
Select Class

- Just like *CheckBox & Radio Buttons*, **List box & Multiple Select List Box** Operations also works together and almost the same way.
- In order to perform any action, the first task is to identify the element group. As *DropDown/Multiple Select* is not a single element, they contains one or more than one elements in them.
- In order to deal with the list box and multi select list box, selenium provides the class called **SELECT CLASS**.

Drop down/List box handling:

- To perform any operation on list/multi select drop down elements we use select class. Select class is available in the following package.

```
import org.openqa.selenium.support.ui.Select;
```

- First identify the select tag element and create the object of select class and pass the identified element as an argument for the select class constructor.

```
WebElement listBox = driver.findElement(By.name("dropdown"));  
Select select = new Select(listBox);
```

- Following are the methods provided by the select class to perform the operation on drop down/list box.
- To select the option from the drop down, we use the following methods.

```
WebElement listBox = driver.findElement(By.name("Mobdevices"));  
Select select = new Select(listBox);  
select.selectByIndex("Index");  
select.selectByValue("value");  
select.selectByVisibleText("Text");
```

Below is the sample HTML code

```

<!DOCTYPE html>
<html>
  <head>
    <title>HTML select Tag</title>
  </head>
  <body>
    <form action="/cgi-bin/dropdown.cgi" method="post">
      <select name="dropdown">
        <option value="Data Structures" >Data Structures</option>
        <option value="Data Mining">Data Mining</option>
        <option value="Java">Core Java</option>
        <option value="Sele" selected>Selenium</option>
        <option value="py">Python</option>
        <option value="app">Appium</option>
        <option value="ptest">PyTest</option>
      </select>
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>

```

SelectByIndex (Index) -> this method takes index as an argument to select the options from the drop down/list box.

Script to select the option using select by index?

```

package select;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class SelectByIndex {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "D:/Softwares/Drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/IMP/HTML%20Pages/SelectDropdown.html");

        WebElement listBox = driver.findElement(By.name("dropdown"));
        Select select = new Select(listBox);
        select.selectByIndex(0);

        driver.close();
    }
}

```

Select_by_value ("Value") -> this method takes the backend attribute value for selecting the option from the list box.

Script to select the option from the list box using value?

```

package select;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class SelectByValue {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "D:/Softwares/Drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/IMP/HTML%20Pages/SelectDropdown.html");

        WebElement listBox = driver.findElement(By.name("dropdown"));
        Select select = new Select(listBox);
        select.selectByValue("ptest");

        driver.close();
    }
}

```

Select_by_visible_text ("text") -> this method select the options from the list box by visible text.

Script to select the option from the list box using visible text?

```
package select;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class SelectByVisibleText {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "D:/Softwares/Drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/IMP/HTML%20Pages/SelectDropdown.html");

        WebElement listBox = driver.findElement(By.name("dropdown"));
        Select select = new Select(listBox);
        select.selectByVisibleText("Appium");
        driver.close();
    }
}
```

Script to verify whether the list box is single select or mutli select?

1. We use **isMultiple ()** method present in Select class to verify if the list box is single select drop down or multi select drop down.
2. Returns **TRUE**, if the list box is multi-select list box
3. Returns **FALSE**, if the list box is single select list box

```
package select;

import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class Is_multiple {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/IMP/HTML%20Pages/multiSelect.html");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

        WebElement listBox = driver.findElement(By.name("Mobdevices"));
        Select select = new Select(listBox);

        Boolean status = select.isMultiple();
        if (status) {
            System.out.println("List box is multi select list box");
        } else {
            System.out.println("List box is single select list box");
        }
        driver.close();
    }
}
```

Write a script to count the number of options present in the drop down box?

- In order to get all the options present in the drop down, Select class provide a method “**getOptions ()**”, getOptions method returns **list of web elements**.

```
package select;

import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class CountNumberOfOptions {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/IMP/HTML%20Pages/SelectDropdown.html");

        WebElement listBox = driver.findElement(By.name("dropdown"));
        Select select = new Select(listBox);

        List allOptions = select.getOptions();
        int count = allOptions.size();
        System.out.println("Total number of options present in a drop down is: " + count);

        driver.close();
    }
}
```

Write a program to print all the options present in the drop down

```
package select;

import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class GetAllOptions {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "D:/Softwares/Drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/IMP/HTML%20Pages/SelectDropdown.html");

        WebElement listBox = driver.findElement(By.name("dropdown"));

        Select select = new Select(listBox);
        List allOptions = select.getOptions();

        for (WebElement option : allOptions) {
            System.out.println(option.getText());
        }

        driver.close();
    }
}
```

Sample HTML for developing the multi-select list box

```
<html>
<head>
<title>Multi select Drop Down List Box</title>
</head>
<body>
<p>What all devices do you listen to music on?</p>
<select name="Mobdevices" multiple="multiple"><option value="0" selected> Please select</option>
<option value="1">iPhone</option>
<option value="2">Nokia</option>
<option value="3">Samsung</option>
<option value="4">HTC</option>
<option value="5">BlackBerry</option>
</select>
</body>
</html>
```

- To deselect the option from the drop down, use the following methods.
 1. DeselectByIndex (Index)
 2. DeselectByValue ("String")
 3. DeselectByVisibleText ("String")
 4. DeselectAll ()
- De-select function has to be used only with multi-select drop downs.
- If we try to use these function with single select drop down. We will get the unsupported tag exception.

Exception in thread "main" java.lang.UnsupportedOperationException: You may only deselect options of a multi-select

Write a script to select multiple options from the drop down and deselect all the option

- In order to deselect all the selected options from the multi select drop down, use "DeselectAll ()" function.

```

package select;

import java.util.List;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class DeselectAll {
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/IMP/HTML%20Pages/multiSelect.html");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        WebElement listBox = driver.findElement(By.name("Mobdevices"));
        Select select = new Select(listBox);

        List allOption = select.getOptions();
        int count = allOption.size();
        for (int i = 1; i < count; i++) {
            select.selectByIndex(i);
            Thread.sleep(1000);
        }
        Thread.sleep(5000);
        select.deselectAll(); // De-selecting all the selected options
        driver.close();
    }
}

```

Activate W

- In order to deselect the single option from the drop down, use below methods.
 1. DeselectByIndex
 2. DeselectByValue
 3. DeselectByVisibleText

Write a program to deselect the option using the above methods

```

package select;

import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class Deselect_single_options {
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/IMP/HTML%20Pages/multiSelect.html");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

        WebElement listBox = driver.findElement(By.name("Mobdevices"));
        Select select = new Select(listBox);
        select.selectByIndex(0);
        select.selectByValue("2");
        select.selectByVisibleText("HTC");

        // De-Selecting the selected options
        select.deselectByIndex(0);
        select.deselectByValue("2");
        select.deselectByVisibleText("HTC");

        driver.close();
    }
}

```

Activate V

Write a script to get the first selected option from the multi select drop down?

- In order to get the first selected option from the multi select drop down, use `getFirstSelectedOption` method.
- `getFirstSelectedOption` returns the first selected option from the drop down.
- Return type of the above method is web element.

```

package select;

import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class Get_first_selected_option {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/IMP/HTML%20Pages/multiSelect.html");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

        WebElement listBox = driver.findElement(By.name("Mobdevices"));
        Select select = new Select(listBox);

        select.selectByIndex(5);
        select.selectByIndex(3);
        WebElement firstSelectedOption = select.getFirstSelectedOption();
        System.out.println("First selected option in the list box is: "+firstSelectedOption.getText());
        driver.close();
    }
}

```

Write a script to get all the selected option from the drop down?

- In order to get all selected option from the multi select drop down, use **"getAllSelectedOptions"** method.
- "getAllSelectedOptions" method gets all the selected option from the drop down.
- Return type of this method is list of web element.

```
package select;

import java.util.List;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class Get_all_selected_option {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/IMP/HTML%20Pages/multiSelect.html");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        WebElement listBox = driver.findElement(By.name("Mobdevices"));
        Select select = new Select(listBox);

        select.selectByIndex(4);
        select.selectByIndex(1);
        select.selectByIndex(5);
        List allSelectedOption = select.getAllSelectedOptions();

        for (WebElement option : allSelectedOption) {
            System.out.println(option.getText());
        }
        driver.close();
    }
}
```

Activate W
Get to Settings

Write a program to search for the specified option from the drop down.


```

package select;

import java.util.List;

public class SearchSpecifiedOption {
    public static void main(String[] args) throws InterruptedException {
        String eText = "Selenium";
        int found = 0;

        WebDriver driver = new ChromeDriver();
        driver.get("file:///D:/HTML%20Pages/Practice/ListBox/DuplicateOption.html");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

        WebElement listBox = driver.findElement(By.name("dropdown"));
        Select select = new Select(listBox);

        List<WebElement> allOptions = select.getOptions();
        System.out.println("Searching...");
        for (WebElement option : allOptions) {
            String value = option.getText();
            if (value.equals(eText)) {
                found++;
            }
        }
        if (found == 0) {
            System.out.println(eText + " is not found");
        } else if (found == 1) {
            System.out.println(eText + " is found");
        } else {
            System.out.println(eText + " is duplicate");
        }
        driver.close();
    }
}

```

Write a program to get all the option from the drop down, sort all the option and print

```
package select;

import java.util.ArrayList;

public class SortAllTheOptions {
    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/HTML/ListBox/SelectDropdown.html");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

        WebElement listBox = driver.findElement(By.name("dropdown"));
        Select select = new Select(listBox);

        List allOptions = select.getOptions();
        ArrayList<String> allText = new ArrayList<String>();
        for (WebElement option : allOptions) {
            String str = option.getText();
            allText.add(str);
        }

        Collections.sort(allText);

        for (String s : allText) {
            System.out.println(s);
        }
        driver.close();
    }
}
```

Note:

1. If the specified option is not in the drop down means, we will get the "NoSuchElementException".
2. If the specified option is duplicate, select class will always performs the operation of the first matched option.

Very Important:

- We can use the select class to perform the operation on the drop downs, if it is only developed using the "Select" tag.
- If the drop down is developed without using the select tag. Than such kind of drop down cannot be handled by the "Select" class.
- If we try to use "Select" class on the drop downs which is not built using the "Select" tag. Selenium web driver will through the below exception.

OSS

n" org.openqa.selenium.support.ui.UnexpectedTagNameException: Element should have been "select" 10.0', revision: '176b4a9', time: '2018-03-02T19:11:12.151Z'

- Then, how to handle the drop downs which is built without using the drop downs.
- Drop downs which is built without using the select tag are called as customised drop downs.

- In order to handle the customized drop downs, we use web elements commands only.

Customized drop down

- Drop downs which is developed without using the select tags are called as customized drop downs.
- In order to handle the customized drop downs, we have to identify each and every elements using “Find Element” or “Find Elements” and have to perform the operations using web element commands.

Sample HTML:

```

<html>
  <head>
    <title>Qspiders</title>
  <style>
    .dropbtn {
      background-color: #4CAF50;
      color: white;
      padding: 16px;
      font-size: 16px;
      border: none;
      cursor: pointer;
    }

    .dropbtn:hover, .dropbtn:focus {
      background-color: #3e8e41;
    }

    .dropdown {
      position: relative;
      display: inline-block;
    }

    .dropdown-content {
      display: none;
      position: absolute;
      background-color: #f9f9f9;
      min-width: 160px;
      overflow: auto;
      box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
      z-index: 1;
    }

    .dropdown-content a {
      color: black;
      padding: 12px 16px;
      text-decoration: none;
      display: block;
    }

    .dropdown a:hover {background-color: #f1f1f1}
    .show {display:block;}
  </style>
</head>

```

```

<body>

<h2>Clickable Dropdown</h2>
<p>Click on the button to open the dropdown menu.</p>
<div class="dropdown">
  <button onclick="myFunction()" class="dropbtn">Dropdown</button>
  <ul id="myDropdown" class="dropdown-content">
    <li><a href="https://www.google.co.in/?gfe_rd=cr&ei=kUJrWfufGrPv8wfc35XACg">Google</a></li>
    <li><a href="https://www.facebook.com/">Facebook</a></li>
  </ul>
</div>

<script>
/* When the user clicks on the button,
toggle between hiding and showing the dropdown content */
function myFunction() {
  document.getElementById("myDropdown").classList.toggle("show");
}

// Close the dropdown if the user clicks outside of it
window.onclick = function(event) {
  if (!event.target.matches('.dropbtn')) {

    var dropdowns = document.getElementsByClassName("dropdown-content");
    var i;
    for (i = 0; i < dropdowns.length; i++) {
      var openDropdown = dropdowns[i];
      if (openDropdown.classList.contains('show')) {
        openDropdown.classList.remove('show');
      }
    }
  }
}
</script>
</body>
</html>

```

Write a script to handle the customized drop down?

```

package select;

import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class Customised_example1 {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "D:/Softwares/Drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/HTML/ListBox/CustomizeListBox.html");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

        WebElement dropDown = driver.findElement(By.xpath("//button[text()='Dropdown']"));
        dropDown.click();
        WebElement googleLink = driver.findElement(By.xpath("//a[text()='Google']"));
        googleLink.click();

        driver.close();
    }
}

```

Handling Suggestions

- Auto suggestions are the automatically generated suggestions by the server based on searching criteria.

Example:

- If we try to search java in google it will give suggestions like
 1. Java tutorial
 2. Java interview questions ETC.
- To handle the auto suggestions, we use find elements to identify all the auto suggestion elements and performs the operation on the single element.

Example:

1. Open the browser
2. Go to google
3. Type JAVA in the search text box
4. Find all the auto suggestion given by the google
5. Print the count
6. Print the text of all the auto suggestion
7. Select the 6th auto suggestion

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import java.util.List;

public class AutoSuggestion_Google {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "D:/Softwares/Drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

        WebElement searchBox = driver.findElement(By.name("q"));
        searchBox.sendKeys("java");
        List allSuggestions = driver.findElements(By.xpath("//ul[@role='listbox']/li"));
        int count = allSuggestions.size();
        System.out.println("Total number options in auto suggestion: " + count);

        for (WebElement option : allSuggestions) {
            String text = option.getText();
            System.out.println(text);
        }
        WebElement SixthElement = allSuggestions.get(5);
        SixthElement.click();

        driver.close();
    }
}
```

Activate Window
Go to Settings to activate

Assignment on autosuggestions

1. Go to "yatra.com"
2. Enter the city name in "From" text box and handle the auto suggestion drop down