HyperText Markup Language

The first version of HTML was written by Tim Berners-Lee in 1993. Since then, there have been many different versions of HTML. The most widely used version throughout the 2000's was HTML 4.01, which became an official standard in December 1999.

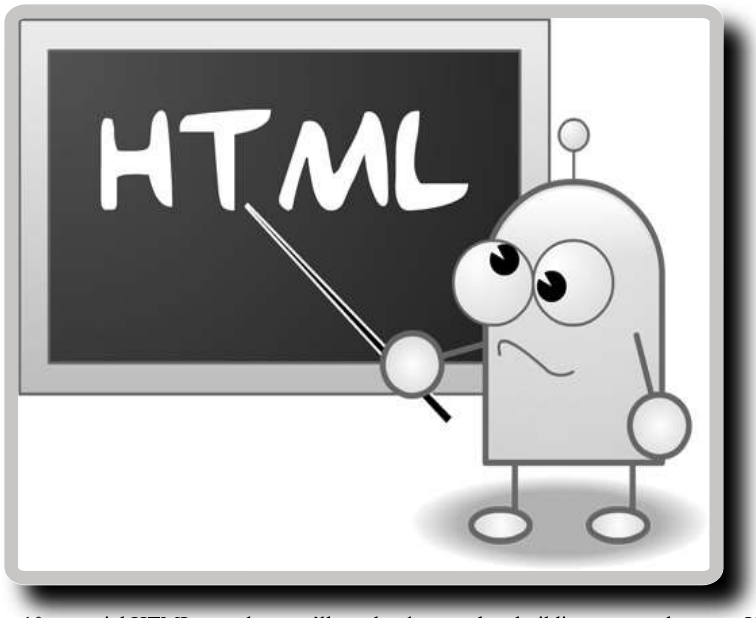
Another version, XHTML, was a rewrite of HTML as an XML language. XML is a standard markup language that is used to create other markup languages. Hundreds of XML languages are in use today, including GML (Geography Markup Language), MathML, MusicML, and RSS (Really Simple Syndication). Since each of these languages was written in a common language (XML), their content can easily be shared across applications. This makes XML potentially very powerful, and it's no surprise that the W3C would create an XML version of HTML (again, called XHTML). XHTML became an official standard in 2000, and was updated in 2002. XHTML is very similar to HTML, but has stricter rules. Strict rules are necessary for all XML languages, because without it, interoperability between applications would be impossible. You'll learn more about the differences between HTML and XHTML in Unit 2.

Headings let you break up your page content into readable chunks. They work much like headings and subheadings in a book or a report.

HTML actually supports 6 heading elements: h1, h2, h3, h4, h5, and h6. h1 is for the most important headings, h2 is for less important subheadings, and so on. Typically you won’t need to use more than h1, h2 and h3, unless your page is very long and complex.

The p element lets you create paragraphs of text. Most browsers display paragraphs with a vertical gap between each paragraph, nicely breaking up the text.

While you can create “paragraphs” of text just by using br tags to insert blank lines between chunks of text, it’s better to use p elements instead. Not only is it neater, but it gives browsers, search engines and other non-humans a better idea of how your page is structured.



Just starting out with HTML? Here are 10 essential HTML tags that you’ll need to know when building your web pages. If you learn how these 10 tags work then you’ll have enough knowledge to put together a basic page.

At the end of the tutorial you’ll find code for an example web page that includes all 10 tags, so that you can see how to use them.

Tags and elements

An HTML tag is a special word or letter surrounded by angle brackets, and >. You use tags to create HTML elements, such as paragraphs or links.

Many elements have an opening tag and a closing tag — for example, a p (paragraph) element has a tag, followed by the paragraph text, followed by a closing p tag.

Some elements don’t have a closing tag. These are called empty elements. For example, the br element for inserting line breaks is simply written br .

If you’re working with XHTML then you write empty elements using self-closing tags — for example, br inside greater and less than.

Berners-Lee

1.html … html — The root element

All web pages start with the html element. It’s also called the root element because it’s at the root of the tree of elements that make up a web page.

To create the html element, you write an opening html> tag followed by a closing /html> tag. Everything else in your web page then goes between these 2 tags:

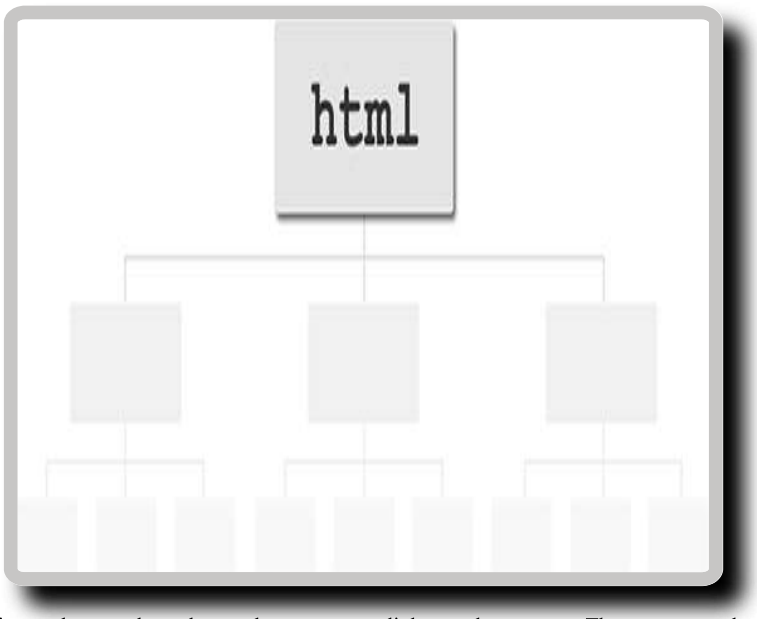
The head element contains information about the web page, as opposed to the web page content itself. There are many elements that you can put inside the head element, such as:

- Title (described below)
- Link, which you can use to add style sheets and favicons to your page
- Meta, for specifying things like character sets, page descriptions, and keywords for search engines
- Script, for adding JavaScript code to the page

The title element contains the title of the page. The title is displayed in the browser’s title bar (the bar at the top of the browser window), as well as in bookmarks, search engine results, and many other places.

The title should describe the page’s content succinctly and accurately. Try to give each page of your site its own unique title.

The body element appears after the head element in the page. It should contain all the content of your web page: text, images, and so on. All web pages have 1 single body element, with the exception of frameset pages, which contain frame elements instead.



One of the most important elements in a web page, the a element lets you create links to other content. The content can be either on your own site or on another site.

To create a link, you wrap a> and /a> tags around the content you want to use for the link, and supply the URL to link to in the a> tag’s href attribute.

The img element lets you insert images into your web pages. To insert an image, you first upload the image to your web server, then use an img tag to reference the uploaded image filename.

he alt attribute is required for all img tags. It’s used by browsers that don’t display images to give alternative text to the visitor.

The div element is a generic container that you can use to add more structure to your page content. For example, you might group several paragraphs or headings that serve a similar purpose together inside a div element. Typically, div elements are used for things like:

- Page headers and footers
- Columns of content and sidebars
- Highlighted boxes within the text flow
- Areas of the page with a specific purpose, such as ad spots
- Image galleries

The span element is similar to div in that it’s used to add structure to your content. The difference is that div is a block-level element, while span is an inline element:

- Block-level elements, such as div, h1, and p, are elements that are designed to hold relatively large or stand-alone blocks of content, such as paragraphs of text. A block-level element always starts on a new line.
- Inline elements, such as span, a, and img, are designed to hold smaller pieces of content — such as a few words or a sentence — within a larger block of content. Adding an inline element doesn’t cause a new line to be created. Block-level elements can contain inline elements, but inline elements can’t contain block-level elements.

fill in form