INTRODUCTION TO HTML

It is used to develop a webpage.

It is made up of tags (inbuild tags).

Tag: Predefined function to perform certain set of actions.

Core Structure of HTML

```
<html>
   <head>
         <title>Selenium</title>
   </head>
   <body>   ⎫
            ⎬   view port area
   </body>  ⎭
</html>
```

We see whatever we write inside Body in View Port area.

Tags:

1. Paired tag

2. Unpaired tag

1. Paired tag:

The tag which has both opening and closing tags is considered as paired tag.

Ex: <html></html>, <body></body>, <p></p> etc.

2. Unpaired tag:

The tag which contains only opening tag is considered as unpaired tag. They are also called as Self closing tags.

Ex: <br>, <img>, <input> etc.

Tag Name:

The very first word after "<" symbol until the first space is called as Tag Name.

“<” →Conical bracket/Angular bracket.

Ex: <div…..></div>

<span class=…..>

<button class=…>

<input type=…..>

Here div, span, button and input are the tag names.

To view tag name of a particular web element--->right click-->inspect.

Attribute:

It provides the additional information about the tag.

Any key value pair inside the opening tag is considered as attribute.

Ex: <input class="username", type="text" ></input>

↓ Attribute value

↓ Attribute Name

Attribute Name: Key of an attribute

Attribute Value: Value of an attribute

Path:

Path is used to find elements in html code.

1. Absolute path

2. Relative path

1. Absolute path

Starting from html tag to the desired element or the required element is called as
Absolute path.
Absolute path starts with "/" (single forward slash).

Syntax:
./html/body/tag_name/child_tag_name
. --> current page/directory/path.

Ex:

```
html
  ↳ body
        ↳ div
             ↳ input
                     ↳ a→desired element
```

path→ ./html/body/div/input/a

Html code for paths

```
<html>
      <head>
               <title> Selenium </title>
       </head>
        <body>
              <div>
                     <input type= "text" value= "A">
                     <input type= "text" value= "B">
              </div>
              <div>
                     <input type= "text" value= "C">
                     <input type= "text" value= "D">
              </div>
              <div>
                     <input type= "text" value= "E">
                     <input type= "text" value= "F">
              </div>
              <div>
                     <a href="Submit">Submit</a>
              </div>
 </body>
```

</html>

1. Absolute path of div

./html/body/div

Where to type this path

Right click on the web element→inspect→click on Ctrl+F in the DOM→Enter the path in search bar.

2. absolute path of div2.

./html/body/div[2]

3. Absolute path of C

./html/body/div[2]/input[1]

4. Absolute path of AB

./html/body/div[1]/input

5. Absolute path of AC

./html/body/div[1]/input[1]|./html/body/div[2]/input[1]

6.Absolute path of BDF

./html/body/div[1]/input[2]|./html/body/div[2]/input[2]|./html/body/div[3]/input[2]

7. Absolute path of AD

./html/body/div[1]/input[1]|./html/body/div[2]/input[2]

8. Absolute path of ADE

./html/body//div[1]/input[1]|./html/body/div[2]/input[2]|./html/body/div[3]/input[1]

9. Absolute path of ACE

./html/body/div[1]/input[1]|./html/body/div[2]/input[1]|./html/body/div[3]/input[1]

10. Absolute path of ACEF

./html/body/div[1]/input[1]|./html/body/div[2]/input[1]|./html/body/div[3]/input[1]|./html/body/div[3]/input[2]

2. Relative path

Relative path starts with "//" (double forward slash) which represents descendants (child, grandchild, great grandchild).

Syntax:
//tag_Name/child_tag_name

Note:
/--->represents immediate child
//--->represents descendants
Here we can jump to any of the grandchildren.
Ex:

1. Relative path for A
//div[1]/input[1]

2. Relative path for D
//div[2]/input[2]

3. Relative path for AC
//div[1]/input[1]|//div[2]/input[1]

4. Relative path for ACF
//div[1]/input[1]|//div[2]/input[1]|//div[3]/input[2]

5. Relative path for CD
//div[2]/input

6. Relative path for ACE
//div/input[1]

7. Relative path for BDF
//div/input[2]

8. Relative path for ABCD
//div[1]/input|//div[2]/input

9. Relative path for BDEF
//div/input[2]|//div[3]/input
or
//div/input[2]|//div[3]/input[1]

10. Relative path for ABCDEF
//input or //div/input

SEARCH CONTEXT METHODS

It is the super most interface of the WebDriver Architectures.

It is used to identify/locate/search elements on the Webpage.

It provides 2 abstract methods.

1. findElement()

2. findElements()

1. findElement(By by):
   It is used to locate/identify one web element at a time.
   It takes argument called By, where we should pass By type object reference.
   Return type is WebElement.
   If we have 100 matching elements-->It returns only the first matching element.
   If there are no matching elements---> It throws "NoSuchElementException".

2. findElements(By by):
   It is used to locate/identify more than 1 web element at a time.
   It takes argument called By, where we should pass By type object reference.
   Return type is List<WebElement>.
   If we have 100 matching elements-->It returns all the 100 web elements at a time.
   If there are no matching elements---> It returns empty list or list with 0 size.

By:

By is a abstract class in Selenium.

It has 8 static methods.

1. id()
2. name()
3. linkText()
4. partialLinkText()
5. tagName()
6. className()
7. cssSelector()
8. xpath()

All these methods accepts String as an argument.

All these 8 methods return By type object reference.

All these methods are considered as locators.

By.id(" ")--->By type object reference

```
driver.findElement(By.id(" "));
driver.findElement(By.name(" "));
driver.findElement(By.linkText(" "));
driver.findElement(By.partialLinkText(" "));
driver.findElement(By.tagName(" "));
driver.findElement(By.className(" "));
driver.findElement(By.cssSelector(" "));
driver.findElement(By.xpath(" "));
```