Document Object Model in JavaScript

**Summary**: in this tutorial, you will learn about the Document Object Model in JavaScript.

What is Document Object Model (DOM)

The Document Object Model (DOM) is an application programming interface (API) for manipulating HTML documents.

The DOM represents an HTML document as a tree of nodes. The DOM provides functions that allow you to add, remove, and modify parts of the document effectively.

Note that the DOM is cross-platform and language-independent ways of manipulating HTML and XML documents.

A document as a hierarchy of nodes

The DOM represents an HTML document as a hierarchy of nodes. Consider the following HTML document:

**<html>**

**<head>**

**<title>**JavaScript DOM**</title>**

**</head>**

**<body>**

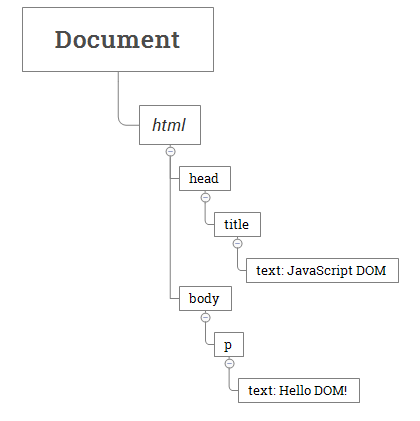
**<p>**Hello DOM!**</p>**

**</body>**

**</html>**

Code language: HTML, XML (xml)

The following tree represents the above HTML document:



In this DOM tree, the document is the root node. The root node has one child node which is the <html> element. The <html> element is called the *document element*.

Each document can have only one document element. In an HTML document, the document element is the <html> element. Each markup can be represented by a node in the tree.

Node Types

Each node in the DOM tree is identified by a node type. JavaScript uses integer numbers to determine the node types. The following table illustrates the node type constants:

| **Constant** | **Value** | **Description** |
| --- | --- | --- |
| Node.ELEMENT\_NODE | 1 | An Element node like <p> or <div>. |
| Node.TEXT\_NODE | 3 | The actual Text inside an Element or Attr. |
| Node.CDATA\_SECTION\_NODE | 4 | A CDATASection, such as <!CDATA[[ … ]]>. |
| Node.PROCESSING\_INSTRUCTION\_NODE | 7 | A ProcessingInstruction of an XML document, such as <?xml-stylesheet … ?>. |
| Node.COMMENT\_NODE | 8 | A Comment node, such as <!-- … -->. |
| Node.DOCUMENT\_NODE | 9 | A Document node. |
| Node.DOCUMENT\_TYPE\_NODE | 10 | A DocumentType node, such as <!DOCTYPE html>. |
| Node.DOCUMENT\_FRAGMENT\_NODE | 11 | A DocumentFragment node. |

To get the type of node, you use the nodeType property:

node.nodeType

Code language: CSS (css)

You can compare the nodeType property with the above constants to determine the node type. For example:

if (node.nodeType == Node.ELEMENT\_NODE) {

*// node is the element node*

}

Code language: JavaScript (javascript)

The nodeName and nodeValue properties

A node has two important properties: nodeName and nodeValue that provide specific information about the node.

The values of these properties depend on the node type. For example, if the node type is the element node, the nodeName is always the same as the element’s tag name and nodeValue is always null.

For this reason, it’s better to test node type before using these properties:

if (node.nodeType == Node.ELEMENT\_NODE) {

let name = node.nodeName; *// tag name like <p>*

}

Code language: JavaScript (javascript)

Node and Element

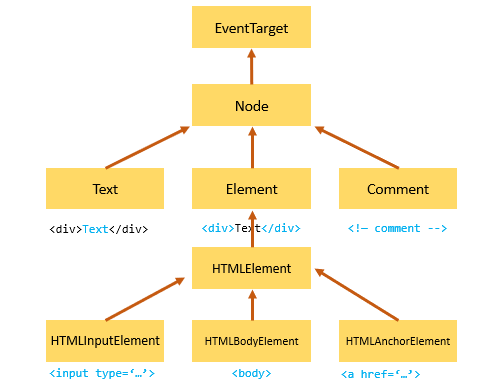
Sometimes it’s easy to confuse between the Node and the Element.

A node is a generic name of any object in the DOM tree. It can be any built-in DOM element such as the document. Or it can be any HTML tag specified in the HTML document like <div> or <p>.

An element is a node with a specific node type Node.ELEMENT\_NODE, which is equal to 1.

In other words, the node is the generic type of element. The element is a specific type of the node with the node type Node.ELEMENT\_NODE.

The following picture illustrates the relationship between the Node and Element types:



Note that the [getElementById()](https://www.javascripttutorial.net/javascript-dom/javascript-getelementbyid/) and [querySelector()](https://www.javascripttutorial.net/javascript-dom/javascript-queryselector/) returns an object with the Element type while [getElementsByTagName()](https://www.javascripttutorial.net/javascript-dom/javascript-getelementsbytagname/) or [querySelectorAll()](https://www.javascripttutorial.net/javascript-dom/javascript-queryselector/) returns NodeList which is a collection of nodes.

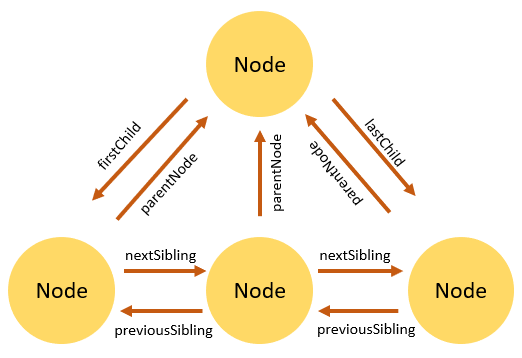
Node Relationships

Any node has relationships to other nodes in the DOM tree. The relationships are the same as the ones described in a traditional family tree.

For example, <body> is a [child node](https://www.javascripttutorial.net/javascript-dom/javascript-get-child-element/) of the <html> node, and <html> is the [parent](https://www.javascripttutorial.net/javascript-dom/javascript-get-parent-element-parentnode/) of the <body> node.

The <body> node is the [sibling](https://www.javascripttutorial.net/javascript-dom/javascript-siblings/) of the <head> node because they share the same immediate parent, which is the <html> element.

The following picture illustrates the relationships between nodes:



Summary

* An HTML or XML document can be represented as a tree of nodes, like a traditional family tree.
* Each markup can be represented as a node with a specific node type.
* Element is a specific type of node with the node type Node.ELEMENT\_NODE.
* In the DOM tree, a node has relationships with other nodes.