4.5 Handling Various Web Elements

This section will guide you to:

* Handle various web elements present on the page.

This section is divided into eleven subsections, namely:

4.5.1 Edit box

4.5.2 Link

4.5.3 Button

4.5.4 Image, image link, and image button

4.5.5 Text area

4.5.6 Checkbox

4.57 Radio button

4.5.8 Dropdown list

4.5.9 Web table/HTML table

4.5.10 Frame

### 4.5.11 Switching between tabs in the same browser window

**Step 4.5.1:** Edit box

* Open Eclipse
* It is a basic text control that enables a user to type a small amount of text.
* Operations on Edit box
  + Enter a Value,
  + Clear the Value,
  + Check enabled status,
  + Check edit box existence,
  + Get the value

**Step 4.5.2:** Link

* Link is more appropriately referred to as a hyperlink and connects one web page to another. It allows the user to click their way from page to page.
* Operations on Link
  + Click the link,
  + Check the link existence,
  + Check the link enabled status,
  + Return the link name

**Step 4.5.3:** Button

* This represents a clickable button, which can be used in forms and places in the document that needs a simple, standard button functionality.
* Operations on Button
  + Click
  + Check Enabled status
  + Display status

**Step 4.5.4:** Image, image link, and image button

* It helps in performing actions on images like clicking on the image link or the image button, etc.
* Operations Image
  + Three types of Image elements in Web Environment
  + General Image (No functionality)
  + Image Button (Submits)
  + Image Link (Redirects to another page/location)

**Step 4.5.5:** Text area

* It is an inline element used to designate a plain-text editing control containing multiple lines.
* Return / Capture Text Area or Error message from a web page

**Step 4.5.6:** Checkbox

* This is a selection box or a tick box which is a small interactive box that can be toggled by the user to indicate an affirmative or a negative choice.
* Operations on checkbox
  + Check if the checkbox is displayed or not
  + Check if the checkbox is enabled or not
  + Check if the checkbox is selected or not
  + Select the checkbox
  + Unselect the checkbox

**Step 4.5.7:** Radio button

* It is an option button which is a graphical control element that allows the user to choose only one predefined set of mutually exclusive options.
* Operations on Radio Button
  + Select Radio Button
  + Verify if the Radio Button is displayed or not
  + Verify if the Radio Button is enabled or not
  + Verify if the Radio Button is selected or not
* Example:

oRadioButton.get(1).click();

**Step 4.5.8:** Drop-down list

* It is a graphical control element, similar to the list box, which allows the user to choose one value from the list. When this drop-down list is inactive, it displays only a single value.
* Operations on drop-down list
  + Check the drop-down box’s existence
  + Check if the drop-down is enabled or not
  + Select an item
  + Get Items Count
* Example:

Select fruits = new Select(driver.findElement(By.id("fruits")));

fruits.selectByVisibleText("Banana");

fruits.selectByIndex(1);

**Step 4.5.9:** Web table/HTML table

* Operations on Web Table/HTML Table
  + Get cell value
  + Get Rows Count
  + Get Cells Count

**Step 4.5.10:** Frame

* Operations on Frame
  + Switch from Top window to a frame
  + Switch from a frame to Top window
* Examples:
  + driver.switchTo().frame("iframe1");
  + driver.switchTo().frame("id of the element");

**Step 4.5.11:** Switching between tabs in the same browser window

* Open a new tab using Ctrl + t.
* Driver control automatically switches to the newly opened tab.
* Perform the required operations here.
* Next, switch back to the old tab using Ctrl + Tab. You need to keep pressing this unless you reach the desired tab.
* Once the desired tab is reached, then perform the operations in that tab.
* Example:

driver.switchTo().window(tabs2.get(1));

driver.switchTo().window(tabs2.get(0));

Here is the code for the above steps:

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

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

import java.util.ArrayList;

public class WebElementsDemo {

    public static void main(String[] args) {

        // Set the path to the chromedriver executable

        System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

        // Launch the Chrome browser

        WebDriver driver = new ChromeDriver();

        // Navigate to Google homepage

        driver.get("https://www.google.com");

        // Step 4.5.1: Edit box

        // Enter a value in the search box

        driver.findElement(By.name("q")).sendKeys("Hello World");

        // Clear the value

        driver.findElement(By.name("q")).clear();

        // Check enabled status of the search box

        boolean isSearchBoxEnabled = driver.findElement(By.name("q")).isEnabled();

        // Check search box existence

        boolean doesSearchBoxExist = driver.findElements(By.name("q")).size() > 0;

        // Get the value from the search box

        String searchBoxValue = driver.findElement(By.name("q")).getAttribute("value");

        // Step 4.5.2: Link

        // Click the "Images" link

        driver.findElement(By.linkText("Images")).click();

        // Check the "Images" link existence

        boolean doesImagesLinkExist = driver.findElements(By.linkText("Images")).size() > 0;

        // Check the "Images" link enabled status

        boolean isImagesLinkEnabled = driver.findElement(By.linkText("Images")).isEnabled();

        // Return the "Images" link name

        String imagesLinkName = driver.findElement(By.linkText("Images")).getText();

        // Step 4.5.3: Button

        // Click the "Google Search" button

        driver.findElement(By.name("btnK")).click();

        // Check enabled status of the "Google Search" button

        boolean isSearchButtonEnabled = driver.findElement(By.name("btnK")).isEnabled();

        // Display status of the "Google Search" button

        boolean isSearchButtonDisplayed = driver.findElement(By.name("btnK")).isDisplayed();

        // Step 4.5.4: Image, image link, and image button

        // Operations on Image

        // Step 4.5.5: Text area

        // Return / Capture Text Area or Error message from a web page

        String textAreaValue = driver.findElement(By.tagName("textarea")).getAttribute("value");

        // Step 4.5.6: Checkbox

        // Check if the "I'm Feeling Lucky" checkbox is displayed or not

        boolean isFeelingLuckyCheckboxDisplayed = driver.findElement(By.name("btnI")).isDisplayed();

        // Check if the "I'm Feeling Lucky" checkbox is enabled or not

        boolean isFeelingLuckyCheckboxEnabled = driver.findElement(By.name("btnI")).isEnabled();

        // Check if the "I'm Feeling Lucky" checkbox is selected or not

        boolean isFeelingLuckyCheckboxSelected = driver.findElement(By.name("btnI")).isSelected();

        // Select the "I'm Feeling Lucky" checkbox

        driver.findElement(By.name("btnI")).click();

        // Unselect the "I'm Feeling Lucky" checkbox

        driver.findElement(By.name("btnI")).click();

        // Step 4.5.7: Radio button

        // Select the "Search" radio button

        driver.findElement(By.cssSelector("input[name='tbm'][value='s']")).click();

        // Verify if the "Search" radio button is displayed or not

        boolean isSearchRadioButtonDisplayed = driver.findElement(By.cssSelector("input[name='tbm'][value='s']")).isDisplayed();

        // Verify if the "Search" radio button is enabled or not

        boolean isSearchRadioButtonEnabled = driver.findElement(By.cssSelector("input[name='tbm'][value='s']")).isEnabled();

        // Verify if the "Search" radio button is selected or not

        boolean isSearchRadioButtonSelected = driver.findElement(By.cssSelector("input[name='tbm'][value='s']")).isSelected();

        // Step 4.5.8: Drop-down list

        // Check the drop-down box existence

        boolean doesDropDownExist = driver.findElements(By.name("lang")).size() > 0;

        // Check if the drop-down box is enabled or not

        boolean isDropDownEnabled = driver.findElement(By.name("lang")).isEnabled();

        // Select an item from the drop-down list

        Select dropdown = new Select(driver.findElement(By.name("lang")));

        dropdown.selectByVisibleText("English");

        dropdown.selectByValue("fr");

        // Get the count of items in the drop-down list

        int itemCount = dropdown.getOptions().size();

        // Step 4.5.9: Web table/HTML table

        // Get cell value from a table

        String cellValue = driver.findElement(By.xpath("//table/tbody/tr[1]/td[1]")).getText();

        // Get the count of rows in a table

        int rowCount = driver.findElements(By.xpath("//table/tbody/tr")).size();

        // Get the count of cells in a table row

        int cellCount = driver.findElements(By.xpath("//table/tbody/tr[1]/td")).size();

        // Step 4.5.10: Frame

        // Switch from the default content to a frame

        driver.switchTo().frame("frameName");

        // Switch back from a frame to the default content

        driver.switchTo().defaultContent();

        // Step 4.5.11: Switching between tabs in the same browser window

        // Open a new tab

        driver.findElement(By.cssSelector("body")).sendKeys(Keys.CONTROL + "t");

        // Switch to the newly opened tab

        ArrayList<String> tabs = new ArrayList<>(driver.getWindowHandles());

        driver.switchTo().window(tabs.get(1));

        // Perform operations in the new tab

        // Switch back to the old tab

        driver.findElement(By.cssSelector("body")).sendKeys(Keys.CONTROL + "\t");

        driver.switchTo().defaultContent();

        // Perform operations in the old tab

        // Close the browser

        driver.quit();

    }

}

Note: Replace the "**path/to/chromedriver**" with the actual path to the chromedriver executable on your system.