HTML

HTML, an initialism for HyperText Markup Language, is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document—by denoting certain text as links, headings, paragraphs, lists, etc.—and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of "tags" that are surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code (such as JavaScript) that can affect the behavior of Web browsers and other HTML processors.

Origins
Tim Berners-Lee

In 1980, physicist Tim Berners-Lee, who was an independent contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use and share documents. In 1989, Berners-Lee and CERN data systems engineer Robert Cailliau each submitted separate proposals for an Internet-based hypertext system providing similar functionality. The following year, they collaborated on a joint proposal, the WorldWideWeb (W3) project, which was accepted by CERN. In his personal notes from 1990 he lists, "some of the many areas in which hypertext is used", and puts an encyclopaedia first.

First specifications

The first publicly available description of HTML was a document called HTML Tags, first mentioned on the Internet by Berners-Lee in late 1991. It describes 22 elements comprising the initial, relatively simple design of HTML. Thirteen of these elements still exist in HTML 4. HTML is a text and image formatting language used by web browsers to dynamically format web pages. The semantics of many of its tags can be traced to early text formatting languages such as that used by the RUNOFF command developed in the early 1960s for the CTSS (Compatible Time-Sharing System) operating system, and its formatting commands were derived from the commands used by typesetters to manually format documents.

Berners-Lee considered HTML to be, at the time, an application of SGML, but it was not formally defined as such until the mid-1993 publication, by the IETF, of the first proposal for an HTML specification: Berners-Lee and Dan Connolly's "Hypertext Markup Language (HTML)" Internet-Draft, which included an SGML Document Type Definition to define the grammar. The draft expired after six months, but was notable for its acknowledgment of the NCSA Mosaic browser's custom tag for embedding in-line images, reflecting the IETF's philosophy of basing standards on successful prototypes. Similarly, Dave Raggett's competing Internet-Draft, "HTML+ (Hypertext Markup Format)", from late 1993, suggested standardizing already-implemented features like tables and fill-out forms.

HTML -
www.viplavkambli.com
After the HTML and HTML+ drafts expired in early 1994, the IETF created an HTML Working Group, which in 1995 completed "HTML 2.0", the first HTML specification intended to be treated as a standard against which future implementations should be based. Published as Request for Comments 1866, HTML 2.0 included ideas from the HTML and HTML+ drafts. There was no "HTML 1.0"; the 2.0 designation was intended to distinguish the new edition from previous drafts.

Further development under the auspices of the IETF was stalled by competing interests. Since 1996, the HTML specifications have been maintained, with input from commercial software vendors, by the World Wide Web Consortium (W3C). However, in 2000, HTML also became an international standard (ISO/IEC 15445:2000). The last HTML specification published by the W3C is the HTML 4.01 Recommendation, published in late 1999. Its issues and errors were last acknowledged by errata published in 2001.

Version history of the standard

HTML

* HTML and HTML 5
* Dynamic HTML
* XHTML
* XHTML Mobile Profile and C-HTML
* Character encodings
* Font family
* HTML editor
* HTML element
* HTML scripting
* Layout engine
* Quirks mode
* Style sheets
* Unicode and HTML
* W3C
* Web colors
* Comparison of
  o document markup languages
  o web browsers

HTML -
www.viplavkambli.com
o layout engines for
   + HTML
   + HTML 5
   + Non-standard HTML
   + XHTML

This box: view • talk • edit

HTML version timeline

November 1995
   HTML 2.0 was published as IETF RFC 1866. Supplemental RFCs added capabilities:

   * November 1995: RFC 1867 (form-based file upload)
   * May 1996: RFC 1942 (tables)
   * August 1996: RFC 1980 (client-side image maps)
   * January 1997: RFC 2070 (internationalization)

In June 2000, all of these were declared obsolete/historic by RFC 2854.

January 1997
   HTML 3.2 was published as a W3C Recommendation. It was the first version developed and
   standardized exclusively by the W3C, as the IETF had closed its HTML Working Group in September
   1997.

   HTML 3.2 dropped math formulas entirely, reconciled overlap among various proprietary
   extensions, and adopted most of Netscape's visual markup tags. Netscape's blink element and
   Microsoft's marquee element were omitted due to a mutual agreement between the two companies. A
   markup for mathematical formulas similar to that in HTML wasn't standardized until 14 months later
   in MathML.

December 1997
   HTML 4.0 was published as a W3C Recommendation. It offers three "flavors":

   HTML -
   www.viplavkambli.com
* Strict, in which deprecated elements are forbidden,
* Transitional, in which deprecated elements are allowed,
* Frameset, in which mostly only frame related elements are allowed;

Initially code-named "Cougar", HTML 4.0 adopted many browser-specific element types and attributes, but at the same time sought to phase out Netscape's visual markup features by marking them as deprecated in favor of style sheets.

April 1998

HTML 4.0 was reissued with minor edits without incrementing the version number.

December 1999

HTML 4.01 was published as a W3C Recommendation. It offers the same three flavors as HTML 4.0, and its last errata were published May 12, 2001.

May 2000


As of mid-2008, HTML 4.01 and ISO/IEC 15445:2000 are the most recent versions of HTML. Development of the parallel, XML-based language XHTML occupied the W3C's HTML Working Group through the early and mid-2000s.

Drafts

October 1991

HTML Tags, an informal CERN document listing twelve HTML tags, was first mentioned in public.


July 1993

Hypertext Markup Language was published by the IETF as an Internet-Draft (a rough proposal for a standard). It expired in January 1994.

November 1993

HTML -
www.viplavkambli.com
HTML+ was published by the IETF as an Internet-Draft and was a competing proposal to the Hypertext Markup Language draft. It expired in May 1994.

April 1995 (authored March 1995)

HTML 3.0 was proposed as a standard to the IETF, but the proposal expired five months later without further action. It included many of the capabilities that were in Raggett's HTML+ proposal, such as support for tables, text flow around figures, and the display of complex mathematical formulas.

A demonstration appeared in W3C's own Arena browser. HTML 3.0 did not succeed for several reasons. The pace of browser development, as well as the number of interested parties, had outstripped the resources of the IETF. Netscape continued to introduce HTML elements that specified the visual appearance of documents, contrary to the goals of the newly-formed W3C, which sought to limit HTML to describing logical structure. Microsoft, a newcomer at the time, played to all sides by creating its own tags, implementing Netscape's elements for compatibility, and supporting W3C features such as Cascading Style Sheets.

January 2008

HTML5 was published as a Working Draft by the W3C.

Although its syntax closely resembles that of SGML, HTML 5 has abandoned any attempt to be an SGML application, and has explicitly defined its own "html" serialization, in addition to an alternative XML-based XHTML 5 serialization.

XHTML versions

Main article: XHTML

XHTML is a separate language that began as a reformulation of HTML 4.01 using XML 1.0. It continues to be developed:

* XHTML 1.0, published January 26, 2000 as a W3C Recommendation, later revised and republished August 1, 2002. It offers the same three flavors as HTML 4.0 and 4.01, reformulated in XML, with minor restrictions.

* XHTML 1.1, published May 31, 2001 as a W3C Recommendation. It is based on XHTML 1.0 Strict, but includes minor changes, can be customized, and is reformulated using modules from Modularization of XHTML, which was published April 10, 2001 as a W3C Recommendation.

* XHTML 2.0, is still a W3C Working Draft. XHTML 2.0 is incompatible with XHTML 1.x and, therefore, would be more accurate to characterize as an XHTML-inspired new language than an

HTML -
www.viplavkambli.com
update to XHTML 1.x.

* XHTML5, which is an update to XHTML 1.x, is being defined alongside HTML5 in the HTML5 draft.

HTML markup

This section does not cite any references or sources. Please help improve this article by adding citations to reliable sources. Unverifiable material may be challenged and removed. (January 2009)

HTML markup consists of several key components, including elements (and their attributes), character-based data types, and character references and entity references. Another important component is the document type declaration, which specifies the Document Type Definition. As of HTML 5, no Document Type Definition will need to be specified, and will only determine the layout mode.

The Hello world program, a common computer program employed for comparing programming languages, scripting languages, and markup languages is made of 9 lines of code in HTML, albeit line breaks and the <!DOCTYPE> tag, or the document type declaration, are optional:

```html
<!DOCTYPE html>
<html>
<head>
<title>Hello HTML</title>
</head>
<body>
<p>Hello World!</p>
</body>
</html>
```

This Document Type Declaration is in HTML5 format.

If the <!DOCTYPE> tag is not included, the layout mode defaults to "quirks mode."

Elements

This section does not cite any references or sources. Please help improve this article by adding citations to reliable sources. Unverifiable material may be challenged and removed. (January 2009)
See HTML elements for more detailed descriptions.

Elements are the basic structure for HTML markup. Elements have two basic properties: attributes and content. Each attribute and each element’s content has certain restrictions that must be followed for an HTML document to be considered valid. An element usually has a start tag (e.g. `<element-name>`) and an end tag (e.g. `</element-name>`). The element’s attributes are contained in the start tag and content is located between the tags (e.g. `<element-name attribute="value">Content</element-name>`). Some elements, such as `<br>`, do not have any content and must not have a closing tag. Listed below are several types of markup elements used in HTML.

Structural markup describes the purpose of text. For example, `<h2>Golf</h2>` establishes "Golf" as a second-level heading, which would be rendered in a browser in a manner similar to the "HTML markup" title at the start of this section. Structural markup does not denote any specific rendering, but most Web browsers have standardized on how elements should be formatted. Text may be further styled with Cascading Style Sheets (CSS).

Presentational markup describes the appearance of the text, regardless of its function. For example `<b>boldface</b>` indicates that visual output devices should render "boldface" in bold text, but gives no indication what devices which are unable to do this (such as aural devices that read the text aloud) should do. In the case of both `<b>bold</b>` and `<i>italic</i>`, there are elements which usually have an equivalent visual rendering but are more semantic in nature, namely `<strong>strong emphasis</strong>` and `<em>emphasis</em>` respectively. It is easier to see how an aural user agent should interpret the latter two elements. However, they are not equivalent to their presentational counterparts: it would be undesirable for a screen-reader to emphasize the name of a book, for instance, but on a screen such a name would be italicized. Most presentational markup elements have become deprecated under the HTML 4.0 specification, in favor of CSS based style design.

Hypertext markup links parts of the document to other documents. HTML up through version XHTML 1.1 requires the use of an anchor element to create a hyperlink in the flow of text: `<a>Wikipedia</a>`. However, the href attribute must also be set to a valid URL so for example the HTML code, `<a href="http://en.wikipedia.org/">Wikipedia</a>`, will render the word "Wikipedia" as a hyperlink. To link on an image, the anchor tag use the following syntax: `<a href="url"><img src="image.gif" alt="alternative text" width="50" height="50"></a>`

Attributes

Most of the attributes of an element are name-value pairs, separated by "="; and written within the start tag of an element, after the element’s name. The value may be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML). Leaving attribute values unquoted is considered unsafe. In contrast with name-value pair
attributes, there are some attributes that affect the element simply by their presence in the start tag of the element (like the ismap attribute for the img element).

Most elements can take any of several common attributes:

* The id attribute provides a document-wide unique identifier for an element. This can be used by stylesheets to provide presentational properties, by browsers to focus attention on the specific element, or by scripts to alter the contents or presentation of an element.

* The class attribute provides a way of classifying similar elements for presentation purposes. For example, an HTML document might use the designation class="notation" to indicate that all elements with this class value are subordinate to the main text of the document. Such elements might be gathered together and presented as footnotes on a page instead of appearing in the place where they occur in the HTML source.

* An author may use the style non-attributal codes presentational properties to a particular element. It is considered better practice to use an element’s id or class attributes to select the element with a stylesheet, though sometimes this can be too cumbersome for a simple ad hoc application of styled properties.

* The title attribute is used to attach subtextual explanation to an element. In most browsers this attribute is displayed as what is often referred to as a tooltip.

The generic inline element span can be used to demonstrate these various attributes:

  <span id="anId" class="aClass" style="color:blue;" title="Hypertext Markup Language">HTML</span>

This example displays as HTML; in most browsers, pointing the cursor at the abbreviation should display the title text "Hypertext Markup Language."

Most elements also take the language-related attributes lang and dir.

Character and entity references

This section does not cite any references or sources. Please help improve this article by adding citations to reliable sources. Unverifiable material may be challenged and removed. (January 2009)
See also: List of XML and HTML character entity references

As of version 4.0, HTML defines a set of 252 character entity references and a set of 1,114,050 numeric character references, both of which allow individual characters to be written via simple markup, rather than literally. A literal character and its markup counterpart are considered equivalent and are rendered identically.

The ability to "escape" characters in this way allows for the characters < and & (when written as &lt; and &amp;, respectively) to be interpreted as character data, rather than markup. For example, a literal < normally indicates the start of a tag, and & normally indicates the start of a character entity reference or numeric character reference; writing it as &amp; or &#x26; or &#38; allows & to be included in the content of elements or the values of attributes. The double-quote character ("), when used to quote an attribute value, must also be escaped as &quot; or &#x22; or &#34; when it appears within the attribute value itself. The single-quote character ('), when used to quote an attribute value, must also be escaped as &apos; or &#39; (should NOT be escaped as &apos; except in XHTML documents) when it appears within the attribute value itself. However, since document authors often overlook the need to escape these characters, browsers tend to be very forgiving, treating them as markup only when subsequent text appears to confirm that intent.

Escaping also allows for characters that are not easily typed or that aren't even available in the document's character encoding to be represented within the element and attribute content. For example, the acute-accented e (é), a character typically found only on Western European keyboards, can be written in any HTML document as the entity reference &eacute; or as the numeric references &#233; or &#x00e9;. The characters comprising those references (that is, the &, the ;, the letters in eacute, and so on) are available on all keyboards and are supported in all character encodings, whereas the literal é is not.

Data types

This section does not cite any references or sources. Please help improve this article by adding citations to reliable sources. Unverifiable material may be challenged and removed. (January 2009)

HTML defines several data types for element content, such as script data and stylesheet data, and a plethora of types for attribute values, including IDs, names, URIs, numbers, units of length, languages, media descriptors, colors, character encodings, dates and times, and so on. All of these data types are specializations of character data.

Document type declaration

This section does not cite any references or sources. Please help improve this article by adding citations to reliable sources. Unverifiable material may be challenged and removed. (January 2009)
HTML documents are required to start with a Document Type Declaration (informally, a “doctype”). In browsers, the function of the doctype is selecting the rendering mode—particularly to avoid the quirks mode.

The original purpose of the doctype is to enable validation based on Document Type Definition (DTD) with SGML tools. The DTD to which the DOCTYPE refers contains machine-readable grammar specifying the permitted and prohibited content for a document conforming to such a DTD. Browsers do not read the DTD, however. HTML5 validation is not DTD-based, so in HTML5 the doctype does not refer to a DTD.

An example of an HTML 4 doctype:

```xml
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

This declaration references the Strict DTD of HTML 4.01, which does not have presentational elements like `<font>`, leaving formatting to Cascading Style Sheets and the span and div tags. SGML-based validators read the DTD in order to properly parse the document and to perform validation. In modern browsers, the HTML 4.01 Strict doctype activates standards layout mode for CSS as opposed to quirks mode.

In addition, HTML 4.01 provides Transitional and Frameset DTDs. The Transitional DTD was intended to gradually phase in the changes made in the Strict DTD, while the Frameset DTD was intended for those documents which contained frames.

Semantic HTML

This section does not cite any references or sources. Please help improve this article by adding citations to reliable sources. Unverifiable material may be challenged and removed. (January 2009)

There is no official specification called "Semantic HTML". Semantic HTML refers to a practice to create documents with HTML that contain only the author's intended meaning, without any reference to how this meaning is presented. It is part of the idea of separation of presentation and content. For example, the emphasis element (`<em>`) and the italics element (`<i>`) are functionally identical at default, but have two different meanings. The meaning behind the markup is important in semantic HTML.

CSS is typically used to create presentational effects for semantic HTML. With CSS, a designer can create the content with the appropriate, semantic markup and then use CSS to alter the appearance of their markup. The idea is that strictly presentational elements (such as italics or bold lettering) should not be used in the code and that any use of presentational elements should have some

HTML -
www.viplavkambli.com
meaning attached to it, such as a citation.

The benefit of semantic HTML is that presentation of the text will be consistently applied, so long as the proper markup is consistently applied. Moreover, it allows for easy change of presentation, by simply editing the style sheets as well as being able to transfer the text from one site to another.

The main disadvantage is that HTML does not contain enough markup tags to describe every single conceivable description or meaning. As such, people will typically use the division (\textless div\textgreater) tag along with a set of pre-defined classes or IDs to properly mark up text for their intended meaning. If the designer has a glut of sections or meanings that don't fit well with HTML's markup, they may be forced to use a lot of division (\textless div\textgreater) tags, which could easily obfuscate the code.

Delivery of HTML

HTML documents can be delivered by the same means as any other computer file; however, they are most often delivered either by HTTP from a Web server or by e-mail.

HTTP

The World Wide Web is composed primarily of HTML documents transmitted from Web servers to Web browsers using the Hypertext Transfer Protocol (HTTP). However, HTTP is used to serve images, sound, and other content in addition to HTML. To allow the Web browser to know how to handle each document it receives, other information is transmitted along with the document. This metadata usually includes the MIME type (e.g. text/html or application/xhtml+xml) and the character encoding (see Character encodings in HTML).

In modern browsers, the MIME type that is sent with the HTML document may affect how the document is initially interpreted. A document sent with the XHTML MIME type is expected to be well-formed XML, and syntax errors may cause the browser to fail to render it. The same document sent with the HTML MIME type might be displayed successfully, since some browsers are more lenient with HTML.

The W3C recommendations state that XHTML 1.0 documents that follow guidelines set forth in the recommendation's Appendix C may be labeled with either MIME Type. The current XHTML 1.1 Working Draft also states that XHTML 1.1 documents should be labeled with either MIME type.

HTML e-mail

This section does not cite any references or sources. Please help improve this article by adding

HTML -
www.viplavkambli.com
citations to reliable sources. Unverifiable material may be challenged and removed. (January 2009)

Main article: HTML e-mail

Most graphical e-mail clients allow the use of a subset of HTML (often ill-defined) to provide formatting and semantic markup not available with plain text. This may include typographic information like coloured headings, emphasized and quoted text, inline images and diagrams. Many such clients include both a GUI editor for composing HTML e-mail messages and a rendering engine for displaying them. Use of HTML in e-mail is controversial because of compatibility issues, because it can help disguise phishing attacks, because it can confuse spam filters and because the message size is larger than plain text.

Naming conventions

The most common filename extension for files containing HTML is .html. A common abbreviation of this is .htm, which originated because some early operating systems and file systems, such as DOS and FAT, limited file extensions to three letters. See 8.3 filename.

HTML Application

Main article: HTML Application

An HTML Application (HTA; file extension ".hta") is a Microsoft Windows application that uses HTML and Dynamic HTML in a browser to provide the application's graphical interface. A regular HTML file is confined to the security model of the web browser, communicating only to web servers and manipulating only webpage objects and site cookies. An HTA runs as a fully trusted application and therefore has more privileges, like creation/editing/removal of files and Windows Registry entries. Because they operate outside the browser's security model, HTAs cannot be executed via HTTP, but must be downloaded (just like an EXE file) and executed from local file system.

Current flavors of HTML

Since its inception, HTML and its associated protocols gained acceptance relatively quickly. However, no clear standards existed in the early years of the language. Though its creators originally conceived of HTML as a semantic language devoid of presentation details, practical uses pushed many presentational elements and attributes into the language, driven largely by the various browser vendors. The latest standards surrounding HTML reflect efforts to overcome the sometimes chaotic development of the language and to create a rational foundation for building both meaningful and well-presented documents. To return HTML to its role as a semantic language, the W3C has developed style languages such as CSS and XSL to shoulder the burden of presentation. In conjunction, the HTML specification has slowly reined in the presentational elements.

HTML -
www.viplavkambli.com
There are two axes differentiating various flavors of HTML as currently specified: SGML-based HTML versus XML-based HTML (referred to as XHTML) on one axis, and strict versus transitional (loose) versus frameset on the other axis.

**SGML-based versus XML-based HTML**

One difference in the latest HTML specifications lies in the distinction between the SGML-based specification and the XML-based specification. The XML-based specification is usually called XHTML to distinguish it clearly from the more traditional definition; however, the root element name continues to be 'html' even in the XHTML-specified HTML. The W3C intended XHTML 1.0 to be identical to HTML 4.01 except where limitations of XML over the more complex SGML require workarounds. Because XHTML and HTML are closely related, they are sometimes documented in parallel. In such circumstances, some authors conflate the two names as (X)HTML or X(HTML).

Like HTML 4.01, XHTML 1.0 has three sub-specifications: strict, loose, and frameset.

Aside from the different opening declarations for a document, the differences between an HTML 4.01 and XHTML 1.0 document—in each of the corresponding DTDs—are largely syntactic. The underlying syntax of HTML allows many shortcuts that XHTML does not, such as elements with optional opening or closing tags, and even EMPTY elements which must not have an end tag. By contrast, XHTML requires all elements to have an opening tag or a closing tag. XHTML, however, also introduces a new shortcut: an XHTML tag may be opened and closed within the same tag, by including a slash before the end of the tag like this: `<br/>`. The introduction of this shorthand, which is not used in the SGML declaration for HTML 4.01, may confuse earlier software unfamiliar with this new convention.

To understand the subtle differences between HTML and XHTML, consider the transformation of a valid and well-formed XHTML 1.0 document that adheres to Appendix C (see below) into a valid HTML 4.01 document. To make this translation requires the following steps:

1. The language for an element should be specified with a lang attribute rather than the XHTML xml:lang attribute. XHTML uses XML's built in language-defining functionality attribute.
2. Remove the XML namespace (xmlns=URI). HTML has no facilities for namespaces.
3. Change the document type declaration from XHTML 1.0 to HTML 4.01. (see DTD section for further explanation).
4. If present, remove the XML declaration. (Typically this is: `<?xml version="1.0" encoding="utf-8"?>`).
5. Ensure that the document's MIME type is set to text/html. For both HTML and XHTML, this comes from the HTTP Content-Type header sent by the server.
6. Change the XML empty-element syntax to an HTML style empty element (`<br/>` to `<br>`).

HTML -
www.viplavkambli.com
Those are the main changes necessary to translate a document from XHTML 1.0 to HTML 4.01. To translate from HTML to XHTML would also require the addition of any omitted opening or closing tags. Whether coding in HTML or XHTML it may just be best to always include the optional tags within an HTML document rather than remembering which tags can be omitted.

A well-formed XHTML document adheres to all the syntax requirements of XML. A valid document adheres to the content specification for XHTML, which describes the document structure.

The W3C recommends several conventions to ensure an easy migration between HTML and XHTML (see HTML Compatibility Guidelines). The following steps can be applied to XHTML 1.0 documents only:

* Include both xml:lang and lang attributes on any elements assigning language.
* Use the empty-element syntax only for elements specified as empty in HTML.
* Include an extra space in empty-element tags: for example <br /> instead of <br/>
* Include explicit close tags for elements that permit content but are left empty (for example, <div></div>, not </div>).
* Omit the XML declaration.

By carefully following the W3C’s compatibility guidelines, a user agent should be able to interpret the document equally as HTML or XHTML. For documents that are XHTML 1.0 and have been made compatible in this way, the W3C permits them to be served either as HTML (with a text/html MIME type), or as XHTML (with an application/xhtml+xml or application/xml MIME type). When delivered as XHTML, browsers should use an XML parser, which adheres strictly to the XML specifications for parsing the document’s contents.

Transitional versus Strict

This section may be confusing or unclear to readers. Please help clarify the article; suggestions may be found on the talk page. (February 2009)

This section does not cite any references or sources. Please help improve this article by adding citations to reliable sources. Unverifiable material may be challenged and removed. (March 2009)

The latest SGML-based specification HTML 4.01 and the earliest XHTML version include three sub-specifications: Strict, Transitional (once called Loose), and Frameset. The Strict variant represents the standard proper, whereas the Transitional and Frameset variants were developed to assist in the transition from earlier versions of HTML (including HTML 3.2). The Transitional and Frameset variants allow for presentational markup whereas the Strict variant encourages the use of style sheets through

HTML -
www.viplavkambli.com
its omission of most presentational markup.

The primary differences which make the Transitional variant more permissive than the Strict variant (the differences are the same in HTML 4 and XHTML 1.0) are:

* A looser content model
  
  o Inline elements and plain text (#PCDATA) are allowed directly in: body, blockquote, form, noscript and noframes

* Presentation related elements
  
  o underline (u)
  o strike-through (s)
  o center
  o font
  o basefont

* Presentation related attributes
  
  o background and bgcolor attributes for body element.
  o align attribute on div, form, paragraph (p), and heading (h1...h6) elements
  o align, noshade, size, and width attributes on hr element
  o align, border, vspace, and hspace attributes on img and object elements
  o align attribute on legend and caption elements
  o align and bgcolor on table element
  o nowrap, bgcolor, width, height on td and th elements
  o bgcolor attribute on tr element
  o clear attribute on br element
  o compact attribute on dl, dir and menu elements
  o type, compact, and start attributes on ol and ul elements
  o type and value attributes on li element
  o width attribute on pre element

* Additional elements in Transitional specification
  
  o menu list (no substitute, though unordered list is recommended; may return in XHTML 2.0 specification)
  o dir list (no substitute, though unordered list is recommended)

HTML -
www.viplavkambli.com
o isindex (element requires server-side support and is typically added to documents server-side)
  o applet (deprecated in favor of object element)
* The language attribute on script element (presumably redundant with type attribute, though this is maintained for legacy reasons).
* Frame related entities
  o frameset element (used in place of body for frameset DTD)
  o frame element
  o iframe
  o noframes
  o target attribute on anchor, client-side image-map (imagemap), link, form, and base elements

Frameset versus transitional

In addition to the above transitional differences, the frameset specifications (whether XHTML 1.0 or HTML 4.01) specifies a different content model:

<html>
<head>
<title></title>
<!-- other head elements -->
</head>

<!-- frameset replaces body -->
<frameset>

<!-- frame definitions -->
<frame></frame>
<!-- , ... -->

<!-- optional: alternate page body for frames-incompatible user agents -->
<noframes>

HTML -
www.viplavkambli.com
Summary of flavors

As this list demonstrates, the loose flavors of the specification are maintained for legacy support. However, contrary to popular misconceptions, the move to XHTML does not imply a removal of this legacy support. Rather the X in XML stands for extensible and the W3C is modularizing the entire specification and opening it up to independent extensions. The primary achievement in the move from XHTML 1.0 to XHTML 1.1 is the modularization of the entire specification. The strict version of HTML is deployed in XHTML 1.1 through a set of modular extensions to the base XHTML 1.1 specification. Likewise someone looking for the loose (transitional) or frameset specifications will find similar extended XHTML 1.1 support (much of it is contained in the legacy or frame modules). The modularization also allows for separate features to develop on their own timetable. So for example XHTML 1.1 will allow quicker migration to emerging XML standards such as MathML (a presentational and semantic math language based on XML) and XForms—a new highly advanced web-form technology to replace the existing HTML forms.

In summary, the HTML 4.01 specification primarily reined in all the various HTML implementations into a single clear written specification based on SGML. XHTML 1.0, ported this specification, as is, to the new XML defined specification. Next, XHTML 1.1 takes advantage of the extensible nature of XML and modularizes the whole specification. XHTML 2.0 will be the first step in adding new features to the specification in a standards-body-based approach.

Hypertext features not in HTML

HTML lacks some of the features found in earlier hypertext systems, such as typed links, source tracking, fat links, and more. Even some hypertext features that were in early versions of HTML have been ignored by most popular web browsers until recently, such as the link element and in-browser Web page editing.

Sometimes Web services or browser manufacturers remedy these shortcomings. For instance, wikis and content management systems allow surfers to edit the Web pages they visit.