HTML

The **HyperText Markup Language** or **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.

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 `<img />` 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] A form of HTML, known as HTML5, is used to display video and audio, primarily using the `<canvas>` element, in collaboration with javascript.

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 `<img>`. 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 shown on browser tabs and window titles, and the tag `<div>` defines a division of the page used for easy styling. Between `<head>` and `</head>`, a `<meta>` element can be used to define webpage metadata.

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

Main article: HTML element

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][*better source needed*]

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 element's *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 `<img>` used to embed images, the reference to the image resource in the format like this: `<img src="example.com/example.jpg">`

Some elements, such as the line break `<br>`, or `<br />` 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 `<br>` tag or the inline `<img>` 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

See also: HTML element

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 1*
>
> *Heading Level 2*
>
> *Heading Level 3*
>
> *Heading Level 4*
>
> *Heading Level 5*
>
> *Heading Level 6*

Note that CSS can drastically change the rendering.

Paragraphs:

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

Line breaks:

`<br>`. The difference between `<br>` and `<p>` is that `<br>` breaks a line without altering the semantic structure of the page, whereas `<p>` sections the page into paragraphs. The element `<br>` 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:

```
<input type="text" /> <!-- This is for text input -->
<input type="file" /> <!-- This is for uploading files -->
<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, `<b>bold text</b>` 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 `<b>bold text</b>` 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>strong text</strong>` and `<em>emphasized text</em>` 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"><img src="image.gif" alt="descriptive text" width="50" height="50" border="0"></a>`.

Attributes

Main article: HTML attribute

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."