menu");
                driver.manage().window().maximize();
                driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
                WebElement oldStyle = driver.findElement(By.id("oldSelectMenu"));
                Select refObj=new Select(oldStyle);
                refObj.selectByVisibleText("Aqua");
                driver.close();
        }
}
```

GetOptions()

```
public class SelectClassGetOptions {
    @Test
    public void singleSelect() throws Exception{
        WebDriverManager.chromedriver().setup();
        WebDriver driver = new ChromeDriver();
        driver.get("https://demoqa.com/select-menu");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        WebElement oldStyle = driver.findElement(By.id("oldSelectMenu"));
        Select refObj=new Select(oldStyle);
        List<WebElement> options = refObj.getOptions();
        for(WebElement option:options) {
            System.out.println(option.getText());
        }
        driver.close();
    }
}
```

isMultiple() & deselectAll()

```
public class SelectClassMultiSelectAndDeselectAll {
    @Test
    public void multiSelect() throws Exception{
        WebDriverManager.chromedriver().setup();
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        driver.get("https://demoqa.com/select-menu");
        WebElement oldStyle = driver.findElement(By.id("cars"));
        Robot robot = new Robot();
        robot.mouseWheel(10);
        Select refObj=new Select(oldStyle);
        System.out.println(refObj.isMultiple());
        refObj.selectByIndex(0);
        refObj.selectByValue("saab");
        refObj.selectByVisibleText("Audi");
        refObj.deselectAll();
        driver.close();
    }
}
```