Locators in Selenium

To perform an action on any element, we first have to locate/find it on the web page through various built-in locator strategies which are being offered by Selenium WebDriver.

WE'LL COVER THE FOLLOWING

How do we identify an element in Selenium?

```
WebElement name = driver.findElement(By.id("username"));
```

Here, the `findElement(By)` method returns a `WebElement` object. **WebDriver** represents the browser and **WebElement** represents a DOM node (a control, a link, field, etc.). `By` abstract class used above also supports a number of additional locator strategies, as described below.

WebDriver has multiple methods to identify the WebElement using a locator. https://selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html

WebElement provides methods to operate upon the locators for performing some actions.

https://selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html

Element selection strategies in WebDriver

Id

The element whose `id` matches the search value.

In this example, the email input field is identified by the *id* attribute.

```
WebElement emailField = driver.findElement(By.id("email"));
```

name

The element whose `name` matches the search value.

In this example, the email input field is identified by the *name* attribute.

```
WebElement emailField = driver.findElement(By.name("email"));
```

class

Locate the element with a `class` name.

In this example, the email input field is identified by the *class* attribute.

```
WebElement emailField = driver.findElement(By.className("inputtext login_form_input_box"));
```

css

The element matching a `css` selector:

In this example, the email input field is identified by the *css selector*, as shown in the screenshot above.

```
WebElement emailField = driver.findElement(By.cssSelector("input#email"));
```

xpath

Locates elements matching an `xpath` expression:

In this example, the email input field is identified by the *xpath*, as shown in the screenshot above.

```
WebElement emailField = driver.findElement(By.xpath("//input[@type = 'emai
l']"));
```

linkText

The element whose visible text matches the search value:

In this example, the forgotten account field is identified by the *visible link text*, as shown in the screenshot above.

```
WebElement forgottenAccount = driver.findElement(By.linkText("Forgotten ac
count?"));
```

partialLinkText

It locates anchor elements whose visible text *contains* the search value.

In this example, the forgotten account field is identified by the *partially visible link text*, as shown in the screenshot above.

```
WebElement forgottenAccount = driver.findElement(By.partialLinkText("Forgo
tten"));
```

tag

It locates elements whose `tag` name matches the search value.

In this example, the Facebook text is identified by the tag name *"h1"* as shown in the screenshot above.

```
WebElement facebookText = driver.findElement(By.tagName("h1"));
```

Now that you're familiar with locating the web elements, it's time to learn how to play around with these web elements.