# Automation Testing

# Handling Various Web Elements

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```

# A Day in the Life of an Automation Test Engineer

Joel as an Automation Test Engineer wants to automate web application.

Now as an Automation Test Engineer, when he automate a web application, he need’s a tool that provides options to interact with many internal web element components, like textbox, radio buttons, dropdowns, tables, etc.

To know about it, let us go through the lesson.

# Learning Objectives

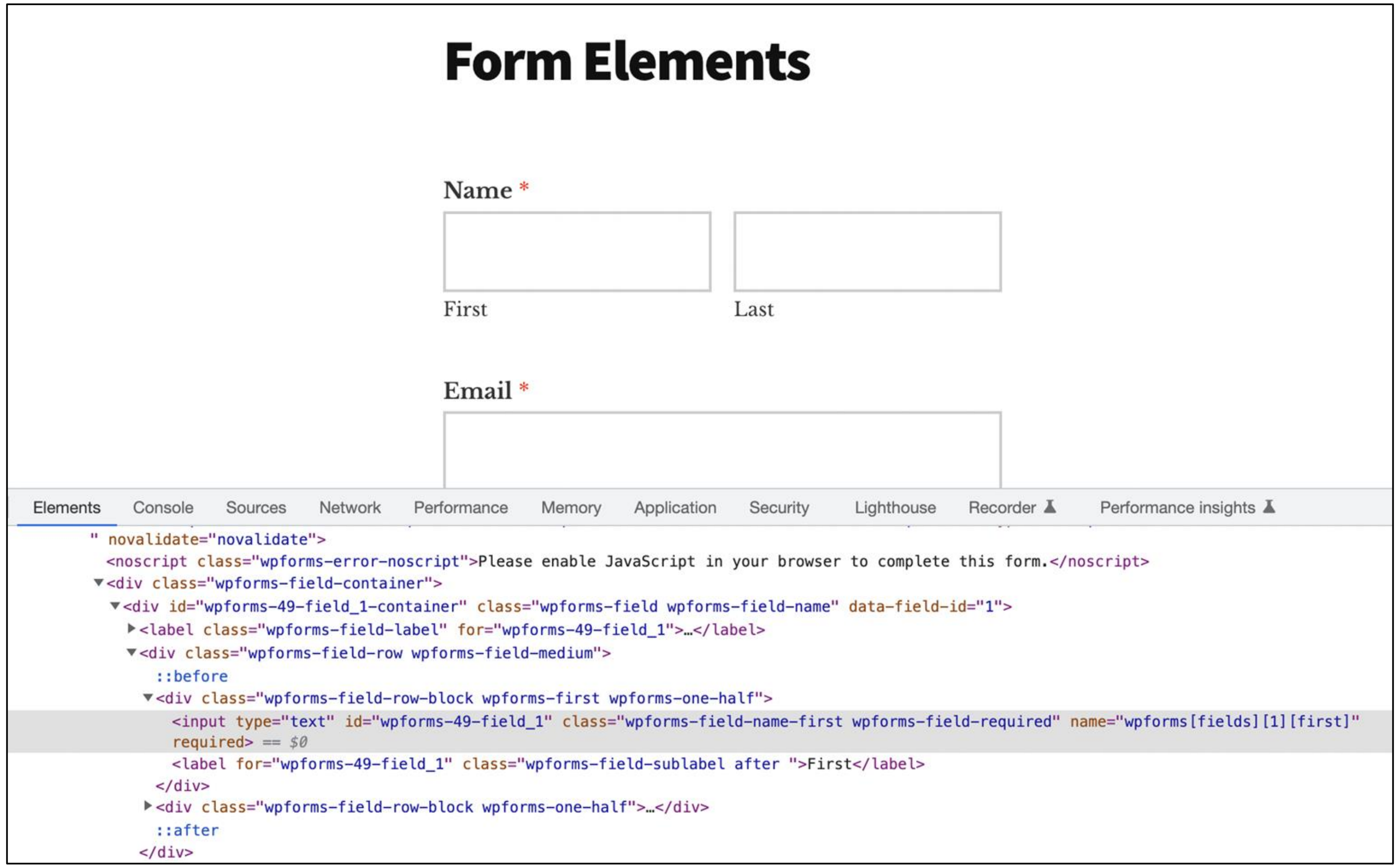
By the end of this lesson, you will be able to:

- Comprehend various WebElements
- Apply the radio button and checkbox WebElements
- Apply dropdown and multiple dropdown selectors

# Various Web Elements

# Text Field (Input)

- Input tag is used in HTML to create a text field or textbox.
- All the Identfiers, including the ID, Name, Classname, Tagname, CSS Selector, and XPath, can be used to identify a text box.



## Form Elements

Name *

First

Last

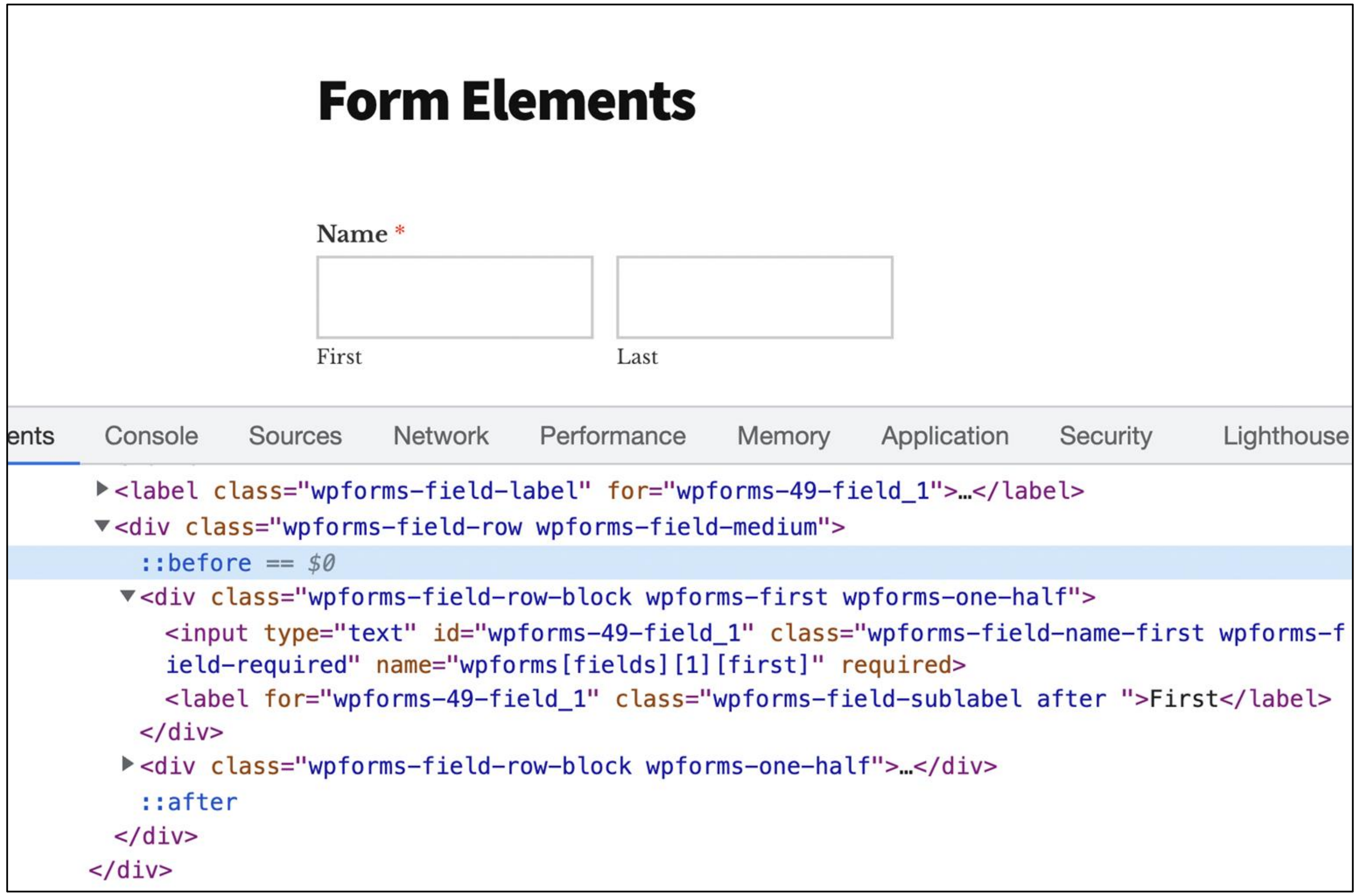
Email *

```
    " novalidate="novalidate">
     <noscript class="wpforms-error-noscript">Please enable JavaScript in your browser to complete this form.</noscript>
    ▼<div class="wpforms-field-container">
     ▼<div id="wpforms-49-field_1-container" class="wpforms-field wpforms-field-name" data-field-id="1">
       ▶<label class="wpforms-field-label" for="wpforms-49-field_1">…</label>
       ▼<div class="wpforms-field-row wpforms-field-medium">
          ::before
         ▼<div class="wpforms-field-row-block wpforms-first wpforms-one-half">
            <input type="text" id="wpforms-49-field_1" class="wpforms-field-name-first wpforms-field-required" name="wpforms[fields][1][first]"
            required> == $0
            <label for="wpforms-49-field_1" class="wpforms-field-sublabel after ">First</label>
          </div>
         ▶<div class="wpforms-field-row-block wpforms-one-half">…</div>
          ::after
        </div>
```

# Text Field (Input)

To interact with a textbox, the following methods can be followed:

- driver.findElement(By.Id(“wpforms-49-field_1”).sendKeys(“John”);
- driver.findElement(By.Xpath(//input[@type=“text”]).sendKeys(“John”);



## Form Elements

Name *

First

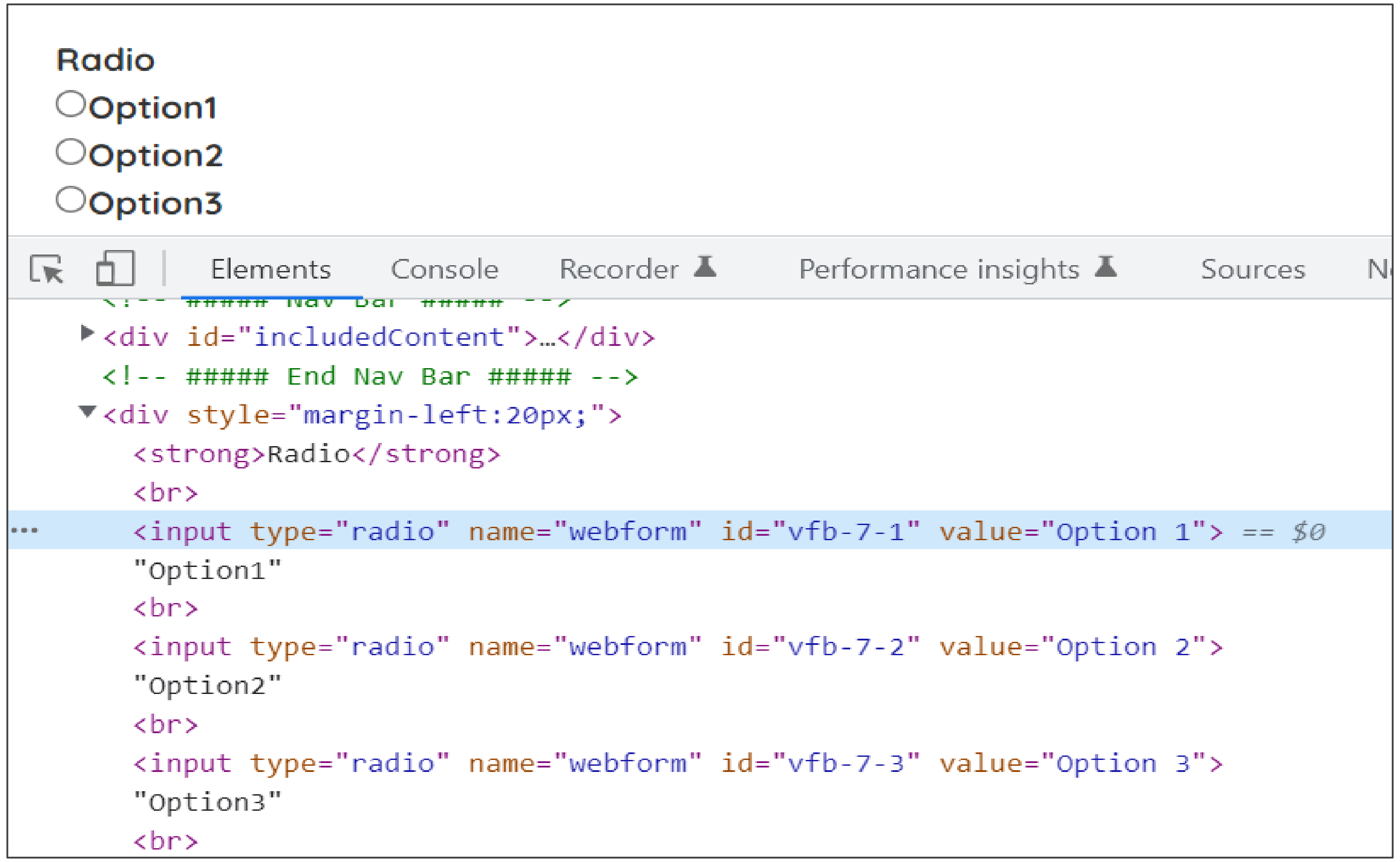
Last

```
▶<label class="wpforms-field-label" for="wpforms-49-field_1">…</label>
▼<div class="wpforms-field-row wpforms-field-medium">
   ::before == $0
  ▼<div class="wpforms-field-row-block wpforms-first wpforms-one-half">
     <input type="text" id="wpforms-49-field_1" class="wpforms-field-name-first wpforms-f
     ield-required" name="wpforms[fields][1][first]" required>
     <label for="wpforms-49-field_1" class="wpforms-field-sublabel after ">First</label>
   </div>
  ▶<div class="wpforms-field-row-block wpforms-one-half">…</div>
   ::after
 </div>
</div>
```

# Radio Button

# Radio Button

A radio button is an HTML element that allows the user to choose only one of the alternatives available.



**Radio**

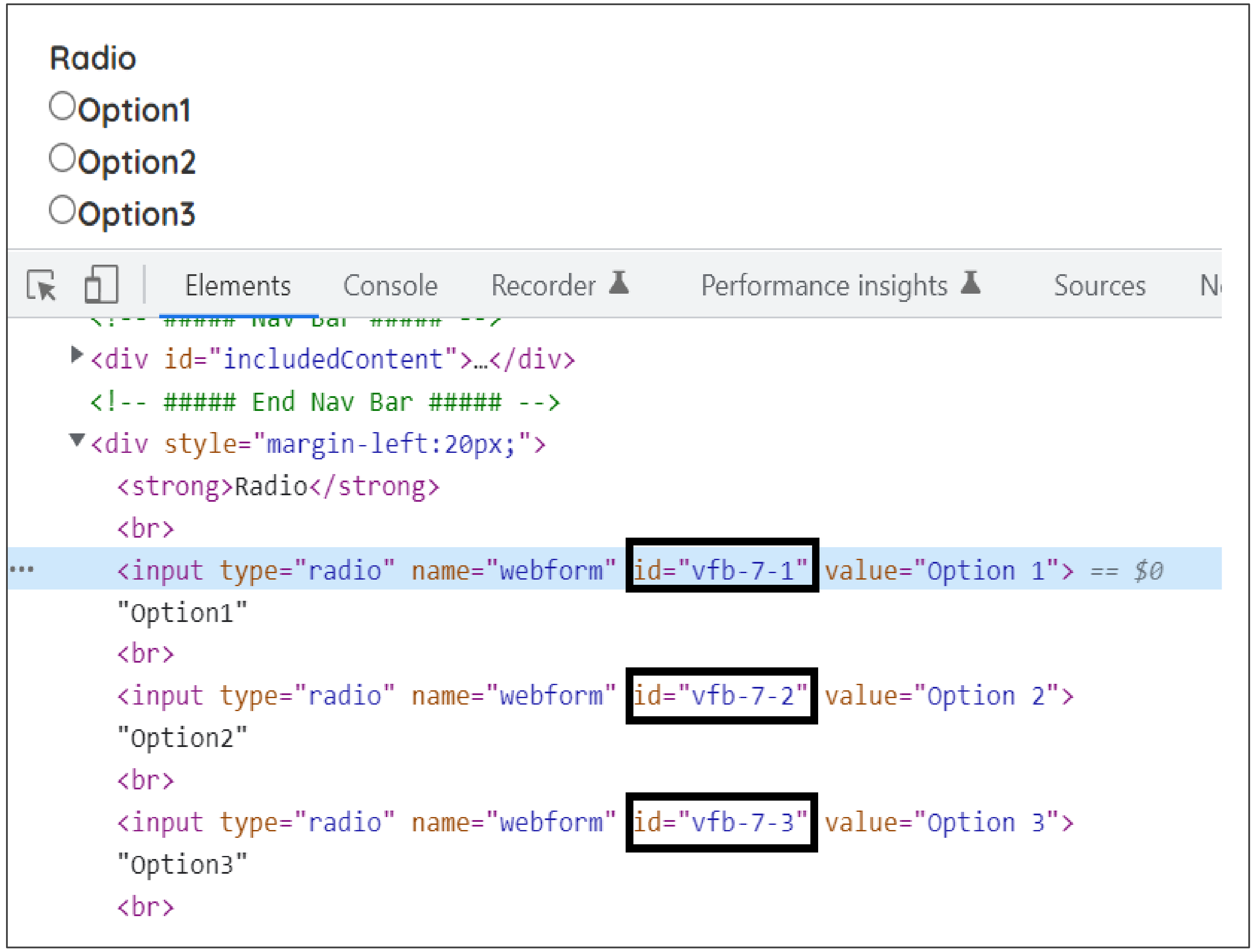
○Option1

○Option2

○Option3

```
  <!-- ##### Nav Bar ##### -->
 ▶<div id="includedContent">…</div>
  <!-- ##### End Nav Bar ##### -->
 ▼<div style="margin-left:20px;">
    <strong>Radio</strong>
    <br>
    <input type="radio" name="webform" id="vfb-7-1" value="Option 1"> == $0
    "Option1"
    <br>
    <input type="radio" name="webform" id="vfb-7-2" value="Option 2">
    "Option2"
    <br>
    <input type="radio" name="webform" id="vfb-7-3" value="Option 3">
    "Option3"
    <br>
```

# Identifying Radio Button



**Radio**

○Option1

○Option2

○Option3

```
 ▸<div id="includedContent">…</div>
  <!-- ##### End Nav Bar ##### -->
 ▾<div style="margin-left:20px;">
    <strong>Radio</strong>
    <br>
    <input type="radio" name="webform" id="vfb-7-1" value="Option 1"> == $0
    "Option1"
    <br>
    <input type="radio" name="webform" id="vfb-7-2" value="Option 2">
    "Option2"
    <br>
    <input type="radio" name="webform" id="vfb-7-3" value="Option 3">
    "Option3"
    <br>
```

Radio Buttons could be identified in different ways, like:

- Using ID
- Using CSS Selector
- Using XPath

# Identifying Radio Button

The driver.findElement() method is used with one of the locator methods, including the **By.id, By.xpath, or By.cssSelector**

```
• driver.findElement(By.id("vfb-7-1")).click();
• driver.findElement(By.xpath("//*[@type='radio'][2]")).click();
• driver.findElement(By.cssSelector("input[id='vfb-7-1']")).click();
```

# Radio Button Methods

Validation methods like isSelected(), isDispalyed(), and isEnabled() are used to verify the status of the radio button.

# Radio Button Methods

- **isSelected():** Checks whether a radio button is selected or not
- **isDisplayed()**: Checks whether a radio button is displayed on the web page or not
- **isEnabled()**: Checks whether a radio button is enabled or not

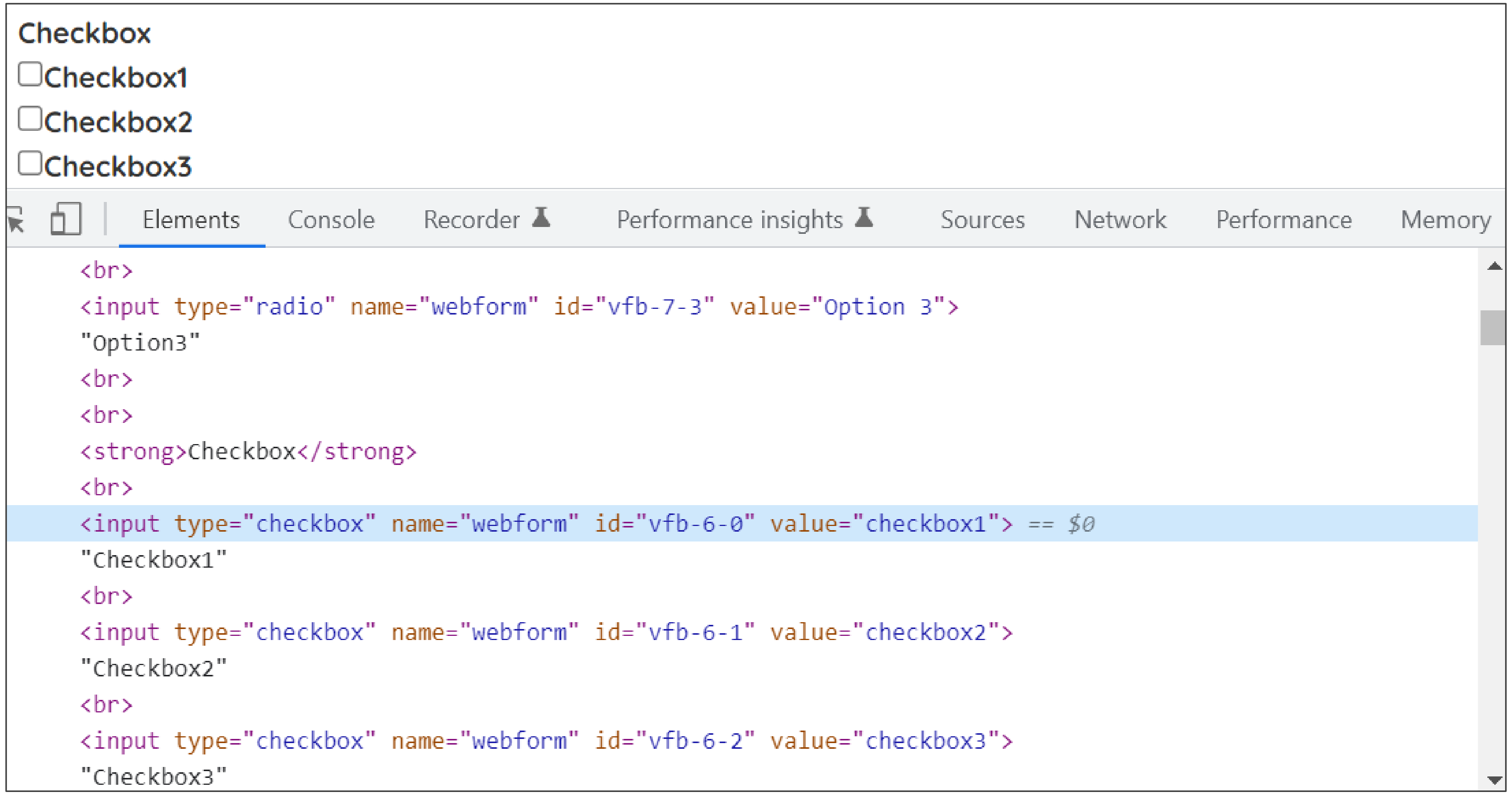
**Example:**

```
    driver.findElement(By.xpath("//*[@type='radio'][2]")).isEnabled()();
    driver.findElement(By.name("radio1")).isDisplayed();
    driver.findElement(By.cssSelector("input[id='vfb-7-1']")).isSelected();
```

# Checkbox

# Checkbox

The checkbox is an HTML element that allows the user to select many options from the list of values.



**Checkbox**

□Checkbox1

□Checkbox2

□Checkbox3

```
<br>
<input type="radio" name="webform" id="vfb-7-3" value="Option 3">
"Option3"
<br>
<br>
<strong>Checkbox</strong>
<br>
<input type="checkbox" name="webform" id="vfb-6-0" value="checkbox1"> == $0
"Checkbox1"
<br>
<input type="checkbox" name="webform" id="vfb-6-1" value="checkbox2">
"Checkbox2"
<br>
<input type="checkbox" name="webform" id="vfb-6-2" value="checkbox3">
"Checkbox3"
```

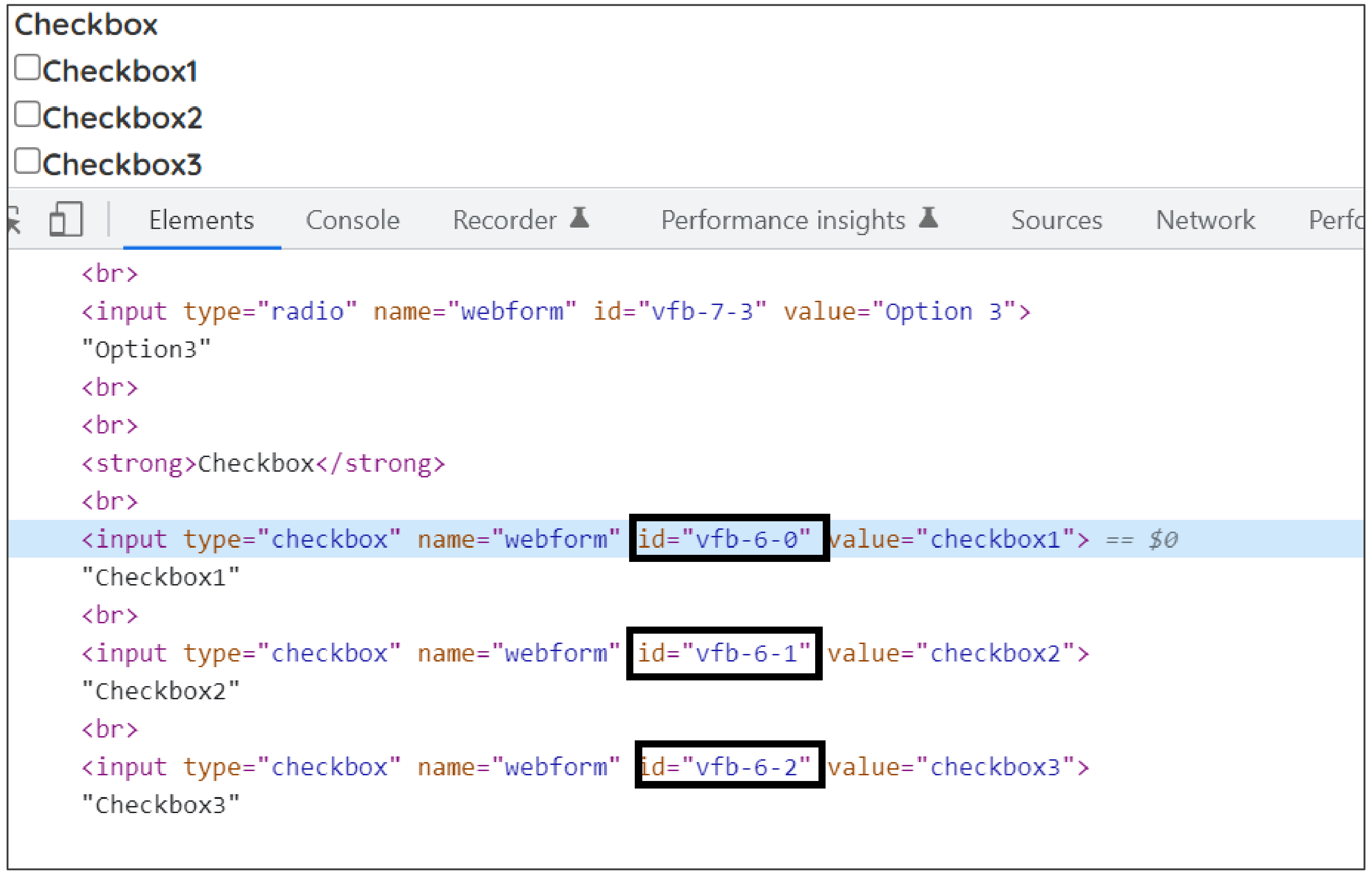
# Identifying Checkbox

Check Buttons could be identified in different ways :

- Using ID
- Using CSS Selector
- Using XPath

# Identifying Checkbox

- driver.findElement(By.id("vfb-6-0")).click();
- driver.findElement(By.xpath("//*[@type='checkbox'][2]")).click();
- driver.findElement(By.cssSelector("input[id='vfb-6-0']")).click();



**Checkbox**

□Checkbox1

□Checkbox2

□Checkbox3

```
<br>
<input type="radio" name="webform" id="vfb-7-3" value="Option 3">
"Option3"
<br>
<br>
<strong>Checkbox</strong>
<br>
<input type="checkbox" name="webform" id="vfb-6-0" value="checkbox1"> == $0
"Checkbox1"
<br>
<input type="checkbox" name="webform" id="vfb-6-1" value="checkbox2">
"Checkbox2"
<br>
<input type="checkbox" name="webform" id="vfb-6-2" value="checkbox3">
"Checkbox3"
```

# Selecting Values in a Checkbox

- **isSelected():** Checks whether a checkbox is selected or not
- **isDisplayed():** Checks whether a checkbox displays on the web page or not
- **isEnabled():** Checks whether a checkbox is enabled or not

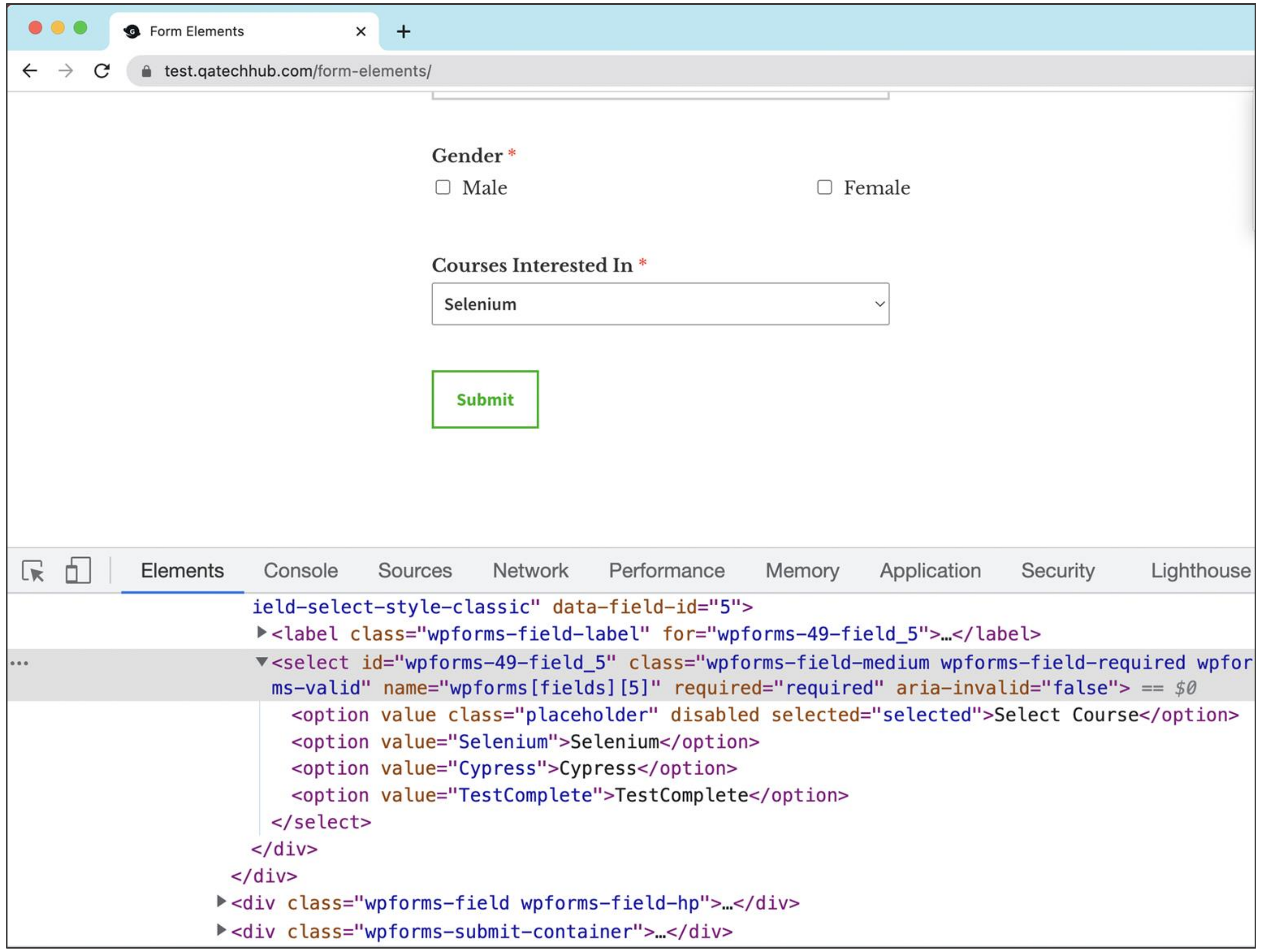
**Example:**

- driver.findElement(By.xpath("//*[@type='checkbox'][2]")).isEnabled();
- driver.findElement(By.name("checkbox1")).isDisplayed();
- driver.findElement(By.cssSelector("input[id='vfb-6-0']")).isSelected();

# DropDown List

# DropDown

Select tag is used in HTML to create a dropdown on a webpage. The values of these tags come under the options tag.



Gender *

☐ Male ☐ Female

Courses Interested In *

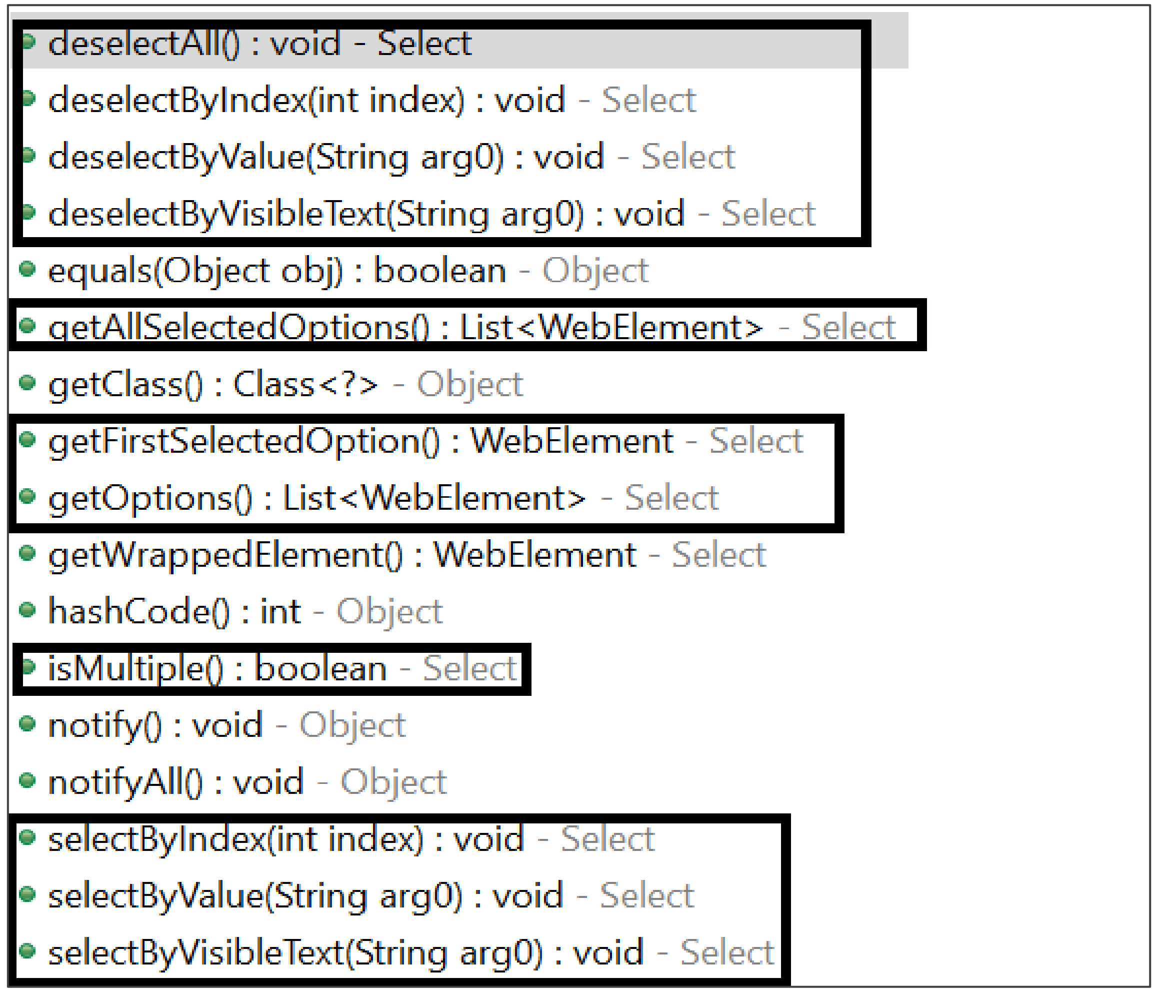
Selenium

Submit

```
ield-select-style-classic" data-field-id="5">
 ▶<label class="wpforms-field-label" for="wpforms-49-field_5">…</label>
 ▼<select id="wpforms-49-field_5" class="wpforms-field-medium wpforms-field-required wpfor
  ms-valid" name="wpforms[fields][5]" required="required" aria-invalid="false"> == $0
    <option value class="placeholder" disabled selected="selected">Select Course</option>
    <option value="Selenium">Selenium</option>
    <option value="Cypress">Cypress</option>
    <option value="TestComplete">TestComplete</option>
  </select>
</div>
</div>
▶<div class="wpforms-field wpforms-field-hp">…</div>
▶<div class="wpforms-submit-container">…</div>
```

# Working With Select Class

To select a value from the dropdowns, Selenium provides a **Select class** and supports the following methods:

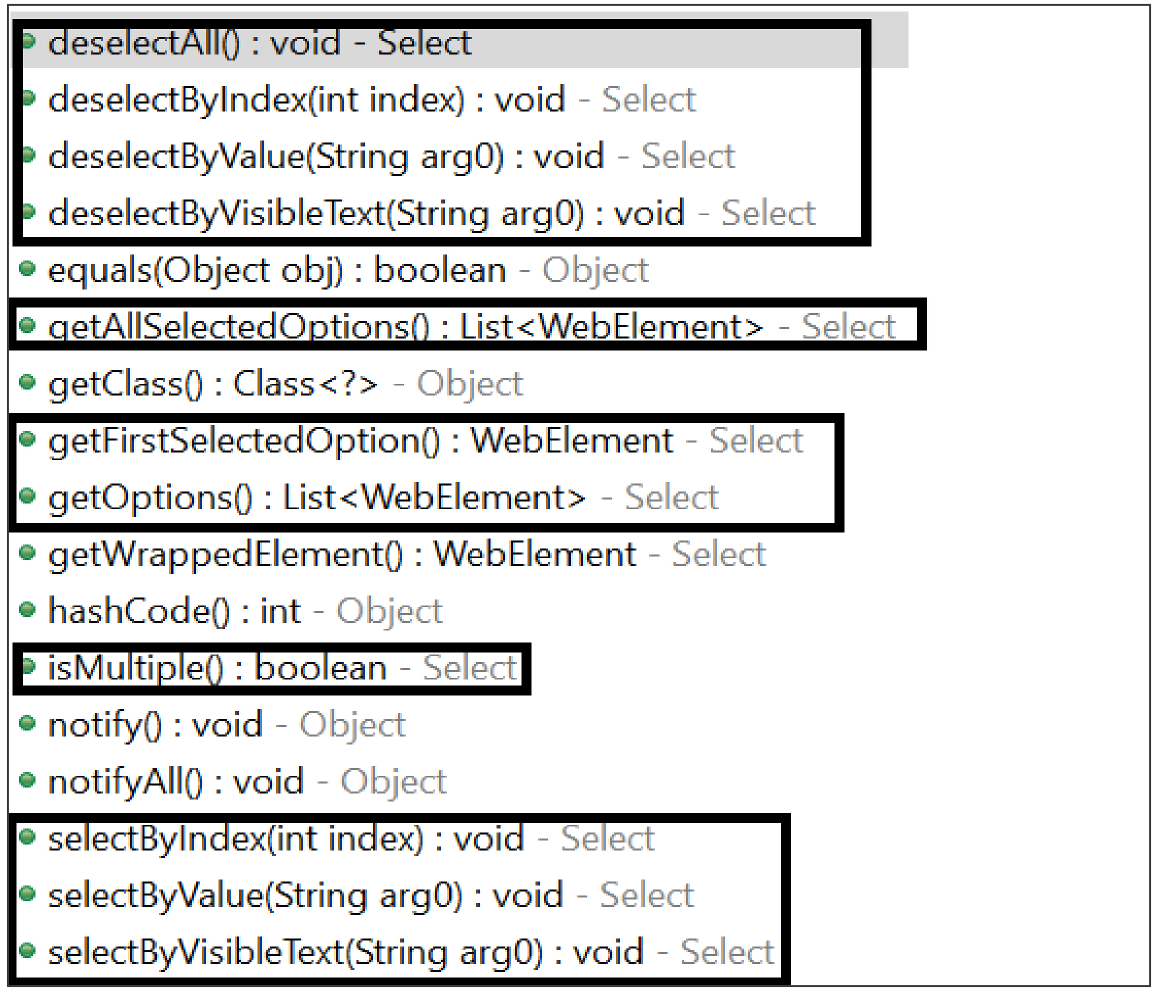


```
deselectAll() : void - Select
deselectByIndex(int index) : void - Select
deselectByValue(String arg0) : void - Select
deselectByVisibleText(String arg0) : void - Select
equals(Object obj) : boolean - Object
getAllSelectedOptions() : List<WebElement> - Select
getClass() : Class<?> - Object
getFirstSelectedOption() : WebElement - Select
getOptions() : List<WebElement> - Select
getWrappedElement() : WebElement - Select
hashCode() : int - Object
isMultiple() : boolean - Select
notify() : void - Object
notifyAll() : void - Object
selectByIndex(int index) : void - Select
selectByValue(String arg0) : void - Select
selectByVisibleText(String arg0) : void - Select
```

The three Select Methods are:

- Select via Index
- Select via visible text
- Select via value

# Working With Select Class



```
deselectAll() : void - Select
deselectByIndex(int index) : void - Select
deselectByValue(String arg0) : void - Select
deselectByVisibleText(String arg0) : void - Select
equals(Object obj) : boolean - Object
getAllSelectedOptions() : List<WebElement> - Select
getClass() : Class<?> - Object
getFirstSelectedOption() : WebElement - Select
getOptions() : List<WebElement> - Select
getWrappedElement() : WebElement - Select
hashCode() : int - Object
isMultiple() : boolean - Select
notify() : void - Object
notifyAll() : void - Object
selectByIndex(int index) : void - Select
selectByValue(String arg0) : void - Select
selectByVisibleText(String arg0) : void - Select
```

- The two Get methods are:
  1.Get Options
  2.Get First Selected Options

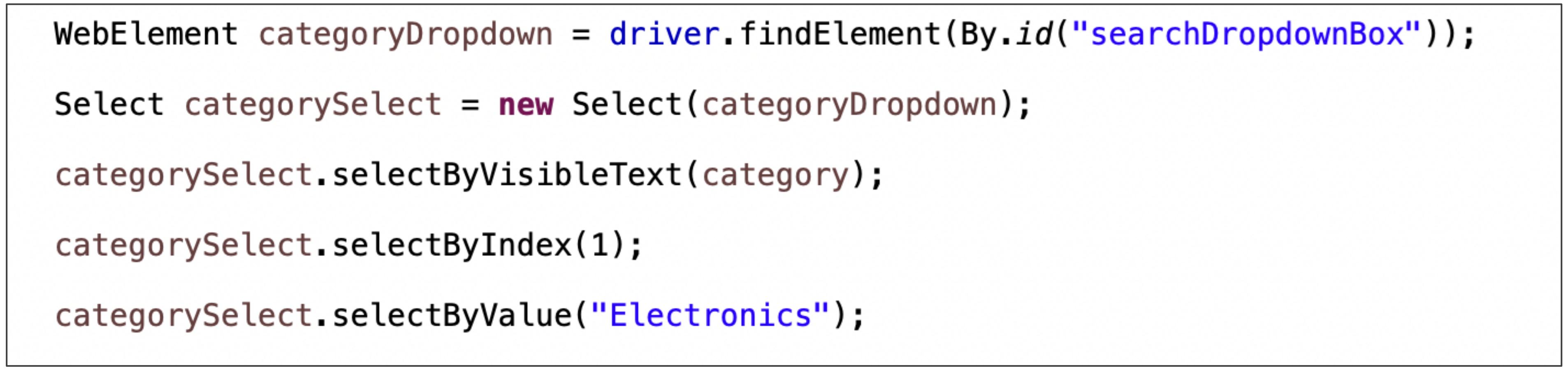
- There is an isMultiple() method that verifies if a dropdown allows multiple selections or not.

- The Deselect Methods are:
  1. Deselect via index
  2. Deselect via Visible text
  3. Deselect via value
  4. Deselect all

# Working With DropDown Methods

The three Select methods are:

- selectByIndex
- selectByValue
- selectByVisibleText



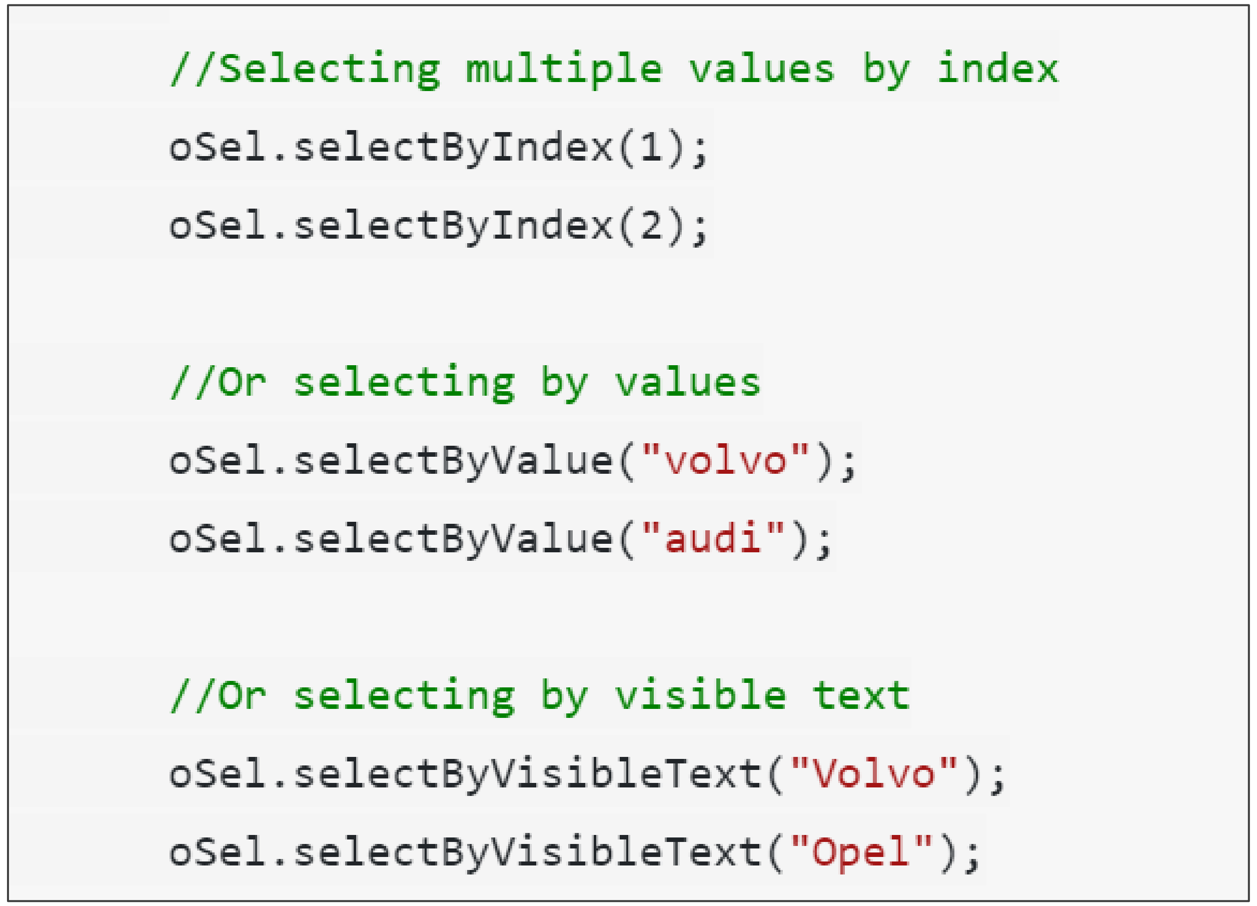
```
WebElement categoryDropdown = driver.findElement(By.id("searchDropdownBox"));

Select categorySelect = new Select(categoryDropdown);

categorySelect.selectByVisibleText(category);

categorySelect.selectByIndex(1);

categorySelect.selectByValue("Electronics");
```

FULL STACK

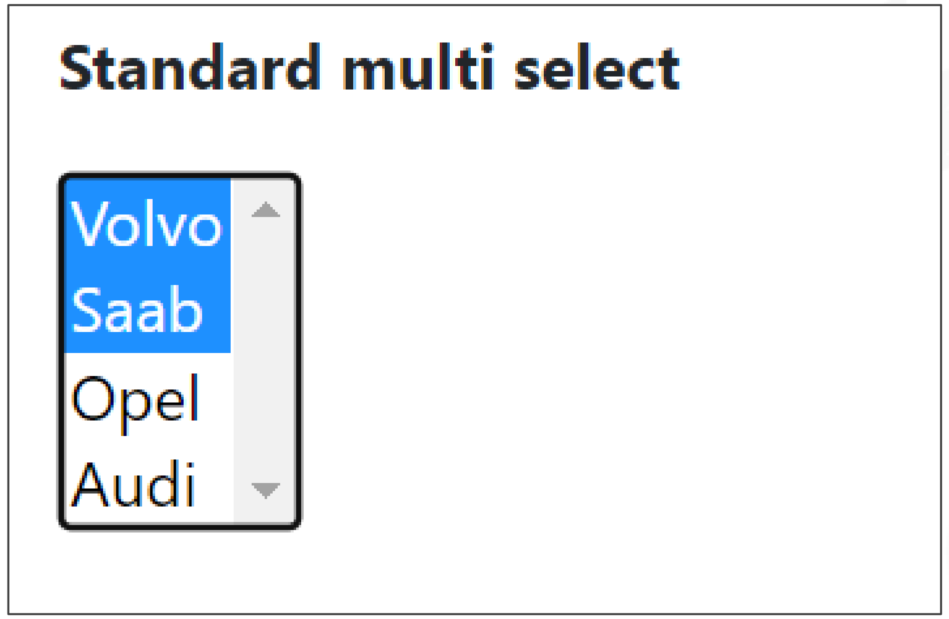
# Multiple Select From Dropdown

# Working With Multiselect DropDown

Multiselect dropdown allows selecting multiple values from a dropdown.

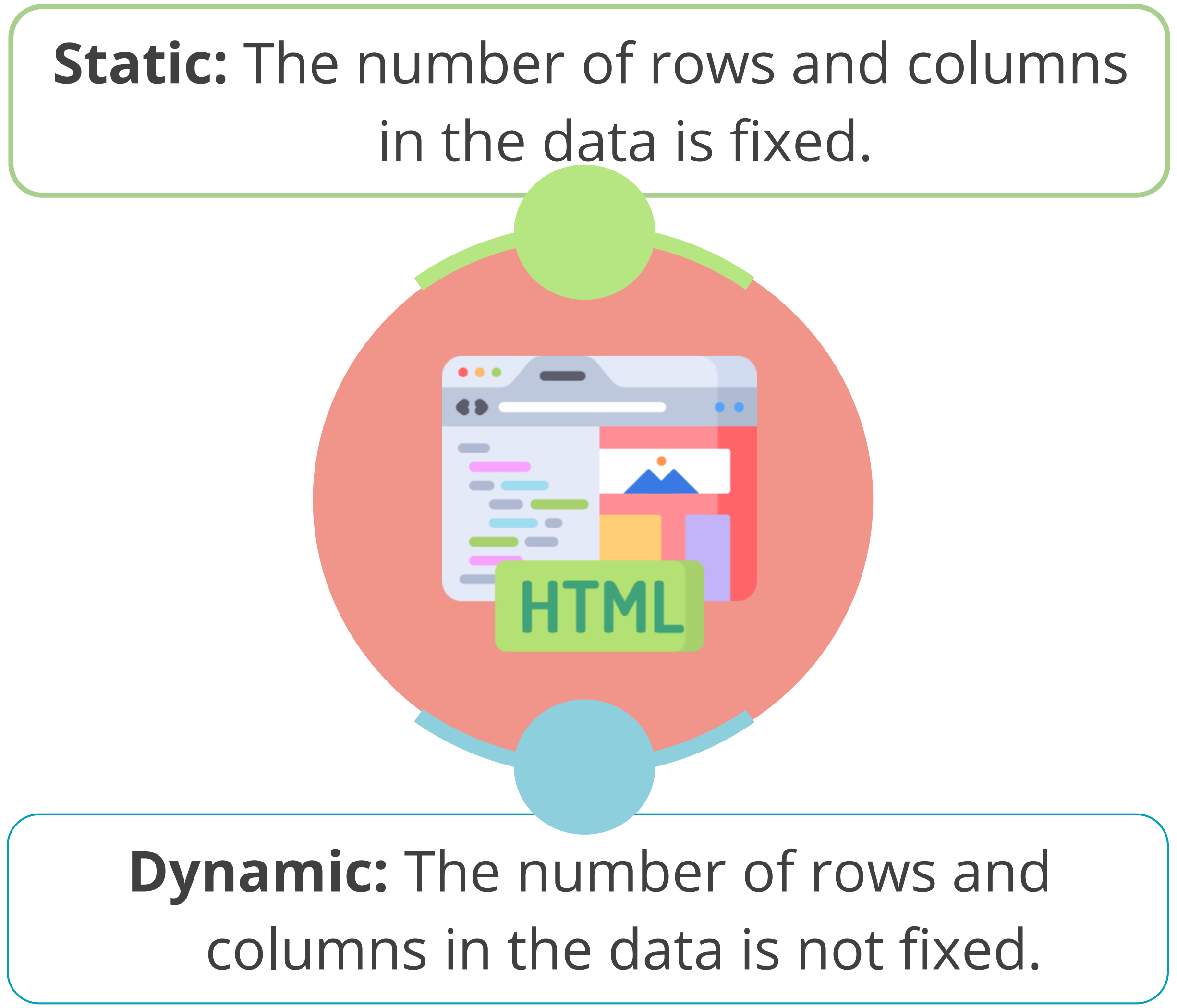


```
//Selecting multiple values by index
oSel.selectByIndex(1);
oSel.selectByIndex(2);

//Or selecting by values
oSel.selectByValue("volvo");
oSel.selectByValue("audi");

//Or selecting by visible text
oSel.selectByVisibleText("Volvo");
oSel.selectByVisibleText("Opel");
```



# Working With Tables

There are two sorts of HTML tables:



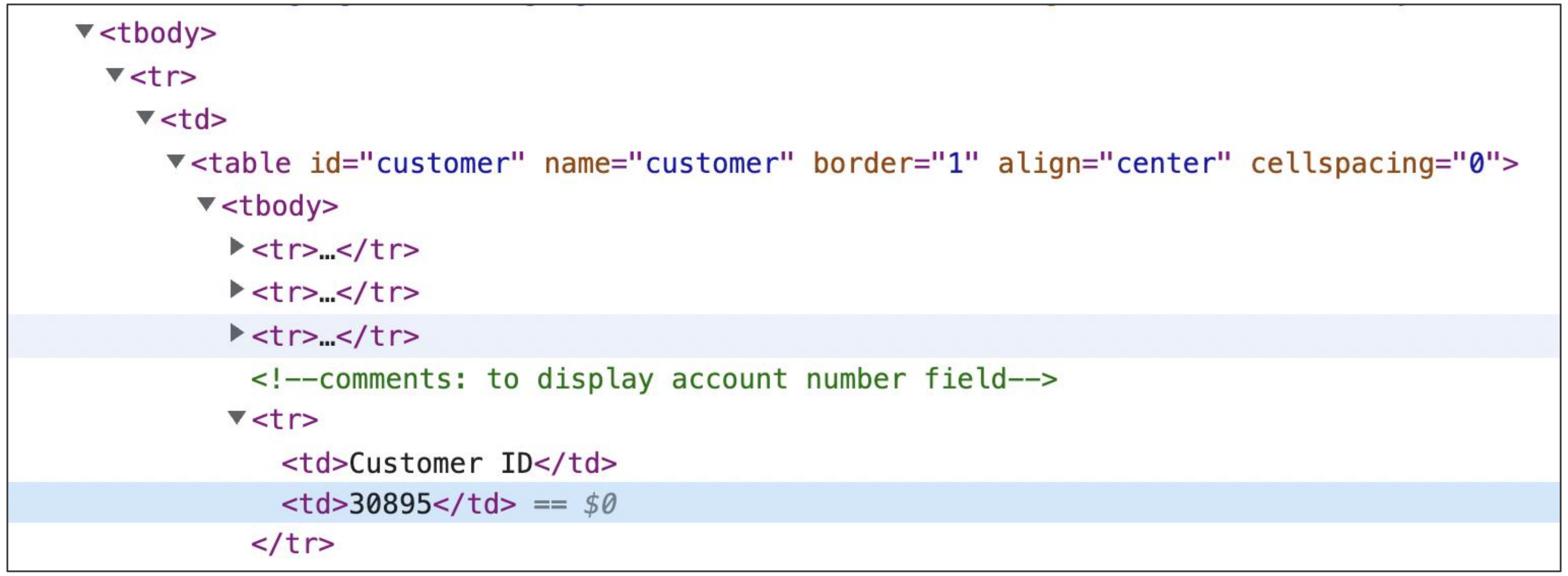
**Static:** The number of rows and columns in the data is fixed.

**Dynamic:** The number of rows and columns in the data is not fixed.

# Working With Tables

- Tables on an HTML page are created with a “table” tag. The rows are represented by “tr” tags, and each row has some values that are represented by “td” tags.
- To get values from the table one can use relative XPath.
- Xpath to get the Customer ID, one can use:

**//table[@id='customer']//td[text()='Customer ID']//following-sibling::td**



```
▼<tbody>
  ▼<tr>
    ▼<td>
      ▼<table id="customer" name="customer" border="1" align="center" cellspacing="0">
        ▼<tbody>
          ▶<tr>…</tr>
          ▶<tr>…</tr>
          ▶<tr>…</tr>
           <!--comments: to display account number field-->
          ▼<tr>
             <td>Customer ID</td>
             <td>30895</td> == $0
           </tr>
```

# Key Takeaways

- Using Selenium, we can interact with different web application components, like radio buttons, textbox, select dropdown, etc.
- The process of interacting with elements is the same. Identify the element first, using one of the locators or identifiers, and perform the operation or action.
- Operations or actions are click, clear, select a value from a dropdown, type, etc.
- The checkbox is an HTML element that allows the user to select any option from the list of values.
- Multiselect dropdown is the one that allows selecting multiple values from a dropdown.

Thank You

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```