

FULL STACK



Automation Testing

Finding Elements



A Day in the Life of an Automation Test Engineer

As Joel has known the basics of Selenium, to perform the automation testing, let us know more.

As an Automation Test Engineer, to automate and interact with web elements of an application, we need to identify these elements with the help of some Locators or Identifiers.

For testing a web-based application, we need to perform specific actions in the HTML elements, such as click, type, and so on.

Using this automation tool to identify the HTML element, he needs to perform a particular operation.

To know about it, let us go through the lesson



Learning Objectives

By the end of this lesson, you will be able to:

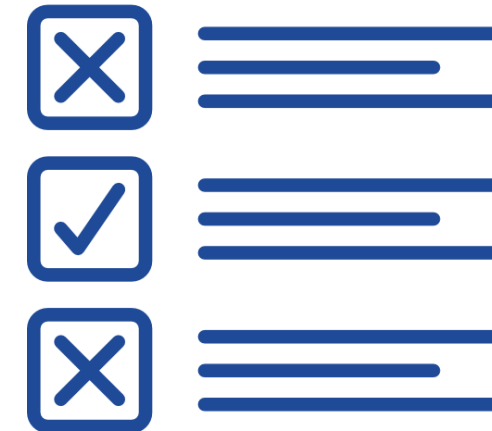
- Comprehend Locators in Selenium
- Comprehend Locators available in the HTML codes
- Locate the elements



Locators or Identifiers

Locators or Identifiers

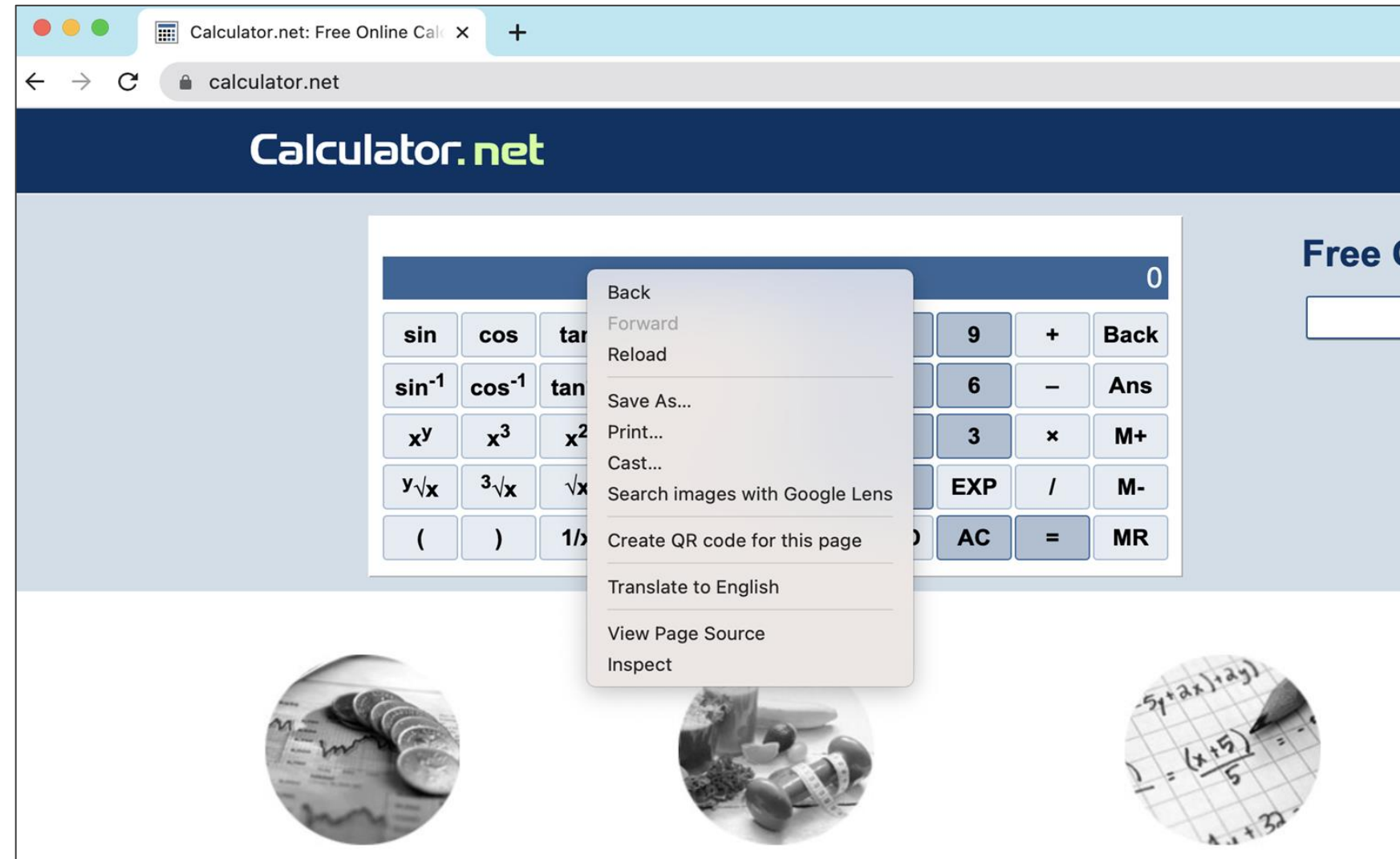
One of the important components of the Selenium infrastructure is the Locator, which enables the scripts to uniquely identify the Web elements on the web page.



Selenium use Locators or Identifiers to interact with elements of a web page like textbox, radio button, checkbox, etc.

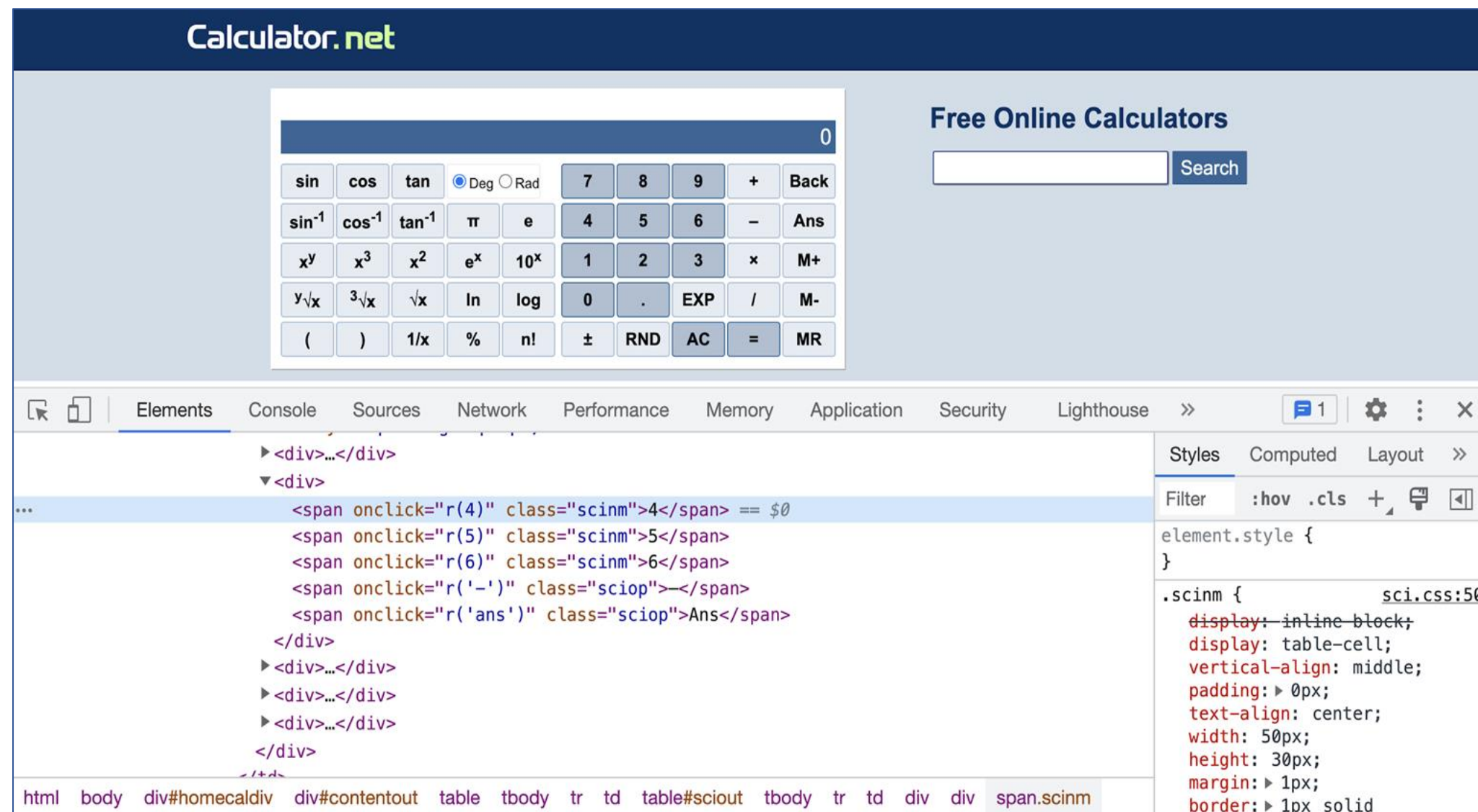
How to Locate Elements?

- To identify the element, you can use the developer's tool provided by every browser vendor.
- Right click and click on Inspect.
- Hover around other elements to change the code.



How to Locate Elements?

Hover around the page to get the HTML code of the elements present on the page. As we hover around, HTML tags change in the developers' tool.



Locators or Identifiers

There are eight Locators or Identifiers in Selenium:

Id

Name

Classname

Linktext

Tagname

Partial Linktext

CSS Selector

XPath

Supported Locators

1. Id

Identifying the web elements using the 'id' attribute

2. Name

Identifying the web elements using the 'name' attribute

3. ClassName

Identifying the web elements using the 'class' attribute

4. LinkText

Querying the XML document, and the same can uniquely identify the web element on any page

Supported Locators

5. Partial Linktext

Identifying web elements using the partially visible text from a link

6. Tagname

Identifying the element using the tag name

7. CSS Selector

Creates style rules for webpages and can be used to identify any web element

8. Xpath

Used to query the XML document, and the same can uniquely identify the web element on any page

Locating Elements

Locating elements in Selenium WebDriver is performed with the help of `findElement()` and `findElements()` methods.

`findElement()`

It returns a `WebElement` object based on specified search criteria or throws an exception if it does not find any element matching the search criteria.

`findElements()`

It returns a list of `WebElements` matching the search criteria. If no elements are found, it returns an empty list.

Find Elements Using ID

Locators Elements Usage - By ID

- ID is the attribute available in the HTML code and can be used to identify the element.
- Syntax: **driver.findElement(By.id("sciOutPut"))**

The screenshot shows the Calculator.net website. The calculator interface is visible, including a display showing '0' and various function buttons like sin, cos, tan, etc. The browser's developer tools are open, showing the 'Elements' tab. The HTML structure is displayed, and the `div#sciOutPut` element is highlighted, showing its content as `<div id="sciOutPut">0</div> == $0`. The breadcrumb at the bottom of the developer tools indicates the path: `html > body > div#homecaldiv > div#contentout > table > tbody > tr > td > table#sciout > tbody > tr > td > div > div#sciOutPut`.

FULL STACK

Find Elements Using Name

Locate Elements Using – By Name

Free C

input#scirdsettingd 13 x 13

0

sin cos tan Deg Rad 7 8 9 + Back

sin⁻¹ cos⁻¹ tan⁻¹ π e 4 5 6 - Ans

x^y x³ x² e^x 10^x 1 2 3 × M+

y√x 3√x √x ln log 0 . EXP / M-

() 1/x % n! ± RND AC = MR

```
<span onclick="r('sin')" class="scifunc">sin</span>
<span onclick="r('cos')" class="scifunc">cos</span>
<span onclick="r('tan')" class="scifunc">tan</span>
<span class="scird">
  <label for="scirdsettingd">
    <input id="scirdsettingd" type="radio" name="scirdsetting" value="deg" onclick="cnDegreeRadians='degree';" checked="" />
    "Deg"
  </label>
  <label for="scirdsettingr">...</label>
</span>
</div>
<div>
  <span onclick="r('asin')" class="scifunc">...</span>
  <span onclick="r('acos')" class="scifunc">...</span>
  <span onclick="r('atan')" class="scifunc">...</span>
  <span onclick="r('pi')" class="scifunc">π</span>
  <span onclick="r('e')" class="scifunc">e</span>
</div>
```

- Name is an attribute available in the HTML code and can be used to identify the element. In this example shown, it is the name attribute of the radio button below.
- Syntax:
driver.findElement(By.name("scirdsetting"))

FULL STACK

Find Elements Using Classname

Locate Elements Using- By Class Name

Free Online Calcul

input#scirdsettingd 13 x 13

0

sin cos tan Deg Rad 7 8 9 + Back

sin⁻¹ cos⁻¹ tan⁻¹ π e 4 5 6 - Ans

x^y x³ x² e^x 10^x 1 2 3 × M+

y√x 3√x √x ln log 0 . EXP / M-

() 1/x % n! ± RND AC = MR

```
<span onclick="r('sin')" class="scifunc">sin</span>
<span onclick="r('cos')" class="scifunc">cos</span>
<span onclick="r('tan')" class="scifunc">tan</span>
<span class="scird">
  <label for="scirdsettingd">
    <input id="scirdsettingd" type="radio" name="scirdsetting" value="deg" onclick="cnDegreeRadians='degree';" checked> == $0
    "Deg"
  </label>
  <label for="scirdsettingr">...</label>
</span>
</div>
<div>
  <span onclick="r('asin')" class="scifunc">...</span>
  <span onclick="r('acos')" class="scifunc">...</span>
  <span onclick="r('atan')" class="scifunc">...</span>
  <span onclick="r('pi')" class="scifunc">π</span>
  <span onclick="r('e')" class="scifunc">e</span>
</div>
```

- Elements are identified with class attribute in the HTML code.
- In the example, the Class name of the button is
driver.findElement(By.className("scifunc"))

Find Elements Using Tagname

Locate Elements Using- By Tagname

- The DOM Tag Name of an element can be used to locate that element in the WebDriver. It is easy to handle tables with the help of this method.
- Syntax:
WebElement table = driver.findElement(By.id("calctable"));
List<WebElement> row = table.findElements(By.tagName("tr"));
int rowcount = row.size();



Find Elements Using Linktext and Partial Linktext

Locate Elements Using – By Linktext

The screenshot displays a web page with four columns of calculator links: Financial Calculators, Health Calculators, Math Calculators, and Other Calculators. The 'BMI Calculator' link under 'Health Calculators' is selected. The browser's developer tools show the corresponding HTML element: `BMI Calculator`. The address bar shows the URL: `https://www.calculator.net/bmi-calculator.html`.

- This method helps to locate a link element with matching visible text.

- Syntax:

`driver.findElement(By.linkText("linkText"))`

Locate Elements Using - Partial Linktext

The screenshot displays a web page with four columns of calculator links: Financial Calculators, Health Calculators, Math Calculators, and Other Calculators. The 'BMI Calculator' link under 'Health Calculators' is highlighted. A tooltip above it shows a mouse cursor and the text '104.92 x 18'. Below the grid, the browser's developer tools are open to the 'Elements' tab, showing the HTML structure. The selected element is an anchor tag: `BMI Calculator == $0`. The URL bar shows `https://www.calculator.net/bmi-calculator.html`.

- This method helps to locate a link element with partial matching visible text.

- Syntax:

`driver.findElement(By.partialLinkText("here")).click();`

Key Takeaways

- Selenium provides us with eight Locators or Identifiers that are ID, Name, Classname, Tag name, Linktext, Partial linktext, CSS Selector, and XPath.
- The first four locators, ID, Name, Classname, and Tagname, are available as attributes in HTML code.
- Linktext and Partial Linktext are used with links and identify elements based on the text associated with them.



FULL STACK

Thank You

