**DOM and Events**

**What is Document Object Model DOM?**

DOM is a cross platform and language independent interface that treats an HTML/XML/SVG document as a tree structure wherein each node is an object representing a part of the document (or you can say each node represents an HTML element).

This interface allows us to create, change or remove elements from the document. The DOM represents the document as nodes and objects; that way, programming languages can interact with the page. Nodes can also have eventhandlers attached to them. Once an event is triggered, the event handler gets executed.

Take a look at this HTML code:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>DOM tree structure</title>

</head>

<body>

<h1>DOM tree structure</h1>

<h2>Learn about the DOM</h2>

</body>

</html>

Our document is called the root node and contains one child node which is the <html> element. The <html> element contains 2 children which are <head> and <body> element.

DOM methods allow programmatic access to the tree (using JS language).

**Fundamental Data Types/Interfaces**

- **Document** : the root document object itself.

- **Node** : Every object located within a document is a node of some kind. In HTML document,an object can be an element node, text node or attr node etc.

- **Element** : The *element* type is based on a *node*. It refers to an element or a node of type *element* returned by a member of DOM API.

- **NodeList** : It is an array of elements, like the kind that is returned by the method document.querySelectorAll().

- **Attr** : When an attribute is returned by a member, it is an object reference that exposes a small special interface for attributes.

**Relation between Objects and Interfaces in DOM**

In a DOM, one object implements either one interface or may be several interfaces. The table object, for e.g, implements a specialized HTMLTableElement interface, which includes such methods as CreateCaption and insertRow. But since it’s an HTML element ultimately, it also implements *Element* interface. And finally, since an HTML element is also, as far as the DOM is concerned, a node in the tree of nodes that make up the object model for an HTML page, the table also implements a more basic *Node* interface, from which *Element* derives.

The document and window objects are the objects whose interfaces you generally use most often in DOM programming. In simple terms, the window object represents something like the browser, and the document object is the root of the document itself. Element inherits from the generic Node interface, and together these two interfaces provide many of the methods and properties you use on individual elements.

**DOM Interfaces**