Manipulating the elements

Test automation basics with Selenium & Java

What Selenium can do with WebElement?

WebElement methods:

- click()
- sendKeys("some string") *or* sendKeys(Keys.RETURN)
- getText()
- getAttribute("value")

click and sendKeys

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Scenario with search button

```
WebDriver driver  = new ChromeDriver();
driver.navigate().to("https://www.google.com");
WebElement searchField = driver.findElement(By.name("q"));
//put 'selenium java' text to text field
searchField.sendKeys("selenium java");
WebElement searchButton = driver.findElement(By.name("btnK"));
//press on search button - IT DOESN'T WORKS! (Why?)
searchButton.click();
driver.quit();
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Scenario with Enter button

```
WebDriver driver  = new ChromeDriver();

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

System.out.println(driver.getTitle());

WebElement searchField = driver.findElement(By. id("lst-ib"));

 //put 'selenium java' text to text field

searchField.sendKeys("selenium java");

 //press ENTER

searchField.sendKeys(Keys. RETURN);

driver.quit();
```

getText and getAttribute

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

WebElement methods

```
<input name="sampleName" value="qwerty">Hello
world!</input>
```

```
driver.findElement(By.name("sampleName"))
      .getAttribute("value");

driver.findElement(By.name("sampleName"))
      .getText();
```

getText and getAttribute example

```
WebDriver driver  = new ChromeDriver();

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

WebElement gmailLink = driver.findElement(By.xpath("//a[@class='gb_d']"));

System.out.println("href attribute contains value: " +
                   gmailLink.getAttribute("href"));

System.out.println("element contains text: " +
                   gmailLink.getText());

driver.quit();
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Exercise

Search with Enter key

Automate steps

1) Open google.com
2) Find search input field
3) Clear input field
4) Put some text to the field
5) Press Enter / Return key
6) Print in console number of results of google search
7) Print in console how many time took the search

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Local Tests

```
public class TestHelper {
    public static String siteUrl() {
        return "d:/qa/site/";
    }
}
```

```
public class LocalTest {
    @Test
    public void testGet() {
        File file = new File(
              "src/test/resources/chromedriver.exe");
        System.setProperty("webdriver.chrome.driver",
              file.getAbsolutePath());
        WebDriver driver = new ChromeDriver();
        driver.get(TestHelper.siteUrl()
        + "locators.html");
    }
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Find element by ID

```
// Link
driver.findElement(By.id("submit_btn")).click();
driver.findElement(By.id("cancel_link")).click();
// Textfield
driver.findElement(By.id("username"))
    .sendKeys("ddosers");
// HTML Div element
driver.findElement(By.id("alert_div")).getText();
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Find element by Name

```
driver.findElement(By.name("comment"))
      .sendKeys("Selenium Cool");
```

Find element by (Partial) Link Text

For Hyperlinks only. Using a link's text is probably the most direct way to click a link, as it is what we see on the page.

```
driver.findElement(By.linkText("Cancel")).click();
```

Selenium allows you to identify a hyperlink control with a partial text. This can be quite useful when the text is dynamically generated. In other words, the text on one web page might be different on your next visit. We might be able to use the common text shared by these dynamically generated link texts to identify them

```
// will click the "Cancel" link
driver.findElement(By.partialLinkText("ance")).click();
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Find element by XPath

- XPath, the XML Path Language, is a query language for selecting nodes from an XML document. When a browser renders a web page, it parses it into a DOM tree or similar. XPath can be used to refer a certain node in the DOM tree.
- If this sounds a little too much technical for you, don’t worry, just remember XPath is the most powerful way to find a specific web control.

```
driver.findElement(By.xpath(
    "//*[@id='div2']/input[@type='checkbox']"))
    .click();
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Find element by Tag Name

- There are a limited set of tag names in HTML. In other words, many elements share the same tag names on a web page.
- We normally don’t use the tag_name locator by itself to locate anelement.
- We often use it with others in a chained locators (see the section below). However, there is an exception

```
driver.findElement(By.tagName("body")).getText();
```

Find element by Tag Name

- There are a limited set of tag names in HTML. In other words, many elements share the same tag names on a web page.
- We normally don’t use the tag_name locator by itself to locate anelement.
- We often use it with others in a chained locators (see the section below). However, there is an exception

```
driver.findElement(By.tagName("body")).getText();
```

The above test statement returns the text view of a web page, this is a very useful one as Selenium WebDriver does not have built-in method return the text of a web page.

Find element by Class

- The class attribute of a HTML element is used for styling. It can also be used for identifying elements.
- Commonly, a HTML element's class attribute has multiple values, like below.

```
<a href="back.html" class="btn btn-default">Cancel</a>
<input type="submit"
    class="btn btn-deault btn-primary">Submit</input>
```

You may use any one of them

```
// Submit button
driver.findElement(By.className("btn-
primary")).click();

// Cancel link
driver.findElement(By.className("btn")).click();
```

Find element by CSS Selector

- You may also use CSS Path to locate a web element.

```
driver.findElement(By.cssSelector(
      "#div2 > input[type='checkbox']")).click();
```

Find element by CSS Selector

- You may also use CSS Path to locate a web element.

```
driver.findElement(By.cssSelector(
      "#div2 > input[type='checkbox']")).click();
```

However, the use of CSS selector is generally more prone to structure changes of a web page

Chain findElement to find child elements

- For a page containing more than one elements with the same attributes, like the one below, we could use XPath locator.

```
<div id="div1">
    <input type="checkbox" name="same" value="on">
    Same checkbox in Div
</div>
<div id="div2">
    <input type="checkbox" name="same" value="on">
    Same checkbox in Div 2
</div>
```

There is another way: chain `findElement` to find a child element.

```
driver.findElement(By.id("div2"))
      .findElement(By.name("same"))
      .click();
```

Find multiple elements

- As its name suggests, findElements return a list of matched elements back. Its syntax is exactly the same as findElement, i.e. can use any of 8 locators.
- The test statements will find two checkboxes under div#container and click the second one.

```
List<WebElement> checkboxElems =
driver.findElements(By.xpath(
    "//div[@id='container']//input[@type='checkbox']"));
System.out.println(checkboxElems); // => 2
checkboxElems.get(1).click();
```