**Chapter 3: Welcome to JavaScript**

**3.1 Entry Point**

Every program needs an entry point. Writing the code directly into the **<script>** tag might cause a problem because as soon as the browser sees the **<script>** tag it will start its execution before it is fully downloaded into the browser. So it is always necessary to use the entry point in the program.

The entry point can be **DOMContentLoaded**, **readyState** or **window.onLoad**

**What is a DOM?**

DOM stands for Document Object Model.

<html>

<head>

<title>My Document</title>

</head>

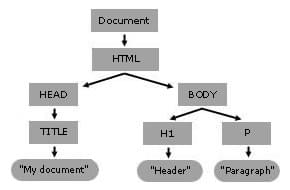
<body>

<h1>Header</h1>

<p>Paragraph</p>

</body>

</html>



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

* JavaScript can change all the HTML elements in the page
* JavaScript can change all the HTML attributes in the page
* JavaScript can change all the CSS styles in the page
* JavaScript can remove existing HTML elements and attributes
* JavaScript can add new HTML elements and attributes
* JavaScript can react to all existing HTML events in the page
* JavaScript can create new HTML events in the page

**DOMContentLoaded:**

The DOMContentLoaded event fires when the initial HTML document has been completely loaded and parsed, without waiting for stylesheets, images and subframes to finish loading.

**Syntax:**

**document.addEventListener(‘DOMContentLoaded’, load);**

**Or**

**document.addEventListener(‘DOMContentLoaded’, (event)=>{});**

**Or**

**document.addEventListener(‘DOMContentLoaded’, (event)=>{**

**console.log(‘DOM fully loaded and parsed’);**

**})**

**readyState:**

The document.readyState property describes the loading state of the document. When the value of this property changes, a readystatechange event fires on the document object.

The value can be:

* **Loading** - the document is still loading
* **Interactive** - the document has finished loading and the document has been parsed but the sub-resources like scripts, images, stylesheets and frames are still loading.
* **Complete** - the document and all sub-resources have finished loading.

**3. Window.onload:**

With this method we can wait until all the media content is downloaded on the browser.

**External javascripts:**

Javascript can be included in the program from an external file through the syntax:

**<script src=”sample.js”></script>**

**Importing and Exporting multiple definitions:**

The **import** and **export** keywords are used to import/export a variable, a class and a function from an external file.

The syntax is **<script type=”module”>**

**Strict Mode:**

Strict mode makes it easier to write “secure” JavaScript. It will throw an error if we assign a value to a variable that is not declared previously.