

2.1: Markup Languages

Document markup is a notation method that defines how particular pieces of information are meant to be formatted. The term comes from the practice of marking up manuscripts to notate changes that need to be made. Markup in terms of programming languages is used to identify a language that specifies how a document is to appear.

If you have ever used multiple colors of ink or highlighter when making notes and ascribed meaning to those colors for yourself (e.g., yellow highlighter is important, red ink is a definition) then you have already practiced document markup. You are providing additional layers of information along with the written text, in this case visual cues as to the purpose of the written information.

Some popular markup languages are hypertext markup language (HTML), extensible markup language (XML) and extensible hypertext markup language (XHTML). These were each created to fulfill particular needs in defining the layout and structure of the material.



Hypertext markup language is used to aid in the publication of web pages by providing a structure that defines elements like tables, forms, lists and headings, and identifies where different portions of our content begin and end. It can be used to embed other file formats like videos, audio files, documents like PDFs and spreadsheets, among others. HTML is the most relied upon language in the creation of web sites. In this text we will focus on HTML5. While it is technically still in draft form, many proposed elements are already supported by the newer versions of most of the popular browsers.

History

In the beginning, back to the first days of the Internet and ARPA, the primary purpose of creating a page was to share research and information. HTML tags were only meant to provide layout and formatting of a page. As such, early implementations of HTML were somewhat limited as there was little demand for features beyond the basics. Headings, bullets, tables, and color were about all developers had to utilize. As sites were created for other more commercial uses, developers found creative ways of using these tools to get their pages looking more like magazines, advertisements, and what they had drawn on paper. Having been one of those developers, I recall the days of just-get-it-looking-right techniques, splicing page-sized images into tables so graphics were (usually) where we wanted them, nesting tables within tables to create complex layouts, and other methods that violate today's best practices.

Current State

While not formally finalized, many browsers are already supporting a number of features proposed in drafts of HTML5, including things like canvas and media support that greatly improve the browser's ability to process and display complex materials without requiring extensive coding and extensions. In the past, sites that used video and audio players had to integrate support for many players, and would have to include the libraries and formatted files for those systems in their sites. By providing a solution to using these media forms within HTML5, we can improve on the user experience and reduce the efforts necessary to provide them.

While these new features do reduce the amount of programming required to implement higher level elements, and include interactive elements that exceed document markup activities, HTML5 is still considered a markup language.

In these languages, we use tags to ascribe additional meaning to our text, which provide instruction to the browser as to how to display the text on the screen, but are not necessarily displayed to the user. In HTML and XHTML these tags are fixed, or predefined, meaning the names that can be used in the tags are limited to what browsers are able to recognize. In XML, tags are defined by the person creating the content as they are typically used in conjunction with data sources and provide information.

W3C Standards

The [World Wide Web Consortium](#),¹ or W3C, is an international community that supports web development through the creation of open standards that provide the best user experience possible for the widest audience of users. This group of professionals and experts come together to determine how CSS and HTML should operate, what tags should be included as features, and more. The W3C is also your best reference point in determining the accessibility of your site through the use of tools that analyze your code for W3C compliance. These tools confirm if you have fully implemented elements in your code, like providing alternate text descriptions of images in the event that the image cannot load, or the user is visually impaired.

In addition to the creation of accessibility standards, among many others, the W3C also provides tutorials and examples and is likely the most exhaustive reference you will find.

CSS



CSS stands for cascading style sheet, and is used to create rules about the color, font, and layout of our pages. It also determines when those rules are to be used, based on information like the device connecting to the page, or in response to a user's action. CSS can be used by not only HTML but any XML based language. By separating as much of the look and feel of a page from HTML as possible, we actually separate content from appearance. This makes it possible to quickly create several different versions of the appearance of our site, without recreating the content in each version. Our best approach is to use HTML to define the structure (and only structure) of our pages whenever possible, laying the groundwork for CSS to know where to apply the actual style.

History

As HTML grew in popularity, demands on its feature set also grew. Combined with the variety of browser implementations and their varied approaches to rendering and support, creating robust, visually appealing sites involved a significant amount of time and effort. To reduce these, and separate the duties of presentation from those of content, proposals were sought to define a new system of managing these features. CSS was born out of CHSS, or Cascading Hypertext Style Script, and extends our capabilities by allowing us to go far beyond the styling limits of HTML by giving us more power over images, making pages appear more newspaper or magazine-like with layout and typography affects, and reducing load time.

Introduced for public use in 1996, CSS1 contained the ability to apply rules by identifying elements (selectors), and most of the properties still in use today. CSS2 added the ability to adapt for different displays and devices, as well as positioning elements by specific values on the page. CSS2.1 followed with the introduction of additional features, but these were not considered substantial enough to warrant a full version number change.

Current State

While commonly referred to as CSS3, the numbering no longer applies to the language as a whole. The developers have decided to break the language into modules, allowing different aspects of the language to be revised and released independent of one another. This allows for stable modules to stay numbered as they are (since they are not actually changing), while those under more active development can be pushed out as needed. At the moment, most of the “current” modules are at version number 3. Some have not really changed from 2.1, while work on version 4 of selected modules is already underway.

Document Object Model

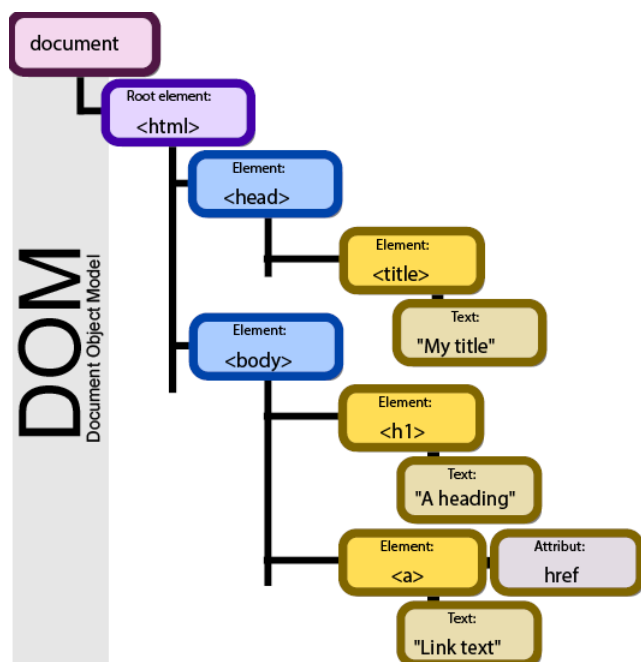


Figure 2.1.1: Document Object Model. (By Birger Eriksson CC-BY-SA-3.0)

Our ability to manipulate and create webpages consistently across formats comes from the document object model API, typically referred to as DOM. This API defines the order and structure of document files as well as how the file is manipulated to create, edit, or remove contents.

The DOM is built to be language and platform independent so any software or programming language can use it to interface with documents. It defines the interface methods and object types that represent elements of documents, the semantics and behavior of attributes of those objects, and also defines how they relate to one another. The DOM, effectively, is what gives rise to the tags we are about to study below. Languages that use the DOM, however, are not required to include all of its features, and may generate additional features of their own.

Figure 20 depicts an example of a document's model in tree format, with nested elements appearing to the right and below their parents. In this example, we are shown an HTML page with a section for the head and the body, which includes a page title and a link as its contents. This structure provides the ability for us to traverse, or move around the document, by referring to an object's name or attribute.

2.1: Markup Languages is shared under a [not declared](#) license and was authored, remixed, and/or curated by LibreTexts.

- 2.1: Markup Languages by Michael Mendez has no license indicated.