**CSS selectors for Selenium**

**CSS** is "**Cascading Style Sheets**" and it is defined to display HTML in structured and colorful styles are applied to webpage.

**Selectors** are patterns that match against elements in a tree, and as such form one of several technologies that can be used to select nodes in an XML document.

When we don't have an option to choose Id or Name, we should prefer using **CSS locators** as the best alternative.

The following are the list of object identifier or **locators supported by selenium.**

* id
* Name
* Linktext
* Partial Linktext
* Tag Name
* class name
* CSS
* XPATH

**CSS VS XPATH**

* CSS has more Advantage than Xpath
* CSS is much more faster and simpler than the Xpath.
* In IE Xpath works very slow, where as CSS works faster when compared to Xpath.

In [xpath](http://seleniumeasy.com/selenium-tutorials/xpath-tutorial-for-selenium) we can use in different ways

1. With ID : - //input[@id='email'] or we can also use as //\*[@id='email']

2. With Name - //input[@name='email'] or we can also use as //\*[@name='email']

In [css](http://seleniumeasy.com/selenium-tutorials/css-selectors-tutorial-for-selenium-with-examples)we can use as below :

1. With ID - css=input#email or css=#email  
2. With Name - css=input[name=email] or css=[name=email]

#### **In CSS there are two special characters which has important role to play.**

##### **1. dot (.) refers to class**

##### **2. Hash (#) refers to Id**

Using CSS locators, we can also locate elements with **sub-strings**. Which are really help full when there are dynamically generated ids in webpage.

There are three important special characters:

**1. '^' symbol, represents the starting text in a string.**  
**2. '$' symbol represents the ending text in a string.**  
**3. '\*' symbol represents contains text in a string.**

**CSS Locators for Sub-string matches (Start, end and containing text) in selenium**

**\***- It will find input tag which contains 'id' attribute **containing** 'mai' text. Email contains 'mai' \*/

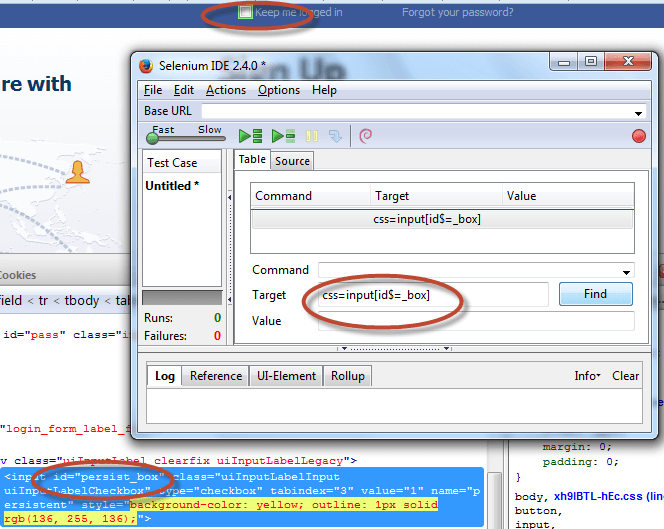
css=input[id\*='mai']

**^**-It will find input tag which contains 'id' attribute **starting** with 'ema' text. Email starts with 'ema' \*/

css=input[id^='ema']

**$**-It will find input tag which contains 'id' attribute **ending** with 'ail' text. Email ends with 'mail' \*/

css=input[id$='mail']



If you remove the symbol and try to find the element with same sub-string, it will display error as "locator not found". We can observe the error in the below screen shot. one with error and the other with success]

