

Web Basics

HTML

HTML or **HyperText Markup Language** is the standard markup language used to create web pages.



HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like `<html>`). HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some tags represent empty elements and so are unpaired, for example ``. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags).

A web browser can read HTML files and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of

HTML web pages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML*

Markup Language

A (document) markup language is a modern system for annotating a document in a way that is syntactically distinguishable from the text. The idea and terminology evolved from the “marking up” of paper manuscripts, i.e., the revision instructions by editors, traditionally written with a blue pencil on authors’ manuscripts. In digital media this “blue pencil instruction text” was replaced by tags, that is, instructions are expressed directly by tags or “instruction text encapsulated by tags”.

HTML Elements

An HTML element is an individual component of an HTML document or web page, once this has been parsed into the Document Object Model. HTML is composed of a tree of HTML elements and other nodes, such as text nodes. Each element can have HTML attributes specified. Elements can also have content, including other elements and text. HTML elements represent semantics, or meaning. For example, the title element represents the title of the document.

*Definitions and some images courtesy of Wikipedia

In the HTML syntax, most elements are written with a start tag and an end tag, with the content in between. An HTML tag is composed of the name of the element, surrounded by angle brackets. An end tag also has a slash after the opening angle bracket, to distinguish it from the start tag. For example, a paragraph, which is represented by the `p` element, would be written as

```
<p>In the HTML syntax, most elements are written ...</p>
```

Presentation Semantics

Semantics is the study of meaning. In computer science, particularly in human-computer interaction, presentation semantics specify how a particular piece of a formal language is represented in a distinguished manner accessible to human senses, usually human vision. For example, saying that ` ... ` must render the text between these constructs using some bold typeface is a specification of presentation semantics for that syntax. An example of interactive presentation semantics is defining the expected behavior of a hypertext link on a suitable syntax.

Cascading Style Sheets

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

Separating Document Content from Presentation

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts.[1] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for tableless web design). It obviates those portions of markup that would specify presentation by instead providing that information in a separate file. For each relevant HTML element (identified by tags), it provides a list of formatting instructions. For example, it might say (in CSS syntax), “All heading 1 elements should be bold.” Therefore, no formatting markup such as bold tags (``) is needed within the content; what is needed is simply semantic markup saying, “this text is a level 1 heading.”

JavaScript

JavaScript (JS) is a dynamic computer programming language.[5] It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.[5] It is also being used in server-side network programming (with Node.js), game development and the creation of desktop and mobile applications.

Additional Definitions

Web browser

A web browser (commonly referred to as a browser or web client) is a software application for retrieving, presenting and traversing information resources on the World Wide Web. An information resource is identified by a Uniform Resource Identifier (URI/URL) and may be a web page, image, video or other piece of content.[1] Hyperlinks present in resources enable users easily to navigate their browsers to related resources.

Asynchronous JavaScript and XML (Ajax)

Ajax is a group of interrelated Web development techniques used on the client-side to create asynchronous Web applications. With Ajax, Web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page.

Client-side scripting

Client-side scripting generally refers to the class of computer programs on the web that are executed client-side, by the user's web browser, instead of server-side (on the web server).[1] This type of computer programming is an important part of the Dynamic HTML (DHTML) concept, enabling web pages to be scripted; that is, to have different and changing content depending on user input, environmental conditions (such as the time of day), or other variables.

Client-side scripts are often embedded within an HTML or XHTML document (hence known as an “embedded script”), but they may also be contained in a separate file, to which the document (or documents) that use it make reference (hence known as an “external script”). Upon request, the necessary files are sent to the user's computer by the web server (or servers) on which they reside. The user's web browser executes the script, then displays the document, including any visible output from the script. Client-side scripts may also contain instructions for the browser to follow in response to certain user actions, (e.g., clicking a button). Often, these instructions can be followed without further communication with the server.

By viewing the file that contains the script, users may be able to see its source code. Many web authors learn how to write client-side scripts partly by examining the source code for other authors' scripts.

Server-side Scripting

In contrast, server-side scripts, written in languages such as PHP, ASP.NET, Java, ColdFusion, Perl, Ruby, Python, and server-side JavaScript,[2] are executed by the web server when the user requests a document. They produce output in a format understandable by web browsers (usually HTML), which is then sent to the user's computer. The user cannot see the script's source code (unless the author publishes the code separately), and may not even be aware that a script was executed. Documents produced by server-side scripts may, in turn, contain client-side scripts.

Hyperlinks



In computing, a hyperlink is a reference to data that the reader can directly follow either by clicking or by hovering or that is followed automatically.[1] A hyperlink points to a whole document or to a specific element within a document. Hypertext is text with hyperlinks. A software system for viewing and creating hypertext is a hypertext system, and to create a hyperlink is to hyperlink (or simply to link). A user following hyperlinks is said to navigate or browse the hypertext.

URL

The uniform resource locator, abbreviated as URL (also known as web address, particularly when used with HTTP), is a specific character string that constitutes a reference to a resource. In most web browsers, the URL of a web page is displayed on top inside an address bar. An example of a typical URL would be “http://en.example.org/wiki/Main_Page”. A URL is technically a type of uniform resource identifier (URI), but in many technical documents and verbal discussions, URL is often used as a synonym for URI, and this is not considered a problem.[1] URLs are commonly used for web pages (http), but can also be used for file transfer (ftp), email (mailto) and many other applications

1984 Web Design:

What's good is bad; what's bad is good

What You Want to Use:

- Redundancy - repeat key elements
- Obviousness - make interactive elements stand out

What You *Don't* Want

- Subtlety - not being able to clearly distinguish items
- Scarceness - not having multiple ways to interact