The HTML DOM (Document Object Model)

When a web page is loaded, the browser creates a **D**ocument **O**bject **M**odel 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

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.**

To understand this more clearly, let's consider the following simple HTML document:

#### Example

<!DOCTYPE html>

<html>

<head>

<title>My Page</title>

</head>

<body>

<h1>Mobile OS</h1>

<ul>

<li>Android</li>

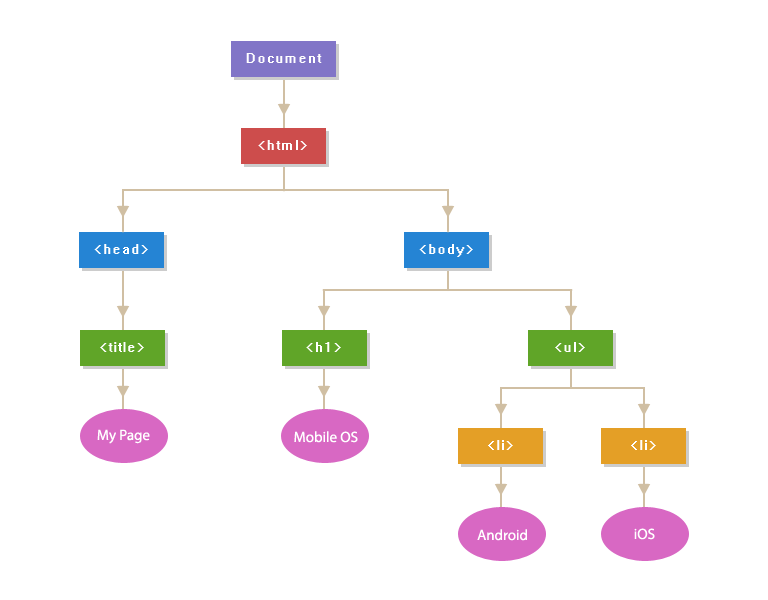
<li>iOS</li>

</ul>

</body>

</html>

The above HTML document can be represented by the following DOM tree:



The above diagram demonstrates the parent/child relationships between the nodes. The topmost node i.e. the Document node is the root node of the DOM tree, which has one child, the <html> element. Whereas, the <head> and <body> elements are the child nodes of the <html> parent node.

The <head> and <body> elements are also siblings since they are at the same level. Further, the text content inside an element is a child node of the parent element. So, for example, "Mobile OS" is considered as a child node of the <h1> that contains it, and so on.

[Comments](https://www.tutorialrepublic.com/javascript-tutorial/javascript-syntax.php#comments) inside the HTML document are nodes in the DOM tree as well, even though it doesn't affect the visual representation of the document in any way. Comments are useful for documenting the code, however, you will rarely need to retrieve and manipulate them.

HTML attributes such as id, class, title, style, etc. are also considered as nodes in DOM hierarchy but they don't participate in parent/child relationships like the other nodes do. They are accessed as properties of the element node that contains them.

Each element in an HTML document such as image, hyperlink, form, button, heading, paragraph, etc. is represented using a JavaScript object in the DOM hierarchy, and each object contains properties and methods to describe and manipulate these objects. For example, the style property of the DOM elements can be used to [get or set the inline style of an element](https://www.tutorialrepublic.com/javascript-tutorial/javascript-dom-styling.php).