29. Handling Dropdowns

Dropdown

- Dropdown is a different web element not like a normal element which contains number of options and we will select an option from it.
- Dropdown tagname is **select**. Again select contains different options each option is a webelement.
- In Selenium Webdriver dropdowns are handled using the **Select class**. This class provides several actions that you can perform on dropdown elements.

select class methods

1. **select_by_visible_text**
    a. Selects an option from a dropdown by matching the visible text of the option.
2. **select_by_value**
    a. Selects an option from the dropdown using the `value` attribute of the `<option>` tag.
3. **select_by_index**
    a. Selects an option from the dropdown based on its index (starting from 0).
4. **options**
    a. Returns a list of all options available in the dropdown.
5. **first_selected_option**
    a. Returns the first selected option in a single-select dropdown.
6. **deselect_all (for Multi-Select)**
    a. Deselects all selected options in a multi-select dropdown.
7. **deselect_by_visible_text (for Multi-Select)**
    a. Deselects an option in a multi-select dropdown by using the visible text of the option.
8. **deselect_by_value (for Multi-Select)**
    a. Deselects an option in a multi-select dropdown using the `value` attribute of the `<option>`.
9. **deselect_by_index (for Multi-Select)**
    a. Deselects an option in a multi-select dropdown based on its index.
10. **all_selected_options (for Multi-Select)**
    a. Returns a list of all currently selected options in a multi-select dropdown.

Dropdowns

- Single-Select Dropdown
- Multi-Select Dropdown
- Bootstrap Dropdown

Single-Select Dropdown

➔ Single-Select Dropdown is a dropdown where you can select only **one option** at a time.

SingleSelectDropDownDemo.py

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.select import Select
opt = webdriver.ChromeOptions()
opt.add_argument("--start-maximized")
opt.add_experimental_option("detach",True)
```

```
driver = webdriver.Chrome(options=opt)
driver.implicitly_wait(10)
driver.get("https://testautomationpractice.blogspot.com/")
```

Locate the dropdown element

```
drop_down_element = driver.find_element(By.XPATH, "//select[@id='country']")
dropdown = Select(drop_down_element)
```

Select by visible text

```
dropdown.select_by_visible_text("France")
```

Select by value

```
dropdown.select_by_value("japan")
```

Select by index

```
dropdown.select_by_index(2)
```

Capture all the options

```
all_options = dropdown.options
print("Total number of countries:", len(all_options))
for option in all_options:
  print(option.text)
```

Get the first selected option

```
first_selected_option = dropdown.first_selected_option
print("First selected option is:", first_selected_option.text)
driver.quit()
```

Multi-Select Dropdown

- ➔ Multi-Select Dropdown is a dropdown where you can select multiple options at the same time.
- ➔ We can determine if a dropdown is multi-select by using the `**is_multiple()**` method, which returns a boolean value.

MultiSelectDropDownDemo.py

```
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.select import Select
opt = webdriver.ChromeOptions()
opt.add_argument("--start-maximized")
opt.add_experimental_option("detach",True)
driver = webdriver.Chrome(options=opt)
driver.implicitly_wait(10)
driver.get("https://testautomationpractice.blogspot.com/")
```

Locate the multi-select dropdown element

```
multi_select_element = driver.find_element(By.XPATH, "//select[@id='colors']")
multi_select = Select(multi_select_element)
```

Check if it's a multi-select dropdown

```
print("Is it multi-select dropdown?", multi_select.is_multiple)
```

Select by visible text

```
multi_select.select_by_visible_text("Green")
```

Select by value

```
multi_select.select_by_value("white")
```

Select by index

```
multi_select.select_by_index(1)  # Blue
```

Capture all the options

```
options = multi_select.options
print("Total number of options:", len(options))
for option in options:
  print(option.text)
```

Get the first selected option

```
first_option = multi_select.first_selected_option
print("First option selected is:", first_option.text)
time.sleep(5)
```

Additional methods available for multi-select dropdown

Deselect by visible text

```
multi_select.deselect_by_visible_text("Green")
```

Deselect by value

```
multi_select.deselect_by_value("white")
```

Deselect by index

```
multi_select.deselect_by_index(1)  # Blue
```

Deselect all the options

```
multi_select.deselect_all()
driver.quit()
```

Bootstrap Dropdown

- ➔ The bootstrap dropdown is an enhanced part of the dropdown where we will deal with UL, LI, DIV, SPAN, etc tags of HTML.
- ➔ Dropdown **without select tag** is Bootstrap Dropdown
- ➔ To handle this kind of drop-down we have to use the **find_elements method** and then we can run a for loop to get specific elements.
- ➔ Great for complex UIs and interactive designs.

BootstrapDropDownDemo.py

```
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
opt = webdriver.ChromeOptions()
opt.add_argument("--start-maximized")
opt.add_experimental_option("detach",True)
driver = webdriver.Chrome(options=opt)
driver.implicitly_wait(10)
```

```
driver.get("https://www.jquery-az.com/boots/demo.php?ex=63.0_2")
```

Open the dropdown

```
driver.find_element(By.XPATH, "//button[contains(@class,'multiselect')]").click()
```

Select a single item (Java)

```
driver.find_element(By.XPATH, "//input[@value='Java']").click()
```

Capture all the items from the dropdown

```
items = driver.find_elements(By.XPATH, "//ul[contains(@class,'multiselect')]//label")
print("Number of Options:", len(items))  # 14
print("Printing items from the dropdown....")
for item in items:
   print(item.text)
```

Select multiple items (Python and Oracle)

```
for item in items:
   item_text = item.text
   if item_text == "Python" or item_text == "Oracle":
      item.click()
driver.quit()
```