

2 Introduction to HTML 4

Contents

2 Introduction to HTML 4	
2.1 What is the World Wide Web?	2
2.1.1 Introduction to URIs	2
2.1.2 Fragment identifiers	3
2.1.3 Relative URIs	3
2.2 What is HTML?	5
2.2.1 A brief history of HTML	5
2.3 HTML 4	6
2.3.1 Internationalization	6
2.3.2 Accessibility	7
2.3.3 Tables	8
2.3.4 Compound documents	8
2.3.5 Style sheets	9
2.3.6 Scripting	9
2.3.7 Printing	10
2.4 Authoring documents with HTML 4	10
2.4.1 Separate structure and presentation	10
2.4.2 Consider universal accessibility to the Web	10
2.4.3 Help user agents with incremental rendering	11

2.1 What is the World Wide Web?

The *World Wide Web (Web)* is a network of information resources. The Web relies on three mechanisms to make these resources readily available to the widest possible audience:

1. A uniform naming scheme for locating resources on the Web (e.g., URIs).
2. Protocols, for access to named resources over the Web (e.g., HTTP).
3. Hypertext, for easy navigation among resources (e.g., HTML).

The ties between the three mechanisms are apparent throughout this specification.

2.1.1 Introduction to URIs

Every resource available on the Web -- HTML document, image, video clip, program, etc. -- has an address that may be encoded by a *Universal Resource Identifier*, or "URI".

URIs typically consist of three pieces:

1. The naming scheme of the mechanism used to access the resource.
2. The name of the machine hosting the resource.
3. The name of the resource itself, given as a path.

Consider the URI that designates the W3C Technical Reports page:

```
http://www.w3.org/TR
```

This URI may be read as follows: There is a document available via the HTTP protocol (see [\[RFC2616\]](#)), residing on the machine `www.w3.org`, accessible via the path `/TR`. Other schemes you may see in HTML documents include `"mailto"` for email and `"ftp"` for FTP.

Here is another example of a URI. This one refers to a user's mailbox:

```
...this is text...  
For all comments, please send email to  
<A href="mailto:joe@someplace.com">Joe Cool</A>.
```

Note. Most readers may be familiar with the term *"URL"* and not the term *"URI"*. URLs form a subset of the more general URI naming scheme.

2.1.2 Fragment identifiers

Some URIs refer to a location within a resource. This kind of URI ends with `"#"` followed by an anchor identifier (called the *fragment identifier*). For instance, here is a URI pointing to an anchor named `section_2`:

```
http://somesite.com/html/top.html#section_2
```

2.1.3 Relative URIs

A *relative URI* doesn't contain any naming scheme information. Its path generally refers to a resource on the same machine as the current document. Relative URIs may contain relative path components (e.g., ".." means one level up in the hierarchy defined by the path), and may contain [fragment identifiers](#).

[Relative URIs](#) are [resolved to full URIs](#) using a base URI. As an example of relative URI resolution, assume we have the base URI "http://www.acme.com/support/intro.html". The relative URI in the following markup for a hypertext link:

```
<A href="suppliers.html">Suppliers</A>
```

would expand to the full URI "http://www.acme.com/support/suppliers.html", while the relative URI in the following markup for an image

```
<IMG src="../../../icons/logo.gif" alt="logo">
```

would expand to the full URI "http://www.acme.com/icons/logo.gif".

In HTML, [URIs](#) are used to:

- Link to another document or resource, (see the [A](#) and [LINK](#) elements).
- Link to an external style sheet or script (see the [LINK](#) and [SCRIPT](#) elements).
- Include an image, object, or applet in a page, (see the [IMG](#), [OBJECT](#), [APPLET](#) and [INPUT](#) elements).
- Create an image map (see the [MAP](#) and [AREA](#) elements).
- Submit a form (see [FORM](#)).
- Create a frame document (see the [FRAME](#) and [IFRAME](#) elements).
- Cite an external reference (see the [Q](#), [BLOCKQUOTE](#), [INS](#) and [DEL](#) elements).
- Refer to metadata conventions describing a document (see the [HEAD](#) element).

Please consult the section on the [URI](#) type for more information about URIs.

2.2 What is HTML?

To publish information for global distribution, one needs a universally understood language, a kind of publishing mother tongue that all computers may potentially understand. The publishing language used by the World Wide Web is HTML (from HyperText Markup Language).

HTML gives authors the means to:

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button.
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
- Include spread-sheets, video clips, sound clips, and other applications directly in their documents.

2.2.1 A brief history of HTML

[HTML](#) was originally developed by Tim Berners-Lee while at [CERN](#), and popularized by the Mosaic browser developed at NCSA. During the course of the 1990s it has blossomed with the explosive growth of the Web. During this time, HTML has been extended in a number of ways. The Web depends on Web page authors and vendors sharing the same conventions for HTML. This has motivated joint work on specifications for HTML.

[HTML 2.0](#) (November 1995, see [\[RFC1866\]](#)) was developed under the aegis of the [Internet Engineering Task Force \(IETF\)](#) to codify common practice in late 1994. [HTML+](#) (1993) and [HTML 3.0](#) (1995, see [\[HTML30\]](#)) proposed much richer versions of HTML. Despite never receiving consensus in standards discussions, these drafts led to the adoption of a range of new

features. The efforts of the World Wide Web Consortium's HTML Working Group to codify common practice in 1996 resulted in [HTML 3.2](#) (January 1997, see [\[HTML32\]](#)). Changes from HTML 3.2 are summarized in [Appendix A](#)

Most people agree that HTML documents should work well across different browsers and platforms. Achieving interoperability lowers costs to content providers since they must develop only one version of a document. If the effort is not made, there is much greater risk that the Web will devolve into a proprietary world of incompatible formats, ultimately reducing the Web's commercial potential for all participants.

Each version of HTML has attempted to reflect greater consensus among industry players so that the investment made by content providers will not be wasted and that their documents will not become unreadable in a short period of time.

HTML has been developed with the vision that all manner of devices should be able to use information on the Web: PCs with graphics displays of varying resolution and color depths, cellular telephones, hand held devices, devices for speech for output and input, computers with high or low bandwidth, and so on.

2.3 HTML 4

HTML 4 extends HTML with mechanisms for style sheets, scripting, frames, embedding objects, improved support for right to left and mixed direction text, richer tables, and enhancements to forms, offering improved accessibility for people with disabilities.

HTML 4.01 is a revision of HTML 4.0 that corrects errors and makes some [changes since the previous revision](#).

2.3.1 Internationalization

This version of HTML has been designed with the help of experts in the field of internationalization, so that documents may be written in every language and be transported easily around the world. This has been accomplished by incorporating [\[RFC2070\]](#), which deals with the internationalization of HTML.

One important step has been the adoption of the ISO/IEC:10646 standard (see [\[ISO10646\]](#)) as the document character set for HTML. This is the world's most inclusive standard dealing with issues of the representation of international characters, text direction, punctuation, and other world language issues.

HTML now offers greater support for diverse human languages within a document. This allows for more effective indexing of documents for search engines, higher-quality typography, better text-to-speech conversion, better hyphenation, etc.

2.3.2 Accessibility

As the Web community grows and its members diversify in their abilities and skills, it is crucial that the underlying technologies be appropriate to their specific needs. HTML has been designed to make Web pages more accessible to those with physical limitations. HTML 4 developments inspired by concerns for accessibility include:

- Better distinction between document structure and presentation, thus encouraging the use of style sheets instead of HTML presentation elements and attributes.
- Better forms, including the addition of access keys, the ability to group form controls semantically, the ability to group [SELECT](#) options semantically, and active labels.
- The ability to markup a text description of an included object (with the [OBJECT](#) element).
- A new client-side image map mechanism (the [MAP](#) element) that allows authors to integrate image and text links.
- The requirement that alternate text accompany images included with the [IMG](#) element and image maps included with the [AREA](#) element.
- Support for the [title](#) and [lang](#) attributes on all elements.
- Support for the [ABBR](#) and [ACRONYM](#) elements.
- A wider range of target media (tty, braille, etc.) for use with style sheets.

- Better tables, including captions, column groups, and mechanisms to facilitate non-visual rendering.
- Long descriptions of tables, images, frames, etc.

Authors who design pages with accessibility issues in mind will not only receive the blessings of the accessibility community, but will benefit in other ways as well: well-designed HTML documents that distinguish structure and presentation will adapt more easily to new technologies.

Note. *For more information about designing accessible HTML documents, please consult [\[WAI\]](#).*

2.3.3 Tables

The new table model in HTML is based on [\[RFC1942\]](#). Authors now have greater control over structure and layout (e.g., column groups). The ability of designers to recommend column widths allows user agents to display table data incrementally (as it arrives) rather than waiting for the entire table before rendering.

Note. *At the time of writing, some HTML authoring tools rely extensively on [tables](#) for formatting, which may easily cause accessibility problems.*

2.3.4 Compound documents

HTML now offers a standard mechanism for embedding generic media objects and applications in HTML documents. The [OBJECT](#) element (together with its more specific ancestor elements [IMG](#) and [APPLET](#)) provides a mechanism for including images, video, sound, mathematics, specialized applications, and other objects in a document. It also allows authors to specify a hierarchy of alternate renderings for user agents that don't support a specific rendering.

2.3.5 Style sheets

Style sheets simplify HTML markup and largely relieve HTML of the responsibilities of presentation. They give both authors and users control over the presentation of documents -- font information, alignment, colors, etc.

Style information can be specified for individual elements or groups of elements. Style information may be specified in an HTML document or in external style sheets.

The mechanisms for associating a style sheet with a document is independent of the style sheet language.

Before the advent of style sheets, authors had limited control over rendering. HTML 3.2 included a number of attributes and elements offering control over alignment, font size, and text color. Authors also exploited tables and images as a means for laying out pages. The relatively long time it takes for users to upgrade their browsers means that these features will continue to be used for some time. However, since style sheets offer more powerful presentation mechanisms, the World Wide Web Consortium will eventually phase out many of HTML's presentation elements and attributes. Throughout the specification elements and attributes at risk are marked as "[deprecated](#)". They are accompanied by examples of how to achieve the same effects with other elements or style sheets.

2.3.6 Scripting

Through scripts, authors may create dynamic Web pages (e.g., "smart forms" that react as users fill them out) and use HTML as a means to build networked applications.

The mechanisms provided to include scripts in an HTML document are independent of the scripting language.

2.3.7 Printing

Sometimes, authors will want to make it easy for users to print more than just the current document. When documents form part of a larger work, the relationships between them can be described using the HTML [LINK](#) element or using W3C's [Resource Description Framework \(RDF\)](#) (see [\[RDF10\]](#)).

2.4 Authoring documents with HTML 4

We recommend that authors and implementors observe the following [general principles](#) when working with HTML 4.

2.4.1 Separate structure and presentation

HTML has its roots in SGML which has always been a language for the specification of structural markup. As HTML matures, more and more of its presentational elements and attributes are being replaced by other mechanisms, in particular style sheets. Experience has shown that separating the structure of a document from its presentational aspects reduces the cost of serving a wide range of platforms, media, etc., and facilitates document revisions.

2.4.2 Consider universal accessibility to the Web

To make the Web more accessible to everyone, notably those with disabilities, authors should consider how their documents may be rendered on a variety of platforms: speech-based browsers, braille-readers, etc. We do not recommend that authors limit their creativity, only that they consider alternate renderings in their design. HTML offers a number of mechanisms to this end (e.g., the [alt](#) attribute, the [accesskey](#) attribute, etc.)

Furthermore, authors should keep in mind that their documents may be reaching a far-off audience with different computer configurations. In order for documents to be interpreted correctly, authors should include in their documents information about the natural language and direction of the text, how the document is encoded, and other issues related to internationalization.

2.4.3 Help user agents with incremental rendering

By carefully designing their tables and making use of new table features in HTML 4, authors can help user agents render documents more quickly. Authors can learn how to design tables for incremental rendering (see the [TABLE](#) element). Implementors should consult the [notes on tables](#) in the appendix for information on incremental algorithms.

[previous](#) [next](#) [contents](#) [elements](#) [attributes](#) [index](#)