XHTML™ 1.0: The Extensible HyperText Markup Language

A Reformulation of HTML 4.0 in XML 1.0

W3C Working Draft 5th May 1999

This version:
http://www.w3.org/TR/1999/xhtml1-19990505
Previous versions:
http://www.w3.org/TR/1999/WD-html-in-xml-19990304
http://www.w3.org/TR/1999/WD-html-in-xml-19990224
http://www.w3.org/TR/1998/WD-html-in-xml-19981205
Newest version:
http://www.w3.org/TR/xhtml1
Also available for local browsing as a Zipped archive
Authors:
See acknowledgements.

Document Status and errata

Status of this document

This document is a Working Draft of the World Wide Web Consortium. Please send detailed comments on this document to www-html-editor@w3.org before 2359Z, June 1st 1999. We cannot guarantee a personal response, but we will try when it is appropriate. Public discussion on HTML features takes place on the mailing list www-html@w3.org (archive). The W3C staff contact for work on HTML is Dave Raggett.

This document has been produced as part of the W3C HTML Activity. The goals of the HTML Working Group *(members only)* are discussed in the HTML Working Group charter *(members only)*.

This specification is a revision of the working draft dated 4th March 1999 incorporating suggestions received during review, comments and further deliberations of the W3C HTML Working Group. The detailed differences are available for reviewers to compare.

Publication as a Working Draft does not imply endorsement by the W3C membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite W3C Drafts as other than "work in progress".

Abstract

This specification defines XHTML 1.0, a reformulation of HTML 4.0 as an XML 1.0 application, and three DTDs corresponding to the ones defined by HTML 4.0. The semantics of the elements and their attributes are defined in the W3C Recommendation for HTML 4.0. These semantics provide the foundation for future extensibility of XHTML. Compatibility with existing HTML user agents is possible by following a small set of guidelines.

Contents

1. What is XHTML?

XHTML is a reformulation of HTML 4.0 [HTML] as an application of XML 1.0 [XML].

XHTML 1.0 specifies three DTDs corresponding to the HTML 4.0 DTDs, and an XML namespace identified by a unique URI.

XHTML 1.0 is the basis for a family of future document types that extend and subset HTML. This idea is discussed in more detail in the section on Future Directions.

1.1 What is HTML 4.0?

HTML 4.0 [HTML] is an SGML (Standard Generalized Markup Language) application conforming to International Standard ISO 8879, and is widely regarded as the standard publishing language of the World Wide Web.

SGML is a language for describing markup languages, particularly those used in electronic document exchange, document management, and document publishing. HTML is an example of a language defined in SGML.

SGML has been around since the middle 1980's and has remained quite stable. Much of this stability stems from the fact that the language is both feature-rich and flexible. This flexibility, however, comes at a price, and that price is a level of complexity that has inhibited its adoption in a diversity of environments, including the World Wide Web.

HTML, as originally conceived, was to be a language for the exchange of scientific and other technical documents, suitable for use by non-document specialists. HTML addressed the problem of SGML complexity by specifying a small set of structural and semantic tags suitable for authoring relatively simple documents. In addition to simplifying the document structure, HTML added support for hypertext. Multimedia capabilities were added later.

In a remarkably short space of time, HTML became wildly popular and rapidly outgrew its original purpose. Since HTML's inception, there has been rapid invention of new elements for use within HTML (as a standard) and for adapting HTML to vertical, highly specialized, markets. This plethora of new elements has led to compatibility problems for documents across different platforms.

As the heterogeneity of both software and platforms rapidly proliferate, it is clear that the suitability of 'classic' HTML 4.0 for use on these platforms is somewhat limited.

1.2 What is XML?

XML™ is the shorthand for Extensible Markup Language, and is an acronym of eXtensible Markup Language [XML].

XML was conceived as a means of regaining the power and flexibility of SGML without most of its complexity. Although a restricted form of SGML, XML nonetheless preserves most of SGML's power and richness, and yet still retains all of SGML's commonly used features.

While retaining these beneficial features, XML removes many of the more complex features of SGML that make the authoring and design of suitable software both difficult and costly.

1.3 Why the need for XHTML?

There are two major reasons for content developers to adopt XHTML:

First, XHTML is designed to be extensible. This extensibility relies upon the XML requirement that documents be well-formed. Under SGML, the addition of a new group of elements would mean alteration of the entire DTD. In an XML-based DTD, all that is required is that the new set of elements be internally consistent and well-formed to be added to an existing DTD. The greatly eases the development and integration of new collections of elements.

Second, XHTML is designed for portability. There will be increasing use of non-desktop user agents to access Internet documents. Some estimates indicate that by the year 2002, 75% of Internet document viewing will be carried out on these alternate platforms. In most cases these platforms will not have the computing power of a desktop platform, and will not be designed to accommodate ill-formed HTML as current user agents tend to do. Indeed if these user agents do not receive well-formed XHTML, they may simply not display the document.

2. Definitions

2.1 Terminology

The following terms are used in this specification. These terms extend the definitions in [RFC2119] in ways based upon similar definitions in ISO/IEC 9945-1:1990 [POSIX.1]:

Implementation-defined
: A value or behavior is implementation-defined when it is left to the implementation to define [and document] the corresponding requirements for correct document construction.

May
: With respect to implementations, the word "may" is to be interpreted as an optional feature that is not required in this specification but can be provided. With respect to Document Conformance, the word "may" means that the optional feature must not be used. The term "optional" has the same definition as "may".

Must
: In this specification, the word "must" is to be interpreted as a mandatory requirement on the implementation or on Strictly Conforming XHTML Documents, depending upon the context. The term "shall" has the same definition as "must".

Reserved
: A value or behavior is unspecified, but it is not allowed to be used by Conforming Documents nor to be supported by a Conforming User Agents.

Should
: With respect to implementations, the word "should" is to be interpreted as an implementation recommendation, but not a requirement. With respect to

documents, the word "should" is to be interpreted as recommended programming practice for documents and a requirement for Strictly Conforming XHTML Documents.

Supported

Certain facilities in this specification are optional. If a facility is supported, it behaves as specified by this specification.

Unspecified

When a value or behavior is unspecified, the specification defines no portability requirements for a facility on an implementation even when faced with a document that uses the facility. A document that requires specific behavior in such an instance, rather than tolerating any behavior when using that facility, is not a Strictly Conforming XHTML Document.

2.2 General Terms

Attribute

An attribute is a parameter to an element declared in the DTD. An attribute's type and value range, including a possible default value, are defined in the DTD.

DTD

A DTD, or document type definition, is a collection of XML declarations that, as a collection, defines the legal structure, elements, and attributes that are available for use in a document that complies to the DTD.

Document

A document is a stream of data that, after being combined with any other streams it references, is structured such that it holds information contained within elements that are organized as defined in the associated DTD. See Document Conformance for more information.

Element

An element is a document structuring unit declared in the DTD. The element's content model is defined in the DTD, and additional semantics may be defined in the prose description of the element.

Facilities

Functionality includes elements, attributes, and the semantics associated with those elements and attributes. An implementation supporting that functionality is said to provide the necessary facilities.

Implementation

An implementation is a system that provides collection of facilities and services that supports this specification. See User Agent Conformance for more information.

Parsing

Parsing is the act whereby a document is scanned, and the information contained within the document is filtered into the context of the elements in which the information is structured.

Rendering

Rendering is the act whereby the information in a document is presented. This presentation is done in the form most appropriate to the environment (e.g. aurally, visually, in print).

User Agent

A user agent is an implementation that retrieves and processes XHTML documents. See User Agent Conformance for more information.

Validation

Validation is a process whereby documents are verified against the associated DTD, ensuring that the structure, use of elements, and use of attributes are consistent with the definitions in the DTD.

Well-formed

A document is well-formed when it is structured according to the rules defined in Section 2.1 of the XML 1.0 Recommendation [XML]. Basically, this definition states that elements, delimited by their start and end tags, are nested properly within one another.

3. Normative Definition of XHTML 1.0

3.1 Document Conformance

This version of XHTML provides a definition of strictly conforming XHTML documents, which are restricted to tags and attributes from the XHTML 1.0 namespace. See Section 3.1.2 for information on using XHTML with other namespaces, for instance, to include metadata expressed in RDF within XHTML documents.

3.1.1 Strictly Conforming Documents

A Strictly Conforming XHTML Document is a document that requires only the facilities described as mandatory in this specification. Such a document must meet all of the following criteria:

1. It must validate against one of the three DTDs found in Appendix A.

2. The root element of the document must be `<html>`.

3. The root element of the document must designate the XHTML 1.0 namespace using the `xmlns` attribute [XMLNAMES]. The namespace for XHTML 1.0 is defined to be:

    - http://www.w3.org/TR/xhtml1

4. There must be a DOCTYPE declaration in the document prior to the root element. If present, the public identifier included in the DOCTYPE declaration must reference one of the three DTDs found in Appendix A using the respective Formal Public Identifier. The system identifier may be modified appropriately.

```
<!DOCTYPE
 html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">

<!DOCTYPE
 html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">

<!DOCTYPE
 html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
 "http://www.w3.org/TR/xhtml1/DTD/frameset.dtd">
```

XHTML Documents may be labeled with the Internet Media Type **`text/html`** or **`text/xml`**. When labeled as **`text/html`**, documents should follow the guidelines set forth in Appendix C. Failure to follow these guidelines will almost certainly ensure that the document will fail to be processed on older implementations.

Here is an example of a minimal XHTML document.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1">
  <head>
    <title>Virtual Library</title>
  </head>
  <body>
    <p>Moved to <a href="http://vlib.org/">vlib.org</a>.</p>
  </body>
</html>
```

3.1.2 Using XHTML with other namespaces

The XHTML 1.0 namespace may be used with other XML namespaces as per [XMLNAMES], although such documents are not strictly conforming XHTML 1.0 documents as defined above. Future work by W3C will address ways to specify conformance for documents involving multiple namespaces.

The following example shows the way in which XHTML 1.0 could be used in conjunction with the MathML Recommendation:

```
<html xmlns="http://www.w3.org/TR/xhtml1">
  <head>
    <title>A Math Example</title>
  </head>
  <body>
    <p>The following is MathML markup:</p>
    <math xmlns="http://www.w3.org/TR/REC-MathML">
      <apply> <log/>
        <logbase>
          <cn> 3 </cn>
        </logbase>
        <ci> x </ci>
      </apply>
    </math>
  </body>
</html>
```

The following example shows the way in which XHTML 1.0 markup could be incorporated into another XML namespace:

```
<?xml version="1.0"?>
<!-- initially, the default namespace is "books" -->
<book xmlns='urn:loc.gov:books'
    xmlns:isbn='urn:ISBN:0-395-36341-6'>
  <title>Cheaper by the Dozen</title>
  <isbn:number>1568491379</isbn:number>
  <notes>
    <!-- make HTML the default namespace for a hypertext commentary -->
    <p xmlns='http://www.w3.org/TR/xhtml1'>
        This is also available <a href="http://www.w3.org/">online</a>.
    </p>
  </notes>
</book>
```

3.2 User Agent Conformance

A conforming user agent must meet all of the following criteria:

1. In order to be consistent with the XML 1.0 Recommendation [XML], the user agent must parse and evaluate an XHTML document for well-formedness. If the user agent claims to be a validating user agent, it must also validate documents against their referenced DTDs according to [XML].
2. When the user agent claims to support facilities defined within this specification or required by this specification through normative reference, it must do so in ways consistent with the facilities' definition.
3. If a user agent encounters an element it does not recognize, it must render the element's content.
4. If a user agent encounters an attribute it does not recognize, it must ignore the entire attribute specification (i.e., the attribute and its value).
5. If a user agent encounters an attribute value it doesn't recognize, it must use the default attribute value.
6. If it encounters an undeclared entity, the entity must be treated as character data.

4. Differences with HTML 4.0

Due to the fact that XHTML is an XML application, certain practices that were perfectly legal in SGML-based HTML 4.0 [HTML] must be changed.

4.1 New Requirements

4.1.1 Documents must be well-formed.

Well-formedness is a new concept introduced by [XML]. Essentially this means that all elements must either have closing tags or be written in a special form (as described below), and that all the elements must nest.

Although overlapping is illegal in SGML, it was widely tolerated in SGML-based browsers.

CORRECT: nested elements.

<p>here is an emphasized <em>paragraph</em>.</p>

INCORRECT: overlapping elements

```
<p>here is an emphasized <em>paragraph.</p></em>
```

4.1.2 Element and attribute names must be in lower case.

XHTML documents must use lower case for all HTML element and attribute names. This difference is necessary because XML is case-sensitive e.g. <li> and <LI> are different tags.

4.1.3 For non-empty elements, end tags are required.

In SGML-based HTML 4.0 certain elements were permitted to omit the end tag; with the elements that followed implying closure. This omission is not permitted in XML-based XHTML. All elements other than those declared in the DTD as **`EMPTY`** must have an end tag.

CORRECT: terminated elements

```
<p>here is a paragraph.</p><p>here is another paragraph.</p>
```

INCORRECT: unterminated elements

```
<p>here is a paragraph.<p>here is another paragraph.
```

4.1.4 Attribute values must always be quoted.

All attribute values must be quoted, even those which appear to be numeric.

CORRECT: quoted attribute values

```
<table rows="3">
```

INCORRECT: unquoted attribute values

```
<table rows=3>
```

4.1.5 Attribute Minimization

XML does not support attribute minimization. Attribute-value pairs must be written in full. Attribute names such as **`compact`** and **`checked`** cannot occur in elements without their value being specified.

CORRECT: unminimized attributes

```
<dl compact="compact">
```

INCORRECT: minimized attributes

```
<dl compact>
```

4.1.6 Empty Elements

Empty elements must end with `/>`. For instance, `<br/>` or `<hr/>`.

CORRECT: terminated empty tags

```
<br/><hr/>
```

INCORRECT: unterminated empty tags

```
<br><hr>
```

4.1.7 White space handling in attribute values

In attribute values, user agents will strip leading and trailing white-space from attribute values and and map sequences of one or more white space characters (including line breaks) to a single inter-word space (an ASCII space character for western scripts). See Section 3.3.3 of [XML].

4.1.8 Script and Style elements

In XHTML, the script and style elements are declared as having `#PCDATA` content. As a result, `<` and `&` will be treated as the start of markup, and entities such as `&lt;` and `&amp;` will be recognized as entity references by the XML processor to `<` and `&` respectively. Wrapping the content of the script or style element within a `CDATA` marked section avoids the expansion of these entities.

```
<script>
 <![CDATA[
 ... unescaped script content ...
 ]]>
 </script>
```

CDATA sections are recognized by the XML processor and appear as nodes in the Document Object Model, see Section 1.3 of the DOM Level 1 Recommendation [DOM].

An alternative is to use external script and style documents.

4.1.9 SGML exclusions

SGML gives the writer of a DTD the ability to exclude specific elements from being contained within an element. Such prohibitions (called "exclusions") are not possible in XML.

For example, the HTML 4.0 Strict DTD forbids the nesting of an 'a' element within another 'a' element to any descendant depth. It is not possible to spell out such prohibitions in XML. Even though these prohibitions cannot be defined in the DTD, certain elements should not be nested. A summary of such elements and the elements that should not be nested in them is found in the normative Appendix B.

4.1.10 HTML 4.0 errata

The current HTML 4.0 DTDs do not reflect errata changes made to the HTML 4.0 Recommendation [HTML]. The XHTML DTDs incorporate these errata, and thus errors in HTML 4.0 DTDs are corrected in the XHTML DTDs. The errata can be found at [ERRATA].

4.2 Converting existing content to XHTML

HTML Tidy is W3C sample code that automatically converts existing web content to XHTML. It can cope with a wide range of markup errors, and offers a means to smoothly transition existing HTML documents to XHTML. For more information, see [TIDY].

5. Compatibility Issues

Although there is no requirement for XHTML 1.0 documents to be compatible with existing user agents, in practice this is easy to accomplish. Guidelines for creating compatible documents can be found in Appendix C.

5.1 Internet Media Type

Work is currently in progress to determine how Internet media types [RFC2046] should be used when delivering XML documents, and this will be the subject of a future W3C document.

Since XHTML is an XML application, XHTML documents may be delivered using the Internet media type `text/xml`. Additionally, since one of the aims of XHTML is to allow migration from existing HTML user agents to XHTML user agents, XHTML documents may be delivered using the Internet media type `text/html`. In this case, it is recommended that the documents follow the guidelines in Appendix C to decrease the chance of document processing failure.

6. Future Directions

XHTML 1.0 provides the basis for a family of document types that will extend and subset XHTML, in order to support a wide range of new devices and applications, by defining modules and specifying a mechanism for combining these modules. This mechanism will enable the extension and subsetting of XHTML 1.0 in a uniform way through the definition of new modules.

6.1 Modularizing HTML

As the use of XHTML moves from the traditional desktop user agents to other platforms, it is clear that not all of the XHTML elements will be required on all platforms. For example a hand held device or a cell-phone may only support a subset of XHTML elements.

The process of modularization breaks XHTML up into a series of smaller element sets. These elements can then be recombined to meet the needs of different communities.

These modules will be defined in a later W3C document.

6.2 Subsets and Extensibility

Modularization brings with it several advantages:

- It provides a formal mechanism for subsetting XHTML.
- It provides a formal mechanism for extending XHTML.
- It simplifies the transformation between document types.
- It promotes the reuse of modules in new document types.

6.3 Document Profiles

A document profile specifies the syntax and semantics of a set of documents. Conformance to a document profile provides a basis for interoperability guarantees. The document profile specifies the facilities required to process documents of that type, e.g. which image formats can be used, levels of scripting, style sheet support, and so on.

For product designers this enables various groups to define their own standard profile.

For authors this will obviate the need to write several different versions of documents for different clients.

For special groups such as chemists, medical doctors, or mathematicians this allows a special profile to be built using standard HTML elements plus a group of elements geared to the specialist's needs.

Appendices

Appendix A. DTDs

This appendix is normative.

These DTDs and entity sets form a normative part of this specification. The complete set of DTD files together with an XML declaration and SGML Open Catalog is included in the zip file for this specification.

A.1 Document Type Definitions

These DTDs approximate the HTML 4.0 DTDs. It is likely that when the DTDs are modularized, a method of DTD construction will be employed that corresponds more closely to HTML 4.0.

- XHTML-1.0-Strict
- XHTML-1.0-Transitional
- XHTML-1.0-Frameset

A.2 Entity Sets

The XHTML entity sets are the same as for HTML 4.0, but have been modified to be valid XML 1.0 entity declarations. Note the entity for the Euro currency sign (`&euro;` or `&#8364;` or `&#x20AC;`) is defined as part of the special characters.

- Latin-1 characters
- Special characters
- Symbols

Appendix B. Element Prohibitions

This appendix is normative.

The following elements have prohibitions on which elements they can contain (see Section 4.1.9). This prohibition applies to all depths of nesting, i.e. it contains all the descendant elements.

`a`
: cannot contain other `a` elements.

`pre`
: cannot contain the `img`, `object`, `big`, `small`, `sub`, or `sup` elements.

`button`
: cannot contain the `input`, `select`, `textarea`, `label`, `button`, `form`, `fieldset`, `iframe` or `isindex` elements.

`label`
: cannot contain other `label` elements.

`form`
: cannot contain other `form` elements.

Appendix C. Guidelines

This appendix is informative.

This appendix summarizes design guidelines for authors who wish their XHTML documents to render on existing HTML user agents.

- Be aware that processing instructions are rendered on some user agents.
- Include a space before the trailing `/` and `>` of empty elements, e.g. `<br />`, `<hr />` and `<img src="karen.jpg" alt="Karen" />`. Also, use the minimized tag syntax for empty elements, e.g. `<br />`, as the alternative syntax `<br></br>` allowed by XML gives uncertain results in many existing user agents.
- Given an empty instance of an element whose content model is not `EMPTY` (for example, an empty title or paragraph) do not use the minimized form (e.g. use `<p> </p>` and not `<p />`).
- Use external style sheets if your style sheet uses `<` or `&` or `]]>`. Use external scripts if your script uses `<` or `&` or `]]>`.
- Avoid line breaks and multiple white space characters within attribute values. These are handled inconsistently by user agents.
- Don't include more than one `isindex` element in the document `head`. The `isindex` element is deprecated in favor of the `input` element.
- Use both the `lang` and `xml:lang` attributes when specifying the language of an element. The value of the `xml:lang` attribute takes precedence.
- In XML, URIs that end with fragment identifiers of the form `"#foo"` do not refer to elements with an attribute `name="foo"`; rather, they refer to elements with an attribute defined to be of type `ID`, e.g., the `id` attribute in HTML 4.0. Many existing HTML clients don't support the use of `ID`-type attributes in this way, so if you want to be able to process the document on HTML clients, you may wish to supply both `id` and `name` values on the target element, e.g., `<a id="foo" name="foo">...</a>`

- To specify a character encoding in the document, use both the encoding attribute specification on the xml declaration (e.g. `<?xml version="1.0" encoding="EUC-JP"?>`) and a meta http-equiv statement (e.g. `<meta http-equiv="Content-type" content='text/html; charset="EUC-JP"' />`). The value of the encoding attribute of the xml processing instruction takes precedence.
- Some HTML user agents are unable to interpret boolean attributes when these appear in their full (non-minimized) form, as required by XML 1.0. Note this problem doesn't effect user agents compliant with HTML 4.0. The following attributes are involved: `compact`, `nowrap`, `ismap`, `declare`, `noshade`, `checked`, `disabled`, `readonly`, `multiple`, `selected`, `noresize`, `defer`.

Appendix D. Acknowledgements

This appendix is informative.

This specification was written with the participation of the members of the W3C HTML working group:

Steven Pemberton, CWI (HTML Working Group Chair)
Murray Altheim, Sun Microsystems
Daniel Austin, CNET: The Computer Network
Frank Boumphrey, HTML Writers Guild
John Burger, Mitre
Andrew W. Donoho, IBM
Sam Dooley, IBM
Klaus Hofrichter, GMD
Philipp Hoschka, W3C
Masayasu Ishikawa, W3C
Warner ten Kate, Philips Electronics
Peter King, Phone.com
Paula Klante, JetForm
Shin'ichi Matsui, W3C/Panasonic
Shane McCarron, The Open Group
Ann Navarro, HTML Writers Guild
Zach Nies, Quark
Dave Raggett, W3C/HP (W3C lead for HTML)
Patrick Schmitz, Microsoft
Chris Wilson, Microsoft
Ted Wugofski, Gateway 2000
Dan Zigmond, WebTV Networks

Appendix E. References

This appendix is informative.

[CC/PP]
"Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation", F. Reynolds, J. Hjelm, S. Dawkins, S. Singhal, 30 November 1998.

This document describes a method for using the Resource Description Format (RDF) to create a general, yet extensible framework for describing user preferences and device capabilities. Servers can exploit this to customize the service or content provided.
Available at: http://www.w3.org/TR/NOTE-CCPP

[CSS2]
"Cascading Style Sheets, level 2 (CSS2) Specification", B. Bos, H. W. Lie, C. Lilley, I. Jacobs, 12 May 1998.
Available at: http://www.w3.org/TR/REC-CSS2

[DOM]
"Document Object Model (DOM) Level 1 Specification", Lauren Wood *et al.*, 1 October 1998.
Available at: http://www.w3.org/TR/REC-DOM-Level-1

[ERRATA]
"HTML 4.0 Specification Errata".
This document lists the errata for the HTML 4.0 specification.
Available at: http://www.w3.org/MarkUp/html40-updates/REC-html40-19980424-errata.html

[HTML]
"HTML 4.0 Specification", D. Raggett, A. Le Hors, I. Jacobs, 18 December 1997, revised 24 April 1998.
Available at: http://www.w3.org/TR/REC-html40

[POSIX.1]
"ISO/IEC 9945-1:1990 Information Technology - Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) [C Language]", Institute of Electrical and Electronics Engineers, Inc, 1990.

[RFC2045]
"RFC2045: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", N. Freed and N. Borenstein, November 1996.
Available at http://www.ietf.org/rfc/rfc2045.txt. Note that this RFC obsoletes RFC1521, RFC1522, and RFC1590.

[RFC2046]
"RFC2046: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", N. Freed and N. Borenstein, November 1996.
Available at http://www.ietf.org/rfc/rfc2046.txt. Note that this RFC obsoletes RFC1521, RFC1522, and RFC1590.

[RFC2068]
"RFC2068: HTTP Version 1.1 ", R. Fielding, J. Gettys, J. Mogul, H. Frystyk Nielsen, and T. Berners-Lee, January 1997.
Available at http://www.ietf.org/rfc/rfc2068.txt.

[RFC2119]
"RFC2119: Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997.
Available at: http://www.ietf.org/rfc/rfc2119.txt

[RFC2376]
"RFC2376: XML Media Types", E. Whitehead, M. Murata, July 1998.
Available at: http://www.ietf.org/rfc/rfc2376.txt

[RFC2396]
"RFC2396: Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter, August 1998.
This document updates RFC1738 and RFC1808.

Available at: http://www.ietf.org/rfc/rfc2396.txt

[TIDY]

"HTML Tidy" is a tool for detecting and correcting a wide range of markup errors prevalent in HTML. It can also be used as a tool for converting existing HTML content to be well formed XML. Tidy is being made available on the same terms as other W3C sample code, i.e. free for any purpose, and entirely at your own risk.
It is available from: http://www.w3.org/Status.html#TIDY

[XML]

"Extensible Markup Language (XML) 1.0 Specification", T. Bray, J. Paoli, C. M. Sperberg-McQueen, 10 February 1998.
Available at: http://www.w3.org/TR/REC-xml

[XMLNAMES]

"Namespaces in XML", T. Bray, D. Hollander, A. Layman, 14 January 1999.
XML namespaces provide a simple method for qualifying names used in XML documents by associating them with namespaces identified by URI.
Available at: http://www.w3.org/TR/REC-xml-names

[XMLSTYLE]

"Associating stylesheets with XML documents Version 1.0", J. Clark, 14 January 1999.
This document describes a means for a stylesheet to be associated with an XML document by including one or more processing instructions with a target of xml-stylesheet in the document's prolog.
Available at: http://www.w3.org/TR/PR-xml-stylesheet

THIS VERSION IS OUTDATED!For the latest version, please look at https://www.w3.org/TR/xhtml1/.