Selenium Dropdowns Select Class

In **Selenium WebDriver**, interacting with dropdowns is commonly done using the **`Select`** class, which provides methods to select or deselect options in **`<select>`** HTML tags.

🔷 What is a Dropdown?

In HTML:

```
html

<select id="dropdown">
  <option value="1">Option 1</option>
  <option value="2">Option 2</option>
</select>
```

Selenium provides the **`Select`** class to handle such dropdowns.

🔧 Importing `Select` Class in Java

```
java

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

🧪 Example Code: Handling Dropdowns

✅ Full Example with Explanation

```
java

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 DropdownExample {
    public static void main(String[] args) {
        // Set path to ChromeDriver
        System.setProperty("webdriver.chrome.driver", "path_to_chromedriver");

        // Launch browser
        WebDriver driver = new ChromeDriver();

        // Navigate to webpage with dropdown
        driver.get("https://example.com/dropdown");

        // Locate the dropdown element
        WebElement dropdownElement = driver.findElement(By.id("dropdown"));

        // Create Select object
        Select dropdown = new Select(dropdownElement);

        // Select by visible text
        dropdown.selectByVisibleText("Option 1");

        // Select by index (starting from 0)
        dropdown.selectByIndex(1);

        // Select by value (matches value="2" from option tag)
        dropdown.selectByValue("2");

        // Get selected option
        WebElement selectedOption = dropdown.getFirstSelectedOption();
        System.out.println("Selected option: " + selectedOption.getText());

        // Get all options
        for (WebElement option : dropdown.getOptions()) {
            System.out.println("Available option: " + option.getText());
        }

        // Close browser
        driver.quit();
    }
}
```

🔹 Common Methods in `Select` Class

Method	Description
`selectByVisibleText(String text)`	Selects dropdown option by the text visible to the user
`selectByIndex(int index)`	Selects dropdown option by index (starting from 0)
`selectByValue(String value)`	Selects option by matching the **`value`** attribute
`getFirstSelectedOption()`	Returns the currently selected option
`getOptions()`	Returns a list of all options in the dropdown
`deselectAll()`	Deselects all selected options (only for multi-select dropdowns)
`isMultiple()`	Checks if dropdown supports multiple selections

✅ Handling Multi-Select Dropdowns

If the dropdown allows selecting **multiple options** (using **`multiple`** attribute), use:

```java
if (dropdown.isMultiple()) {
    dropdown.selectByIndex(0);
    dropdown.selectByIndex(2);
    dropdown.deselectByIndex(0);
    dropdown.deselectAll();
}
```

⚠️ Notes

- **`Select`** class only works with **`<select>`** tags. For custom dropdowns (div-based), you need to use standard click and XPath logic.
- Always use **`getFirstSelectedOption()`** to verify the selected value.
- Always check with **`isMultiple()`** before calling **`deselect`** methods.

Would you like an example for **custom dropdowns (not using `<select>`)** as well?