

indistinctly from non-nested elements:

plain ***bold <I>italic</I>*** may be rendered
the same as plain ***bold <I>italic</I>***

5.7.1. Idiomatic Elements

Phrases may be marked up to indicate certain idioms.

NOTE - User agents may support the `<DFN>` element, not included in this specification, as it has been deployed to some extent. It is used to indicate the defining instance of a term, and it is typically rendered in italic or bold italic.

5.7.1.1. Citation: CITE

The `<CITE>` element is used to indicate the title of a book or other citation. It is typically rendered as italics. For example:

He just couldn't get enough of `<cite>The Grapes of Wrath</cite>`.

5.7.1.2. Code: CODE

The `<CODE>` element indicates an example of code, typically rendered in a mono-spaced font. The `<CODE>` element is intended for short words or phrases of code; the `<PRE>` block structuring element (5.5.2, "Preformatted Text: PRE") is more appropriate for multiple-line listings. For example:

The expression `<code>x += 1</code>`
is short for `<code>x = x + 1</code>`.

5.7.1.3. Emphasis: EM

The `` element indicates an emphasized phrase, typically rendered as italics. For example:

A singular subject `always` takes a singular verb.

5.7.1.4. Keyboard: KBD

The `<KBD>` element indicates text typed by a user, typically rendered in a mono-spaced font. This is commonly used in instruction manuals. For example:

Enter `<kbd>FIND IT</kbd>` to search the database.

5.7.1.5. Sample: SAMP

The <SAMP> element indicates a sequence of literal characters, typically rendered in a mono-spaced font. For example:

The only word containing the letters <samp>mt</samp> is dreamt.

5.7.1.6. Strong Emphasis: STRONG

The element indicates strong emphasis, typically rendered in bold. For example:

STOP, or I'll say "STOP" again!

5.7.1.7. Variable: VAR

The <VAR> element indicates a placeholder variable, typically rendered as italic. For example:

Type <SAMP>html-check <VAR>file</VAR> | more</SAMP>
to check <VAR>file</VAR> for markup errors.

5.7.2. Typographic Elements

Typographic elements are used to specify the format of marked text.

Typical renderings for idiomatic elements may vary between user agents. If a specific rendering is necessary -- for example, when referring to a specific text attribute as in "The italic parts are mandatory" -- a typographic element can be used to ensure that the intended typography is used where possible.

NOTE - User agents may support some typographic elements not included in this specification, as they have been deployed to some extent. The <STRIKE> element indicates horizontal line through the characters, and the <U> element indicates an underline.

5.7.2.1. Bold: B

The element indicates bold text. Where bold typography is unavailable, an alternative representation may be used.

5.7.2.2. Italic: I

The <I> element indicates italic text. Where italic typography is unavailable, an alternative representation may be used.

5.7.2.3. Teletype: TT

The <TT> element indicates teletype (monospaced)text. Where a teletype font is unavailable, an alternative representation may be used.

5.7.3. Anchor: A

The <A> element indicates a hyperlink anchor (see 7, "Hyperlinks"). At least one of the NAME and HREF attributes should be present. Attributes of the <A> element:

HREF

gives the URI of the head anchor of a hyperlink.

NAME

gives the name of the anchor, and makes it available as a head of a hyperlink.

TITLE

suggests a title for the destination resource -- advisory only. The TITLE attribute may be used:

- * for display prior to accessing the destination resource, for example, as a margin note or on a small box while the mouse is over the anchor, or while the document is being loaded;

- * for resources that do not include a title, such as graphics, plain text and Gopher menus, for use as a window title.

REL

The REL attribute gives the relationship(s) described by the hyperlink. The value is a whitespace separated list of relationship names. The semantics of link relationships are not specified in this document.

REV

same as the REL attribute, but the semantics of the relationship are in the reverse direction. A link from A to B with REL="X" expresses the same relationship as a link from B to A with REV="X". An anchor may have both REL and REV attributes.

URN

specifies a preferred, more persistent identifier for the head anchor of the hyperlink. The syntax and

semantics of the URN attribute are not yet specified.

METHODS

specifies methods to be used in accessing the destination, as a whitespace-separated list of names. The set of applicable names is a function of the scheme of the URI in the HREF attribute. For similar reasons as for the TITLE attribute, it may be useful to include the information in advance in the link. For example, the HTML user agent may chose a different rendering as a function of the methods allowed; for example, something that is searchable may get a different icon.

5.8. Line Break: BR

The
 element specifies a line break between words (see 6, "Characters, Words, and Paragraphs"). For example:

```
<P> Pease porridge hot<BR>
Pease porridge cold<BR>
Pease porridge in the pot<BR>
Nine days old.
```

5.9. Horizontal Rule: HR

The <HR> element is a divider between sections of text; typically a full width horizontal rule or equivalent graphic. For example:

```
<HR>
<ADDRESS>February 8, 1995, CERN</ADDRESS>
</BODY>
```

5.10. Image: IMG

The element refers to an image or icon via a hyperlink (see 7.3, "Simultaneous Presentation of Image Resources").

HTML user agents may process the value of the ALT attribute as an alternative to processing the image resource indicated by the SRC attribute.

NOTE - Some HTML user agents can process graphics linked via anchors, but not graphics. If a graphic is essential, it should be referenced from an <A> element rather than an element. If the graphic is not essential, then the element is appropriate.

Attributes of the element:

ALIGN

alignment of the image with respect to the text baseline.

* `TOP' specifies that the top of the image aligns with the tallest item on the line containing the image.

* `MIDDLE' specifies that the center of the image aligns with the baseline of the line containing the image.

* `BOTTOM' specifies that the bottom of the image aligns with the baseline of the line containing the image.

ALT

text to use in place of the referenced image resource, for example due to processing constraints or user preference.

ISMAP

indicates an image map (see 7.6, "Image Maps").

SRC

specifies the URI of the image resource.

NOTE - In practice, the media types of image resources are limited to a few raster graphic formats: typically `image/gif', `image/jpeg'. In particular, `text/html' resources are not intended to be used as image resources.

Examples of use:

```
<IMG SRC="triangle.xbm" ALT="Warning:"> Be sure  
to read these instructions.
```

```
<a href="http://machine/htbin/imap/sample">  
<IMG SRC="sample.xbm" ISMAP>  
</a>
```

6. Characters, Words, and Paragraphs

An HTML user agent should present the body of an HTML document as a collection of typeset paragraphs and preformatted text. Except for preformatted elements (<PRE>, <XMP>, <LISTING>, <TEXTAREA>), each block structuring element is regarded as a paragraph by taking the

data characters in its content and the content of its descendant elements, concatenating them, and splitting the result into words, separated by space, tab, or record end characters (and perhaps hyphen characters). The sequence of words is typeset as a paragraph by breaking it into lines.

6.1. The HTML Document Character Set

The document character set specified in 9.5, "SGML Declaration for HTML" must be supported by HTML user agents. It includes the graphic characters of Latin Alphabet No. 1, or simply Latin-1. Latin-1 comprises 191 graphic characters, including the alphabets of most Western European languages.

NOTE - Use of the non-breaking space and soft hyphen indicator characters is discouraged because support for them is not widely deployed.

NOTE - To support non-western writing systems, a larger character repertoire will be specified in a future version of HTML. The document character set will be [ISO-10646], or some subset that agrees with [ISO-10646]; in particular, all numeric character references must use code positions assigned by [ISO-10646].

In SGML applications, the use of control characters is limited in order to maximize the chance of successful interchange over heterogeneous networks and operating systems. In the HTML document character set only three control characters are allowed: Horizontal Tab, Carriage Return, and Line Feed (code positions 9, 13, and 10).

The HTML DTD references the Added Latin 1 entity set, to allow mnemonic representation of selected Latin 1 characters using only the widely supported ASCII character repertoire. For example:

Kurt Göl;del was a famous logician and mathematician.

See 9.7.2, "ISO Latin 1 Character Entity Set" for a table of the "Added Latin 1" entities, and 13, "The HTML Coded Character Set" for a table of the code positions of [ISO 8859-1] and the control characters in the HTML document character set.

7. Hyperlinks

In addition to general purpose elements such as paragraphs and lists, HTML documents can express hyperlinks. An HTML user agent allows the user to navigate these hyperlinks.

A hyperlink is a relationship between two anchors, called the head and the tail of the hyperlink[DEXTER]. Anchors are identified by an anchor address: an absolute Uniform Resource Identifier (URI), optionally followed by a '#' and a sequence of characters called a fragment identifier. For example:

```
http://www.w3.org/hypertext/WWW/TheProject.html
http://www.w3.org/hypertext/WWW/TheProject.html#z31
```

In an anchor address, the URI refers to a resource; it may be used in a variety of information retrieval protocols to obtain an entity that represents the resource, such as an HTML document. The fragment identifier, if present, refers to some view on, or portion of the resource.

Each of the following markup constructs indicates the tail anchor of a hyperlink or set of hyperlinks:

- * <A> elements with HREF present.
- * <LINK> elements.
- * elements.
- * <INPUT> elements with the SRC attribute present.
- * <ISINDEX> elements.
- * <FORM> elements with `METHOD=GET'.

These markup constructs refer to head anchors by a URI, either absolute or relative, or a fragment identifier, or both.

In the case of a relative URI, the absolute URI in the address of the head anchor is the result of combining the relative URI with a base absolute URI as in [RELURL]. The base document is taken from the document's <BASE> element, if present; else, it is determined as in [RELURL].

7.1. Accessing Resources

Once the address of the head anchor is determined, the user agent may obtain a representation of the resource.

For example, if the base URI is `http://host/x/y.html' and the document contains:

```

```

then the user agent uses the URI ``http://host/icons/abc.gif'` to access the resource, as in [URL]..

7.2. Activation of Hyperlinks

An HTML user agent allows the user to navigate the content of the document and request activation of hyperlinks denoted by `<A>` elements. HTML user agents should also allow activation of `<LINK>` element hyperlinks.

To activate a link, the user agent obtains a representation of the resource identified in the address of the head anchor. If the representation is another HTML document, navigation may begin again with this new document.

7.3. Simultaneous Presentation of Image Resources

An HTML user agent may activate hyperlinks indicated by `` and `<INPUT>` elements concurrently with processing the document; that is, image hyperlinks may be processed without explicit request by the user. Image resources should be embedded in the presentation at the point of the tail anchor, that is the `` or `<INPUT>` element.

`<LINK>` hyperlinks may also be processed without explicit user request; for example, style sheet resources may be processed before or during the processing of the document.

7.4. Fragment Identifiers

Any characters following a ``#'` character in a hypertext address constitute a fragment identifier. In particular, an address of the form ``#fragment'` refers to an anchor in the same document.

The meaning of fragment identifiers depends on the media type of the representation of the anchor's resource. For ``text/html'` representations, it refers to the `<A>` element with a `NAME` attribute whose value is the same as the fragment identifier. The matching is case sensitive. The document should have exactly one such element. The user agent should indicate the anchor element, for example by scrolling to and/or highlighting the phrase.

For example, if the base URI is ``http://host/x/y.html'` and the user activated the link denoted by the following markup:

```
<p> See: <a href="app1.html#bananas">appendix 1</a>  
for more detail on bananas.
```

Then the user agent accesses the resource identified by ``http://host/x/app1.html'`. Assuming the resource is represented using the ``text/html'` media type, the user agent must locate the `<A>` element whose NAME attribute is ``bananas'` and begin navigation there.

7.5. Queries and Indexes

The `<ISINDEX>` element represents a set of hyperlinks. The user can choose from the set by providing keywords to the user agent. The user agent computes the head URI by appending ``?'` and the keywords to the base URI. The keywords are escaped according to [URL] and joined by ``+'`. For example, if a document contains:

```
<BASE HREF="http://host/index">
<ISINDEX>
```

and the user provides the keywords ``apple'` and ``berry'`, then the user agent must access the resource ``http://host/index?apple+berry'`.

`<FORM>` elements with ``METHOD=GET'` also represent sets of hyperlinks. See 8.2.2, "Query Forms: METHOD=GET" for details.

7.6. Image Maps

If the ISMAP attribute is present on an `` element, the `` element must be contained in an `<A>` element with an HREF present. This construct represents a set of hyperlinks. The user can choose from the set by choosing a pixel of the image. The user agent computes the head URI by appending ``?'` and the x and y coordinates of the pixel to the URI given in the `<A>` element. For example, if a document contains:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<head><title>ImageMap Example</title>
<BASE HREF="http://host/index"></head>
<body>
<p> Choose any of these icons:<br>
<a href="/cgi-bin/imapemap"></a>
```

and the user chooses the upper-leftmost pixel, the chosen hyperlink is the one with the URI ``http://host/cgi-bin/imapemap?0,0'`.

8. Forms

A form is a template for a form data set and an associated method and action URI. A form data set is a sequence of name/value pair fields. The names are specified on the NAME attributes of form input elements, and the values are given initial values by various forms of markup and edited by the user. The resulting form data set is used to access an information service as a function of the action and method.

Forms elements can be mixed in with document structuring elements. For example, a <PRE> element may contain a <FORM> element, or a <FORM> element may contain lists which contain <INPUT> elements. This gives considerable flexibility in designing the layout of forms.

Form processing is a level 2 feature.

8.1. Form Elements

8.1.1. Form: FORM

The <FORM> element contains a sequence of input elements, along with document structuring elements. The attributes are:

ACTION

specifies the action URI for the form. The action URI of a form defaults to the base URI of the document (see 7, "Hyperlinks").

METHOD

selects a method of accessing the action URI. The set of applicable methods is a function of the scheme of the action URI of the form. See 8.2.2, "Query Forms: METHOD=GET" and 8.2.3, "Forms with Side-Effects: METHOD=POST".

ENCTYPE

specifies the media type used to encode the name/value pairs for transport, in case the protocol does not itself impose a format. See 8.2.1, "The form-urlencoded Media Type".

8.1.2. Input Field: INPUT

The <INPUT> element represents a field for user input. The TYPE attribute discriminates between several variations of fields.

The <INPUT> element has a number of attributes. The set of applicable attributes depends on the value of the TYPE attribute.

8.1.2.1. Text Field: INPUT TYPE=TEXT

The default value of the TYPE attribute is 'TEXT', indicating a single line text entry field. (Use the <TEXTAREA> element for multi-line text fields.)

Required attributes are:

NAME

name for the form field corresponding to this element.

The optional attributes are:

MAXLENGTH

constrains the number of characters that can be entered into a text input field. If the value of MAXLENGTH is greater than the value of the SIZE attribute, the field should scroll appropriately. The default number of characters is unlimited.

SIZE

specifies the amount of display space allocated to this input field according to its type. The default depends on the user agent.

VALUE

The initial value of the field.

For example:

```
<p>Street Address: <input name=street><br>
Postal City code: <input name=city size=16 maxlength=16><br>
Zip Code: <input name=zip size=10 maxlength=10 value="99999-9999"><br>
```

8.1.2.2. Password Field: INPUT TYPE=PASSWORD

An <INPUT> element with 'TYPE=PASSWORD' is a text field as above, except that the value is obscured as it is entered. (see also: 10, "Security Considerations").

For example:

```
<p>Name: <input name=login> Password: <input type=password name=passwd>
```

8.1.2.3. Check Box: INPUT TYPE=CHECKBOX

An <INPUT> element with `TYPE=CHECKBOX' represents a boolean choice. A set of such elements with the same name represents an n-of-many choice field. Required attributes are:

NAME

symbolic name for the form field corresponding to this element or group of elements.

VALUE

The portion of the value of the field contributed by this element.

Optional attributes are:

CHECKED

indicates that the initial state is on.

For example:

```
<p>What flavors do you like?
<input type=checkbox name=flavor value=vanilla>Vanilla<br>
<input type=checkbox name=flavor value=strawberry>Strawberry<br>
<input type=checkbox name=flavor value=chocolate checked>Chocolate<br>
```

8.1.2.4. Radio Button: INPUT TYPE=RADIO

An <INPUT> element with `TYPE=RADIO' represents a boolean choice. A set of such elements with the same name represents a 1-of-many choice field. The NAME and VALUE attributes are required as for check boxes. Optional attributes are:

CHECKED

indicates that the initial state is on.

At all times, exactly one of the radio buttons in a set is checked. If none of the <INPUT> elements of a set of radio buttons specifies `CHECKED', then the user agent must check the first radio button of the set initially.

For example:

```
<p>Which is your favorite?
<input type=radio name=flavor value=vanilla>Vanilla<br>
<input type=radio name=flavor value=strawberry>Strawberry<br>
<input type=radio name=flavor value=chocolate>Chocolate<br>
```


8.1.2.5. Image Pixel: INPUT TYPE=IMAGE

An `<INPUT>` element with ``TYPE=IMAGE'` specifies an image resource to display, and allows input of two form fields: the x and y coordinate of a pixel chosen from the image. The names of the fields are the name of the field with ``x'` and ``y'` appended. ``TYPE=IMAGE'` implies ``TYPE=SUBMIT'` processing; that is, when a pixel is chosen, the form as a whole is submitted.

The NAME attribute is required as for other input fields. The SRC attribute is required and the ALIGN is optional as for the `` element (see 5.10, "Image: IMG").

For example:

```
<p>Choose a point on the map:  
<input type=image name=point src="map.gif">
```

8.1.2.6. Hidden Field: INPUT TYPE=HIDDEN

An `<INPUT>` element with ``TYPE=HIDDEN'` represents a hidden field. The user does not interact with this field; instead, the VALUE attribute specifies the value of the field. The NAME and VALUE attributes are required.

For example:

```
<input type=hidden name=context value="l2k3j4l2k3j4l2k3j4lk23">
```

8.1.2.7. Submit Button: INPUT TYPE=SUBMIT

An `<INPUT>` element with ``TYPE=SUBMIT'` represents an input option, typically a button, that instructs the user agent to submit the form. Optional attributes are:

NAME

indicates that this element contributes a form field whose value is given by the VALUE attribute. If the NAME attribute is not present, this element does not contribute a form field.

VALUE

indicates a label for the input (button).

You may submit this request internally:

```
<input type=submit name=recipient value=internal><br>
```

or to the external world:

```
<input type=submit name=recipient value=world>
```

8.1.2.8. Reset Button: INPUT TYPE=RESET

An <INPUT> element with `TYPE=RESET' represents an input option, typically a button, that instructs the user agent to reset the form's fields to their initial states. The VALUE attribute, if present, indicates a label for the input (button).

When you are finished, you may submit this request:

```
<input type=submit><br>
```

You may clear the form and start over at any time: <input type=reset>

8.1.3. Selection: SELECT

The <SELECT> element constrains the form field to an enumerated list of values. The values are given in <OPTION> elements. Attributes are:

MULTIPLE

indicates that more than one option may be included in the value.

NAME

specifies the name of the form field.

SIZE

specifies the number of visible items. Select fields of size one are typically pop-down menus, whereas select fields with size greater than one are typically lists.

For example:

```
<SELECT NAME="flavor">
<OPTION>Vanilla
<OPTION>Strawberry
<OPTION value="RumRasin">Rum and Raisin
<OPTION selected>Peach and Orange
</SELECT>
```

The initial state has the first option selected, unless a SELECTED attribute is present on any of the <OPTION> elements.

8.1.3.1. Option: OPTION

The Option element can only occur within a Select element. It represents one choice, and has the following attributes:

SELECTED

Indicates that this option is initially selected.

VALUE

indicates the value to be returned if this option is chosen. The field value defaults to the content of the <OPTION> element.

The content of the <OPTION> element is presented to the user to represent the option. It is used as a returned value if the VALUE attribute is not present.

8.1.4. Text Area: TEXTAREA

The <TEXTAREA> element represents a multi-line text field. Attributes are:

COLS

the number of visible columns to display for the text area, in characters.

NAME

Specifies the name of the form field.

ROWS

The number of visible rows to display for the text area, in characters.

For example:

```
<TEXTAREA NAME="address" ROWS=6 COLS=64>
HaL Computer Systems
1315 Dell Avenue
Campbell, California 95008
</TEXTAREA>
```

The content of the <TEXTAREA> element is the field's initial value.

Typically, the ROWS and COLS attributes determine the visible dimension of the field in characters. The field is typically rendered in a fixed-width font. HTML user agents should allow text to extend beyond these limits by scrolling as needed.

8.2. Form Submission

An HTML user agent begins processing a form by presenting the document with the fields in their initial state. The user is allowed to modify the fields, constrained by the field type etc. When the user indicates that the form should be submitted (using a submit button or image input), the form data set is processed according to its method, action URI and enctype.

When there is only one single-line text input field in a form, the user agent should accept Enter in that field as a request to submit the form.

8.2.1. The form-urlencoded Media Type

The default encoding for all forms is `'application/x-www-form-urlencoded'`. A form data set is represented in this media type as follows:

1. The form field names and values are escaped: space characters are replaced by `'+'`, and then reserved characters are escaped as per [URL]; that is, non-alphanumeric characters are replaced by `'%HH'`, a percent sign and two hexadecimal digits representing the ASCII code of the character. Line breaks, as in multi-line text field values, are represented as CR LF pairs, i.e. `'%0D%0A'`.
2. The fields are listed in the order they appear in the document with the name separated from the value by `'='` and the pairs separated from each other by `'&'`. Fields with null values may be omitted. In particular, unselected radio buttons and checkboxes should not appear in the encoded data, but hidden fields with VALUE attributes present should.

NOTE - The URI from a query form submission can be used in a normal anchor style hyperlink. Unfortunately, the use of the `'&'` character to separate form fields interacts with its use in SGML attribute values as an entity reference delimiter. For example, the URI `'http://host/?x=1&y=2'` must be written `'<a href="http://host/?x=1&y=2"'` or `''`.

HTTP server implementors, and in particular, CGI implementors are encouraged to support the use of `';'` in place of `'&'` to save users the trouble of escaping `'&'` characters this way.

8.2.2. Query Forms: METHOD=GET

If the processing of a form is idempotent (i.e. it has no lasting observable effect on the state of the world), then the form method should be `'GET'`. Many database searches have no visible side-effects and make ideal applications of query forms.

To process a form whose action URL is an HTTP URL and whose method is ``GET'`, the user agent starts with the action URI and appends a ``?'` and the form data set, in ``application/x-www-form-urlencoded'` format as above. The user agent then traverses the link to this URI just as if it were an anchor (see 7.2, "Activation of Hyperlinks").

NOTE - The URL encoding may result in very long URIs, which cause some historical HTTP server implementations to exhibit defective behavior. As a result, some HTML forms are written using ``METHOD=POST'` even though the form submission has no side-effects.

8.2.3. Forms with Side-Effects: METHOD=POST

If the service associated with the processing of a form has side effects (for example, modification of a database or subscription to a service), the method should be ``POST'`.

To process a form whose action URL is an HTTP URL and whose method is ``POST'`, the user agent conducts an HTTP POST transaction using the action URI, and a message body of type ``application/x-www-form-urlencoded'` format as above. The user agent should display the response from the HTTP POST interaction just as it would display the response from an HTTP GET above.

8.2.4. Example Form Submission: Questionnaire Form

Consider the following document:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<title>Sample of HTML Form Submission</title>
<H1>Sample Questionnaire</H1>
<P>Please fill out this questionnaire:
<FORM METHOD="POST" ACTION="http://www.w3.org/sample">
<P>Your name: <INPUT NAME="name" size="48">
<P>Male <INPUT NAME="gender" TYPE=RADIO VALUE="male">
<P>Female <INPUT NAME="gender" TYPE=RADIO VALUE="female">
<P>Number in family: <INPUT NAME="family" TYPE=text>
<P>Cities in which you maintain a residence:
<UL>
<LI>Kent <INPUT NAME="city" TYPE=checkbox VALUE="kent">
<LI>Miami <INPUT NAME="city" TYPE=checkbox VALUE="miami">
<LI>Other <TEXTAREA NAME="other" cols=48 rows=4></textarea>
</UL>
Nickname: <INPUT NAME="nickname" SIZE="42">
<P>Thank you for responding to this questionnaire.
<P><INPUT TYPE=SUBMIT> <INPUT TYPE=RESET>
</FORM>
```

The initial state of the form data set is:

```
name      ""

gender    "male"

family    ""

other     ""

nickname  ""
```

Note that the radio input has an initial value, while the checkbox has none.

The user might edit the fields and request that the form be submitted. At that point, suppose the values are:

```
name      "John Doe"

gender    "male"

family    "5"

city      "kent"

city      "miami"

other     "abc\ndefk"

nickname  "J&D"
```

The user agent then conducts an HTTP POST transaction using the URI ``http://www.w3.org/sample'``. The message body would be (ignore the line break):

name=John+Doe&gender=male&family=5&city=kent&city=miami&
other=abc%0D%0Adef&nickname=J%26D

9. HTML Public Text

9.1. HTML DTD

This is the Document Type Definition for the HyperText Markup Language, level 2.

```
<!--      html.dtd

      Document Type Definition for the HyperText Markup Language
      (HTML DTD)

      $Id: html.dtd,v 1.30 1995/09/21 23:30:19 connolly Exp $

      Author: Daniel W. Connolly <connolly@w3.org>
      See Also: html.decl, html-1.dtd
      http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html
-->

<!ENTITY % HTML.Version
      "-//IETF//DTD HTML 2.0//EN"

      -- Typical usage:

      <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
      <html>
      ...
      </html>
--
>

<!--===== Feature Test Entities =====-->

<!ENTITY % HTML.Recommended "IGNORE"
      -- Certain features of the language are necessary for
      compatibility with widespread usage, but they may
      compromise the structural integrity of a document.
      This feature test entity enables a more prescriptive
      document type definition that eliminates
      those features.
-->

<![ %HTML.Recommended [
      <!ENTITY % HTML.Deprecated "IGNORE">
```

]]>

<!ENTITY % HTML.Deprecated "INCLUDE"

-- Certain features of the language are necessary for compatibility with earlier versions of the specification, but they tend to be used and implemented inconsistently, and their use is deprecated. This feature test entity enables a document type definition that eliminates these features.

-->

<!ENTITY % HTML.Highlighting "INCLUDE"

-- Use this feature test entity to validate that a document uses no highlighting tags, which may be ignored on minimal implementations.

-->

<!ENTITY % HTML.Forms "INCLUDE"

-- Use this feature test entity to validate that a document contains no forms, which may not be supported in minimal implementations

-->

<!--===== Imported Names =====-->

<!ENTITY % Content-Type "CDATA"

-- meaning an internet media type
(aka MIME content type, as per RFC1521)

-->

<!ENTITY % HTTP-Method "GET | POST"

-- as per HTTP specification, in progress

-->

<!--===== DTD "Macros" =====-->

<!ENTITY % heading "H1|H2|H3|H4|H5|H6">

<!ENTITY % list " UL | OL | DIR | MENU " >

<!--===== Character mnemonic entities =====-->

<!ENTITY % ISOLat1 PUBLIC

"ISO 8879-1986//ENTITIES Added Latin 1//EN//HTML">
%ISOLat1;

<!ENTITY amp CDATA "&" -- ampersand -->


```

<!ENTITY gt CDATA "&#62;"      -- greater than      -->
<!ENTITY lt CDATA "&#60;"      -- less than        -->
<!ENTITY quot CDATA "&#34;"    -- double quote     -->

```

```

<!--===== SGML Document Access (SDA) Parameter Entities =====>

```

```

<!-- HTML 2.0 contains SGML Document Access (SDA) fixed attributes
in support of easy transformation to the International Committee
for Accessible Document Design (ICADD) DTD

```

```

    "-//EC-USA-CDA/ICADD//DTD ICADD22//EN".

```

```

ICADD applications are designed to support usable access to
structured information by print-impaired individuals through
Braille, large print and voice synthesis. For more information on
SDA & ICADD:

```

- ISO 12083:1993, Annex A.8, Facilities for Braille, large print and computer voice
- ICADD ListServ
 <ICADD%ASUACAD.BITNET@ARIZVM1.ccit.arizona.edu>
- Usenet news group bit.listserv.easi
- Recording for the Blind, +1 800 221 4792

```

-->

```

```

<!ENTITY % SDAFORM "SDAFORM CDATA #FIXED"
    -- one to one mapping      -->
<!ENTITY % SDARULE "SDARULE CDATA #FIXED"
    -- context-sensitive mapping -->
<!ENTITY % SDAPREF "SDAPREF CDATA #FIXED"
    -- generated text prefix    -->
<!ENTITY % SDASUFF "SDASUFF CDATA #FIXED"
    -- generated text suffix     -->
<!ENTITY % SDASUSP "SDASUSP NAME #FIXED"
    -- suspend transform process -->

```

```

<!--===== Text Markup =====>

```

```

<![ %HTML.Highlighting [

```

```

<!ENTITY % font " TT | B | I ">

```

```

<!ENTITY % phrase "EM | STRONG | CODE | SAMP | KBD | VAR | CITE ">

```

```

<!ENTITY % text "#PCDATA | A | IMG | BR | %phrase | %font">

```

```

<!ELEMENT (%font;|%phrase) - - (%text)*>

```

```

<!ATTLIST ( TT | CODE | SAMP | KBD | VAR )
    %SDAFORM; "Lit"

```

```

    >
<!ATTLIST ( B | STRONG )
    %SDAFORM; "B"
    >
<!ATTLIST ( I | EM | CITE )
    %SDAFORM; "It"
    >

<!-- <TT>      Typewriter text          -->
<!-- <B>       Bold text                  -->
<!-- <I>       Italic text                -->

<!-- <EM>      Emphasized phrase          -->
<!-- <STRONG>  Strong emphasis            -->
<!-- <CODE>    Source code phrase         -->
<!-- <SAMP>    Sample text or characters  -->
<!-- <KBD>     Keyboard phrase, e.g. user input -->
<!-- <VAR>     Variable phrase or substitutable -->
<!-- <CITE>    Name or title of cited work -->

<!ENTITY % pre.content "#PCDATA | A | HR | BR | %font | %phrase">

]]>

<!ENTITY % text "#PCDATA | A | IMG | BR">

<!ELEMENT BR      - O EMPTY>
<!ATTLIST BR
    %SDAPREF; "&#RE;"
    >

<!-- <BR>      Line break          -->

<!--===== Link Markup =====>

<!ENTITY % linkType "NAMES">

<!ENTITY % linkExtraAttributes
    "REL %linkType #IMPLIED
    REV %linkType #IMPLIED
    URN CDATA #IMPLIED
    TITLE CDATA #IMPLIED
    METHODS NAMES #IMPLIED
    ">

<![ %HTML.Recommended [
    <!ENTITY % A.content "(%text)*"

```

```

    -- <H1><a name="xxx">Heading</a></H1>
       is preferred to
       <a name="xxx"><H1>Heading</H1></a>
    -->
]]>

<!ENTITY % A.content    "(%heading|%text)*">

<!ELEMENT A      - - %A.content -(A)>
<!ATTLIST A
    HREF CDATA #IMPLIED
    NAME CDATA #IMPLIED
    %linkExtraAttributes;
    %SDAPREF; "<Anchor: #AttList>"
    >
<!-- <A>                Anchor; source/destination of link      -->
<!-- <A NAME="...">    Name of this anchor                    -->
<!-- <A HREF="...">    Address of link destination            -->
<!-- <A URN="...">    Permanent address of destination        -->
<!-- <A REL=...>        Relationship to destination             -->
<!-- <A REV=...>        Relationship of destination to this     -->
<!-- <A TITLE="...">   Title of destination (advisory)        -->
<!-- <A METHODS="..."> Operations on destination (advisory)   -->

<!--===== Images =====>

<!ELEMENT IMG      - 0 EMPTY>
<!ATTLIST IMG
    SRC CDATA #REQUIRED
    ALT CDATA #IMPLIED
    ALIGN (top|middle|bottom) #IMPLIED
    ISMAP (ISMAP) #IMPLIED
    %SDAPREF; "<Fig><?SDATrans Img: #AttList>#AttVal(Alt)</Fig>"
    >

<!-- <IMG>                Image; icon, glyph or illustration    -->
<!-- <IMG SRC="...">    Address of image object                -->
<!-- <IMG ALT="...">    Textual alternative                    -->
<!-- <IMG ALIGN=...>     Position relative to text              -->
<!-- <IMG ISMAP>         Each pixel can be a link                -->

<!--===== Paragraphs=====>

<!ELEMENT P      - 0 (%text)*>
<!ATTLIST P
    %SDAFORM; "Para"
    >

```

RFC 1866

Hypertext Markup Language - 2.0

November 1995

```
<!-- <P>           Paragraph           -->
```

```
<!--===== Headings, Titles, Sections =====-->
```

```
<!ELEMENT HR      - O EMPTY>
```

```
<!ATTLIST HR
```

```
    %SDAPREF; "&#RE;&#RE;"
```

```
>
```

```
<!-- <HR>           Horizontal rule -->
```

```
<!ELEMENT ( %heading ) - - (%text;)*>
```

```
<!ATTLIST H1
```

```
    %SDAFORM; "H1"
```

```
>
```

```
<!ATTLIST H2
```

```
    %SDAFORM; "H2"
```

```
>
```

```
<!ATTLIST H3
```

```
    %SDAFORM; "H3"
```

```
>
```

```
<!ATTLIST H4
```

```
    %SDAFORM; "H4"
```

```
>
```

```
<!ATTLIST H5
```

```
    %SDAFORM; "H5"
```

```
>
```

```
<!ATTLIST H6
```

```
    %SDAFORM; "H6"
```

```
>
```

```
<!-- <H1>           Heading, level 1 -->
```

```
<!-- <H2>           Heading, level 2 -->
```

```
<!-- <H3>           Heading, level 3 -->
```

```
<!-- <H4>           Heading, level 4 -->
```

```
<!-- <H5>           Heading, level 5 -->
```

```
<!-- <H6>           Heading, level 6 -->
```

```
<!--===== Text Flows =====-->
```

```
<![ %HTML.Forms [
```

```
    <!ENTITY % block.forms "BLOCKQUOTE | FORM | ISINDEX">
```

```
]]>
```

```
<!ENTITY % block.forms "BLOCKQUOTE">
```

```

<![ %HTML.Deprecated [
    <!ENTITY % preformatted "PRE | XMP | LISTING">
]]>

<!ENTITY % preformatted "PRE">

<!ENTITY % block "P | %list | DL
    | %preformatted
    | %block.forms">

<!ENTITY % flow "(%text|%block)*">

<!ENTITY % pre.content "#PCDATA | A | HR | BR">
<!ELEMENT PRE - - (%pre.content)*>
<!ATTLIST PRE
    WIDTH NUMBER #IMPLIED
    %SDAFORM; "Lit"
>

<!-- <PRE>                Preformatted text                -->
<!-- <PRE WIDTH=...>      Maximum characters per line      -->

<![ %HTML.Deprecated [

<!ENTITY % literal "CDATA"
    -- historical, non-conforming parsing mode where
       the only markup signal is the end tag
       in full
    -->

<!ELEMENT (XMP|LISTING) - - %literal>
<!ATTLIST XMP
    %SDAFORM; "Lit"
    %SDAPREF; "Example:&#RE;"
>
<!ATTLIST LISTING
    %SDAFORM; "Lit"
    %SDAPREF; "Listing:&#RE;"
>

<!-- <XMP>                Example section                -->
<!-- <LISTING>            Computer listing                -->

<!ELEMENT PLAINTEXT - O %literal>
<!-- <PLAINTEXT>         Plain text passage                -->

<!ATTLIST PLAINTEXT
    %SDAFORM; "Lit"

```

```

    >
]]>

<!--===== Lists =====>

<!ELEMENT DL      - - (DT | DD)+>
<!ATTLIST DL
    COMPACT (COMPACT) #IMPLIED
    %SDAFORM; "List"
    %SDAPREF; "Definition List:"
>

<!ELEMENT DT      - 0 (%text)*>
<!ATTLIST DT
    %SDAFORM; "Term"
>

<!ELEMENT DD      - 0 %flow>
<!ATTLIST DD
    %SDAFORM; "LItem"
>

<!-- <DL>           Definition list, or glossary      -->
<!-- <DL COMPACT>   Compact style list               -->
<!-- <DT>           Term in definition list           -->
<!-- <DD>           Definition of term                -->

<!ELEMENT (OL|UL) - - (LI)+>
<!ATTLIST OL
    COMPACT (COMPACT) #IMPLIED
    %SDAFORM; "List"
>
<!ATTLIST UL
    COMPACT (COMPACT) #IMPLIED
    %SDAFORM; "List"
>

<!-- <UL>           Unordered list                   -->
<!-- <UL COMPACT>   Compact list style               -->
<!-- <OL>           Ordered, or numbered list        -->
<!-- <OL COMPACT>   Compact list style               -->

<!ELEMENT (DIR|MENU) - - (LI)+ -(%block)>
<!ATTLIST DIR
    COMPACT (COMPACT) #IMPLIED
    %SDAFORM; "List"
    %SDAPREF; "<LHead>Directory</LHead>"
>

```

```

<!ATTLIST MENU
    COMPACT (COMPACT) #IMPLIED
    %SDAFORM; "List"
    %SDAPREF; "<LHead>Menu</LHead>"
    >

<!-- <DIR>          Directory list          -->
<!-- <DIR COMPACT>   Compact list style      -->
<!-- <MENU>          Menu list               -->
<!-- <MENU COMPACT>  Compact list style      -->

<!ELEMENT LI        - 0 %flow>
<!ATTLIST LI
    %SDAFORM; "LItem"
    >

<!-- <LI>           List item                -->

<!--===== Document Body =====>

<![ %HTML.Recommended [
    <!ENTITY % body.content "(%heading|%block|HR|ADDRESS|IMG)*"
    -- <h1>Heading</h1>
       <p>Text ...
          is preferred to
       <h1>Heading</h1>
       Text ...
    -->
]]>

<!ENTITY % body.content "(%heading | %text | %block |
                        HR | ADDRESS)*">

<!ELEMENT BODY 0 0 %body.content>

<!-- <BODY>        Document body    -->

<!ELEMENT BLOCKQUOTE - - %body.content>
<!ATTLIST BLOCKQUOTE
    %SDAFORM; "BQ"
    >

<!-- <BLOCKQUOTE>   Quoted passage  -->

<!ELEMENT ADDRESS - - (%text|P)*>
<!ATTLIST ADDRESS
    %SDAFORM; "Lit"
    %SDAPREF; "Address:&#RE;"

```

```

    >

<!-- <ADDRESS> Address, signature, or byline -->

<!--===== Forms =====>

<![ %HTML.Forms [

<!ELEMENT FORM - - %body.content -(FORM) +(INPUT|SELECT|TEXTAREA)>
<!ATTLIST FORM
    ACTION CDATA #IMPLIED
    METHOD (%HTTP-Method) GET
    ENCTYPE %Content-Type; "application/x-www-form-urlencoded"
    %SDAPREF; "<Para>Form:</Para>"
    %SDASUFF; "<Para>Form End.</Para>"
    >

<!-- <FORM>                                Fill-out or data-entry form      -->
<!-- <FORM ACTION="...">                 Address for completed form      -->
<!-- <FORM METHOD=...>                     Method of submitting form        -->
<!-- <FORM ENCTYPE="...">               Representation of form data      -->

<!ENTITY % InputType "(TEXT | PASSWORD | CHECKBOX |
                        RADIO | SUBMIT | RESET |
                        IMAGE | HIDDEN )">

<!ELEMENT INPUT - O EMPTY>
<!ATTLIST INPUT
    TYPE %InputType TEXT
    NAME CDATA #IMPLIED
    VALUE CDATA #IMPLIED
    SRC CDATA #IMPLIED
    CHECKED (CHECKED) #IMPLIED
    SIZE CDATA #IMPLIED
    MAXLENGTH NUMBER #IMPLIED
    ALIGN (top|middle|bottom) #IMPLIED
    %SDAPREF; "Input: "
    >

<!-- <INPUT>                                Form input datum                -->
<!-- <INPUT TYPE=...>                       Type of input interaction       -->
<!-- <INPUT NAME=...>                       Name of form datum             -->
<!-- <INPUT VALUE="...">                 Default/initial/selected value -->
<!-- <INPUT SRC="...">                   Address of image                -->
<!-- <INPUT CHECKED>                       Initial state is "on"          -->
<!-- <INPUT SIZE=...>                     Field size hint                 -->
<!-- <INPUT MAXLENGTH=...>               Data length maximum            -->
<!-- <INPUT ALIGN=...>                   Image alignment                 -->

```



```

<!ELEMENT SELECT - - (OPTION+) -(INPUT|SELECT|TEXTAREA)>
<!ATTLIST SELECT
    NAME CDATA #REQUIRED
    SIZE NUMBER #IMPLIED
    MULTIPLE (MULTIPLE) #IMPLIED
    %SDAFORM; "List"
    %SDAPREF;
    "<LHead>Select #AttVal(Multiple)</LHead>"
>

<!-- <SELECT>                Selection of option(s)                -->
<!-- <SELECT NAME=...>        Name of form datum                  -->
<!-- <SELECT SIZE=...>        Options displayed at a time         -->
<!-- <SELECT MULTIPLE>        Multiple selections allowed         -->

<!ELEMENT OPTION - 0 (#PCDATA)*>
<!ATTLIST OPTION
    SELECTED (SELECTED) #IMPLIED
    VALUE CDATA #IMPLIED
    %SDAFORM; "LItem"
    %SDAPREF;
    "Option: #AttVal(Value) #AttVal(Selected)"
>

<!-- <OPTION>                A selection option                    -->
<!-- <OPTION SELECTED>        Initial state                        -->
<!-- <OPTION VALUE="...">   Form datum value for this option-->

<!ELEMENT TEXTAREA - - (#PCDATA)* -(INPUT|SELECT|TEXTAREA)>
<!ATTLIST TEXTAREA
    NAME CDATA #REQUIRED
    ROWS NUMBER #REQUIRED
    COLS NUMBER #REQUIRED
    %SDAFORM; "Para"
    %SDAPREF; "Input Text -- #AttVal(Name): "
>

<!-- <TEXTAREA>                An area for text input                -->
<!-- <TEXTAREA NAME=...>        Name of form datum                  -->
<!-- <TEXTAREA ROWS=...>        Height of area                      -->
<!-- <TEXTAREA COLS=...>        Width of area                      -->

]]>

```

```

<!--===== Document Head =====>

```

```

<![ %HTML.Recommended [

```

```

        <!ENTITY % head.extra "">
    ]]>
    <!ENTITY % head.extra "& NEXTID?">

    <!ENTITY % head.content "TITLE & ISINDEX? & BASE? %head.extra">

    <!ELEMENT HEAD O O (%head.content) +(META|LINK)>

    <!-- <HEAD>      Document head      -->

    <!ELEMENT TITLE - - (#PCDATA)* -(META|LINK)>
    <!ATTLIST TITLE
        %SDAFORM; "Ti"      >

    <!-- <TITLE>      Title of document -->

    <!ELEMENT LINK - O EMPTY>
    <!ATTLIST LINK
        HREF CDATA #REQUIRED
        %linkExtraAttributes;
        %SDAPREF; "Linked to : #AttVal (TITLE) (URN) (HREF)">    >

    <!-- <LINK>          Link from this document          -->
    <!-- <LINK HREF="..."> Address of link destination    -->
    <!-- <LINK URN="..."> Lasting name of destination     -->
    <!-- <LINK REL=...> Relationship to destination        -->
    <!-- <LINK REV=...> Relationship of destination to this -->
    <!-- <LINK TITLE="..."> Title of destination (advisory) -->
    <!-- <LINK METHODS="..."> Operations allowed (advisory) -->

    <!ELEMENT ISINDEX - O EMPTY>
    <!ATTLIST ISINDEX
        %SDAPREF;
        "<Para>[Document is indexed/searchable.]</Para>">

    <!-- <ISINDEX>      Document is a searchable index      -->

    <!ELEMENT BASE - O EMPTY>
    <!ATTLIST BASE
        HREF CDATA #REQUIRED      >

    <!-- <BASE>          Base context document          -->
    <!-- <BASE HREF="..."> Address for this document    -->

    <!ELEMENT NEXTID - O EMPTY>
    <!ATTLIST NEXTID
        N CDATA #REQUIRED      >

```

RFC 1866

Hypertext Markup Language - 2.0

November 1995

```

<!-- <NEXTID>           Next ID to use for link name      -->
<!-- <NEXTID N=...>     Next ID to use for link name      -->

<!ELEMENT META - O EMPTY>
<!ATTLIST META
    HTTP-EQUIV  NAME      #IMPLIED
    NAME        NAME      #IMPLIED
    CONTENT     CDATA     #REQUIRED  >

<!-- <META>             Generic Meta-information         -->
<!-- <META HTTP-EQUIV=...> HTTP response header name     -->
<!-- <META NAME=...>    Meta-information name           -->
<!-- <META CONTENT="..."> Associated information        -->

<!--===== Document Structure =====>

<![ %HTML.Deprecated [
    <!ENTITY % html.content "HEAD, BODY, PLAINTEXT?">
]]>
<!ENTITY % html.content "HEAD, BODY">

<!ELEMENT HTML O O (%html.content)>
<!ENTITY % version.attr "VERSION CDATA #FIXED '%HTML.Version;'">

<!ATTLIST HTML
    %version.attr;
    %SDAFORM; "Book"
    >

<!-- <HTML>             HTML Document      -->

```

9.2. Strict HTML DTD

This document type declaration refers to the HTML DTD with the `HTML.Recommended' entity defined as `INCLUDE' rather than IGNORE; that is, it refers to the more structurally rigid definition of HTML.

```

<!--    html-s.dtd

    Document Type Definition for the HyperText Markup Language
    with strict validation (HTML Strict DTD).

    $Id: html-s.dtd,v 1.3 1995/06/02 18:55:46 connolly Exp $

    Author: Daniel W. Connolly <connolly@w3.org>
    See Also: http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html

-->

```

```

<!ENTITY % HTML.Version
    "-//IETF//DTD HTML 2.0 Strict//EN"

    -- Typical usage:

        <!DOCTYPE HTML PUBLIC
            "-//IETF//DTD HTML Strict//EN">
        <html>
        ...
        </html>
    --
    >

<!-- Feature Test Entities -->
<!ENTITY % HTML.Recommended "INCLUDE">

<!ENTITY % html PUBLIC "-//IETF//DTD HTML 2.0//EN">
%html;

```

9.3. Level 1 HTML DTD

This document type declaration refers to the HTML DTD with the 'HTML.Forms' entity defined as 'IGNORE' rather than 'INCLUDE'. Documents which contain <FORM> elements do not conform to this DTD, and must use the level 2 DTD.

```

<!--      html-1.dtd

Document Type Definition for the HyperText Markup Language
with Level 1 Extensions (HTML Level 1 DTD).

$Id: html-1.dtd,v 1.2 1995/03/29 18:53:10 connolly Exp $

Author: Daniel W. Connolly <connolly@w3.org>
See Also: http://info.cern.ch/hypertext/WWW/MarkUp/MarkUp.html

-->

```

```

<!ENTITY % HTML.Version
    "-//IETF//DTD HTML 2.0 Level 1//EN"

    -- Typical usage:

        <!DOCTYPE HTML PUBLIC
            "-//IETF//DTD HTML Level 1//EN">
        <html>
        ...
        </html>

```

RFC 1866

Hypertext Markup Language - 2.0

November 1995

```
--
>
```

```
<!-- Feature Test Entities -->
<!ENTITY % HTML.Forms "IGNORE">
```

```
<!ENTITY % html PUBLIC "-//IETF//DTD HTML 2.0//EN">
%html;
```

9.4. Strict Level 1 HTML DTD

This document type declaration refers to the level 1 HTML DTD with the `HTML.Recommended' entity defined as `INCLUDE' rather than IGNORE; that is, it refers to the more structurally rigid definition of HTML.

```
<!--      html-1s.dtd

      Document Type Definition for the HyperText Markup Language
      Struct Level 1

      $Id: html-1s.dtd,v 1.3 1995/06/02 18:55:43 connolly Exp $

      Author: Daniel W. Connolly <connolly@w3.org>
      See Also: http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html
-->
```

```
<!ENTITY % HTML.Version
      "-//IETF//DTD HTML 2.0 Strict Level 1//EN"

      -- Typical usage:

      <!DOCTYPE HTML PUBLIC
          "-//IETF//DTD HTML Strict Level 1//EN">
      <html>
      ...
      </html>
      --
      >
```

```
<!-- Feature Test Entities -->
```

```
<!ENTITY % HTML.Recommended "INCLUDE">
```

```
<!ENTITY % html-1 PUBLIC "-//IETF//DTD HTML 2.0 Level 1//EN">
%html-1;
```

9.5. SGML Declaration for HTML

This is the SGML Declaration for HyperText Markup Language.

```
<!SGML "ISO 8879:1986"
```

```
--
```

```
    SGML Declaration for HyperText Markup Language (HTML).
```

```
--
```

CHARSET

```
    BASESET "ISO 646:1983//CHARSET
              International Reference Version
              (IRV)//ESC 2/5 4/0"
```

```
    DESCSET 0  9  UNUSED
              9  2  9
              11 2  UNUSED
              13 1  13
              14 18 UNUSED
              32 95 32
              127 1  UNUSED
```

```
    BASESET "ISO Registration Number 100//CHARSET
              ECMA-94 Right Part of
              Latin Alphabet Nr. 1//ESC 2/13 4/1"
```

```
    DESCSET 128 32  UNUSED
              160 96  32
```

CAPACITY

```
    SGMLREF
    TOTALCAP      150000
    GRPCAP        150000
    ENTCAP        150000
```

SCOPE DOCUMENT SYNTAX

```
    SHUNCHAR CONTROLS 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
                  17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 127
```

```
    BASESET "ISO 646:1983//CHARSET
              International Reference Version
              (IRV)//ESC 2/5 4/0"
```

```
    DESCSET 0 128 0
```

FUNCTION

```
    RE          13
    RS          10
    SPACE       32
    TAB SEPCHAR 9
```

```
    NAMING      LCNMSTRT ""
                  UCNMSTRT ""
```

```

        LCNMCHAR ".-"
        UCNMCHAR ".-"
        NAMECASE GENERAL YES
                ENTITY NO
    DELIM    GENERAL SGMLREF
        SHORTREF SGMLREF
    NAMES    SGMLREF
    QUANTITY SGMLREF
        ATTSPLN 2100
        LITLEN  1024
        NAMELEN 72    -- somewhat arbitrary; taken from
                        internet line length conventions --
        PILEN   1024
        TAGLVL  100
        TAGLEN  2100
        GRPGTCNT 150
        GRPCNT  64

```

FEATURES

MINIMIZE

```

    DATATAG NO
    OMITTAG YES
    RANK    NO
    SHORTTAG YES

```

LINK

```

    SIMPLE NO
    IMPLICIT NO
    EXPLICIT NO

```

OTHER

```

    CONCUR NO
    SUBDOC NO
    FORMAL YES

```

```

APPINFO    "SDA"  -- conforming SGML Document Access application
            --

```

>

<!--

```

    $Id: html.decl,v 1.17 1995/06/08 14:59:32 connolly Exp $

```

```

    Author: Daniel W. Connolly <connolly@w3.org>

```

```

    See also: http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html

```

-->

9.6. Sample SGML Open Entity Catalog for HTML

The SGML standard describes an "entity manager" as the portion or component of an SGML system that maps SGML entities into the actual storage model (e.g., the file system). The standard itself does not

define a particular mapping methodology or notation.

To assist the interoperability among various SGML tools and systems, the SGML Open consortium has passed a technical resolution that defines a format for an application-independent entity catalog that maps external identifiers and/or entity names to file names.

Each entry in the catalog associates a storage object identifier (such as a file name) with information about the external entity that appears in the SGML document. In addition to entries that associate public identifiers, a catalog entry can associate an entity name with a storage object identifier. For example, the following are possible catalog entries:

```
-- catalog: SGML Open style entity catalog for HTML --
-- $Id: catalog,v 1.3 1995/09/21 23:30:23 connolly Exp $ --

-- Ways to refer to Level 2: most general to most specific --
PUBLIC "-//IETF//DTD HTML//EN"          html.dtd
PUBLIC "-//IETF//DTD HTML 2.0//EN"       html.dtd
PUBLIC "-//IETF//DTD HTML Level 2//EN"   html.dtd
PUBLIC "-//IETF//DTD HTML 2.0 Level 2//EN" html.dtd

-- Ways to refer to Level 1: most general to most specific --
PUBLIC "-//IETF//DTD HTML Level 1//EN"    html-1.dtd
PUBLIC "-//IETF//DTD HTML 2.0 Level 1//EN" html-1.dtd

-- Ways to refer to
    Strict Level 2: most general to most specific --
PUBLIC "-//IETF//DTD HTML Strict//EN"      html-s.dtd
PUBLIC "-//IETF//DTD HTML 2.0 Strict//EN"   html-s.dtd
PUBLIC "-//IETF//DTD HTML Strict Level 2//EN" html-s.dtd
PUBLIC "-//IETF//DTD HTML 2.0 Strict Level 2//EN" html-s.dtd

-- Ways to refer to
    Strict Level 1: most general to most specific --
PUBLIC "-//IETF//DTD HTML Strict Level 1//EN" html-1s.dtd
PUBLIC "-//IETF//DTD HTML 2.0 Strict Level 1//EN" html-1s.dtd

-- ISO latin 1 entity set for HTML --
PUBLIC "ISO 8879-1986//ENTITIES Added Latin 1//EN//HTML" ISolat1\
sgml
```

9.7. Character Entity Sets

The HTML DTD defines the following entities. They represent particular graphic characters which have special meanings in places in the markup, or may not be part of the character set available to

the writer.

9.7.1. Numeric and Special Graphic Entity Set

The following table lists each of the characters included from the Numeric and Special Graphic entity set, along with its name, syntax for use, and description. This list is derived from `ISO Standard 8879:1986//ENTITIES Numeric and Special Graphic//EN'. However, HTML does not include for the entire entity set -- only the entities listed below are included.

GLYPH	NAME	SYNTAX	DESCRIPTION
<	lt	<	Less than sign
>	gt	>	Greater than sign
&	amp	&	Ampersand
"	quot	"	Double quote sign

9.7.2. ISO Latin 1 Character Entity Set

The following public text lists each of the characters specified in the Added Latin 1 entity set, along with its name, syntax for use, and description. This list is derived from ISO Standard 8879:1986//ENTITIES Added Latin 1//EN. HTML includes the entire entity set.

```
<!-- (C) International Organization for Standardization 1986
      Permission to copy in any form is granted for use with
      conforming SGML systems and applications as defined in
      ISO 8879, provided this notice is included in all copies.
-->
<!-- Character entity set. Typical invocation:
      <!ENTITY % ISLat1 PUBLIC
          "ISO 8879-1986//ENTITIES Added Latin 1//EN//HTML">
      %ISLat1;
-->
<!--      Modified for use in HTML
          $Id: ISLat1.sgml,v 1.2 1994/11/30 23:45:12 connolly Exp $ -->
<!ENTITY AElig CDATA "&#198;" -- capital AE diphthong (ligature) -->
<!ENTITY Aacute CDATA "&#193;" -- capital A, acute accent -->
<!ENTITY Acirc CDATA "&#194;" -- capital A, circumflex accent -->
<!ENTITY Agrave CDATA "&#192;" -- capital A, grave accent -->
<!ENTITY Aring CDATA "&#197;" -- capital A, ring -->
<!ENTITY Atilde CDATA "&#195;" -- capital A, tilde -->
<!ENTITY Auml CDATA "&#196;" -- capital A, dieresis or umlaut mark -->
<!ENTITY Ccedil CDATA "&#199;" -- capital C, cedilla -->
<!ENTITY ETH CDATA "&#208;" -- capital Eth, Icelandic -->
<!ENTITY Eacute CDATA "&#201;" -- capital E, acute accent -->
<!ENTITY Ecirc CDATA "&#202;" -- capital E, circumflex accent -->
```

```
<!ENTITY Egrave CDATA "&#200;" -- capital E, grave accent -->
<!ENTITY Euml CDATA "&#203;" -- capital E, dieresis or umlaut mark -->
<!ENTITY Iacute CDATA "&#205;" -- capital I, acute accent -->
<!ENTITY Icirc CDATA "&#206;" -- capital I, circumflex accent -->
<!ENTITY Igrave CDATA "&#204;" -- capital I, grave accent -->
<!ENTITY Iuml CDATA "&#207;" -- capital I, dieresis or umlaut mark -->
<!ENTITY Ntilde CDATA "&#209;" -- capital N, tilde -->
<!ENTITY Oacute CDATA "&#211;" -- capital O, acute accent -->
<!ENTITY Ocirc CDATA "&#212;" -- capital O, circumflex accent -->
<!ENTITY Ograve CDATA "&#210;" -- capital O, grave accent -->
<!ENTITY Oslash CDATA "&#216;" -- capital O, slash -->
<!ENTITY Otilde CDATA "&#213;" -- capital O, tilde -->
<!ENTITY Ouml CDATA "&#214;" -- capital O, dieresis or umlaut mark -->
<!ENTITY THORN CDATA "&#222;" -- capital THORN, Icelandic -->
<!ENTITY Uacute CDATA "&#218;" -- capital U, acute accent -->
<!ENTITY Ucirc CDATA "&#219;" -- capital U, circumflex accent -->
<!ENTITY Ugrave CDATA "&#217;" -- capital U, grave accent -->
<!ENTITY Uuml CDATA "&#220;" -- capital U, dieresis or umlaut mark -->
<!ENTITY Yacute CDATA "&#221;" -- capital Y, acute accent -->
<!ENTITY aacute CDATA "&#225;" -- small a, acute accent -->
<!ENTITY acirc CDATA "&#226;" -- small a, circumflex accent -->
<!ENTITY aelig CDATA "&#230;" -- small ae diphthong (ligature) -->
<!ENTITY agrave CDATA "&#224;" -- small a, grave accent -->
<!ENTITY aring CDATA "&#229;" -- small a, ring -->
<!ENTITY atilde CDATA "&#227;" -- small a, tilde -->
<!ENTITY auml CDATA "&#228;" -- small a, dieresis or umlaut mark -->
<!ENTITY ccedil CDATA "&#231;" -- small c, cedilla -->
<!ENTITY eacute CDATA "&#233;" -- small e, acute accent -->
<!ENTITY ecirc CDATA "&#234;" -- small e, circumflex accent -->
<!ENTITY egrave CDATA "&#232;" -- small e, grave accent -->
<!ENTITY eth CDATA "&#240;" -- small eth, Icelandic -->
<!ENTITY euml CDATA "&#235;" -- small e, dieresis or umlaut mark -->
<!ENTITY iacute CDATA "&#237;" -- small i, acute accent -->
<!ENTITY icirc CDATA "&#238;" -- small i, circumflex accent -->
<!ENTITY igrave CDATA "&#236;" -- small i, grave accent -->
<!ENTITY iuml CDATA "&#239;" -- small i, dieresis or umlaut mark -->
<!ENTITY ntilde CDATA "&#241;" -- small n, tilde -->
<!ENTITY oacute CDATA "&#243;" -- small o, acute accent -->
<!ENTITY ocirc CDATA "&#244;" -- small o, circumflex accent -->
<!ENTITY ograve CDATA "&#242;" -- small o, grave accent -->
<!ENTITY oslash CDATA "&#248;" -- small o, slash -->
<!ENTITY otilde CDATA "&#245;" -- small o, tilde -->
<!ENTITY ouml CDATA "&#246;" -- small o, dieresis or umlaut mark -->
<!ENTITY szlig CDATA "&#223;" -- small sharp s, German (sz ligature)-->
<!ENTITY thorn CDATA "&#254;" -- small thorn, Icelandic -->
<!ENTITY uacute CDATA "&#250;" -- small u, acute accent -->
<!ENTITY ucirc CDATA "&#251;" -- small u, circumflex accent -->
<!ENTITY ugrave CDATA "&#249;" -- small u, grave accent -->
```

```
<!ENTITY uuml CDATA "&#252;" -- small u, dieresis or umlaut mark -->
<!ENTITY yacute CDATA "&#253;" -- small y, acute accent -->
<!ENTITY yuml CDATA "&#255;" -- small y, dieresis or umlaut mark -->
```

10. Security Considerations

anchors, embedded images, and all other elements which contain URIs as parameters may cause the URI to be dereferenced in response to user input. In this case, the security considerations of [URL] apply.

The widely deployed methods for submitting forms requests -- HTTP and SMTP -- provide little assurance of confidentiality. Information providers who request sensitive information via forms -- especially by way of the 'PASSWORD' type input field (see 8.1.2, "Input Field: INPUT") -- should be aware and make their users aware of the lack of confidentiality.

11. References

[URI]

Berners-Lee, T., "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World- Wide Web", RFC 1630, CERN, June 1994.
<URL:ftp://ds.internic.net/rfc/rfc1630.txt>

[URL]

Berners-Lee, T., Masinter, L., and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, CERN, Xerox PARC, University of Minnesota, December 1994.
<URL:ftp://ds.internic.net/rfc/rfc1738.txt>

[HTTP]

Berners-Lee, T., Fielding, R., and H. Frystyk Nielsen, "Hypertext Transfer Protocol - HTTP/1.0", Work in Progress, MIT, UC Irvine, CERN, March 1995.

[MIME]

Borenstein, N., and N. Freed. "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, Bellcore, Innosoft, September 1993.
<URL:ftp://ds.internic.net/rfc/rfc1521.txt>

[RELURL]

Fielding, R., "Relative Uniform Resource Locators", RFC 1808, June 1995
<URL:ftp://ds.internic.net/rfc/rfc1808.txt>

[GOLD90]

Goldfarb, C., "The SGML Handbook", Y. Rubinsky, Ed.,
Oxford University Press, 1990.

[DEXTER]

Frank Halasz and Mayer Schwartz, "The Dexter Hypertext
Reference Model", Communications of the ACM, pp.
30-39, vol. 37 no. 2, Feb 1994.

[IMEDIA]

Postel, J., "Media Type Registration Procedure",
RFC 1590, USC/Information Sciences Institute, March 1994.
<URL:ftp://ds.internic.net/rfc/rfc1590.txt>

[IANA]

Reynolds, J., and J. Postel, "Assigned Numbers", STD 2,
RFC 1700, USC/Information Sciences Institute, October
1994. <URL:ftp://ds.internic.net/rfc/rfc1700.txt>

[SQ91]

SoftQuad. "The SGML Primer", 3rd ed., SoftQuad Inc.,
1991. <URL:http://www.sq.com/>

[ISO-646]

ISO/IEC 646:1991 Information technology -- ISO 7-bit
coded character set for information interchange
<URL:http://www.iso.ch/cate/d4777.html>

[ISO-10646]

ISO/IEC 10646-1:1993 Information technology -- Universal
Multiple-Octet Coded Character Set (UCS) -- Part 1:
Architecture and Basic Multilingual Plane
<URL:http://www.iso.ch/cate/d18741.html>

[ISO-8859-1]

ISO 8859. International Standard -- Information
Processing -- 8-bit Single-Byte Coded Graphic Character
Sets -- Part 1: Latin Alphabet No. 1, ISO 8859-1:1987.
<URL:http://www.iso.ch/cate/d16338.html>

[SGML]

ISO 8879. Information Processing -- Text and Office
Systems - Standard Generalized Markup Language (SGML),
1986. <URL:http://www.iso.ch/cate/d16387.html>

12. Acknowledgments

The HTML document type was designed by Tim Berners-Lee at CERN as part of the 1990 World Wide Web project. In 1992, Dan Connolly wrote the HTML Document Type Definition (DTD) and a brief HTML specification.

Since 1993, a wide variety of Internet participants have contributed to the evolution of HTML, which has included the addition of in-line images introduced by the NCSA Mosaic software for WWW. Dave Raggett played an important role in deriving the forms material from the HTML+ specification.

Dan Connolly and Karen Olson Muldrow rewrote the HTML Specification in 1994. The document was then edited by the HTML working group as a whole, with updates being made by Eric Schieler, Mike Knezovich, and Eric W. Sink at Spyglass, Inc. Finally, Roy Fielding restructured the entire draft into its current form.

Special thanks to the many active participants in the HTML working group, too numerous to list individually, without whom there would be no standards process and no standard. That this document approaches its objective of carefully converging a description of current practice and formalization of HTML's relationship to SGML is a tribute to their effort.

12.1. Authors' Addresses

Tim Berners-Lee
Director, W3 Consortium
MIT Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02139, U.S.A.

Phone: +1 (617) 253 9670
Fax: +1 (617) 258 8682
EMail: timbl@w3.org

Daniel W. Connolly
Research Technical Staff, W3 Consortium
MIT Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02139, U.S.A.

Phone: +1 (617) 258 8682
EMail: connolly@w3.org
URI: <http://www.w3.org/hypertext/WWW/People/Connolly/>

13. The HTML Coded Character Set

This list details the code positions and characters of the HTML document character set, specified in 9.5, "SGML Declaration for HTML". This coded character set is based on [ISO-8859-1].

REFERENCE	DESCRIPTION
-----	-----
� - 	Unused
		Horizontal tab

	Line feed
 - 	Unused
	Carriage Return
 - 	Unused
 	Space
!	Exclamation mark
"	Quotation mark
#	Number sign
$	Dollar sign
%	Percent sign
&	Ampersand
'	Apostrophe
(Left parenthesis
)	Right parenthesis
*	Asterisk
+	Plus sign
,	Comma
-	Hyphen
.	Period (fullstop)
/	Solidus (slash)
0 - 9	Digits 0-9
:	Colon
;	Semi-colon
<	Less than
=	Equals sign
>	Greater than
?	Question mark
@	Commercial at
A - Z	Letters A-Z
[Left square bracket
\	Reverse solidus (backslash)
]	Right square bracket
^	Caret
_	Horizontal bar (underscore)
`	Acute accent
a - z	Letters a-z
{	Left curly brace
|	Vertical bar

}	Right curly brace
~	Tilde
 - Ÿ	Unused
 	Non-breaking Space
¡	Inverted exclamation
¢	Cent sign
£	Pound sterling
¤	General currency sign
¥	Yen sign
¦	Broken vertical bar
§	Section sign
¨	Umlaut (dieresis)
©	Copyright
ª	Feminine ordinal
«	Left angle quote, guillemotleft
¬	Not sign
­	Soft hyphen
®	Registered trademark
¯	Macron accent
°	Degree sign
±	Plus or minus
²	Superscript two
³	Superscript three
´	Acute accent
µ	Micro sign
¶	Paragraph sign
·	Middle dot
¸	Cedilla
¹	Superscript one
º	Masculine ordinal
»	Right angle quote, guillemotright
¼	Fraction one-fourth
½	Fraction one-half
¾	Fraction three-fourths
¿	Inverted question mark
À	Capital A, grave accent
Á	Capital A, acute accent
Â	Capital A, circumflex accent
Ã	Capital A, tilde
Ä	Capital A, dieresis or umlaut mark
Å	Capital A, ring
Æ	Capital AE diphthong (ligature)
Ç	Capital C, cedilla
È	Capital E, grave accent
É	Capital E, acute accent
Ê	Capital E, circumflex accent
Ë	Capital E, dieresis or umlaut mark
Ì	Capital I, grave accent

Í	Capital I, acute accent
Î	Capital I, circumflex accent
Ï	Capital I, dieresis or umlaut mark
Ð	Capital Eth, Icelandic
Ñ	Capital N, tilde
Ò	Capital O, grave accent
Ó	Capital O, acute accent
Ô	Capital O, circumflex accent
Õ	Capital O, tilde
Ö	Capital O, dieresis or umlaut mark
×	Multiply sign
Ø	Capital O, slash
Ù	Capital U, grave accent
Ú	Capital U, acute accent
Û	Capital U, circumflex accent
Ü	Capital U, dieresis or umlaut mark
Ý	Capital Y, acute accent
Þ	Capital THORN, Icelandic
ß	Small sharp s, German (sz ligature)
à	Small a, grave accent
á	Small a, acute accent
â	Small a, circumflex accent
ã	Small a, tilde
ä	Small a, dieresis or umlaut mark
å	Small a, ring
æ	Small ae diphthong (ligature)
ç	Small c, cedilla
è	Small e, grave accent
é	Small e, acute accent
ê	Small e, circumflex accent
ë	Small e, dieresis or umlaut mark
ì	Small i, grave accent
í	Small i, acute accent
î	Small i, circumflex accent
ï	Small i, dieresis or umlaut mark
ð	Small eth, Icelandic
ñ	Small n, tilde
ò	Small o, grave accent
ó	Small o, acute accent
ô	Small o, circumflex accent
õ	Small o, tilde
ö	Small o, dieresis or umlaut mark
÷	Division sign
ø	Small o, slash
ù	Small u, grave accent
ú	Small u, acute accent
û	Small u, circumflex accent
ü	Small u, dieresis or umlaut mark

ý	Small y, acute accent
þ	Small thorn, Icelandic
ÿ	Small y, dieresis or umlaut mark

14. Proposed Entities

The HTML DTD references the "Added Latin 1" entity set, which only supplies named entities for a subset of the non-ASCII characters in [ISO-8859-1], namely the accented characters. The following entities should be supported so that all ISO 8859-1 characters may only be referenced symbolically. The names for these entities are taken from the appendixes of [SGML].

```

<!ENTITY nbsp    CDATA "&#160;" -- no-break space -->
<!ENTITY iexcl   CDATA "&#161;" -- inverted exclamation mark -->
<!ENTITY cent    CDATA "&#162;" -- cent sign -->
<!ENTITY pound   CDATA "&#163;" -- pound sterling sign -->
<!ENTITY curren  CDATA "&#164;" -- general currency sign -->
<!ENTITY yen     CDATA "&#165;" -- yen sign -->
<!ENTITY brvbar  CDATA "&#166;" -- broken (vertical) bar -->
<!ENTITY sect    CDATA "&#167;" -- section sign -->
<!ENTITY uml     CDATA "&#168;" -- umlaut (dieresis) -->
<!ENTITY copy    CDATA "&#169;" -- copyright sign -->
<!ENTITY ordf    CDATA "&#170;" -- ordinal indicator, feminine -->
<!ENTITY laquo   CDATA "&#171;" -- angle quotation mark, left -->
<!ENTITY not     CDATA "&#172;" -- not sign -->
<!ENTITY shy     CDATA "&#173;" -- soft hyphen -->
<!ENTITY reg     CDATA "&#174;" -- registered sign -->
<!ENTITY macr    CDATA "&#175;" -- macron -->
<!ENTITY deg     CDATA "&#176;" -- degree sign -->
<!ENTITY plusmn  CDATA "&#177;" -- plus-or-minus sign -->
<!ENTITY sup2    CDATA "&#178;" -- superscript two -->
<!ENTITY sup3    CDATA "&#179;" -- superscript three -->
<!ENTITY acute   CDATA "&#180;" -- acute accent -->
<!ENTITY micro   CDATA "&#181;" -- micro sign -->
<!ENTITY para    CDATA "&#182;" -- pilcrow (paragraph sign) -->
<!ENTITY middot  CDATA "&#183;" -- middle dot -->
<!ENTITY cedil   CDATA "&#184;" -- cedilla -->
<!ENTITY sup1    CDATA "&#185;" -- superscript one -->
<!ENTITY ordm    CDATA "&#186;" -- ordinal indicator, masculine -->
<!ENTITY raquo   CDATA "&#187;" -- angle quotation mark, right -->
<!ENTITY frac14  CDATA "&#188;" -- fraction one-quarter -->
<!ENTITY frac12  CDATA "&#189;" -- fraction one-half -->
<!ENTITY frac34  CDATA "&#190;" -- fraction three-quarters -->
<!ENTITY iquest  CDATA "&#191;" -- inverted question mark -->
<!ENTITY Agrave  CDATA "&#192;" -- capital A, grave accent -->
<!ENTITY Aacute  CDATA "&#193;" -- capital A, acute accent -->
<!ENTITY Acirc   CDATA "&#194;" -- capital A, circumflex accent -->

```

```
<!ENTITY Atilde CDATA "&#195;" -- capital A, tilde -->
<!ENTITY Auml CDATA "&#196;" -- capital A, dieresis or umlaut mark -->
<!ENTITY Aring CDATA "&#197;" -- capital A, ring -->
<!ENTITY Aelig CDATA "&#198;" -- capital AE diphthong (ligature) -->
<!ENTITY Ccedil CDATA "&#199;" -- capital C, cedilla -->
<!ENTITY Egrave CDATA "&#200;" -- capital E, grave accent -->
<!ENTITY Eacute CDATA "&#201;" -- capital E, acute accent -->
<!ENTITY Ecirc CDATA "&#202;" -- capital E, circumflex accent -->
<!ENTITY Euml CDATA "&#203;" -- capital E, dieresis or umlaut mark -->
<!ENTITY Igrave CDATA "&#204;" -- capital I, grave accent -->
<!ENTITY Iacute CDATA "&#205;" -- capital I, acute accent -->
<!ENTITY Icirc CDATA "&#206;" -- capital I, circumflex accent -->
<!ENTITY Iuml CDATA "&#207;" -- capital I, dieresis or umlaut mark -->
<!ENTITY ETH CDATA "&#208;" -- capital Eth, Icelandic -->
<!ENTITY Ntilde CDATA "&#209;" -- capital N, tilde -->
<!ENTITY Ograve CDATA "&#210;" -- capital O, grave accent -->
<!ENTITY Oacute CDATA "&#211;" -- capital O, acute accent -->
<!ENTITY Ocirc CDATA "&#212;" -- capital O, circumflex accent -->
<!ENTITY Otilde CDATA "&#213;" -- capital O, tilde -->
<!ENTITY Ouml CDATA "&#214;" -- capital O, dieresis or umlaut mark -->
<!ENTITY times CDATA "&#215;" -- multiply sign -->
<!ENTITY Oslash CDATA "&#216;" -- capital O, slash -->
<!ENTITY Ugrave CDATA "&#217;" -- capital U, grave accent -->
<!ENTITY Uacute CDATA "&#218;" -- capital U, acute accent -->
<!ENTITY Ucirc CDATA "&#219;" -- capital U, circumflex accent -->
<!ENTITY Uuml CDATA "&#220;" -- capital U, dieresis or umlaut mark -->
<!ENTITY Yacute CDATA "&#221;" -- capital Y, acute accent -->
<!ENTITY THORN CDATA "&#222;" -- capital THORN, Icelandic -->
<!ENTITY szlig CDATA "&#223;" -- small sharp s, German (sz ligature) -->
<!ENTITY agrave CDATA "&#224;" -- small a, grave accent -->
<!ENTITY aacute CDATA "&#225;" -- small a, acute accent -->
<!ENTITY acirc CDATA "&#226;" -- small a, circumflex accent -->
<!ENTITY atilde CDATA "&#227;" -- small a, tilde -->
<!ENTITY auml CDATA "&#228;" -- small a, dieresis or umlaut mark -->
<!ENTITY aring CDATA "&#229;" -- small a, ring -->
<!ENTITY aelig CDATA "&#230;" -- small ae diphthong (ligature) -->
<!ENTITY ccedil CDATA "&#231;" -- small c, cedilla -->
<!ENTITY egrave CDATA "&#232;" -- small e, grave accent -->
<!ENTITY eacute CDATA "&#233;" -- small e, acute accent -->
<!ENTITY ecirc CDATA "&#234;" -- small e, circumflex accent -->
<!ENTITY euml CDATA "&#235;" -- small e, dieresis or umlaut mark -->
<!ENTITY igrave CDATA "&#236;" -- small i, grave accent -->
<!ENTITY iacute CDATA "&#237;" -- small i, acute accent -->
<!ENTITY icirc CDATA "&#238;" -- small i, circumflex accent -->
<!ENTITY iuml CDATA "&#239;" -- small i, dieresis or umlaut mark -->
<!ENTITY eth CDATA "&#240;" -- small eth, Icelandic -->
<!ENTITY ntilde CDATA "&#241;" -- small n, tilde -->
<!ENTITY ograve CDATA "&#242;" -- small o, grave accent -->
```

```
<!ENTITY oacute CDATA "&#243;" -- small o, acute accent -->
<!ENTITY ocirc CDATA "&#244;" -- small o, circumflex accent -->
<!ENTITY otilde CDATA "&#245;" -- small o, tilde -->
<!ENTITY ouml CDATA "&#246;" -- small o, dieresis or umlaut mark -->
<!ENTITY divide CDATA "&#247;" -- divide sign -->
<!ENTITY oslash CDATA "&#248;" -- small o, slash -->
<!ENTITY ugrave CDATA "&#249;" -- small u, grave accent -->
<!ENTITY uacute CDATA "&#250;" -- small u, acute accent -->
<!ENTITY ucirc CDATA "&#251;" -- small u, circumflex accent -->
<!ENTITY uuml CDATA "&#252;" -- small u, dieresis or umlaut mark -->
<!ENTITY yacute CDATA "&#253;" -- small y, acute accent -->
<!ENTITY thorn CDATA "&#254;" -- small thorn, Icelandic -->
<!ENTITY yuml CDATA "&#255;" -- small y, dieresis or umlaut mark -->
```