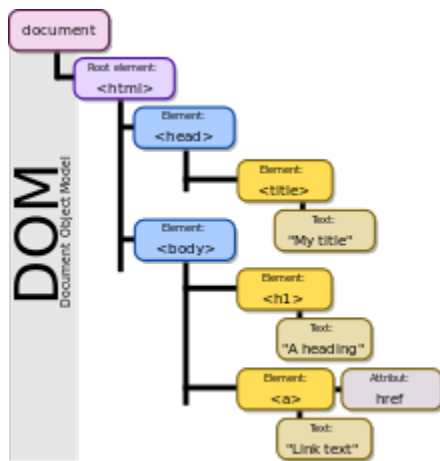


en.wikipedia.org

Document Object Model - Wikipedia

7-9 minutes

Document Object Model



Example of DOM hierarchy in an HTML document

First published October 1, 1998; 18 years ago (1998-10-01)

DOM4^[1]

Latest version November 19, 2015; 16 months ago (2015-11-19)

Organization [W3C](#), [WHATWG](#)

Base standards [WHATWG DOM Living Standard](#)
[W3C DOM4](#)

Abbreviation DOM

The **Document Object Model (DOM)** is a [cross-platform](#) and

[language](#)-independent [application programming interface](#) that treats an [HTML](#), [XHTML](#), or [XML](#) document as a [tree structure](#) wherein each [node](#) is an [object](#) representing a part of the document. The objects can be manipulated programmatically and any visible changes occurring as a result may then be reflected in the display of the document.^[2]

The principal standardization of DOM was handled by the [W3C](#), which last developed a recommendation in 2004. [WHATWG](#) took over development of the standard, publishing it as a [living document](#). The W3C now publishes stable snapshots of the WHATWG standard.

History^{[[edit](#)]}

The history of the Document Object Model is intertwined with the history of the "[browser wars](#)" of the late 1990s between [Netscape Navigator](#) and [Microsoft Internet Explorer](#), as well as with that of [JavaScript](#) and [JScript](#), the first [scripting languages](#) to be widely implemented in the [layout engines](#) of [web browsers](#).

JavaScript was released by [Netscape Communications](#) in 1995 within Netscape Navigator 2.0. Netscape's competitor, [Microsoft](#), released [Internet Explorer 3.0](#) the following year with a [port](#) of JavaScript called JScript. JavaScript and JScript let [web developers](#) create web pages with [client-side](#) interactivity. The limited facilities for detecting user-generated [events](#) and modifying the HTML document in the first

generation of these languages eventually became known as "DOM Level 0" or "Legacy DOM." No independent standard was developed for DOM Level 0, but it was partly described in the specifications for [HTML 4](#).

Legacy DOM was limited in the kinds of [elements](#) that could be accessed. [Form](#), [link](#) and image elements could be referenced with a hierarchical name that began with the root document object. A hierarchical name could make use of either the names or the sequential index of the traversed elements. For example, a [form input element](#) could be accessed as either `document.formName.inputName` or `document.forms[0].elements[0]`.

The Legacy DOM enabled client-side form validation and the popular "[rollover](#)" effect.

In 1997, Netscape and Microsoft released version 4.0 of Netscape Navigator and Internet Explorer respectively, adding support for [Dynamic HTML](#) (DHTML) functionality enabling changes to a loaded HTML document. DHTML required extensions to the rudimentary document object that was available in the Legacy DOM implementations. Although the Legacy DOM implementations were largely compatible since JScript was based on JavaScript, the DHTML DOM extensions were developed in parallel by each browser maker and remained incompatible. These versions of the DOM became known as the "Intermediate DOM."

After the standardization of [ECMAScript](#), the [W3C](#) DOM

Working Group began drafting a standard DOM specification. The completed specification, known as "DOM Level 1", was recommended by W3C in late 1998. By 2005, large parts of W3C DOM were well-supported by common ECMAScript-enabled browsers, including [Microsoft Internet Explorer version 6](#) (from 2001), [Opera](#), [Safari](#) and [Gecko](#)-based browsers (like [Mozilla](#), [Firefox](#), [SeaMonkey](#) and [Camino](#)).

Standards[\[edit\]](#)

The [W3C](#) DOM Working Group published its final recommendation and subsequently disbanded in 2004. Development efforts migrated to the [WHATWG](#), which continues to maintain a living standard.^[3] In 2009, the Web Applications group reorganized DOM activities at the W3C.^[4] In 2013, due to a lack of progress and the impending release of [HTML5](#), the DOM Level 4 specification was reassigned to the [HTML Working Group](#) to expedite its completion.^[5] Meanwhile, in 2015, the Web Applications group was disbanded and DOM stewardship passed to the Web Platform group.^[6] Beginning with the publication of DOM Level 4 in 2015, the W3C creates new recommendations based on snapshots of the WHATWG standard.

- DOM Level 1 provided a complete model for an entire HTML or [XML](#) document, including the means to change any portion of the document.
- DOM Level 2 was published in late 2000. It introduced the

getElementById function as well as an [event model](#) and support for [XML namespaces](#) and CSS.

- DOM Level 3, published in April 2004, added support for [XPath](#) and keyboard [event handling](#), as well as an interface for [serializing](#) documents as XML.
- DOM Level 4 was published in 2015. It is a snapshot of the WHATWG living standard.^[7]

Applications^[edit]

Web browsers^[edit]

To [render](#) a document such as an HTML page, most web browsers use an internal model similar to the **DOM**. The nodes of every document are organized in a [tree structure](#), called the *DOM tree*, with the topmost node named as "Document object". When an HTML page is rendered in browsers, the browser downloads the HTML into local memory and automatically parses it to display the page on screen. The DOM is also the way that JavaScript transmits the state of the browser in HTML pages.

JavaScript^[edit]

When a web page is loaded, the browser creates a Document Object Model of the page.

With the object model, JavaScript is fully enabled to create dynamic HTML:^[8]

- JavaScript can add, change, and remove all of the HTML elements and attributes in the page.
- JavaScript can change all of the CSS styles in the page.
- JavaScript can react to all of the existing events in the page.
- JavaScript can create new events within the page.

Implementations[\[edit\]](#)

Because DOM supports navigation in any direction (e.g., parent and previous sibling) and allows for arbitrary modifications, an implementation must at least buffer the document that has been read so far (or some parsed form of it).^{[\[citation needed\]](#)}

Layout engines[\[edit\]](#)

Web browsers rely on layout engines to parse HTML into a DOM. Some layout engines, such as [Trident/MSHTML](#), are associated primarily or exclusively with a particular browser, such as Internet Explorer. Others, including [Blink](#), [WebKit](#), and [Gecko](#), are shared by a number of browsers, such as [Google Chrome](#), [Opera](#), [Safari](#), and [Firefox](#). The different layout engines implement the DOM standards to varying degrees of compliance.

Libraries[\[edit\]](#)

DOM implementations:

- [libxml2](#)
- [MSXML](#)
- [Xerces](#) is a collection of DOM implementations written in C++, Java and Perl
- XML for <SCRIPT> is a JavaScript-based DOM implementation^[9]

APIs that expose DOM implementations:

- [JAXP](#) (Java API for XML Processing) is an API for accessing DOM providers
- [Lazarus](#) ([Free Pascal](#) IDE) contains two variants of the DOM - with UTF-8 and ANSI format

Inspection tools:

- [DOM Inspector](#) is a web developer tool

Sources^{[[edit](#)]}

- Flanagan, David (2006). *JavaScript: The Definitive Guide*. O'Reilly & Associates. pp. 312–313. [ISBN 0-596-10199-6](#).
- Koch, Peter-Paul (May 14, 2001). ["The Document Object Model: an Introduction"](#). *Digital Web Magazine*. Retrieved January 10, 2009.
- Le Hégarét, Philippe (2002). ["The W3C Document Object Model \(DOM\)"](#). *World Wide Web Consortium*. Retrieved January 10, 2009.

- Guisset, Fabian. *["What does each DOM Level bring?"](#)*. Mozilla Developer Center. Mozilla Project. Retrieved January 10, 2009.

References[\[edit\]](#)

External links[\[edit\]](#)

- [DOM Living Standard](#) by the WHATWG
- [Original W3C DOM hub](#) by the W3C DOM Working Group (outdated)
- [Web Platform Working Group](#) (current steward of W3C DOM)