What's Wrong with HTML

March 1998 / Cover Story / Weaving a Better Web / What's Wrong with HTML

The main thing that has made HTML so popular -- its simple syntax -- is also what has turned it into our biggest headache. Here are the main trouble spots.

Link tracking.

Web pages move constantly, and Webmasters can't keep up with the changing URLs. Sure, there are automatic link checkers that will tell you when a link is broken. But the real problem is that HTML does not have the notion of a central link repository.

Syntax checking.

HTML obstructs validation because it is not a rigid specification. Rather than checking documents for validity, HTML browsers specifically ignore syntax violations to make the display process more robust.

Extensibility.

Because HTML is not extensible, developers cannot create their own tags to reflect their content's semantic relationships. HTML extensions are either proprietary features of the client (which leads to "browser wars" and unreadable documents) or require approval by a committee. They also fatten the specification because they cannot be imported as needed.

Structure.

HTML lacks support for structure, such as nested information hierarchies. Documents are relatively flat, which limits searching to full-text searches and makes navigation cumbersome. (Wouldn't it be nice to have not just "Back" and "Forward" buttons but be able to traverse hierarchies with "Up" and "Down"? To automatically create site maps and tables of content? To "collapse" a page, showing just headings?)

Content-awareness.

HTML searches have to look at all the content of every page. Therefore, they come up with too many hits. This is because HTML jumbles information and meta-information. Style and logic are hard-coded inside the document. Different views and presentations of the information (e.g., a large-print version) have to be generated by the server. Fancy formatting, such as two-column text, requires hacks by the content developer. (Cascading style sheets are an approach to solve this problem.)

Internationalization.

Support for special and international characters (particularly characters with 2 or more bytes and mathematical formulae) is lacking or, at best, inconsistent in HTML. Where provided, it sometimes breaks when changing platforms.

Data interchange.

Similarly, HTML does not help with automatic, reliable data interchange. Its markup controls the appearance of a document but does not provide for tagged data fields.

Reuse.

HTML makes it difficult to reuse information. For the same data to be published on the Web, printed as a catalog, and maintained in a database, conversion and sometimes manual reformatting is necessary. Worse, this has to be repeated each time the information changes.

Dynamic content.

Today's HTML-created pages don't let you refresh the look of a Web page -- attributes like its color, font properties, font size, or background images -- without loading a new page or invoking Java. Any data stored in Java becomes inaccessible from search engines. For any number of reasons, Java hasn't proven to be a panacea for serving up dynamic Web content.

Object orientation.

Developers are hungry to seize the power of object orientation. Today's HTML tags don't map into an object model that would allow any part of a Web page to be treated as an object.
