**Select** is a class provided by Selenium that is used to work with a dropdown element.

Syntax:

Select select = new Select(WebElement element);

If select Tag is not used, **[UnexpectedTagNameException](https://selenium.dev/selenium/docs/api/java/org/openqa/selenium/support/ui/UnexpectedTagNameException.html)** will throw

You can access all the methods provided by the SELECT class by just typing select + dot.

Methods of Select Class

**selectByValue**

selectByValue(String value)

WebElement element = driver.findElement(By.xpath("//select[@name='country']"));

Select select = new Select(element);

select.selectByValue("AR"); // Argentina will get selected.

### selectByIndex

selectByIndex(int index) #Note: Index ranges from 0 to n-1

WebElement element = driver.findElement(By.xpath("//select[@name='country']"));

Select select = new Select(element);

select.selectByIndex(3); // Algeria will get selected.

### selectByVisibleText

selectByVisibleText(String text)

WebElement element = driver.findElement(By.xpath("//select[@name='country']"));

Select select = new Select(element);

select.selectByVisibleText("Albania"); // Albania will get selected.

### deselectAll

deselectAll() method clears all selected entries. It is only valid when the drop-down element list supports multiple selections.

It neither accept anything as a parameter nor returns anything.

Syntax:

Select.deSelectAll();

### deselectByIndex

The deselectByIndex() method deselects or clears an option at the specified index.

Syntax:

select.deSelectByIndex(int index);

### deselectByValue

This method is used for deselecting all the options whose “value” attribute matches the specified argument.

Syntax:

select.deSelectByValue(String value);

### deselectByVisibleText

This method is used to deselect all options that display text matching the parameter.

Syntax:

select.deSelectByVisibleText(String text);

### getFirstSelectedOption

This method is used to get the first selected option in the list.

Syntax:

select.getFirstSelectedOption();

It doesn’t accept anything as parameter,but returns first selected web element from the dropdown list

WebElement element = driver.findElement(By.xpath("//select[@name='country']"));

Select select = new Select(element);

select.getFirstSelectedOption().getText();

System.out.println(select.getFirstSelectedOption().getText());

### getAllSelectedOptions

This method is used to get all selected options in the dropdown list.

Syntax:

select.getAllSelectedOptions();

It does not accept any parameter but returns List of webelements(List<WebElement>)

### isMultiple

This method is used to check that the list supports multiple options at the same time or not.

Syntax:

isMultiple();

It accepts no parameter but returns Boolean. If the list supports multiple options, it will return true otherwise false returns.

getOptions() ->to get all the options of the drop down list.

size() 🡪 to get the size of the list.