



# Handling Dynamic Dropdown Using Selenium Java

Swipe →

## What is a drop-down element

A drop-down element is a form element that presents a list of options to users as a drop-down menu.

Users can select single/multiple option(s) from the menu, the selected option(s) is/are then displayed in the drop-down field.

# What are multi-select drop-down elements

With multi-select dropdown elements, users can select 'multiple' values from the field.

Multi Select List Demo

By clicking on the buttons, you can get value from the list which will display just below the buttons

California

Florida

New Jersey

New York

Ohio

Texas

Pennsylvania

Washington

First Selected

Get All Selected

First selected option is :

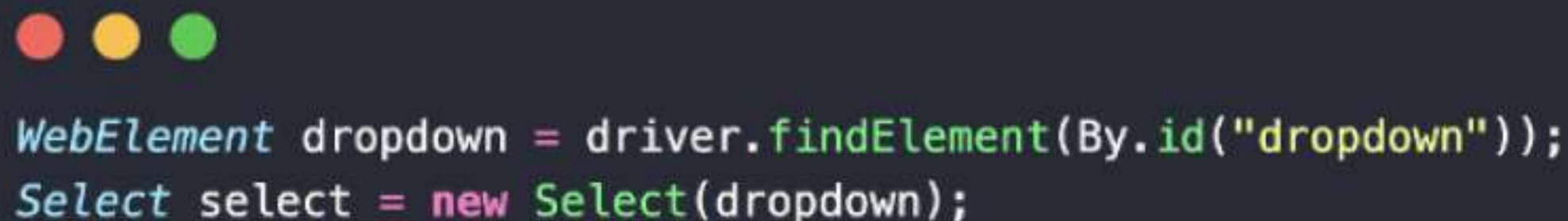
Options selected are :



## Select Class in Selenium WebDriver

*Select* class of the *org.openqa.selenium.support.ui* package lets you handle interactions with dropdown elements on a web page.

### Syntax



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

## **(1/2) Methods of Select class for performing actions on drop-down elements**

- isMultiple()
- selectByValue(String value)
- selectByIndex(int index)
- selectByVisibleText(String text)
- getOptions()
- getAllSelectedOptions()

## **(2/2) Methods of Select class for performing actions on drop-down elements**

- `getFirstSelectedOption()`
- `deselectAll()`
- `deselectByValue(String value)`
- `deselectByIndex(int index)`
- `deselectByVisibleText(String text)`

## Selecting values from dynamic drop-down using 'selectByVisibleText'

```
private Select multiSelectDropdownList ()
{
    WebElement multiSelectDropdown = getDriver ().findElement (By.id ("multi-select"));
    return new Select (multiSelectDropdown);
}

public void selectMultipleOptionByVisibleText (String[] text)
{
    for (int i = 0; i < text.length; i++)
    {
        multiSelectDropdownList ().selectByVisibleText (text[i]);
    }
}
```



## Selecting values from dynamic drop-down using 'selectByIndex'

```
private Select multiSelectDropdownList ()
{
    WebElement multiSelectDropdown = getDriver ().findElement (By.id ("multi-select"));
    return new Select (multiSelectDropdown);
}

public void selectMultipleOptionByIndex (int[] index)
{
    for (int i = 0; i < index.length; i++)
    {
        multiSelectDropdownList ().selectByIndex (index[i]);
    }
}
```



## Selecting values from dynamic drop-down using 'selectByValue'

```
private Select multiSelectDropdownList ()
{
    WebElement multiSelectDropdown = getDriver ().findElement (By.id ("multi-select"));
    return new Select (multiSelectDropdown);
}

public void multiSelectByValues (String[] values)
{
    for (int i = 0; i < values.length; i++)
    {
        multiSelectDropdownList ().selectByValue (values[i]);
    }
}
```

## De-selecting values from dynamic drop-down using 'deselectByVisibleText'

```
private Select multiSelectDropdownList ()
{
    WebElement multiSelectDropdown = getDriver ().findElement (By.id ("multi-select"));
    return new Select (multiSelectDropdown);
}

public void deselectByVisibleText (String text)
{
    multiSelectDropdownList ().deselectByVisibleText (text);
}
```



## Returning all the option elements using 'getAllSelectedOptions'

```
private Select multiSelectDropdownList ()
{
    WebElement multiSelectDropdown = getDriver ().findElement (By.id ("multi-select"));
    return new Select (multiSelectDropdown);
}

public List<String> getAllSelectedOptions ()
{
    final List<WebElement> allOptions = multiSelectDropdownList ().getAllSelectedOptions ();
    final int size = allOptions.size ();
    ArrayList<String> options = new ArrayList<> ();
    for (int i = 0; i < size; i++)
    {
        options.add (allOptions.get (i)
            .getText ());
    }
    return options;
}
```



**Follow**



**Blog**



**[lambdatest.com/blog/](https://lambdatest.com/blog/)**