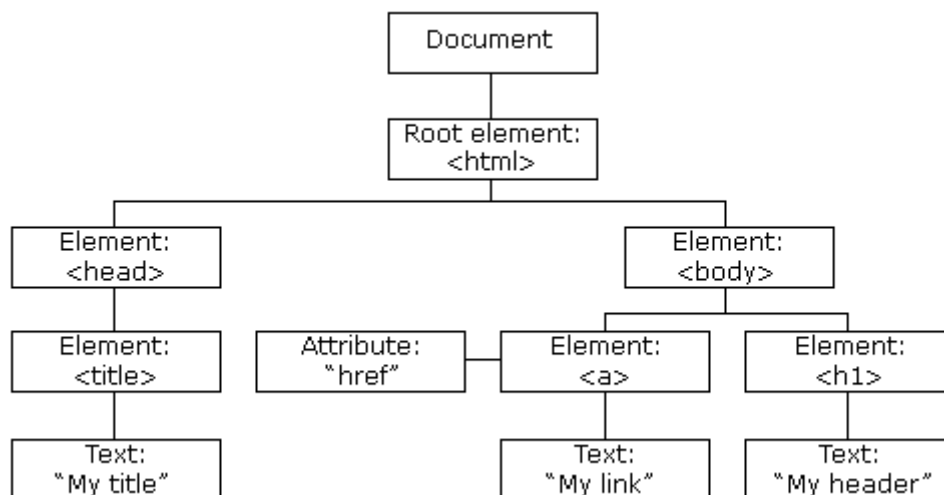


# The HTML DOM (Document Object Model)

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:

- 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

## What is the DOM?

The DOM is a W3C (World Wide Web Consortium) standard.

The DOM defines a standard for accessing documents:

*"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*

The W3C DOM standard is separated into 3 different parts:

- Core DOM - standard model for all document types
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents

## Accessing the DOM

You don't have to do anything special to begin using the DOM. Different browsers have different implementations of the DOM, and these implementations exhibit varying degrees of conformance to the actual DOM standard (a subject we try to avoid in this documentation), but every web browser uses some document object model to make web pages accessible via JavaScript.

When you create a script—whether it's inline in a `<script>` element or included in the web page by means of a script loading instruction—you can immediately begin using the API for the `document` or `window` elements to manipulate the document itself or to get at the children of that document, which are the various elements in the web page. Your DOM programming may be something as simple as the following, which displays an alert message by using the `alert()` function from the `window` object, or it may use more sophisticated DOM methods to actually create new content, as in the longer example below.

This following JavaScript will display an alert when the document is loaded (and when the whole DOM is available for use):

```
<body onload="window.alert('Welcome to my home page!');">
```

## Testing the DOM API

This document provides samples for every interface that you can use in your own web development. In some cases, the samples are complete HTML pages, with the DOM access in a `<script>` element, the interface (e.g, buttons) necessary to fire up the script in a form, and the HTML elements upon which the DOM operates listed as well. When this is the case, you can cut and paste the example into a new HTML document, save it, and run the example from the browser.

There are some cases, however, when the examples are more concise. To run examples that only demonstrate the basic relationship of the interface to the HTML elements, you may want to set up a test page in which interfaces can be easily accessed from scripts. The following very simple web page provides a `<script>` element in the header in which you can place functions that test the interface, a few HTML elements with attributes that you can retrieve, set, or otherwise manipulate, and the web user interface necessary to call those functions from the browser.

You can use this test page or create a similar one to test the DOM interfaces you are interested in and see how they work on the browser platform. You can update the contents of the `test()` function as needed, create more buttons, or add elements as necessary.

What DOM is not ?

- 1.The Document Object Model is not a binary description where it does not define any binary source code in its interfaces.
- 2.The Document Object Model is not used to describe objects in XML or HTML whereas the DOM describes XML and HTML documents as objects.
- 3.The Document Object Model is not represented by set of data structures; it is an interface which specifies object representation.
- 4.The Document Object Model does not show criticality of objects in documents i.e it doesn't have information about which object in document is appropriate to the context and which is not.