Drop-Down’s

A webpage consists of multiple elements such as text fields, buttons, menus, drop-down options. As every tester is aware, [Selenium](https://www.browserstack.com/selenium) is the most widely used tool for end-to-end [automation testing](https://www.browserstack.com/guide/automation-testing-tutorial) of a website. This article discusses how Selenium is used to find web elements in a drop-down menu on a website.

The article covers the following topics:

* Introduction to Select Class in Selenium
* Different Select Methods
* Selecting Multiple items with select command
* Demo: Illustrating the use of Select class in Selenium

**Introduction to Select Class in Selenium**

In Selenium, the Select class provides the implementation of the HTML SELECT tag. A Select tag provides the helper methods with select and deselect options. As Select is an ordinary class, its object is created by the keyword New and also specifies the location of the web element.

**Syntax:**

Select objSelect = new Select();

In the syntax above, it clearly states that Select is asking for an element type object for its constructor, i.e it will create an object of the select class.

**Different Select Methods**

The following are the most commonly used methods to deal with a drop-down list:

1. **selectByVisibleText**: *select\_by\_visible\_text (String arg0): void*

This method is used to select one of the options in a drop-down box or an option among multiple selection boxes. It takes a parameter of String which is one of the values of Select element and it returns nothing.

**Syntax:**

obj.Select. select\_by\_visible\_text (“text”);

**Example:**

objSelect =Select(driver.findElement(By.id("search-box")));

objSelect. select\_by\_visible\_text ("Automation");

**2. selectByIndex: *select\_by\_index (int arg0) : void***

This method is similar to ‘*selectByVisibleText*’, but the difference here is that the user has to provide the index number for the option rather than text. It takes the integer parameter which is the index value of Select element and it returns nothing.

**Syntax:**

oSelect. select\_by\_index (int);

**Example:**

objSelect = Select(driver.findElement(By.id("Seacrch-box")));

Select. select\_by\_index (4);

**3. selectByValue: *selectByValue(String arg0) : void***

This method asks for the value of the desired option rather than the option text or an index. It takes a String parameter which is one of the values of Select element and it does not return anything.

**Syntax:**

oSelect.select\_by\_value(“text”);

**Example:**

objSelect = Select(driver.findElement(By.id("Search-box")));

objSelect.select\_by\_value("Automation Testing");

**4. getOptions: *getOptions( ) : List<WebElement>***

This method gets all the options belonging to the Select tag. It takes no parameter and returns List<WebElements>.

**Syntax:**

oSelect.getOptions();

**Example:**

objSelect = Select(driver.findElement(By.id("Search-box")));

elementCount = oSelect.getOptions();

print(len(elementCount))

**5. deselectAll()**

This method clears all the selected entries. This is only valid when the drop-down element supports multiple selections.

**Syntax:**

objSelect.deselectAll();