Locators are the way to identify an HTML element on a web page. Locators are one of the essential components of Selenium infrastructure, which help Selenium scripts in uniquely identifying the WebElements(such as text box, button, etc.) present of the web page.

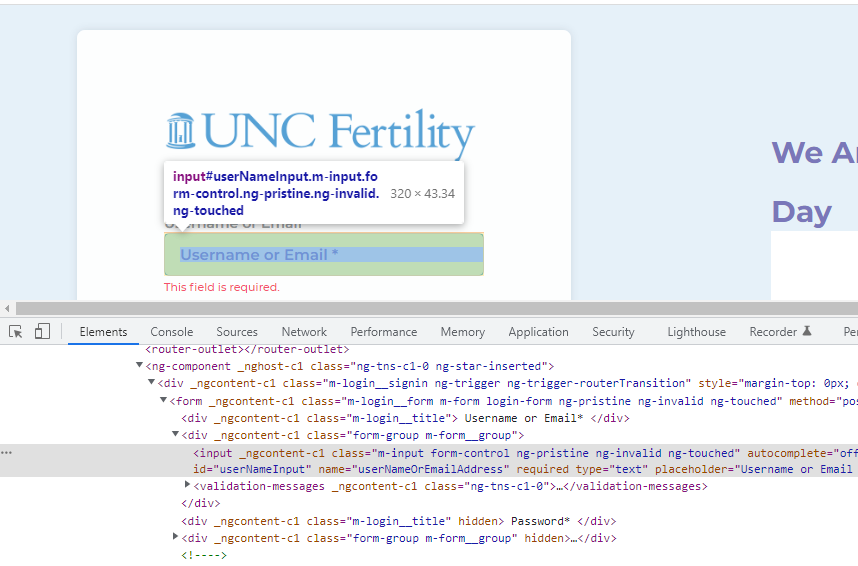
The first thing to start with is to find the *HTML* element in [**DOM**](https://www.w3.org/TR/WD-DOM/introduction.html) (Document Object Model).

1. **DOM** can be accessed in Google Chrome either by pressing **F12** or by **right click** on the web page and then by selecting **Inspect** (as shown in the screenshot below).

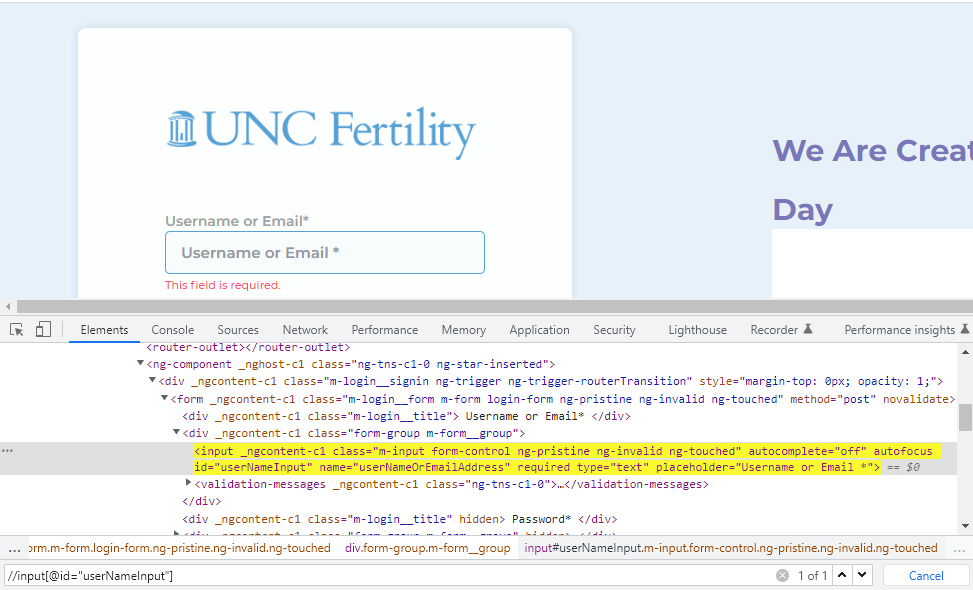
Graphical user interface, application

Description automatically generated

1. Once we click on the "**Inspect option**", it will open the **Developer Tools console**, as shown below. By default, it will open the "Elements" tab, which represents the complete DOM structure of the web page. Now, if we hover the mouse pointer over the HTML tags in the DOM, it will highlight the corresponding elements it represents on the webpage.



1. Then you can start writing your locators and the web elements will get highlighted in the application like below. The tool will let you know how many matchings found for the locator you have written.



**Selenium mainly provides 8 types of locators:**

* **Id**
* **Name**
* **Link Text**
* **Partial Link Text**
* **Tag Name**
* **Class Name**
* **CSS**
* **XPATH**

We can separate the above locators as Attribute based (id,name,class), Link based (linkText,partialLinkText) and Tag based (tagName,xpath and css). We can use xpath in different ways, few examples are below.

* **Attribute Based**

**Syntax= //tagName[@attribute=’value’]**

//input[@id=’userNameInput’]

* **Text Based**

**Syntax=//tagName[text()=’text value’]**

//button[text() =’Next’]

* **Partial text based**

**Syntax=//tagName[contains(text(),’partial value’)]**

//span[contains(text(),' Manage users')]

**Example :**

<!**DOCTYPE** html>

<**html** lang="en">

<**head**>

    <**meta** charset="UTF-8">

    <**title**>Example page for Selenium Test</**title**>

</**head**>

<**body**>

<**p**>

<ul> <li>

<**iframe** name="tree" src="tree.html"></**iframe**>

</li> </ul>

<**div** id="general">

    <**div** class="form">

        <**form** about="/" method="POST">

            User name : <**input** name="username" type="text">

            <**a** href="/agreements.html">User agreements</**a**>

        </**form**>

    </**div**>

</**div**>

</**p**>

</**body**>

</**html**>

In the above example we have a frame so first we must switch to the frame according to selenium to interact with the elements inside the frame. We have a method called switchTo() in selenium. We can make use of that and work with the web elements inside frame. There are 3 ways to handle frames in Selenium.

* By Index
* By Name or Id
* By Web Element

Once you done with frames you have to come back to the main page using a method called defaultContent(), then only you can work with other web elements.

**Locators for the above example:**

driver.findElement(By.tagName(“iframe”))

//input[@name=’username’]

//a[text()=’User agreements’]