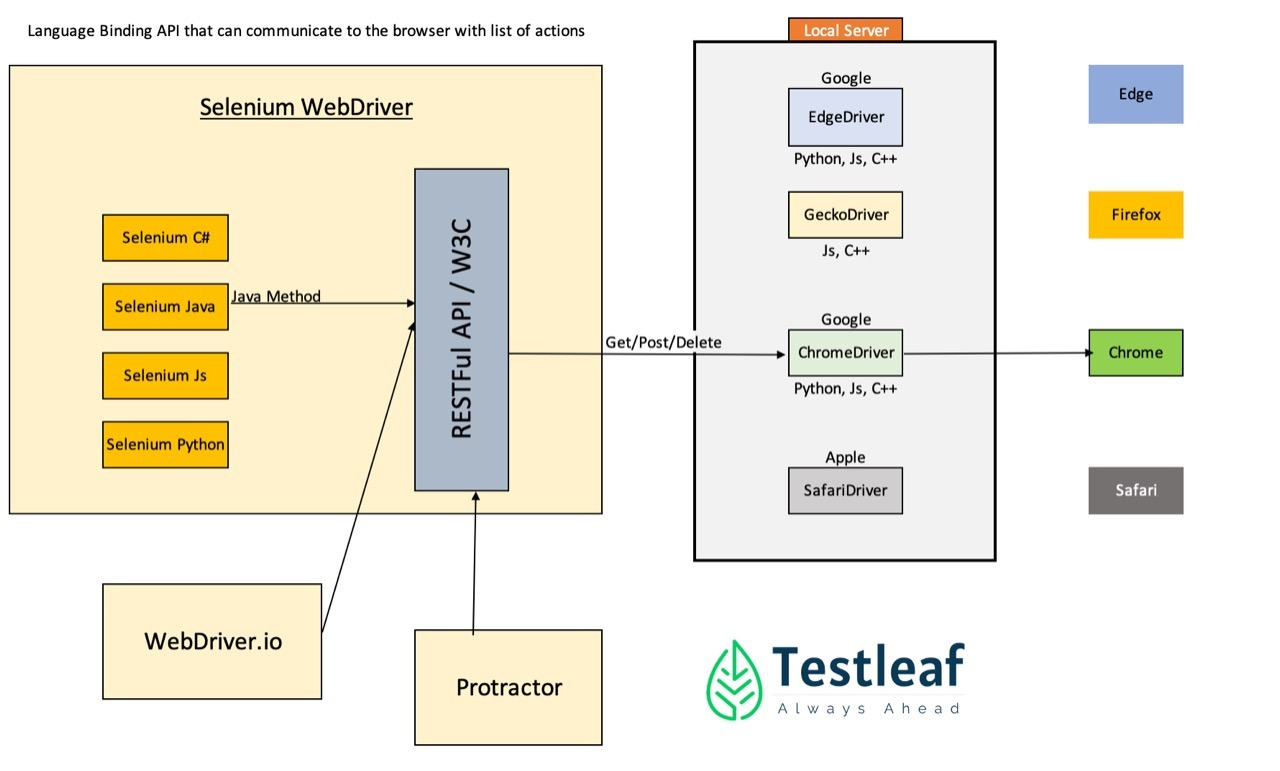
**How to Locate an Element in the Web Page:**

Selenium provides a good platform to automate the Web applications. Selenium is an open source tool which internally works with an API call to communicate with browser using actions and events and makes the job easy to automate the web applications. The components needed for the communication with the browser are programming language, an API, Selenium client library, drivers and browsers.

Internal Working of Selenium



Using selenium client library like Java, python, c# selenium connects the browser drivers which are locally present in your server with the required browser like chrome, firefox, edge, safari through W3c Protocol which is an API call sends and receive the http response for the communication with the web based application.

To establish connection with the browser, the below class and method is called.For example to communicate with the chrome browser,

WebDriverManager.*chromedriver*().setup();

Class which calls the method chromedriver to establish the connection.

ChromeDriver driver= **new** ChromeDriver();

Instantiating the driver class provides access to all the methods of the respective class and has to be imported from the class   
**import** org.openqa.selenium.chrome.ChromeDriver;

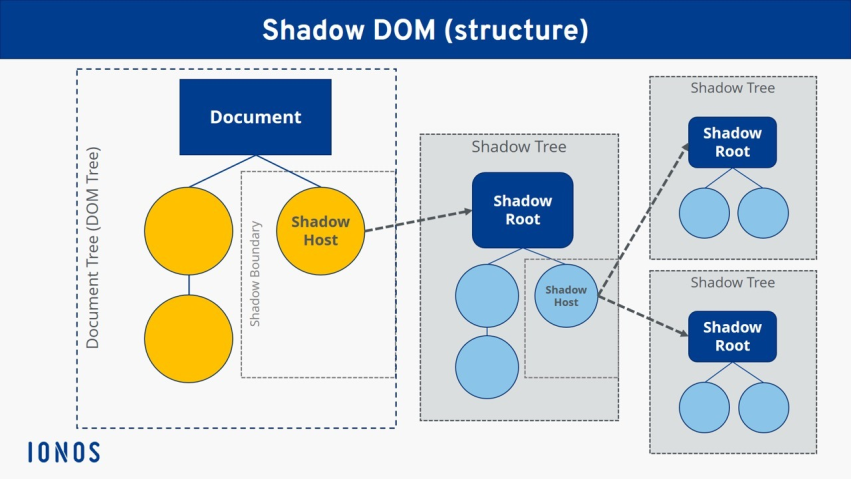
For example:

driver.get("http http://leaftaps.com/opentaps/control/main");

get method helps to load and launch the required web application.

***DOM Elements:***

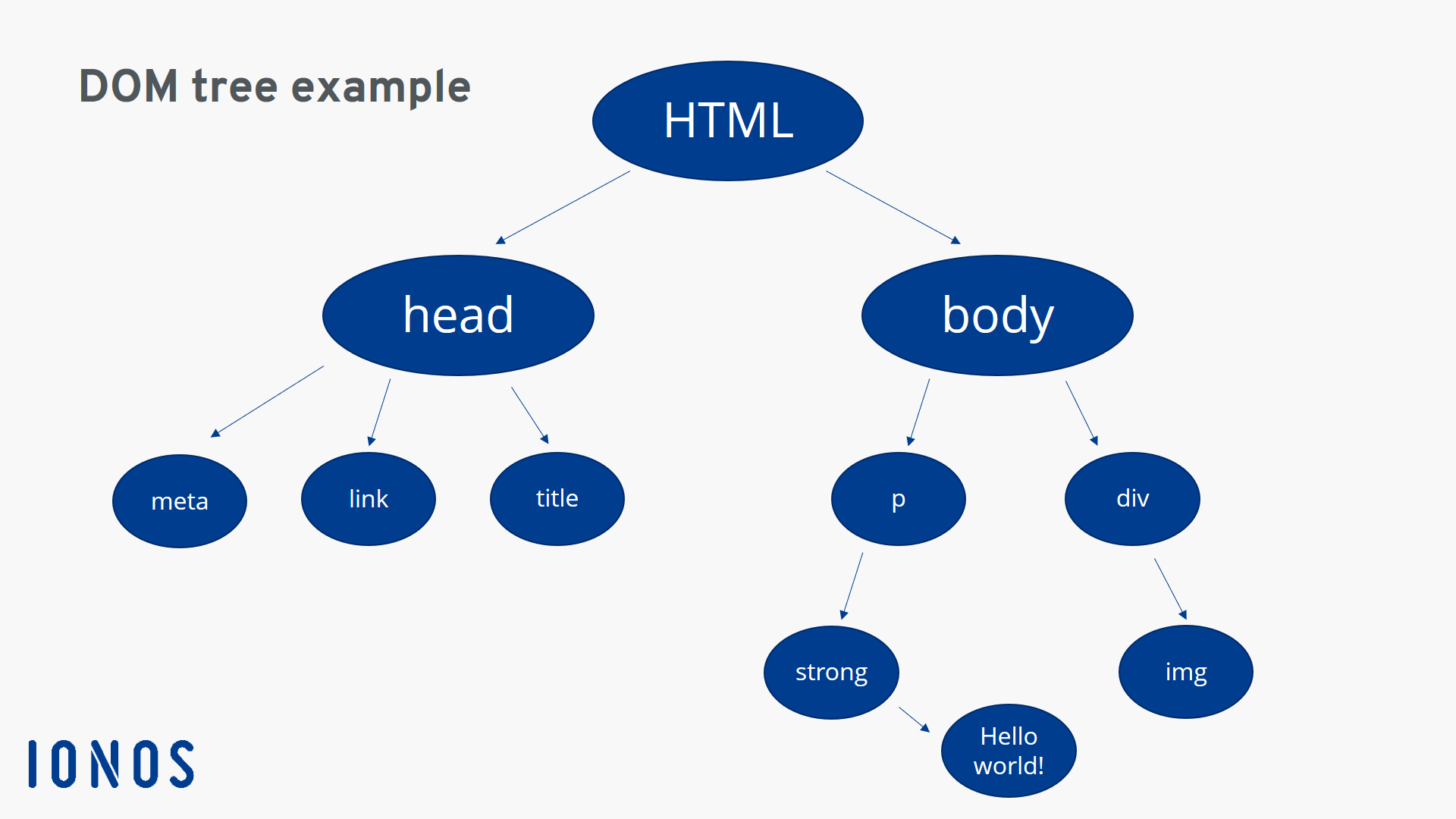
Once the Web page is launched, the user has to access the elements in the webpage. The web page is designed with the HTML Script. The HTML script in the Web page is simply called as DOM (Document Object Model).

 The structure of DOM will be

HTML-refers the root element

Head-> refers with the basic structure like link, meta, title of the webpage and design of the webpage like font, color, size(css sheet)

Body ->refers actual content of the elements of the Web page.



Structure of HTML be like in the above fig:

<! DOCTYPE html> ->refers the type of document  
<html>  
<body>->includes the actual content of the web page

<h1>My First Heading</h1>-> refers the headings in the page  
 <p>My first paragraph.</p>->refers the paragraphs in the page

</body>  
</html>

The element with in <> (angular braces) are called as tag name. Some of the tag names are

<> </>=>represents the start and end of the content

<div> => divides the content of the page

<input>=> refers the input control like textbox

<span>=> refers to the sections in the page

<a>=> refers the hyperlink in the page

<img>=>refers the image file

<iframe>=>refers the HTML document inside the HTML Document

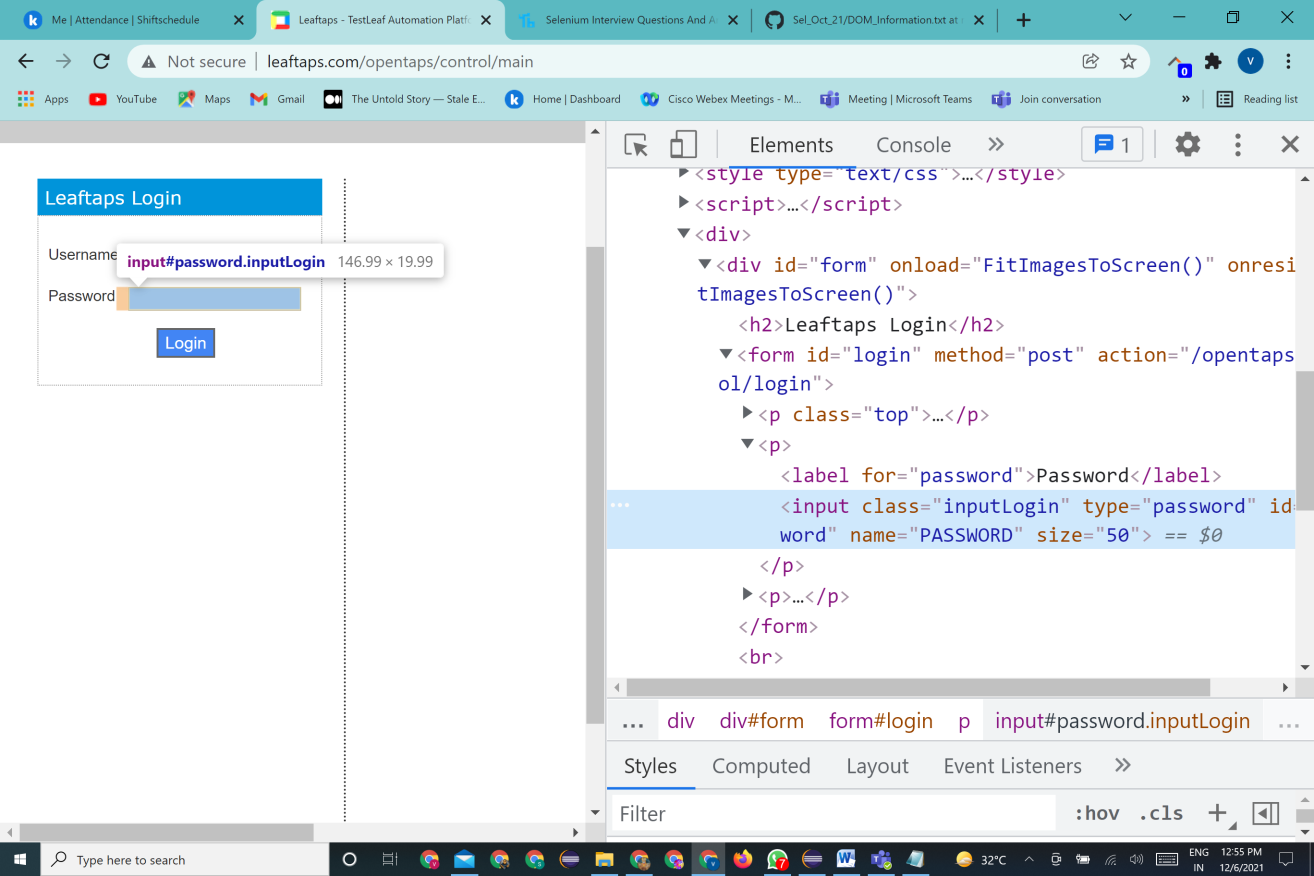
<label> refers the label for the input tag content.

***DOM and HTML attributes:***

To get the information of WebElements from the DOM, the user needs to right click and inspect the required element in the Webpage

The shortcut keys to get open the developer tools(DOM)

\*F12

 \*ctrnl+shift+IFor example, Lets take the web application –Leaftaps

When inspecting the password input field, the web element structure will be like in the below given format

<label for=" password" Password </ label >

<input class=" inputLogin" type= " password" id= " password" name= " PASSWORD" size= "50" < input >

The elements in brown refer the attribute(class) of the given tag name(input) and the blue text(inputLogin) refers the value of the attribute. The black color text(Password) refers the actual text name of the label field. Some of the Attributes of HTML are id, class, name, for, type, value, for, text.

To interact with these elements in the DOM, selenium provides locators to identify the particular element with the help of the selenium webDriver methods findElement() and findElements(). The method findElement() helps to locate a single web element in the Web page and its return type is **WebElement** whereas the findElements returns **List** of webElements.

The element in webpage is identified with the syntax:

driver.findElement(By.by)

*uses locators to identify the web elements*

*Webdriver methods Sets the method to find the Elements and it has to be imported from the class*

**Import** org.openqa.selenium.By;

Let’s explore about selenium Locators:

Selenium locates the element in two ways

1. Methods using Unique Locators

2. Methods using XPath

Identifying Web Elements using Locators:

Selenium Locators are of 8 types. Each locator identifies the element uniquely. They are

**Id -> uses the id attribute**

**Name -> uses the name attribute**

**className -> uses the class attribute**

**linkText -> uses the hyperlink text from the tag name<a>**

**partialLinkText -> uses the partial hyperlink text**

**tagName ->uses the tag name of the web element**

**xpath -> uses the all attributes with the help of XML document.**

**cssSelector ->uses the style rules of the web page**

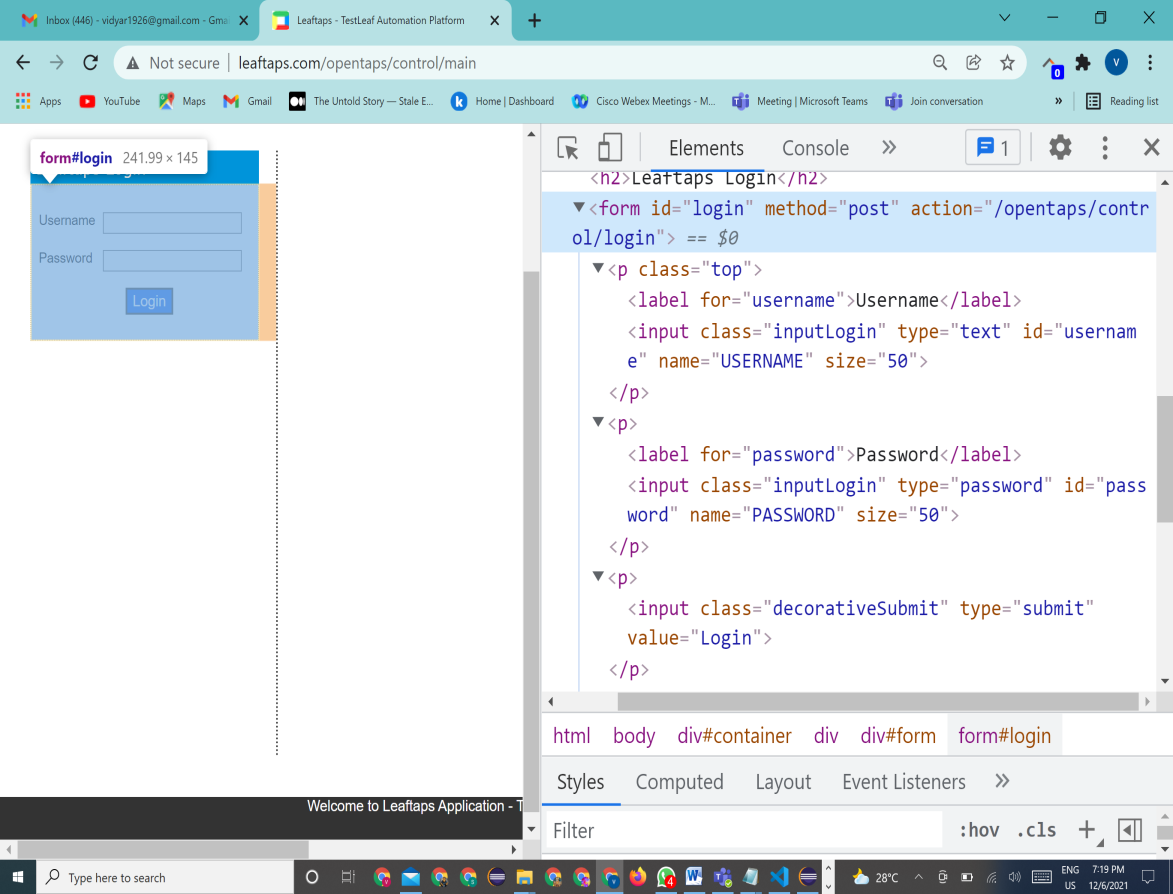
**syntax to use the attributes and attribute values of dom as locator is as**

driver.findElement(By.*locator*("attribute value")) ;

**Let’s dive into how to use each locators and when to use!!**

|  |  |  |  |
| --- | --- | --- | --- |
| **Locators** | **WebDriver** | **When do you use?** | **When you cannot use?** |
| Id | id | Always (Most preferred) | Numbers |
| Name | name | If “id” does not exist | Duplicate |
| Link | linkText | If it is a link | Duplicate |
| Tag name | tagName | For collection of objects | Duplicate |
| Class name | className | When class is unique w/o spaces | White spaces, Duplicate |
| Xpath | xpath | If none of the above, works | Xpath can change |
| Style sheet | cssSelector | This is the last option you have! | CSS can change |

In general when the values of attributes are static, the basic locators are used to identify the webelement.Now will write the selenium script to automate the Login page of Leaftaps application.



The application here involves the actions entering the username and password. After that we need to click the login button. To facilitate the actions selenium provides the method sendKeys(“value “) and to click the button we have the method click().

The following sample code shows the way to identify each web element and its actions to be done into it.

**package** seleniumSample1;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** io.github.bonigarcia.wdm.WebDriverManager;

**public** **class** CreateLead {

**public** **static** **void** main(String[] args) {

// Setting up the browser

WebDriverManager.chromedriver().setup();

//create the instance

ChromeDriver driver=**new** ChromeDriver();

//load the url

driver.get("http://leaftaps.com/opentaps/control/main");

// maximize the webpage

driver.manage().window().maximize();

//Enter the username

driver.findElement(By.id("username")).sendKeys("DemoSalesManager");

//Enter the password

driver.findElement(By.id("password")).sendKeys("crmsfa");

//click the Login button

driver.findElement(By.className("decorativeSubmit")).click();

}

}

Whenever the element could not found in the, then **NoSuchElementException** will be thrown which is a Webdriver exception.