Loactors:

It helps driver to locate the web elements to interact with and to do actions.

As a rule of thumb, your **interrogation strategy** should be in below order:

* First try to use **Id**, **name**, **class**, etc.
* Then, try to interrogate by **CSS**
* Then, use [**XPath**](http://www.swtestacademy.com/xpath-selenium/) to find elements.

Xpath:

Xpath are two types absolute and relative.

Absolute :

It starts with a single slash and provides absolute path of the element.

Ex: /html/body/div[7]/div[3]/div/div[2]/button

Relative XPath:

Starts from the middle of the DOM structure.

//tagName[@attribute=’value’]

We can also use multiple attributes

//input[@type='text' and @name='email']

//input[@type='text' or @name='email'] 🡪 this gets an element if one or the other attributes are matched

We can use below for links

//a[text()='Forgot your password?']

//a[contains(text(),'Forgot your password?')]

Ancestor Xpath: This will identify the ancestor for an element

//input[@name='password']//ancestor::div

Following Xpath: This fetches next div location

//input[@name='password']//following::div

Child element :

//div[@class='ui message']/child::a

Preceding element:

//div[@class='ui message']//preceding::div

Descendant :

Identifies and returns all the element descendants to the current element which means traverse down under the current element’s node.

//div[@id='radio-btn-example']/descendant::label[1]

Following-sibling:

Select the following siblings of the context node.

//\*[@class=’panel-body’]//following-sibling::li

Starts with:

***tag[starts-with( @attribute, ‘starting text’)]***

**Parent XPath Examples**

Returns the parent of the current node as shown in the below example.

**Example**: *.//\*[@id=’get-input’]/button//parent::form*

CSS Selectors:

Basic syntax

### ****Basic Syntax****

|  |  |
| --- | --- |
| **ID** | **#idname** |
| **Class** | **.classname** |

Tag and attribute :

Tag[attribute=’value’]

Ex: select[id='dropdown-class-example']

Contains:

select[id\*='dropdown-clas']

starts With:

**tag[attribute^=’starting text’]**

ex: select[id^='dropdown-clas']

Ends With:

Tag[attribute$=’ending text’]

Tag and ID:

Syntax: Tag#id

Tag and class:

Syntax: Tag.class

### ****Using first-of-type CSS Selector in Selenium****

You can use **“Tag:first-of-type”**. It will select the first tag element.

**Example**

**Syntax**: **.tree-branch>ul>li:first-of-type**

### ****Using last-of-type CSS Selector in Selenium****

You can use **“Tag:last-of-type”**. It will select the last tag element.

**Example**

**Syntax**: **.tree-branch>ul>li:first-of-type**

### ****Using \*:last-of-type CSS Selector in Selenium****

You can use **“\*last-of-type”**. It will select the last child of the parent tag.

**Example**

**Syntax**: **.tree-branch>ul>\*:last-of-type** (Selects the last child of parent tag “ul”.)

### ****Using tag:nth-of-type(****n****) CSS Selector in Selenium****

You can use **“tag:nth-of-type(n)”**. It will select the nth tag element of the list.

**Example**

**Syntax**: **.tree-branch>ul>li:nth-of-type(3)**(Selects 3rd li element.)

### ****Using tag:nth-child(****n****) CSS Selector in Selenium****

You can use **“tag:nth-child(n)”**. It will select the nth child.

**Example**

**Syntax**: **.tree-branch>ul>li:nth-child(3)**(It will select the nth child.)