1. Handling Buttons in Selenium

Definition

A **button** triggers an action (e.g., submit a form, navigate to another page) when clicked.

Locating & Clicking a Button

```
// Click a button using ID
driver.findElement(By.id("loginBtn")).click();

// Click using XPath (example with text)
driver.findElement(By.xpath("//button[text()='Submit']")).click();

// Click using CSS Selector
driver.findElement(By.cssSelector("button.btn-primary")).click();
```

Example Scenario – Login Button

```
driver.get("https://opensource-demo.orangehrmlive.com/");
driver.findElement(By.id("txtUsername")).sendKeys("Admin");
driver.findElement(By.id("txtPassword")).sendKeys("admin123");
driver.findElement(By.id("btnLogin")).click();
```

□ **Tip:** Use `.click()` method for button clicks. If a button is hidden under another element, use `JavascriptExecutor` to click.

---

2. Handling Input Boxes (Text Fields)

Definition

An **input box** allows the user to enter text or numbers.

Entering Text

```
// Using sendKeys()
driver.findElement(By.name("username")).sendKeys("myUser");

// Clearing first, then typing
WebElement email = driver.findElement(By.id("email"));
email.clear();
email.sendKeys("test@example.com");
```

Getting Value from Input

```
String value = driver.findElement(By.id("email")).getAttribute("value");
System.out.println("Entered email: " + value);
```

Example Scenario – Google Search

```
driver.get("https://www.google.com");
WebElement searchBox = driver.findElement(By.name("q"));
searchBox.sendKeys("Selenium WebDriver");
searchBox.submit();
```

□ **Tip:** `.submit()` can be used instead of clicking the search button if the element is inside a form.

---

3. Handling Checkboxes

Definition

A **checkbox** lets the user select multiple options independently.

Selecting & Deselecting a Checkbox

```
WebElement checkbox = driver.findElement(By.id("subscribeCheckbox"));

// Select if not already selected
if (!checkbox.isSelected()) {
    checkbox.click();
}

// Deselect if selected
if (checkbox.isSelected()) {
    checkbox.click();
}
```

Selecting Multiple Checkboxes

```
List<WebElement> checkboxes =
driver.findElements(By.xpath("//input[@type='checkbox']"));
for (WebElement cb : checkboxes) {
    if (!cb.isSelected()) {
        cb.click();
    }
}
```

Example Scenario – Newsletter Subscription

```
driver.get("https://example.com/newsletter");
WebElement subscribe = driver.findElement(By.id("subscribe"));
if (!subscribe.isSelected()) {
    subscribe.click();
}
```

□ **Tip:** Always check `isSelected()` before clicking to avoid toggling unnecessarily.

---

4. Handling Radio Buttons

Definition

A **radio button** allows only **one option** to be selected from a group.

Selecting a Radio Button

```
WebElement maleOption = driver.findElement(By.id("genderMale"));
maleOption.click();
```

Selecting by Value

```
List<WebElement> radios = driver.findElements(By.name("gender"));
for (WebElement radio : radios) {
    if (radio.getAttribute("value").equalsIgnoreCase("Female")) {
        radio.click();
        break;
    }
}
```

Example Scenario – Gender Selection

```
driver.get("https://example.com/register");
driver.findElement(By.id("genderFemale")).click();
```

□ **Tip:** Use `isSelected()` to verify the chosen radio button.

---

5. Handling Select Box (Dropdowns)

Definition

A **select box** lets the user pick one or more values from a dropdown list.

Using the Select Class

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

// Locate dropdown
Select countryDropdown = new Select(driver.findElement(By.id("country")));

// Select by visible text
countryDropdown.selectByVisibleText("India");

// Select by value
countryDropdown.selectByValue("US");

// Select by index (0-based)
countryDropdown.selectByIndex(3);
```

Getting All Options

```
List<WebElement> allOptions = countryDropdown.getOptions();
for (WebElement option : allOptions) {
    System.out.println(option.getText());
}
```

Example Scenario – Country Selection

```
driver.get("https://example.com/signup");
Select country = new Select(driver.findElement(By.id("country")));
country.selectByVisibleText("India");
```

□ **Tip:** For multi-select dropdowns, you can use `.deselectAll()`, `.deselectByIndex()`, etc.

---

6. End-to-End Example – Handling All Controls

```
import org.openqa.selenium.*;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
import io.github.bonigarcia.wdm.WebDriverManager;

public class FormAutomation {
    public static void main(String[] args) {
        WebDriverManager.chromedriver().setup();
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();

        driver.get("https://example.com/form");

        // Input box
        driver.findElement(By.id("firstName")).sendKeys("John");

        // Radio button
        WebElement maleRadio = driver.findElement(By.id("genderMale"));
        if (!maleRadio.isSelected()) maleRadio.click();

        // Checkbox
        WebElement agreeTerms = driver.findElement(By.id("agree"));
        if (!agreeTerms.isSelected()) agreeTerms.click();

        // Select dropdown
        Select country = new Select(driver.findElement(By.id("country")));
        country.selectByVisibleText("India");

        // Button
        driver.findElement(By.id("submitBtn")).click();

        driver.quit();
    }
}
```