

HTML

Hypertext Markup Language (HTML) is the standard [markup language](#) for documents designed to be displayed in a [web browser](#). It can be assisted by technologies such as [Cascading Style Sheets \(CSS\)](#) and [scripting languages](#) such as [JavaScript](#).

HTML

(Hypertext Markup Language)



The official logo of the latest version, [HTML5](#)^[1]

Filename extension	<div>.html</div> <div>.htm</div>
Internet media type	text/html
Type code	TEXT
Developed by	WHATWG
Initial release	1993
Latest release	<div>Living Standard ↗</div> <div>(2020)</div>
Type of format	Document file format
Container for	HTML elements
Contained by	Web browser
Extended from	SGML
Extended to	XHTML
Open format?	Yes
Website	html.spec.whatwg.org ↗

[Web browsers](#) receive HTML documents from a [web server](#) or from local storage and [render](#) the documents into multimedia web pages. HTML describes the structure of a [web page semantically](#) and originally included cues for the appearance of the document.

[HTML elements](#) are the building blocks of HTML pages. With HTML constructs, [images](#) and other objects such as [interactive forms](#) may be embedded into the rendered page. HTML provides a means to create [structured documents](#) by denoting structural [semantics](#) for text such as headings, paragraphs, lists, [links](#), quotes and other items. HTML elements are delineated by *tags*, written using [angle brackets](#). Tags such as `` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a [scripting language](#) such as [JavaScript](#), which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The [World Wide Web Consortium](#) (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.^[2]

History

Development



[Tim Berners-Lee](#) in April 2009

In 1980, physicist [Tim Berners-Lee](#), a contractor at [CERN](#), proposed and prototyped [ENQUIRE](#), a system for CERN researchers to use and share documents. In 1989, Berners-Lee wrote a memo proposing an [Internet](#)-based [hypertext](#) system.^[3] Berners-Lee specified HTML and wrote the browser and server software in late 1990. That year, Berners-Lee and CERN data systems engineer [Robert Cailliau](#) collaborated on a joint request for funding, but the project was not formally adopted by CERN. In his personal notes^[4] from 1990 he listed^[5] "some of the many areas in which hypertext is used" and put an encyclopedia first.

The first publicly available description of HTML was a document called "[HTML Tags](#)" [🔗](#), first mentioned on the Internet by Tim Berners-Lee in late 1991.^{[6][7]} It describes 18 elements comprising the initial, relatively simple design of HTML. Except for the hyperlink tag, these were strongly influenced by [SGMLguid](#), an in-house [Standard Generalized Markup Language](#) (SGML)-based documentation format at CERN. Eleven of these elements still exist in HTML 4.^[8]

HTML is a [markup language](#) that [web browsers](#) use to interpret and [compose](#) text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of [CSS](#). Many of the text elements are found in the 1988 ISO technical report TR 9537 *Techniques for using SGML*, which in turn covers the features of early text formatting languages such as that used by the [RUNOFF command](#) developed in the early 1960s for the [CTSS](#) (Compatible Time-Sharing System) operating system: these formatting commands were derived from the commands used by typesetters to manually format documents. However, the SGML concept of generalized markup is based on elements (nested annotated ranges with attributes) rather than merely print effects, with also the separation of structure and markup; HTML has been progressively moved in this direction with CSS.

Berners-Lee considered HTML to be an application of SGML. It was formally defined as such by the [Internet Engineering Task Force](#) (IETF) with the mid-1993 publication of the first proposal for an HTML specification, the "Hypertext Markup Language (HTML)" Internet Draft by Berners-Lee and [Dan Connolly](#), which included an SGML [Document type definition](#) to define the grammar.^{[9][10]} The draft expired after six months, but was notable for its acknowledgment of the [NCSA Mosaic](#) browser's custom tag for embedding in-line images, reflecting the IETF's philosophy of basing standards on successful prototypes. Similarly, [Dave Raggett](#)'s competing Internet-Draft, "HTML+ (Hypertext Markup Format)", from late 1993, suggested standardizing already-implemented features like tables and fill-out forms.^[11]

After the HTML and HTML+ drafts expired in early 1994, the IETF created an HTML Working Group, which in 1995 completed "HTML 2.0", the first HTML specification intended to be treated as a standard against which future implementations should be based.^[12]

Further development under the auspices of the IETF was stalled by competing interests. Since 1996, the HTML specifications have been maintained, with input from commercial software vendors, by the [World Wide Web Consortium](#) (W3C).^[13] However, in 2000, HTML also became an international standard ([ISO/IEC 15445:2000](#)). HTML 4.01 was published in late 1999, with further errata published

through 2001. In 2004, development began on HTML5 in the [Web Hypertext Application Technology Working Group](#) (WHATWG), which became a joint deliverable with the W3C in 2008, and completed and standardized on 28 October 2014.^[14]

HTML versions timeline

HTML 2

November 24, 1995

HTML 2.0 was published as [RFC 1866](#). Supplemental [RFCs](#) added capabilities:

- November 25, 1995: [RFC 1867](#) (form-based file upload)
- May 1996: [RFC 1942](#) (tables)
- August 1996: [RFC 1980](#) (client-side image maps)
- January 1997: [RFC 2070](#) (internationalization)

HTML 3

January 14, 1997

HTML 3.2^[15] was published as a [W3C Recommendation](#).

It was the first version developed and standardized exclusively by the W3C, as the IETF had closed its HTML Working Group on September 12, 1996.^[16]

Initially code-named "Wilbur",^[17] HTML 3.2 dropped math formulas entirely, reconciled overlap among various proprietary extensions and adopted most of [Netscape's](#)

visual markup tags. Netscape's [blink element](#) and [Microsoft's marquee element](#) were omitted due to a mutual agreement between the two companies.^[13] A markup for mathematical formulas similar to that in HTML was not standardized until 14 months later in [MathML](#).

HTML 4

December 18, 1997

HTML 4.0^[18] was published as a W3C Recommendation. It offers three variations:

- Strict, in which deprecated elements are forbidden
- Transitional, in which deprecated elements are allowed
- Frameset, in which mostly only [frame](#) related elements are allowed.

Initially code-named "Cougar",^[17] HTML 4.0 adopted many browser-specific element types and attributes, but at the same time sought to phase out Netscape's visual markup features by marking them as [deprecated](#) in favor of style sheets. HTML 4 is an SGML application conforming to ISO 8879 – SGML.^[19]

April 24, 1998

HTML 4.0^[20] was reissued with minor edits without incrementing the version number.

December 24, 1999

HTML 4.01^[21] was published as a W3C Recommendation. It offers the same three variations as

HTML 4.0 and its last [errata](#) were published on May 12, 2001.

May 2000

ISO/IEC 15445:2000^{[22][23]} ("ISO HTML", based on HTML 4.01 Strict) was published as an ISO/IEC international standard. In the ISO this standard falls in the domain of the [ISO/IEC JTC1/SC34](#) (ISO/IEC Joint Technical Committee 1, Subcommittee 34 – Document description and processing languages).^[22]

After HTML 4.01, there was no new version of HTML for many years as development of the parallel, XML-based language XHTML occupied the W3C's HTML Working Group through the early and mid-2000s.

HTML 5

October 28, 2014

HTML5^[24] was published as a W3C Recommendation.^[25]

November 1, 2016

HTML 5.1^[26] was published as a W3C Recommendation.^{[27][28]}

December 14, 2017

HTML 5.2^[29] was published as a W3C Recommendation.^{[30][31]}

HTML draft version timeline

October 1991

HTML Tags,^[6] an informal CERN document listing 18 HTML tags, was first mentioned in public.

June 1992

First informal draft of the HTML DTD,^[32] with seven^{[33][34][35]} subsequent revisions (July 15, August 6, August 18, November 17, November 19, November 20, November 22)

November 1992

HTML DTD 1.1 (the first with a version number, based on RCS revisions, which start with 1.1 rather than 1.0), an informal draft^[35]

June 1993

Hypertext Markup Language^[36] was published by the IETF IIR Working Group as an Internet Draft (a rough proposal for a standard). It was replaced by a second version^[37] one month later.

November 1993

HTML+ [↗](#) was published by the IETF as an Internet Draft and was a competing proposal to the Hypertext Markup Language draft. It expired in July 1994.^[38]

November 1994

First draft (revision 00) of HTML 2.0 published by IETF itself^[39] (called as "HTML 2.0" from revision 02^[40]), that finally led to publication of [RFC 1866](#) [↗](#) in November 1995.^[41]

April 1995 (authored March 1995)

HTML 3.0^[42] was proposed as a standard to the IETF, but the proposal expired five months later (28 September 1995)^[43] without further action. It included many of the capabilities that were in Raggett's HTML+ proposal, such

as support for tables, text flow around figures and the display of complex mathematical formulas.^[43]

W3C began development of its own [Arena browser](#) as a [test bed](#) for HTML 3 and Cascading Style Sheets,^{[44][45][46]} but HTML 3.0 did not succeed for several reasons. The draft was considered very large at 150 pages and the pace of browser development, as well as the number of interested parties, had outstripped the resources of the IETF.^[13] Browser vendors, including Microsoft and Netscape at the time, chose to implement different subsets of HTML 3's draft features as well as to introduce their own extensions to it.^[13] (see [Browser wars](#)). These included extensions to control stylistic aspects of documents, contrary to the "belief [of the academic engineering community] that such things as text color, background texture, font size and font face were definitely outside the scope of a language when their only intent was to specify how a document would be organized."^[13] Dave Raggett, who has been a W3C Fellow for many years, has commented for example: "To a certain extent, Microsoft built its business on the Web by extending HTML features."^[13]



Logo of HTML5

January 2008

[HTML5](#) was published as a [Working Draft](#) by the W3C.^[47] Although its syntax closely resembles that of [SGML](#), [HTML5](#) has abandoned any attempt to be an SGML application and has explicitly defined its own "html" serialization, in addition to an alternative XML-based XHTML5 serialization.^[48]

2011 HTML5 – Last Call

On 14 February 2011, the W3C extended the charter of its HTML Working Group with clear milestones for HTML5. In May 2011, the working group advanced HTML5 to "Last Call", an invitation to communities inside and outside W3C to confirm the technical soundness of the specification. The W3C developed a comprehensive test suite to achieve broad interoperability for the full specification by 2014, which was the target date for recommendation.^[49] In January 2011, the WHATWG renamed its "HTML5" living standard to "HTML". The W3C nevertheless continues its project to release HTML5.^[50]

2012 HTML5 – Candidate Recommendation

In July 2012, WHATWG and [W3C](#) decided on a degree of separation. W3C will continue the HTML5 specification work, focusing on a single definitive standard, which is considered as a "snapshot" by WHATWG. The WHATWG organization will continue its work with HTML5 as a "Living Standard". The concept of a living standard is that it is never complete and is always being updated and improved. New features can be added but functionality will not be removed.^[51]

In December 2012, W3C designated HTML5 as a Candidate Recommendation.^[52] The criterion for advancement to [W3C Recommendation](#) is "two 100% complete and fully interoperable implementations".^[53]

2014 HTML5 – Proposed Recommendation and Recommendation

In September 2014, W3C moved HTML5 to Proposed Recommendation.^[54]

On 28 October 2014, HTML5 was released as a stable W3C Recommendation,^[55] meaning the specification process is complete.^[56]

XHTML versions

XHTML is a separate language that began as a reformulation of HTML 4.01 using [XML](#) 1.0. It is no longer being developed as a separate standard.

- XHTML 1.0 was published as a W3C Recommendation on January 26, 2000,^[57] and was later revised and republished

on August 1, 2002. It offers the same three variations as HTML 4.0 and 4.01, reformulated in XML, with minor restrictions.

- XHTML 1.1^[58] was published as a W3C Recommendation on May 31, 2001. It is based on XHTML 1.0 Strict, but includes minor changes, can be customized, and is reformulated using modules in the W3C recommendation "Modularization of XHTML", which was published on April 10, 2001.^[59]
- XHTML 2.0 was a working draft, work on it was abandoned in 2009 in favor of work on [HTML5](#) and [XHTML5](#).^{[60][61][62]} XHTML 2.0 was incompatible with XHTML 1.x and, therefore, would be more accurately characterized as an XHTML-inspired new language than an update to XHTML 1.x.
- An XHTML syntax, known as "XHTML5.1", is being defined alongside [HTML5](#) in the HTML5 draft.^[63]

Transition of HTML Publication to WHATWG

On 28 May 2019, the W3C announced that WHATWG would be the sole publisher of the HTML and DOM standards.^{[64][65][66][67]} The W3C and WHATWG had been publishing competing standards since 2012. While the W3C standard was identical to the WHATWG in 2007 the standards have since progressively diverged due to different design

decisions.^[68] The WHATWG "Living Standard" had been the de facto web standard for some time.^[69]

Markup

HTML markup consists of several key components, including those called *tags* (and their *attributes*), character-based *data types*, *character references* and *entity references*. HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some represent *empty elements* and so are unpaired, for example ``. The first tag in such a pair is the *start tag*, and the second is the *end tag* (they are also called *opening tags* and *closing tags*).

Another important component is the HTML *document type declaration*, which triggers *standards mode* rendering.

The following is an example of the classic "Hello, World!" program:

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <div>
```

```
<p>Hello world!</p>
</div>
</body>
</html>
```

The text between `<html>` and `</html>` describes the web page, and the text between `<body>` and `</body>` is the visible page content. The markup text `<title>This is a title</title>` defines the browser page title, and the tag `<div>` defines a division of the page used for easy styling.

The Document Type Declaration `<!DOCTYPE html>` is for HTML5. If a declaration is not included, various browsers will revert to "quirks mode" for rendering.^[70]

Elements

HTML documents imply a structure of nested [HTML elements](#). These are indicated in the document by HTML *tags*, enclosed in angle brackets thus: `<p>`.^[71]

In the simple, general case, the extent of an element is indicated by a pair of tags: a "start tag" `<p>` and "end tag" `</p>`. The text content of the element, if any, is placed between these tags.

Tags may also enclose further tag markup between the start and end, including a mixture of tags and text. This indicates further (nested) elements, as children of the parent element.

The start tag may also include *attributes* within the tag. These indicate other information, such as identifiers for sections within the document, identifiers used to bind style information to the presentation of the document, and for some tags such as the `` used to embed images, the reference to the image resource in the format like this: ``

Some elements, such as the *line break* `
`, or `
` do not permit *any* embedded content, either text or further tags. These require only a single empty tag (akin to a start tag) and do not use an end tag.

Many tags, particularly the closing end tag for the very commonly used paragraph element `<p>`, are optional. An HTML browser or other agent can infer the closure for the end of an element from the context and the structural rules defined by the HTML standard. These rules are complex and not widely understood by most HTML coders.

The general form of an HTML element is therefore: `<tag attribute1="value1" attribute2="value2">'content'</tag>`. Some HTML elements are defined as *empty elements* and take the

form `<tag attribute1="value1" attribute2="value2">`. Empty elements may enclose no content, for instance, the `
` tag or the inline `` tag. The name of an HTML element is the name used in the tags. Note that the end tag's name is preceded by a slash character, `/`, and that in empty elements the end tag is neither required nor allowed. If attributes are not mentioned, default values are used in each case.

Element examples

Header of the HTML document: `<head>...</head>`. The title is included in the head, for example:

```
<head>
  <title>The Title</title>
  <link rel="stylesheet"
href="stylebyjimbowales.css" /> <!--
Imports Stylesheets -->
</head>
```

Headings: HTML headings are defined with the `<h1>` to `<h6>` tags with H1 being the highest (or most important) level and H6 the least:

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
```

```
<h3>Heading level 3</h3>
<h4>Heading level 4</h4>
<h5>Heading level 5</h5>
<h6>Heading level 6</h6>
```

The effects are:

Heading Level 2

Heading Level 3

Heading Level 4

Note that CSS can drastically change the rendering.

Paragraphs:

```
<p>Paragraph 1</p> <p>Paragraph 2</p>
```

Line breaks: `
`. The difference between `
` and `<p>` is that `
` [breaks a line](#) without altering the semantic structure of the page, whereas `<p>` sections the page into [paragraphs](#). The element `
` is an *empty element* in that, although it may have attributes, it can take no content and it may not have an end tag.

```
<p>This <br> is a paragraph <br> with <br>
line breaks</p>
```

This is a link in HTML. To create a link the `<a>` tag is used. The `href` attribute holds the URL address of the link.

```
<a href="https://www.wikipedia.org/">A link
to Wikipedia!</a>
```

Inputs:

There are many possible ways a user can give input/s like:

```
1 <input type="text" /> <!-- This is for
   text input -->
2 <input type="file" /> <!-- This is for
   uploading files -->
3 <input type="checkbox" /> <!-- This is
   for checkboxes -->
```

Comments:

```
<!-- This is a comment -->
```

Comments can help in the understanding of the markup and do not display in the webpage.

There are several types of markup elements used in HTML:

Structural markup indicates the purpose of text

For example, `<h2>Golf</h2>` establishes "Golf" as a second-level [heading](#). Structural markup does not denote any specific rendering, but most web browsers have default styles for element formatting. Content may be further styled using [Cascading Style Sheets](#) (CSS).^[72]

Presentational markup indicates the appearance of the text, regardless of its purpose

For example, `bold text` indicates that visual output devices should render "boldface" in bold text, but gives little indication what devices that are unable to do this (such as aural devices that read the text aloud) should do.

In the case of both `bold text` and

`<i>italic text</i>`, there are other elements that

may have equivalent visual renderings but that are more semantic in nature, such as `strong`

`text` and `emphasized text`

respectively. It is easier to see how an aural user agent should interpret the latter two elements. However, they are not equivalent to their presentational counterparts: it would be undesirable for a screen-reader to emphasize the name of a book, for instance, but on a screen such a name would be italicized. Most presentational markup elements have become [deprecated](#) under the HTML 4.0 specification in favor of using [CSS](#) for styling.

Hypertext markup makes parts of a document into links to other documents

An anchor element creates a [hyperlink](#) in the document and its `href` attribute sets the link's target [URL](#). For example, the HTML markup `<a`

```
href="https://www.google.com/">Wikipedia</a>
```

, will render the word "[Wikipedia](#)" as a hyperlink. To render an image as a hyperlink, an `img` element is inserted as content into the `a` element. Like `br`, `img` is an empty element with attributes but no content or closing tag. `<a`

```
href="https://example.org"></a> .
```

Attributes

Most of the attributes of an element are [name-value pairs](#), separated by `=` and written within the start tag of an element after the element's name. The value may be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML).^{[73][74]} Leaving attribute values unquoted is considered unsafe.^[75] In contrast with name-value pair attributes, there are some attributes that affect the element simply by their presence in the start tag of the element,^[6] like the `ismap` attribute for the `img` element.^[76]

There are several common attributes that may appear in many elements :

- The `id` attribute provides a document-wide unique identifier for an element. This is used to identify the element so that stylesheets can alter its presentational properties, and scripts may alter, animate or delete its contents or presentation. Appended to the URL of the page, it provides a globally unique identifier for the element, typically a sub-section of the page. For example, the ID "Attributes" in `https://en.wikipedia.org/wiki/HTML#Attributes`.
- The `class` attribute provides a way of classifying similar elements. This can be used for [semantic](#) or presentation purposes. For example, an HTML document might semantically use the designation `<class="notation">` to indicate that all elements with this class value are subordinate to the main text of the document. In presentation, such elements might be gathered together and presented as footnotes on a page instead of appearing in the place where they occur in the HTML source. Class attributes are used semantically in [microformats](#). Multiple class values may be specified; for example `<class="notation important">` puts the element into both the `notation` and the `important` classes.

- An author may use the `style` attribute to assign presentational properties to a particular element. It is considered better practice to use an element's `id` or `class` attributes to select the element from within a [stylesheet](#), though sometimes this can be too cumbersome for a simple, specific, or ad hoc styling.
- The `title` attribute is used to attach subtextual explanation to an element. In most [browsers](#) this attribute is displayed as a [tooltip](#).
- The `lang` attribute identifies the natural language of the element's contents, which may be different from that of the rest of the document. For example, in an English-language document:

```
<p>Oh well, <span lang="fr">c'est la  
vie</span>, as they say in France.</p>
```

The abbreviation element, `abbr`, can be used to demonstrate some of these attributes:

```
<abbr id="anId" class="jargon"  
style="color:purple;" title="Hypertext  
Markup Language">HTML</abbr>
```

This example displays as [HTML](#); in most browsers, pointing the cursor at the abbreviation should display the title text

"Hypertext Markup Language."

Most elements take the language-related attribute `dir` to specify text direction, such as with `rtl` for right-to-left text in, for example, [Arabic](#), [Persian](#) or [Hebrew](#).^[77]

Character and entity references

As of version 4.0, HTML defines a set of 252 [character entity references](#) and a set of 1,114,050 [numeric character references](#), both of which allow individual characters to be written via simple markup, rather than literally. A literal character and its markup counterpart are considered equivalent and are rendered identically.

The ability to "escape" characters in this way allows for the characters `<` and `&` (when written as `<` and `&`, respectively) to be interpreted as character data, rather than markup. For example, a literal `<` normally indicates the start of a tag, and `&` normally indicates the start of a character entity reference or numeric character reference; writing it as `&` or `&` or `&` allows `&` to be included in the content of an element or in the value of an attribute. The double-quote character (`"`), when not used to quote an attribute value, must also be escaped as `"` or `"` or `"` when it appears within the attribute value itself. Equivalently, the single-quote character (`'`), when not used to quote an

attribute value, must also be escaped as `'` or `'` (or as `'` in HTML5 or XHTML documents^{[78][79]}) when it appears within the attribute value itself. If document authors overlook the need to escape such characters, some browsers can be very forgiving and try to use context to guess their intent. The result is still invalid markup, which makes the document less accessible to other browsers and to other [user agents](#) that may try to parse the document for [search and indexing](#) purposes for example.

Escaping also allows for characters that are not easily typed, or that are not available in the document's [character encoding](#), to be represented within element and attribute content. For example, the acute-accented `e` (`é`), a character typically found only on Western European and South American keyboards, can be written in any HTML document as the entity reference `é` or as the numeric references `é` or `é`, using characters that are available on all keyboards and are supported in all character encodings. [Unicode](#) character encodings such as [UTF-8](#) are compatible with all modern browsers and allow direct access to almost all the characters of the world's writing systems.^[80]

Example HTML Escape Sequences

Named	Decimal	Hexadecimal	Result	Description
<code>&amp;</code>	<code>&#38;</code>	<code>&#x26;</code>	<code>&</code>	Ampersand
<code>&lt;</code>	<code>&#60;</code>	<code>&#x3C;</code>	<code><</code>	Less Than
<code>&gt;</code>	<code>&#62;</code>	<code>&#x3E;</code>	<code>></code>	Greater Than
<code>&quot;</code>	<code>&#34;</code>	<code>&#x22;</code>	<code>"</code>	Double Quote
<code>&apos;</code>	<code>&#39;</code>	<code>&#x27;</code>	<code>'</code>	Single Quote
<code>&nbsp;</code>	<code>&#160;</code>	<code>&#xA0;</code>	<code> </code>	Non-Breaking Space
<code>&copy;</code>	<code>&#169;</code>	<code>&#xA9;</code>	<code>©</code>	Copyright
<code>&reg;</code>	<code>&#174;</code>	<code>&#xAE;</code>	<code>®</code>	Registered Trademark
<code>&dagger;</code>	<code>&#8224;</code>	<code>&#x2020;</code>	<code>†</code>	Dagger
<code>&Dagger;</code>	<code>&#8225;</code>	<code>&#x2021;</code>	<code>‡</code>	Double dagger
<code>&ddagger;</code>	<code>&#8225;</code>	<code>&#x2021;</code>	<code>‡</code>	Double dagger
<code>&trade;</code>	<code>&#8482;</code>	<code>&#x2122;</code>	<code>™</code>	Trademark

Data types

HTML defines several [data types](#) for element content, such as script data and stylesheet data, and a plethora of types for attribute values, including IDs, names, URIs, numbers, units of length, languages, media descriptors, colors, character encodings, dates and times, and so on. All of these data types are specializations of character data.

Document type declaration

HTML documents are required to start with a [Document Type Declaration](#) (informally, a "doctype"). In browsers, the doctype helps to define the rendering mode—particularly whether to use [quirks mode](#).

The original purpose of the doctype was to enable parsing and validation of HTML documents by SGML tools based on the [Document Type Definition](#) (DTD). The DTD to which the DOCTYPE refers contains a machine-readable grammar specifying the permitted and prohibited content for a document conforming to such a DTD. Browsers, on the other hand, do not implement HTML as an application of SGML and by consequence do not read the DTD.

[HTML5](#) does not define a DTD; therefore, in HTML5 the doctype declaration is simpler and shorter:^[81]

```
<!DOCTYPE html>
```

An example of an HTML 4 doctype

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01//EN"  
"https://www.w3.org/TR/html4/strict.dtd">
```

This declaration references the DTD for the "strict" version of HTML 4.01. SGML-based validators read the DTD in order to properly parse the document and to perform validation. In modern browsers, a valid doctype activates standards mode as opposed to [quirks mode](#).

In addition, HTML 4.01 provides Transitional and Frameset DTDs, [as explained below](#). Transitional type is the most inclusive, incorporating current tags as well as older or "deprecated" tags, with the Strict DTD excluding deprecated tags. Frameset has all tags necessary to make frames on a page along with the tags included in transitional type.

Semantic HTML

Semantic HTML is a way of writing HTML that emphasizes the meaning of the encoded information over its presentation (look). HTML has included semantic markup from its

inception,^[82] but has also included presentational markup, such as ``, `<i>` and `<center>` tags. There are also the semantically neutral `span` and `div` tags. Since the late 1990s, when [Cascading Style Sheets](#) were beginning to work in most browsers, web authors have been encouraged to avoid the use of presentational HTML markup with a view to the [separation of presentation and content](#).^[83]

In a 2001 discussion of the [Semantic Web](#), [Tim Berners-Lee](#) and others gave examples of ways in which intelligent software "agents" may one day automatically crawl the web and find, filter and correlate previously unrelated, published facts for the benefit of human users.^[84] Such agents are not commonplace even now, but some of the ideas of [Web 2.0](#), [mashups](#) and [price comparison websites](#) may be coming close. The main difference between these web application hybrids and Berners-Lee's semantic agents lies in the fact that the current [aggregation](#) and hybridization of information is usually designed in by [web developers](#), who already know the web locations and the [API semantics](#) of the specific data they wish to mash, compare and combine.

An important type of web agent that does crawl and read web pages automatically, without prior knowledge of what it might find, is the [web crawler](#) or search-engine spider. These software agents are dependent on the semantic clarity of web pages they find as they use various techniques and [algorithms](#) to read and index millions of web pages a day and provide

web users with [search facilities](#) without which the World Wide Web's usefulness would be greatly reduced.

In order for search-engine spiders to be able to rate the significance of pieces of text they find in HTML documents, and also for those creating mashups and other hybrids as well as for more automated agents as they are developed, the semantic structures that exist in HTML need to be widely and uniformly applied to bring out the meaning of published text.^[85]

Presentational markup tags are [deprecated](#) in current HTML and [XHTML](#) recommendations. The majority of presentational features from previous versions of HTML are no longer allowed as they lead to poorer accessibility, higher cost of site maintenance, and larger document sizes.^[86]

Good semantic HTML also improves the [accessibility](#) of web documents (see also [Web Content Accessibility Guidelines](#)). For example, when a screen reader or audio browser can correctly ascertain the structure of a document, it will not waste the visually impaired user's time by reading out repeated or irrelevant information when it has been marked up correctly.

Delivery

HTML documents can be delivered by the same means as any other computer file. However, they are most often delivered either by [HTTP](#) from a [web server](#) or by [email](#).

HTTP

The [World Wide Web](#) is composed primarily of HTML documents transmitted from web servers to web browsers using the [Hypertext Transfer Protocol](#) (HTTP). However, HTTP is used to serve images, sound, and other content, in addition to HTML. To allow the web browser to know how to handle each document it receives, other information is transmitted along with the document. This [meta data](#) usually includes the [MIME type](#) (e.g., `text/html` or `application/xhtml+xml`) and the character encoding (see [Character encoding in HTML](#)).

In modern browsers, the MIME type that is sent with the HTML document may affect how the document is initially interpreted. A document sent with the XHTML MIME type is expected to be [well-formed](#) XML; syntax errors may cause the browser to fail to render it. The same document sent with the HTML MIME type might be displayed successfully, since some browsers are more lenient with HTML.

The W3C recommendations state that XHTML 1.0 documents that follow guidelines set forth in the recommendation's Appendix C may be labeled with either MIME Type.^[87] XHTML

1.1 also states that XHTML 1.1 documents should^[88] be labeled with either MIME type.^[89]

HTML e-mail

Most graphical email clients allow the use of a subset of HTML (often ill-defined) to provide formatting and [semantic](#) markup not available with [plain text](#). This may include typographic information like coloured headings, emphasized and quoted text, inline images and diagrams. Many such clients include both a [GUI](#) editor for composing HTML e-mail messages and a rendering engine for displaying them. Use of HTML in e-mail is criticized by some because of compatibility issues, because it can help disguise [phishing](#) attacks, because of accessibility issues for blind or visually impaired people, because it can confuse [spam](#) filters and because the message size is larger than plain text.

Naming conventions

The most common [filename extension](#) for [files](#) containing HTML is `.html`. A common abbreviation of this is `.htm`, which originated because some early operating systems and file systems, such as [DOS](#) and the limitations imposed by [FAT](#) data structure, limited file extensions to [three letters](#).^[90]

HTML Application

An HTML Application (HTA; file extension ".hta") is a [Microsoft Windows](#) application that uses HTML and Dynamic HTML in a [browser](#) to provide the application's graphical interface. A regular HTML file is confined to the security model of the [web browser's security](#), communicating only to web servers and manipulating only web page objects and [site cookies](#). An HTA runs as a fully trusted application and therefore has more privileges, like creation/editing/removal of files and [Windows Registry](#) entries. Because they operate outside the browser's security model, HTAs cannot be executed via HTTP, but must be downloaded (just like an [EXE file](#)) and executed from local file system.

HTML4 variations

Since its inception, HTML and its associated protocols gained acceptance relatively quickly. However, no clear standards existed in the early years of the language. Though its creators originally conceived of HTML as a semantic language devoid of presentation details,^[91] practical uses pushed many presentational elements and attributes into the language, driven largely by the various browser vendors. The latest standards surrounding HTML reflect efforts to overcome the sometimes chaotic development of the language^[92] and to create a rational foundation for building both meaningful and well-presented documents. To return HTML to its role as a

semantic language, the [W3C](#) has developed style languages such as [CSS](#) and [XSL](#) to shoulder the burden of presentation. In conjunction, the HTML specification has slowly reined in the presentational elements.

There are two axes differentiating various variations of HTML as currently specified: SGML-based HTML versus XML-based HTML (referred to as XHTML) on one axis, and strict versus transitional (loose) versus frameset on the other axis.

SGML-based versus XML-based HTML

One difference in the latest HTML specifications lies in the distinction between the SGML-based specification and the XML-based specification. The XML-based specification is usually called [XHTML](#) to distinguish it clearly from the more traditional definition. However, the root element name continues to be "html" even in the XHTML-specified HTML. The W3C intended XHTML 1.0 to be identical to HTML 4.01 except where limitations of XML over the more complex SGML require workarounds. Because XHTML and HTML are closely related, they are sometimes documented in parallel. In such circumstances, some authors [conflate the two names](#) as (X)HTML or X(HTML).

Like HTML 4.01, XHTML 1.0 has three sub-specifications: strict, transitional and frameset.

Aside from the different opening declarations for a document, the differences between an HTML 4.01 and XHTML 1.0 document—in each of the corresponding DTDs—are largely syntactic. The underlying syntax of HTML allows many shortcuts that XHTML does not, such as elements with optional opening or closing tags, and even empty elements which must not have an end tag. By contrast, XHTML requires all elements to have an opening tag and a closing tag. XHTML, however, also introduces a new shortcut: an XHTML tag may be opened and closed within the same tag, by including a slash before the end of the tag like this: `
`. The introduction of this shorthand, which is not used in the SGML declaration for HTML 4.01, may confuse earlier software unfamiliar with this new convention. A fix for this is to include a space before closing the tag, as such: `
`.^[93]

To understand the subtle differences between HTML and XHTML, consider the transformation of a valid and well-formed XHTML 1.0 document that adheres to Appendix C (see below) into a valid HTML 4.01 document. To make this translation requires the following steps:

1. **The language for an element should be specified with a `lang` attribute rather than the XHTML `xml:lang` attribute.** XHTML uses XML's built in language-defining functionality attribute.

2. **Remove the XML namespace** (`xmlns=URI`). HTML has no facilities for namespaces.
3. **Change the document type declaration** from XHTML 1.0 to HTML 4.01. (see [DTD section](#) for further explanation).
4. If present, **remove the XML declaration**. (Typically this is: `<?xml version="1.0" encoding="utf-8"?>`).
5. **Ensure that the document's MIME type is set to** `text/html` . For both HTML and XHTML, this comes from the HTTP `Content-Type` header sent by the server.
6. **Change the XML empty-element syntax to an HTML style empty element** (`
` to `
`).

Those are the main changes necessary to translate a document from XHTML 1.0 to HTML 4.01. To translate from HTML to XHTML would also require the addition of any omitted opening or closing tags. Whether coding in HTML or XHTML it may just be best to always include the optional tags within an HTML document rather than remembering which tags can be omitted.

A well-formed XHTML document adheres to all the syntax requirements of XML. A valid document adheres to the content specification for XHTML, which describes the document structure.

The W3C recommends several conventions to ensure an easy migration between HTML and XHTML (see [HTML Compatibility Guidelines](#)). The following steps can be applied to XHTML 1.0 documents only:

- Include both `xml:lang` and `lang` attributes on any elements assigning language.
- Use the empty-element syntax only for elements specified as empty in HTML.
- Include an extra space in empty-element tags: for example `
` instead of `
`.
- Include explicit close tags for elements that permit content but are left empty (for example, `<div></div>`, not `<div />`).
- Omit the XML declaration.

By carefully following the W3C's compatibility guidelines, a user agent should be able to interpret the document equally as HTML or XHTML. For documents that are XHTML 1.0 and have been made compatible in this way, the W3C permits them to be served either as HTML (with a `text/html` MIME type), or as XHTML (with an `application/xhtml+xml` or `application/xml` MIME type). When delivered as XHTML, browsers should use an XML parser, which adheres strictly to the XML specifications for parsing the document's contents.

Transitional versus strict

HTML 4 defined three different versions of the language: Strict, Transitional (once called Loose) and Frameset. The Strict version is intended for new documents and is considered best practice, while the Transitional and Frameset versions were developed to make it easier to transition documents that conformed to older HTML specification or didn't conform to any specification to a version of HTML 4. The Transitional and Frameset versions allow for presentational markup, which is omitted in the Strict version. Instead, [cascading style sheets](#) are encouraged to improve the presentation of HTML documents. Because XHTML 1 only defines an XML syntax for the language defined by HTML 4, the same differences apply to XHTML 1 as well.

The Transitional version allows the following parts of the vocabulary, which are not included in the Strict version:

- **A looser content model**

- Inline elements and plain text are allowed directly in:

`body`, `blockquote`, `form`, `noscript` and `noframes`

- **Presentation related elements**

- underline (`u`) (Deprecated. can confuse a visitor with a hyperlink.)
- strike-through (`s`)

- `center` (Deprecated. use CSS instead.)
- `font` (Deprecated. use CSS instead.)
- `basefont` (Deprecated. use CSS instead.)

- **Presentation related attributes**

- `background` (Deprecated. use CSS instead.) and `bgcolor` (Deprecated. use CSS instead.) attributes for `body` (required element according to the W3C.) element.
- `align` (Deprecated. use CSS instead.) attribute on `div`, `form`, paragraph (`p`) and heading (`h1` ... `h6`) elements
- `align` (Deprecated. use CSS instead.), `noshade` (Deprecated. use CSS instead.), `size` (Deprecated. use CSS instead.) and `width` (Deprecated. use CSS instead.) attributes on `hr` element
- `align` (Deprecated. use CSS instead.), `border`, `vspace` and `hspace` attributes on `img` and `object` (caution: the `object` element is only supported in Internet Explorer (from the major browsers)) elements
- `align` (Deprecated. use CSS instead.) attribute on `legend` and `caption` elements

- `align` (Deprecated. use CSS instead.) and `b bgcolor` (Deprecated. use CSS instead.) on `table` element
- `nowrap` (Obsolete), `b bgcolor` (Deprecated. use CSS instead.), `width`, `height` on `td` and `th` elements
- `b bgcolor` (Deprecated. use CSS instead.) attribute on `tr` element
- `clear` (Obsolete) attribute on `br` element
- `compact` attribute on `dl`, `dir` and `menu` elements
- `type` (Deprecated. use CSS instead.), `compact` (Deprecated. use CSS instead.) and `start` (Deprecated. use CSS instead.) attributes on `ol` and `ul` elements
- `type` and `value` attributes on `li` element
- `width` attribute on `pre` element
- **Additional elements in Transitional specification**
 - `menu` (Deprecated. use CSS instead.) list (no substitute, though unordered list is recommended)
 - `dir` (Deprecated. use CSS instead.) list (no substitute, though unordered list is recommended)

- `isindex` (Deprecated.) (element requires server-side support and is typically added to documents server-side, `form` and `input` elements can be used as a substitute)
- `applet` (Deprecated. use the `object` element instead.)
- **The `language` (Obsolete) attribute on script element** (redundant with the `type` attribute).
- **Frame related entities**
 - `iframe`
 - `noframes`
 - `target` (Deprecated in the `map`, `link` and `form` elements.) attribute on `a`, client-side image-map (`map`), `link`, `form` and `base` elements

The Frameset version includes everything in the Transitional version, as well as the `frameset` element (used instead of `body`) and the `frame` element.

Frameset versus transitional

In addition to the above transitional differences, the frameset specifications (whether XHTML 1.0 or HTML 4.01) specify a different content model, with `frameset` replacing `body`,

that contains either `frame` elements, or optionally `noframes` with a `body`.

Summary of specification versions

As this list demonstrates, the loose versions of the specification are maintained for legacy support. However, contrary to popular misconceptions, the move to XHTML does not imply a removal of this legacy support. Rather the X in XML stands for extensible and the W3C is modularizing the entire specification and opening it up to independent extensions. The primary achievement in the move from XHTML 1.0 to XHTML 1.1 is the modularization of the entire specification. The strict version of HTML is deployed in XHTML 1.1 through a set of modular extensions to the base XHTML 1.1 specification. Likewise, someone looking for the loose (transitional) or frameset specifications will find similar extended XHTML 1.1 support (much of it is contained in the legacy or frame modules). The modularization also allows for separate features to develop on their own timetable. So for example, XHTML 1.1 will allow quicker migration to emerging XML standards such as [MathML](#) (a presentational and semantic math language based on XML) and [XForms](#)—a new highly advanced web-form technology to replace the existing HTML forms.

In summary, the HTML 4 specification primarily reined in all the various HTML implementations into a single clearly written specification based on SGML. XHTML 1.0, ported this specification, as is, to the new XML defined specification. Next, XHTML 1.1 takes advantage of the extensible nature of XML and modularizes the whole specification. XHTML 2.0 was intended to be the first step in adding new features to the specification in a standards-body-based approach.

WHATWG HTML versus HTML5

The HTML Living Standard, which is developed by WHATWG, is the official version, while W3C HTML5 is no longer separate from WHATWG.

WYSIWYG editors

There are some [WYSIWYG](#) editors (What You See Is What You Get), in which the user lays out everything as it is to appear in the HTML document using a [graphical user interface](#) (GUI), often similar to [word processors](#). The editor renders the document rather than show the code, so authors do not require extensive knowledge of HTML.

The WYSIWYG editing model has been criticized,^{[94][95]} primarily because of the low quality of the generated code;

there are voices advocating a change to the **WYSIWYM** model (What You See Is What You Mean).

WYSIWYG editors remain a controversial topic because of their perceived flaws such as:

- Relying mainly on layout as opposed to meaning, often using markup that does not convey the intended meaning but simply copies the layout.^[96]
- Often producing extremely verbose and redundant code that fails to make use of the cascading nature of HTML and CSS.
- Often producing ungrammatical markup, called **tag soup** or semantically incorrect markup (such as `` for italics).
- As a great deal of the information in HTML documents is not in the layout, the model has been criticized for its "what you see is all you get"-nature.^[97]

See also

- [Breadcrumb navigation](#)
- [Comparison of HTML parsers](#)
- [Dynamic web page](#)
- [HTML character references](#)
- [List of document markup languages](#)

- [List of XML and HTML character entity references](#)
- [Microdata \(HTML\)](#)
- [Microformat](#)
- [Polyglot markup](#)
- [Semantic HTML](#)
- [W3C \(X\)HTML Validator](#)
- [Web colors](#)

References

1. ["W3C Html"](#) ↗.
2. ["HTML 4.0 Specification — W3C Recommendation — Conformance: requirements and recommendations"](#) ↗. w3. World Wide Web Consortium. December 18, 1997. [Archived](#) ↗ from the original on July 5, 2015. Retrieved July 6, 2015.
3. Tim Berners-Lee, "Information Management: A Proposal." CERN (March 1989, May 1990). [W3.org](#) ↗
4. Tim Berners-Lee, ["Design Issues"](#) ↗
5. Tim Berners-Lee, ["Design Issues"](#) ↗
6. ["Tags used in HTML"](#) ↗. w3. World Wide Web Consortium. November 3, 1992. [Archived](#) ↗ from the

original on January 31, 2010. Retrieved November 16, 2008.

7. Berners-Lee, Tim (October 29, 1991). ["First mention of HTML Tags on the www-talk mailing list"](#) [↗](#). w3. World Wide Web Consortium. [Archived](#) [↗](#) from the original on May 24, 2007. Retrieved April 8, 2007.
8. ["Index of elements in HTML 4"](#) [↗](#). w3. World Wide Web Consortium. December 24, 1999. [Archived](#) [↗](#) from the original on May 5, 2007. Retrieved April 8, 2007.
9. Berners-Lee, Tim (December 9, 1991). ["Re: SGML/HTML docs, X Browser \(archived www-talk mailing list post\)"](#) [↗](#). w3. [Archived](#) [↗](#) from the original on December 22, 2007. Retrieved June 16, 2007. "SGML is very general. HTML is a specific application of the SGML basic syntax applied to hypertext documents with simple structure."
10. Berners-Lee, Tim; Connolly, Daniel (June 1993). ["Hypertext Markup Language \(HTML\): A Representation of Textual Information and MetaInformation for Retrieval and Interchange"](#) [↗](#). w3. [Archived](#) [↗](#) from the original on January 3, 2017. Retrieved January 4, 2017.
11. Raggett, Dave. ["A Review of the HTML+ Document Format"](#) [↗](#). w3. [Archived](#) [↗](#) from the original on February 29, 2000. Retrieved May 22, 2020. "The hypertext markup language HTML was developed as a simple non-proprietary delivery format for global hypertext. HTML+ is

a set of modular extensions to HTML and has been developed in response to a growing understanding of the needs of information providers. These extensions include text flow around floating figures, fill-out forms, tables and mathematical equations."

12. Berners-Lee, Tim; Connelly, Daniel (November 1995). ["Hypertext Markup Language – 2.0"](#) [↗](#). *ietf.org*. Internet Engineering Task Force. [RFC 1866](#) [↗](#). [Archived](#) [↗](#) from the original on August 11, 2010. Retrieved December 1, 2010. "This document thus defines an HTML 2.0 (to distinguish it from the previous informal specifications). Future (generally upwardly compatible) versions of HTML with new features will be released with higher version numbers."
13. Raggett, Dave (1998). [Raggett on HTML 4](#) [↗](#). Archived from [the original](#) [↗](#) on August 9, 2007. Retrieved July 9, 2007.
14. ["HTML5 – Hypertext Markup Language – 5.0"](#) [↗](#). Internet Engineering Task Force. 28 October 2014. [Archived](#) [↗](#) from the original on October 28, 2014. Retrieved November 25, 2014. "This document recommends HTML 5.0 after completion."
15. ["HTML 3.2 Reference Specification"](#) [↗](#). World Wide Web Consortium. January 14, 1997. Retrieved November 16, 2008.

16. ["IETF HTML WG"](#) . Retrieved June 16, 2007. "Note: This working group is closed"
17. Arnoud Engelfriet. ["Introduction to Wilbur"](#) . Web Design Group. Retrieved June 16, 2007.
18. ["HTML 4.0 Specification"](#) . World Wide Web Consortium. December 18, 1997. Retrieved November 16, 2008.
19. ["HTML 4 – 4 Conformance: requirements and recommendations"](#) . Retrieved December 30, 2009.
20. ["HTML 4.0 Specification"](#) . World Wide Web Consortium. April 24, 1998. Retrieved November 16, 2008.
21. ["HTML 4.01 Specification"](#) . World Wide Web Consortium. December 24, 1999. Retrieved November 16, 2008.
22. ISO (2000). ["ISO/IEC 15445:2000 – Information technology – Document description and processing languages – HyperText Markup Language \(HTML\)"](#) . Retrieved December 26, 2009.
23. [Cs.Tcd.ie](#) . Cs.Tcd.ie (2000-05-15). Retrieved on 2012-02-16.
24. ["HTML5: A vocabulary and associated APIs for HTML and XHTML"](#) . World Wide Web Consortium. 28 October 2014. Retrieved 31 October 2014.

25. ["Open Web Platform Milestone Achieved with HTML5 Recommendation"](#) [↗](#) (Press release). World Wide Web Consortium. 28 October 2014. Retrieved 31 October 2014.
26. ["HTML 5.1"](#) [↗](#). World Wide Web Consortium. 1 November 2016. Retrieved 6 January 2017.
27. ["HTML 5.1 is a W3C Recommendation"](#) [↗](#). World Wide Web Consortium. 1 November 2016. Retrieved 6 January 2017.
28. Philippe le Hegaret (17 November 2016). ["HTML 5.1 is the gold standard"](#) [↗](#). World Wide Web Consortium. Retrieved 6 January 2017.
29. ["HTML 5.2"](#) [↗](#). World Wide Web Consortium. 14 December 2017. Retrieved 15 December 2017.
30. ["HTML 5.2 is now a W3C Recommendation"](#) [↗](#). World Wide Web Consortium. 14 December 2017. Retrieved 15 December 2017.
31. Charles McCathie Nevile (14 December 2017). ["HTML 5.2 is done, HTML 5.3 is coming"](#) [↗](#). World Wide Web Consortium. Retrieved 15 December 2017.
32. Connolly, Daniel (6 June 1992). ["MIME as a hypertext architecture"](#) [↗](#). CERN. Retrieved 24 October 2010.
33. Connolly, Daniel (15 July 1992). ["HTML DTD enclosed"](#) [↗](#). CERN. Retrieved 24 October 2010.

34. [Connolly, Daniel](#) (18 August 1992). "document type declaration subset for Hyper Text Markup Language as defined by the World Wide Web project" [↗](#). CERN. Retrieved 24 October 2010.
35. [Connolly, Daniel](#) (24 November 1992). "Document Type Definition for the Hyper Text Markup Language as used by the World Wide Web application" [↗](#). CERN. Retrieved 24 October 2010. See section "Revision History"
36. [Berners-Lee, Tim](#); [Connolly, Daniel](#) (June 1993). "Hyper Text Markup Language (HTML) Internet Draft version 1.1" [↗](#). IETF IIR Working Group. Retrieved 18 September 2010.
37. [Berners-Lee, Tim](#); [Connolly, Daniel](#) (June 1993). "Hypertext Markup Language (HTML) Internet Draft version 1.2" [↗](#). IETF IIR Working Group. Retrieved 18 September 2010.
38. "History for draft-raggett-www-html-00" [↗](#). *datatracker.ietf.org*. 1993-11-08. Retrieved 2019-11-18.
39. [Berners-Lee, Tim](#); [Connolly, Daniel](#) (28 November 1994). "HyperText Markup Language Specification – 2.0 INTERNET DRAFT" [↗](#). IETF. Retrieved 24 October 2010.
40. Connolly <connolly@w3.org>, Daniel W. (1995-05-16). "Hypertext Markup Language - 2.0" [↗](#). *tools.ietf.org*. Retrieved 2019-11-18.

41. ["History for draft-ietf-html-spec-05"](#) . datatracker.ietf.org. Retrieved 2019-11-18.
42. ["HTML 3.0 Draft \(Expired!\) Materials"](#) . World Wide Web Consortium. December 21, 1995. Retrieved November 16, 2008.
43. ["HyperText Markup Language Specification Version 3.0"](#) . Retrieved June 16, 2007.
44. Raggett, Dave (28 March 1995). ["HyperText Markup Language Specification Version 3.0"](#) . *HTML 3.0 Internet Draft Expires in six months*. [World Wide Web Consortium](#). Retrieved 17 June 2010.
45. Bowers, Neil. "Weblint: Just Another Perl Hack". [CiteSeerX 10.1.1.54.7191](#) .
46. Lie, Håkon Wium; Bos, Bert (April 1997). *Cascading style sheets: designing for the Web* . Addison Wesley Longman. p. [263](#) . Retrieved 9 June 2010.
47. ["HTML5"](#) . World Wide Web Consortium. June 10, 2008. Retrieved November 16, 2008.
48. ["HTML5, one vocabulary, two serializations"](#) . Retrieved February 25, 2009.
49. ["W3C Confirms May 2011 for HTML5 Last Call, Targets 2014 for HTML5 Standard"](#) . [World Wide Web Consortium](#). 14 February 2011. Retrieved 18 February 2011.

50. Hickson, Ian. ["HTML Is the New HTML5"](#) . Retrieved 21 January 2011.
51. ["HTML5 gets the splits"](#) . netmagazine.com. Retrieved 23 July 2012.
52. ["HTML5"](#) . W3.org. 2012-12-17. Retrieved 2013-06-15.
53. ["When Will HTML5 Be Finished?"](#) . FAQ. WHAT Working Group. Retrieved 29 November 2009.
54. ["Call for Review: HTML5 Proposed Recommendation Published W3C News"](#) . W3.org. 2014-09-16. Retrieved 2014-09-27.
55. ["Open Web Platform Milestone Achieved with HTML5 Recommendation"](#) . W3C. 28 October 2014. Retrieved 29 October 2014.
56. ["HTML5 specification finalized, squabbling over specs continues"](#) . *Ars Technica*. 2014-10-29. Retrieved 2014-10-29.
57. ["XHTML 1.0: The Extensible HyperText Markup Language \(Second Edition\)"](#) . World Wide Web Consortium. January 26, 2000. Retrieved November 16, 2008.
58. ["XHTML 1.1 – Module-based XHTML — Second Edition"](#) . World Wide Web Consortium. February 16, 2007. Retrieved November 16, 2008.






59. ["Modularization of XHTML"](#) [↗](#). [www.w3.org](#). Retrieved 2017-01-04.
60. ["XHTM 2.0"](#) [↗](#). World Wide Web Consortium. July 26, 2006. Retrieved November 16, 2008.
61. ["XHTML 2 Working Group Expected to Stop Work End of 2009, W3C to Increase Resources on HTML5"](#) [↗](#). World Wide Web Consortium. July 17, 2009. Retrieved November 16, 2008.
62. ["W3C XHTML FAQ"](#) [↗](#).
63. ["HTML5"](#) [↗](#). W3C. 19 October 2013.
64. Jaffe, Jeff (28 May 2019). ["W3C and WHATWG to Work Together to Advance the Open Web Platform"](#) [↗](#). W3C Blog. [Archived](#) [↗](#) from the original on 29 May 2019. Retrieved 29 May 2019.
65. ["W3C and the WHATWG Signed an Agreement to Collaborate on a Single Version of HTML and DOM"](#) [↗](#). W3C. 28 May 2019. [Archived](#) [↗](#) from the original on 29 May 2019. Retrieved 29 May 2019.
66. ["Memorandum of Understanding Between W3C and WHATWG"](#) [↗](#). W3C. 28 May 2019. [Archived](#) [↗](#) from the original on 29 May 2019. Retrieved 29 May 2019.
67. Cimpanu, Catalin (29 May 2019). ["Browser vendors Win War with W3C over HTML and DOM standards"](#) [↗](#).

ZDNet. Archived from [the original](#) on 29 May 2019.
Retrieved 29 May 2019.

68. "W3C - WHATWG Wiki" . *WHATWG Wiki*. Archived from [the original](#) on 29 May 2019. Retrieved 29 May 2019.
69. Shankland, Stephen (July 9, 2009). "An epitaph for the Web standard, XHTML 2" . *CNET*. CBS INTERACTIVE INC.
70. [Activating Browser Modes with Doctype](#) .
Hsivonen.iki.fi. Retrieved on 2012-02-16.
71. "HTML Elements" . w3schools. Retrieved 16 March 2015.
72. "CSS Introduction" . W3schools. Retrieved 16 March 2015.
73. "On SGML and HTML" . World Wide Web Consortium.
Retrieved November 16, 2008.
74. "XHTML 1.0 – Differences with HTML 4" . World Wide Web Consortium. Retrieved November 16, 2008.
75. Korpela, Jukka (July 6, 1998). "Why attribute values should always be quoted in HTML" . Cs.tut.fi. Retrieved November 16, 2008.
76. "Objects, Images, and Applets in HTML documents" .
World Wide Web Consortium. December 24, 1999.
Retrieved November 16, 2008.

77. ["H56: Using the dir attribute on an inline element to resolve problems with nested directional runs"](#) [↗](#). *Techniques for WCAG 2.0*. W3C. Retrieved 18 September 2010.
78. ["Character Entity Reference Chart"](#) [↗](#). World Wide Web Consortium. October 24, 2012.
79. ["The Named Character Reference '"](#) [↗](#). World Wide Web Consortium. January 26, 2000.
80. ["The Unicode Standard: A Technical Introduction"](#) [↗](#). Retrieved 2010-03-16.
81. ["HTML: The Markup Language \(an HTML language reference\)"](#) [↗](#). Retrieved 2013-08-19.
82. Berners-Lee, Tim; Fischetti, Mark (2000). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor* [↗](#). San Francisco: Harper. ISBN 978-0-06-251587-2.
83. Raggett, Dave (2002). ["Adding a touch of style"](#) [↗](#). W3C. Retrieved October 2, 2009. This article notes that presentational HTML markup may be useful when targeting browsers "before Netscape 4.0 and Internet Explorer 4.0". See the [list of web browsers](#) to confirm that these were both released in 1997.
84. Tim Berners-Lee, James Hendler and Ora Lassila (2001). ["The Semantic Web"](#) [↗](#). *Scientific American*. Retrieved

October 2, 2009.

85. Nigel Shadbolt, Wendy Hall and Tim Berners-Lee (2006). ["The Semantic Web Revisited"](#)  (PDF). IEEE Intelligent Systems. Retrieved October 2, 2009.
86. ["HTML: The Living Standard"](#) . WHATWG. Retrieved 27 September 2018.
87. ["XHTML 1.0 The Extensible HyperText Markup Language \(Second Edition\)"](#) . World Wide Web Consortium. 2002 [2000]. Retrieved December 7, 2008. "XHTML Documents which follow the guidelines set forth in Appendix C, "HTML Compatibility Guidelines" may be labeled with the Internet Media Type "text/html" [RFC2854], as they are compatible with most HTML browsers. Those documents, and any other document conforming to this specification, may also be labeled with the Internet Media Type "application/xhtml+xml" as defined in [RFC3236]."
88. ["Key words for use in RFCs to Indicate Requirement Levels"](#) . Harvard University. 1997. [RFC 2119](#) . Retrieved December 7, 2008. "3. SHOULD This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course."

89. ["XHTML 1.1 – Module-based XHTML – Second Edition"](#) [↗](#). World Wide Web Consortium. 2007. Retrieved December 7, 2008. "XHTML 1.1 documents SHOULD be labeled with the Internet Media Type text/html as defined in [RFC2854] or application/xhtml+xml as defined in [RFC3236]."
90. ["Naming Files, Paths, and Namespaces"](#) [↗](#). Microsoft. Retrieved 16 March 2015.
91. [HTML Design Constraints](#) [↗](#), W3C Archives
92. [WWW:BTB – HTML](#) [↗](#), Pris Sears
93. Freeman, E (2005). Head First HTML. O'Reilly.
94. Sauer, C.: WYSIWIKI – Questioning WYSIWYG in the Internet Age. In: Wikimania (2006)
95. Spiesser, J., Kitchen, L.: Optimization of HTML automatically generated by WYSIWYG programs. In: 13th International Conference on World Wide Web, pp. 355–364. WWW '04. ACM, New York, NY (New York, NY, U.S., May 17–20, 2004)
96. [XHTML Reference: blockquote](#) [↗](#) [Archived](#) [↗](#) 2010-03-25 at the [Wayback Machine](#). Xhtml.com. Retrieved on 2012-02-16.
97. [Doug Engelbart's INVISIBLE REVOLUTION](#) [↗](#). Invisiblerevolution.net. Retrieved on 2012-02-16.

External links

HTML

at Wikipedia's [sister projects](#)



[Definitions](#) from
Wiktionary



[Media](#) from Wikimedia
Commons



[Textbooks](#) from
Wikibooks



[Resources](#) from
Wikiversity



[Data](#) from Wikidata



[Discussion](#) from Meta-
Wiki



[Documentation](#) from
MediaWiki

- [HTML](#) at [Curlie](#)
- [WHATWG's HTML Living Standard](#)
- [W3C's HTML specification \(latest published version\)](#)
- [Dave Raggett's Introduction to HTML](#)

- [Tim Berners-Lee Gives the Web a New Definition](#) 
- [HTML Entities](#) 