**DOM**

DOM Stands for Document Object Model

can change,add,remove all the HTML elements,attributes,css styles in the page.

In DOM all the elements are defined as objects

property= can set and get(innerHTML)

method= action can be performed(getElementById)

<htm>

<body>

<p id= "intro">Jokes Over</p>

<p id="demo"></p>

<script>

document.getElementById("demo").innerHTML=document.getElementById("intro").innerHTML;

</script>

</body>

</htm>

Node:

<script>

var para = document.createElement("p");

var node = document.createTextNode("This is new.");

para.appendChild(node);

var element = document.getElementById("div1");

element.appendChild(para);

</script>

EventListener:

<button id="myBtn">Try it</button>

<p id="demo"></p>

<script>

document.getElementById("myBtn").addEventListener("click", displayDate);

function displayDate() {

document.getElementById("demo").innerHTML = Date();

}

</script>

**BOM:**

BOM stands for browser object model

<html>

<body>

<p id="demo"></p>

<script>

document.getElementById("demo").innerHTML =

"Screen height is " + screen.height;

</script>

</body>

</html>

**AJAX**

AJAX = Asynchronous JavaScript And XML.

Update a web page without reloading the page

Send data to a web server - in the background

**How it works:**

Browser:

An event occurs

create an XMLHttpRequest object

send HttpRequest

Server:

Process HTTPRequest

create a response and send data back to the browser

Browser:

Process the returned data using JS

Update page content

XMLHttp:

The XMLHttpRequest object can be used to exchange data with a web server behind the scenes.

This means that it is possible to update parts of a web page, without reloading the whole page.

The readyState property holds the status of the XMLHttpRequest.

The onreadystatechange property defines a function to be executed when the readyState changes.

The status property and the statusText property holds the status of the XMLHttpRequest object.

<html>

    <body>

        <h1>Click to change</h1>

        <p id="demo"></p>

        <button type="button" onclick="loadDoc()"> Change Content</button>

        <script>

            function loadDoc(){

                var hi=new XMLHttpRequest();

                hi.onreadystatechange=function(){

                    if(this.readyState == 4 && this.status==200){

                      document.getElementById("demo").innerHTML=this.responseText

                    }

                }

                hi.open("GET","Hello.txt",true);

                hi.send();

            }

  </script>

    </body>

</html>

**JSON**

JSON stands for JavaScript Object Notation

When exchanging data between a browser and a server, the data can only be text.

JSON is text, and we can convert any JavaScript object into JSON, and send JSON to the server.

We can also convert any JSON received from the server into JavaScript objects.

<!DOCTYPE html>

<html>

<body>

<p id="demo"></p>

<p id="demo2"></p>

<script>

var obj = { name: "Saif", age: 24, city: "Dhaka" };

var myJSON = JSON.stringify(obj);

document.getElementById("demo").innerHTML = myJSON;

document.getElementById("demo2").innerHTML = typeof (myJSON);

</script>

</body>

</html>

Nested objects:

var myObj = {

"name":"John",

"age":30,

"cars": {

"car1":"Toyota",

"car2":"Lamborghini",

"car3":"Range Rover"

}

}

Parsing:

<script>

var txt = '{"name":"Saif", "age":24, "city":"Dhaka"}'

var obj = JSON.parse(txt);

document.getElementById("demo").innerHTML = obj.name + ", " + obj.age;

</script>