DOM Javascript HTML

The Document Object Model (DOM) is a programming interface for HTML and XML documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects. That way, programming languages can connect to the page.

A Web page is a document. This document can be either displayed in the browser window or as the HTML source. But it is the same document in both cases. The Document Object Model (DOM) represents that same document so it can be manipulated. The DOM is an object-oriented representation of the web page, which can be modified with a scripting language such as JavaScript.

The W3C DOM and WHATWG DOM standards are implemented in most modern browsers. Many browsers extend the standard, so care must be exercised when using them on the web where documents may be accessed by various browsers with different DOMs

All of the properties, methods, and events available for manipulating and creating web pages are organized into objects (for example, the document object that represents the document itself, the table object that implements the special HTMLTableElement DOM interface for accessing HTML tables, and so forth). This documentation provides an object-by-object reference to the DOM.

The modern DOM is built using multiple APIs that work together. The core DOM defines the objects that fundamentally describe a document and the objects within it. This is expanded upon as needed by other APIs that add new features and capabilities to the DOM. For example, the HTML DOM API adds support for representing HTML documents to the core DOM

The short example above, like nearly all of the examples in this reference, is JavaScript. That is to say, it's written in JavaScript, but it uses the DOM to access the document and its elements. The DOM is not a programming language, but without it, the JavaScript language wouldn't have any model or notion of web pages, HTML documents, XML documents, and their component parts (e.g. elements). Every element in a document—the document as a whole, the head, tables within the document, table headers, text within the table cells—is part of the document object model for that document, so they can all be accessed and manipulated using the DOM and a scripting language like JavaScript.

In the beginning, JavaScript and the DOM were tightly intertwined, but eventually, they evolved into separate entities. The page content is stored in the DOM and may be accessed and manipulated via JavaScript, so that we may write this approximative equation:

API = DOM + JavaScript

The DOM was designed to be independent of any particular programming language, making the structural representation of the document available from a single, consistent API

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 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.

The HTML DOM Tree of Objects

