**Sorting the Dropdown elements in Ascending order**

There are different ways to sort the drop down elements

1)By using for loop

2)By using addAll()

3)By using Treeset

4)stream().collect

Check the below link for other ways:

<https://chercher.tech/java/dropdown-sorting>

Below is the example using for Loop

Select sel=**new** Select(driver.findElement(By.*xpath*("//form[@name='callForm']//select[@name='flag']")));

//Get all options from the dropdown

List<WebElement> allOptions=sel.getOptions();

//creating a list to store dropdown options

List<String> storeOptions=**new** ArrayList<String>();

//Storing in a list

**for**(WebElement optionElement : allOptions)

{

storeOptions.add(optionElement.getText());

}

//HEre in the dropdown the first option is space so i am removing that as it is not an option

storeOptions.remove("");

//if Select is the first option. Remove "Select" option as it is not actual option

//storeOptions.remove("Select");

// Default order of option in drop down

System.***out***.println("Options in dropdown with Default order :"+storeOptions);

// Creating a temp list to sort

List<String> tempList = **new** ArrayList<String>(storeOptions);

// Sort list ascending

Collections.*sort*(tempList);

System.***out***.println("Sorted List "+ tempList);

// equals() method checks for two lists if they contain the same elements in the same order.

**boolean** ifSortedAscending = storeOptions.equals(tempList);

**if**(ifSortedAscending)

{

System.***out***.println("List is sorted");

}

**else**

System.***out***.println("List is not sorted.");

}

**By using Treeset:**

Before we proceed with the process lets see the trait of Treelist:

1. The objects of the TreeSet class are stored in ascending order
2. The TreeSet stores the objects based on the comparator provided, if there is no comparator then it is stored in ascending order.
3. We can use the TreeSet for verification dropdown option order, we have to create a Treeset object using the list as a parameter for it's constructor.  
   Compare the values of the treeSet and the List using [Assert methods](https://chercher.tech/java/testng-assertions-selenium-webdriver).

**WebElement element = driver.findElement(By.xpath("//select[@id='animals']"));**

**Select se = new Select(element);**

**List<String> originalList = new ArrayList();**

**for (WebElement e : se.getOptions()) {**

**originalList.add(e.getText());**

**}**

**System.out.println("\n this is originalList before Sorting tempList"+ originalList);**

**Set<String> treeset = new TreeSet(originalList);**

**System.out.println("\n this is TreeSet "+ treeset);**

**System.out.println("\n this is originalList after sorting tempList"+ originalList);**

**Assert.assertEquals(treeset, originalList);**

[Selenium](https://chercher.tech/java/index-selenium-webdriver) provides Select class, with help of select class we can handle dropdowns on the webpage

**Steps to select a value in the dropdown in selenium:**  
1. [Find the dropdown using findElement](https://chercher.tech/java/find-elements-in-selenium)  
2. Create an object for **Select class** passing dropdown element as a parameter  
3. Select the value from the dropdown values  
  
**There are two types of dropdowns :**  
1. Single value Dropdown  
2. Multi-Value Dropdown

**Methods Present in Select Class ( All methods are non-static )**

* *selectByIndex(int index)*
* *selectByValue(String value)*
* *selectByVisibleText(String text)*
* *getOptions()*
* *getFirstSelectedOption()*
* *getAllSelectedOptions()*
* *isMultiple()*
* *deselectByIndex(int index)*
* *deselectByValue(String value)*
* *deselectByVisibleText(String text)*
* *deselectAll()*

**Dropdown are formed using select tag in html, if dropdown is not formed with select tag you cannot use this Select Class methods in selenium**

**We will see how to handle such kind of dropdowns in selenium on**[Custom Dropdown in Selenium webdriver](https://chercher.tech/java/dropdown-select-class-in-selenium-webdriver#custom)**tutorial**

**Dropdown Details in selenium webdriver:**

* *Index starts from 0 in dropdowns*
* *Value is nothing but 'value' attributes value*
* *Value could be different from Visible text(Note : In below dropdown image, value is Apple but the Visible Text is 'iPhone' )*
* *Visible text is nothing but text between****'>'****and****'<'***

**Code to select option by using index, value and visibleText**

Select sel=**new** Select(driver.findElement(By.*xpath*("//form[@name='callForm']//select[@name='flag']")));

sel.selectByIndex(0);

sel.selectByValue("");

sel.selectByVisibleText("");

**Custom dropdowns in selenium**

Dropdowns are formed using td, button, span, option, submenu tags, so in those cases, we cannot use our Select class from the selenium webdriver

**// find the dropdown using xpath**

**WebElement dropdownElement = driver.findElement(By.xpath("//button[@id='custom']"));**

**// click the dropdown element**

**dropdownElement.click();**

**Thread.sleep(3000);**

**// find and click the dropdown element**

**driver.findElement(By.xpath("//b[contains(text(),'submenu')]")).click();**

How to check whether dropdown is Singl value or Multi value drop down

1. **isMultiple** - this method is a non-static method of **Select class** in selenium.  
   **isMultiple** method returns True if the dropdown is multi-value dropdown otherwise it returns false for single value dropdown
2. **WebElement dropdownElement = driver.findElement(By.xpath("//select[@id='second']"));**
3. **// create object for Select class**
4. **Select dropdown = new Select(dropdownElement);**
5. **// check single or multi dropdown**
6. **boolean singleOrMultiple = dropdown.isMultiple();**
7. **if (singleOrMultiple) {**
8. **System.out.println("Dropdown is Multi value dropdown");**
9. **}else{**
10. **System.out.println("Dropdown is Single value dropdown");**
11. **}**

**2. getAttribute("multiple")** method return true if the dropdown is multiple and returns false in case of single value dropdown

**// find the dropdown using xpath**

**WebElement dropdownElement = driver.findElement(By.xpath("//select[@id='first']"));**

**// create object for Select class**

**String singleOrMulti = dropdownElement.getAttribute("multiple");**

**// compare the result**

**if (singleOrMulti != null) {**

**System.out.println("Dropdown is Multi value dropdown");**

**}else{**

**System.out.println("Dropdown is Single value dropdown");**

**}**

**Check the value present in dropdown or not?**

**WebElement dropdownElement = driver.findElement(By.xpath("//select[@id='animals']"));**

**// create object for Select class**

**Select dropdown = new Select(dropdownElement);**

**// get all the options and store it in list**

**List allElements = dropdown.getOptions();**

**System.out.println("Values present in Single Value Dropdown");**

**for (WebElement element : allElements) {**

**// iterate over each element and print the text**

**String dropdownOptionValue = element.getText();**

**if (dropdownOptionValue.equals("Avatar")) {**

**System.out.println("Avatar Option is present though it is not a cat family :)");**

**}**

**}**

We can select the dropdown values in few ways:

* *Select class methods ( selectByIndex, selectByValue, selectByVisibleText )s*
* *Click method*
* [*Actions class*](https://chercher.tech/java/mouse-keyboard-actions-class-selenium-webdriver)*click methods*
* [*Find element*](https://chercher.tech/java/find-elements-in-selenium)*method*
* *Sendkeys method*
* *Javascript method*

Method 1, 4, 5 are not applicable for custom dropdowns.

**By using click method**

**driver.get("https://chercher.tech/java/practice-dropdowns");**

**// find the dropdown using xpath**

**WebElement dropdownElement = driver.findElement(By.xpath("//select[@id='first']"));**

**// click the dropdown element**

**dropdownElement.click();**

**Thread.sleep(3000);**

**// find and click the dropdown element**

**driver.findElement(By.xpath("//option[text()='Bing']")).click();**

**By using Action class**

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

**// click the dropdown**

**act.click(driver.findElement(By.xpath("//select[@id='first']")))**

**// click the option**

**.click(driver.findElement(By.xpath("//option[text()='Bing']")))**

**// perform the operation**

**.build().perform();**

**By using FindElement Method**

**// open webpage**

**driver.get("https://chercher.tech/java/practice-dropdowns");**

**// find the option and click it..ta..da**

**driver.findElement(By.xpath("//option[text()='Bing']")).click();**

**BY using sendKeys method**

**driver.findElement(By.xpath("//select[@id='first']")).sendKeys("Bing");**

**By using JavaScript Executor**

**WebElement dropdownElement = driver.findElement(By.xpath("//button[@id='custom']"));**

**//cast driver object to JavaScriptExecutor**

**JavascriptExecutor jse = (JavascriptExecutor) driver;**

**// set the dropdown value t0 'Bing' using javascript**

**jse.executeScript("arguments[0].value='Bing'", dropdownElement);**

**Different ways to check whether the Option is selected or not in Dropdown with Selenium Webdriver**

We can verify whether an option is selected or not in a dropdown using selenium, and there are different ways to do it.

* *isSelect() method*
* *Verify with****getFirstSelectedOption()***
* *Use****getAttribute()****method*
* *Use****Java script****Method*
* **WebElement dropdownElement = driver.findElement(By.xpath("//select[@id='first']//option[text()='Google']"));**
* **// returns true if selected ,false if not selected**
* **boolean selectedOrNot = dropdownElement.isSelected();**
* **// print the text**
* **System.out.println("is Google option selected : "+selectedOrNot);**

**WebElement dropdownElement = driver.findElement(By.xpath("//select[@id='first']"));**

**// create object for Select class**

**Select dropdown = new Select(dropdownElement);**

**// select 1st selected option from the dropdown options**

**WebElement selectedOption = dropdown.getFirstSelectedOption();**

**// get the text from the option**

**String selectedOptionText = selectedOption.getText();**

**// print the text**

**System.out.println("Selected option from the dropdown is : "+selectedOptionText);**

**WebElement dropdownElement = driver.findElement(By.xpath("//select[@id='first']"));**

**// returns the value of selected options**

**String selectedOrNot = dropdownElement.getAttribute("value");**

**System.out.println("Selected Option Values is : "+selectedOrNot);**