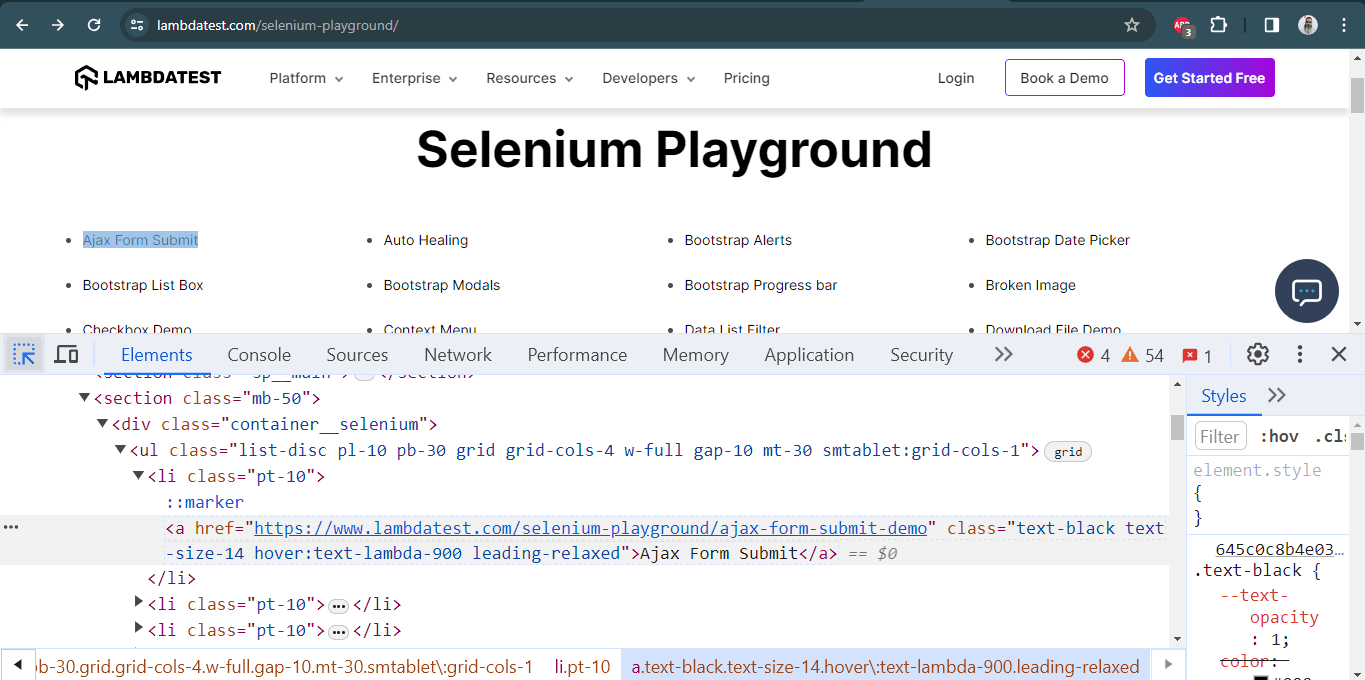
Source: <https://www.lambdatest.com/blog/locators-in-selenium-webdriver-with-examples/#:~:text=CSS%20(Cascading%20Style%20Sheets)%20is,chosen%20over%20the%20XPath%20locator>.

**LinkText Locator in Selenium**

Elements can be located by LinkText, which is present in the hyperlinks. For example, the first link would be selected in a scenario with multiple links of the same text.

However, this identifier strategy can only be used for elements with an anchor( < a > ) tag.

Below is an example of the LambdaTest Selenium Playground website showcasing the selection of the Ajax Form Submit link that is available on the home page. The DOM below shows the highlighted element:



Below is the DOM structure of the same:

|  |  |
| --- | --- |
| 1 | <a href="https://www.lambdatest.com/selenium-playground/ajax-form-submit-demo" class="text-black text-size-14 hover:text-lambda-900 leading-relaxed">Ajax Form Submit&lt;/a> |

Here is how the desired WebElement was located using the *linkText*locator in Selenium:

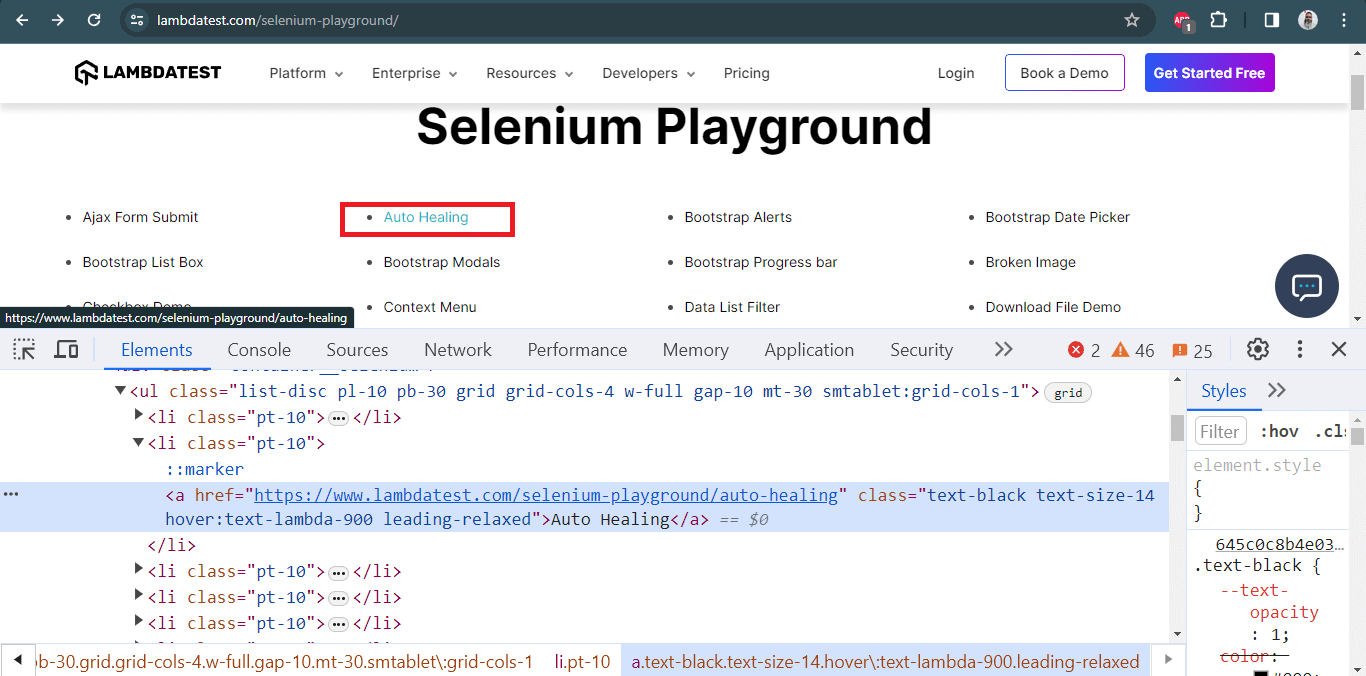
|  |  |
| --- | --- |
| 1 | driver.findElement(By.linkText("Ajax Form Submit"));  NOTE : Link and Partial link text should be used when links are UNIQUE  How to find unique? Go to inpect element and search with the text if its 1of1 then it is unique |

**Partial LinkText Locator in Selenium**

There is a provision to locate a WebElement using Partial LinkText akin to the normal LinkText locator in Selenium. Locating WebElements using Partial LinkText is preferred when the link text is too long.

Here, the partial text helps identify and use a unique element to perform further actions on it. Sometimes, this can also be locating multiple links on a page with a common partial text.

Below is a snapshot of the LambdaTest DOM highlighting the element with the link name as Auto Healing. Instead of using the complete link text, I used the partial link text locator to locate the element using the Healing link text.



Here is the DOM structure of the element:

|  |  |
| --- | --- |
| 1 | <href="https://www.lambdatest.com/selenium-playground/auto-healing" class="text-black text-size-14 hover:text-lambda-900 leading-relaxed">Auto Healing&lt;/a> |

Here is how the desired WebElement was located using the *partialLinkText*locator in Selenium:

The syntax for locating elements by *partialLinkText*is:

|  |  |
| --- | --- |
| 1 | driver.findElement(By.partialLinkText ("Healing")); |

For more information on Link Text and Partial LinkText locators, we recommend you check this article to [find elements with linkText and partialLinkText in Selenium](https://www.lambdatest.com/blog/using-link-text-and-partial-link-text-in-selenium/).

CLassname should be used when space is there in the value

**How to find dynamic elements in selenium?**

We can find it using

1. Contains method
2. Starts-with method
3. Text method
4. Xpath axes
5. Index

A diagram of a web

Description automatically generated

Source: <https://www.youtube.com/watch?v=H1_rmdsZ4gI>

**Contains Method:**

* "contains()" is used to identify an element, when we are familiar with some part of the attributes value of an element.

**Syntax:**

//\*[contains(@attribute\_name,'attribute\_value')]

**Starts-with Method:**

* "starts-with()" is used to identify an element, when we are familiar with the starting attributes value

**Syntax:**

//\*[starts-with(@attribute\_name,'attribute\_value')]

**Text method:**

* "text() method" is used to identify an element based on the text available on the web page.

**Syntax:**

//\*[text()='Google offered in']