Programme	: B. Sc.
Subject	: Computer Science
Semester	: IV
Course Code	: CSS104/CSS105/CSS106
Course Title	: Web Application Development using Flask
Unit IV	: JavaScript
Module Name	: DOM, Introduction to JavaScript, Statements, Syntax, Variables.
Module No	: 14
Presenter	: Ms. Prajoti Chimulkar, Assistant Professor
	St. Xavier's College, Mapusa - Goa.

The HTML DOM (Document Object Model)

- What is the HTML DOM?

The HTML DOM is a standard object model and programming interface for HTML. It defines:

The HTML elements as objects

The properties of all HTML elements

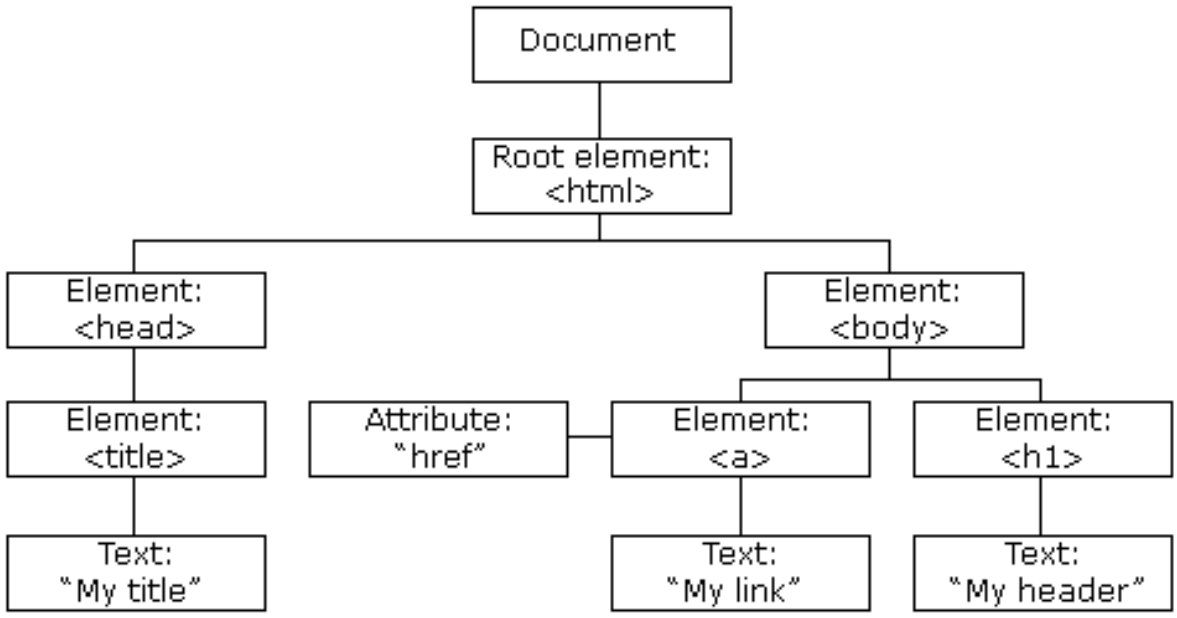
The methods to access all HTML elements

The events for all HTML elements

In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

When a web page is loaded, the browser creates a Document Object Model of the page.The HTML DOM model is constructed as a tree of Objects:

The HTML DOM Tree of Objects



With the object model, JavaScript gets all the power it needs to create dynamic HTML:

It can change all the HTML elements in the page

Change all the HTML attributes in the page

Change all the CSS styles in the page

Remove existing HTML elements and attributes

Add new HTML elements and attributes

React to all existing HTML events in the page

Create new HTML events in the page

HTML DOM Methods :

HTML DOM methods are actions you can perform (on HTML Elements).

HTML DOM properties are values (of HTML Elements) that you can set or change.

The getElementById Method

The most common way to access an HTML element is to use the id of the element.

In the example below the getElementById method used id="demo" to find the element.

The innerHTML Property

The easiest way to get the content of an element is by using the innerHTML property.

The innerHTML property is useful for getting or replacing the content of HTML elements.

The following example changes the content (the innerHTML) of the <p> element with id="demo":

```
<!DOCTYPE html>
<html><body>
<h2>My First Page</h2>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>
</body></html>
```

Output :

My First Page

Hello World!

Note : In the example above, getElementById is a method, while innerHTML is a property.

Changing the HTML Output Stream

In JavaScript, document.write() can be used to write directly to the HTML output stream:

```
<!DOCTYPE html>
<html><body>
<script>
document.write(Date());
</script>
</body></html>
```

Output :

Sat Mar 13 2021 23:21:29 GMT+0530 (India Standard Time)

Changing HTML Content

The easiest way to modify the content of an HTML element is by using the innerHTML property.

To change the content of an HTML element, use this syntax:

```
document.getElementById(id).innerHTML = new HTML
```

This example changes the content of a <p> element:

```
<!DOCTYPE html>
<html><body>
<h2>JavaScript can Change HTML</h2>
<p id="p1">Hello World!</p>
<script>
document.getElementById("p1").innerHTML = "New text!";
</script>
<p>The paragraph above was changed by a script.</p>
</body></html>
```

Output :

JavaScript can Change HTML

New text!

The paragraph above was changed by a script.

This example changes the content of an <h1> element:

```
<!DOCTYPE html>
<html>
<body>
<h1 id="id1">Old Heading</h1>
<script>
var element = document.getElementById("id1");
element.innerHTML = "New Heading";
</script>
<p>JavaScript changed "Old Heading" to "New Heading".</p>
</body>
</html>
```

Changing the Value of an Attribute

To change the value of an HTML attribute, use this syntax:

```
document.getElementById(id).attribute = new value
```

This example changes the value of the src attribute of an <img> element:

```
<!DOCTYPE html>
<html>
<body>
<img id="myImage" src="smiley.gif">
<script>
document.getElementById("myImage").src = "landscape.jpg";
</script>
</body>
</html>
```

Introduction to JavaScript

JavaScript is a cross-platform, object-oriented scripting language developed by Netscape. JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites.

JavaScript is designed for use on web pages and closely integrated with HTML. JavaScript can create applications which run in the browsers such as IE, Opera, FireFox, Google Chrome and other.

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

In HTML, JavaScript code is inserted between <script> and </script> tags. You can place any number of scripts in an HTML document.

Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.

```
<!DOCTYPE html>
<html>
<head><script>
function myFunction() {
document.getElementById("demo").innerHTML = "Paragraph changed."; }
</script></head>
<body>
<h2>JavaScript in Head</h2>
<p id="demo">A Paragraph.</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript in Body</h2>
<p id="demo">A Paragraph.</p>
<button type="button" onclick="myFunction()">Try it</button>
<script>
function myFunction() {
document.getElementById("demo").innerHTML = "Paragraph changed.";}
</script>
</body>
</html>
```

JavaScript Statements :

JavaScript statements are composed of:

Values, Operators, Expressions, Keywords, and Comments.

The statements are executed, one by one, in the same order as they are written.

Each JavaScript statement ends with a semicolon.

Note : JavaScript, however, allows you to omit this semicolon if each of your statements are placed on a separate line.

JavaScript Syntax

JavaScript syntax is the set of rules, how JavaScript programs are constructed:

```
var x, y, z;       // Declare Variables
x = 5; y = 6;      // Assign Values
z = x + y;         // Compute Values
```

JavaScript Operators

JavaScript uses arithmetic operators (+ - * /) to compute values:

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Operators</h2>
<p>JavaScript uses arithmetic operators to compute values (just like algebra).</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = (5 + 6) * 10;
</script>
</body>
</html>
```

JavaScript Comments

Code after double slashes // or between /* and */ is treated as a comment.

Comments are ignored, and will not be executed.

JavaScript is Case Sensitive

All JavaScript identifiers are case sensitive.

The variables lastName and lastname, are two different variables.

JavaScript Variables

JavaScript variables are containers for storing data values.

```
<!DOCTYPE html>
<html><body>
<h2>JavaScript Variables</h2>
<p>In this example, x, y, and z are variables.</p>
<p id="demo"></p>
<script>
```

```
var x = 5;
var y = 6;
var z = x + y;
document.getElementById("demo").innerHTML = "The value of z is: " + z;
</script>
</body></html>
```

JavaScript Identifiers

All JavaScript variables must be identified with unique names.

These unique names are called identifiers.

The general rules for constructing names for variables (unique identifiers) are:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter
- Names can also begin with $ and _ (but we will not use it in this tutorial)
- Names are case sensitive (y and Y are different variables)
- Reserved words (like JavaScript keywords) cannot be used as names

JavaScript Data Types

JavaScript allows you to work with three primitive data types −

Numbers, eg. 123, 120.50 etc.

Strings of text e.g. "This text string" etc.

Boolean e.g. true or false.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Variables</h2>
<p>Strings are written with quotes.</p>
<p>Numbers are written without quotes.</p>
<p id="demo"></p>
<script>
```

```
var pi = 3.14;
var person = "John Dias";
var answer = 'Yes I am!';
document.getElementById("demo").innerHTML = pi + "<br>" + person + "<br>" + answer;
</script>
</body>
</html>
```

Output :

JavaScript Variables

Strings are written with quotes.

Numbers are written without quotes.

3.14

John Dias

Yes I am!