DOM Manipulation

- DOM is the tree structure of the HTML document. The browser understands the tree structure and provides us the output.
- Js helps to manipulate the DOM tree.
- The document is the object that refers to the webpage.

Fetch elements using TagName:

- We can fetch the elements from HTML file usingTagName.
    - *Example:* **`document.getElementsByTagName (" Tag_Name ");`**
- If two or more elements have the same TagName then we use index numbers to get them.
    - *Example:* Let us assume <h1> tag has hii, hello, bye as values in HTML file.

```
< body >
< h1 > hii < /h1 >      // index 0
< h1 > hello < /h1 >    // index 1
< h1 > bye < /h1 >    // index 2
 < script  src = " js_file_name.js " > < /script >
 </body>
```

JS file:

```
var ele = document.getElementsByTagName (" h1 ");
console.log ( ele [2] );    // < h1 > bye < /h1 >
console.log ( ele [2].innerText );    // bye
console.log ( ele [0] );    // < h1 > hii < /h1 >
```

Fetch elements using ClassName:

- we can fetch elements using ClassName from html document. If we have two or more with same ClassName then we can get the element using index number.

Html file:

```
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Thor</td>
            <td class="age">1500</td>
            <td>Asgard</td>
        </tr>
        <tr>
            <td>Ironman</td>
            <td class="age">46</td>
            <td>Earth</td>
        </tr>
    </tbody>
</table>
```

Js file:

```
var ele = document.getElementsByClassName ( " age " );
 console.log ( ele [1] );    // <td class = "age" > 46 </td>
console.log ( ele [1].innerText );    // 46
```

Fetch elements using Id:

We can fetch elements using Id from an HTML document. If we have two or more with the same Id then we can get the element using an index number.

Html file:

```
< body >
 < p id = "para" > hey LetsUpgraders < /p >
 < /body >
```

Js file:

```
var ele = document.getElementById ( "para" );
console.log ( ele ); // < p id = "para" > hey LetsUpgraders < /p >
console.log ( ele.innetText ); // hey LetsUpgraders
```

- innerText is only for text that is it takes everything as text whereas innerHTML is for both text and HTML that is it converts everything into HTML format in a javascript document.

Functions:

- The function is a block of code that has a name that performs a particular task.
- We can use the function multiple times so that it reduces complexity with less lines of code and saves time.

Non – Parameterized function

Syntax:

```
 function function_name ( )
```

```
  {
  // code here
    }
```

Example:

```
function addition ( )
  {
   var a = 15;
   var b = 8;
     var c = a + b;
    console.log ( c );
   }
  addition ( );    // 23
addition ( );    // 23
```

Parameterized function

Syntax:

```
function function_name ( parameter 1, parameter 2 …….)
  {
   //code
  }
```

Example:

```
function addition ( num1 , num2 )    // num1, num2 are called parameters
   {
 var c = num1 + num2;
console.log ( c );
  }
addition ( 2, 3 );    // 5
  addition ( 7, 8 );    // 15
```

Event Handling

- Any action that a user performs on a web page whether it is a click or double click or selection of text by using mouse or keyword is called event handling.

Example:

Html file:

```
< body >
< button onclick = " dosomething ( ) " > click me < /button >
< /body >
```

Js file:

```
function dosomething ( )
  {
 Console.log ( " hello LetsUpgraders " );
  }
```

- Now when I click on the *click me* button then "*hello LetsUpgraders*" is displayed in the console.