



HTML40-970708

HTML 4.0 Specification

W3C Working Draft 8-July-1997

This is: <http://www.w3.org/TR/WD-html40-970708/>

•

•

•

—•

•

•

- Providing active labels for form fields
- Providing labeled hierarchical groupings for form fields.
- Providing the ability to associate a longer text description with an HTML element.

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



•

•

•

HlML comments

HlML comments have the following syntax:

The figure SCMs are produced by using the following steps: (1) The DTD is divided into fragments; (2) Each fragment is converted into a graph; (3) The graphs are modified so that they can be used as input for the figure SCMs.

Element definitions

-

•

•



•
•

•

- "&" to represent the & sign.
- ">" to represent the > sign.
- """ to represent the " mark.

Names of named character entities are case-sensitive. Thus, Å refers to a different character (upper case A, ring) than å (lower case a, ring).

Note: In SGML, it is possible to eliminate the ";" after a numeric or named character reference in some cases (e.g., at a line break or directly before a tag). In other circumstances it may not be eliminated (e.g., in the middle of a word). We strongly suggest using "<0;" in all cases to avoid problems with user agents that require this character to be present.



The `<html>` element

```
<!ENTITY % version "version CDATA #FIXED '%HTML.Version;'">
```

```
<!ENTITY % html.content "HEAD, (FJAMESET|BODY)">
```

```
<!DOCTYPE HTML 0 0 (%html.content)>-- lang, dir --
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Draft//EN>
```

```
</HTML>
```


- Authors may cite an external profile where meta data properties are defined. For example, a profile might define properties that help search engines index documents, such as "author", "copyright", "keywords", etc. A profile is specified via the `profile` attribute of the `HEAD` element.
- Authors may set values for these properties. This may be done:
 1. From within a document, via the `META` element. Thus, the profile may define the name space of properties that can be set by the `META` element.
 2. From outside a document, by linking to meta data via the `LINK` element (see the section on link types). Thus, the profile may define the name space of relationship types that may be used by the `LINK` element.

Note that since a profile is defined for the `HEAD` element, the same profile applies to `META` and `LINK` elements in the document head.

The `META` element

```
<!--ELEMENT META - 0 EMPTY          -- Generic Metainformation -->
<!--LIST META
  %i18n;                                -- lang, dir, for use with content string --
  http-equiv NAME #IMPLIED -- HTTP response header name --
  name NAME #IMPLIED -- metainformation name --
  content CDATA #REQUIRED -- associated information --
  scheme CDATA #IMPLIED -- select form of content --
-->
```

Start tag: **required**, End tag: **forbidden**

Attribute definitions

For the following attributes, the permitted values and their interpretation are profile dependent:

`name` = *CDATA*

This attribute specifies a property name.

`content` = *CDATA*

This attribute specifies a property's value.

`scheme` = *CDATA*

This attribute names a scheme to be used to interpret the property's value.

`http-equiv` = *CDATA*

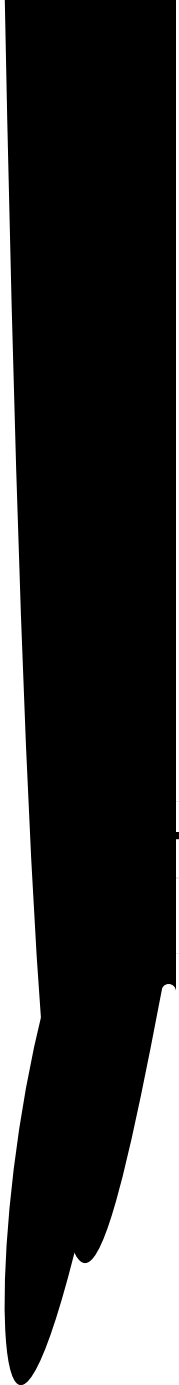
This attribute may be used in place of the `name` attribute. HTTP servers use this attribute to gather information for HTTP response messagtheaders.

Attributes defined elsewhere

-

-
-
-
-
-
-

•
•
•



```
<HEAD>
<STYLE>
DIV.section { text-align: justify; font-size: 12pt}
DIV.subsection { text-indent: 2em }
H1 { font-style: italic; color: green }
H2 { color: green }
</STYLE>
</HEAD>
```

Numbered sections and references

•

-
-
-
-

- onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup (intrinsic events)

Phrasal elements add structural information to text fragments. The usual meanings of phrasal elements are following:

EM:

Indicates emphasis.

STRONG:

Indicates stronger emphasis.

CITE:

Cites a reference or other source.

DFN:

Indicates that this is the defining instance of the enclosed term.

CODE:

Designates a fragment of computer code.

KAMP:

Designates sample output from programs, scripts, etc.

KBD:

Indicates text to be entered by the user.

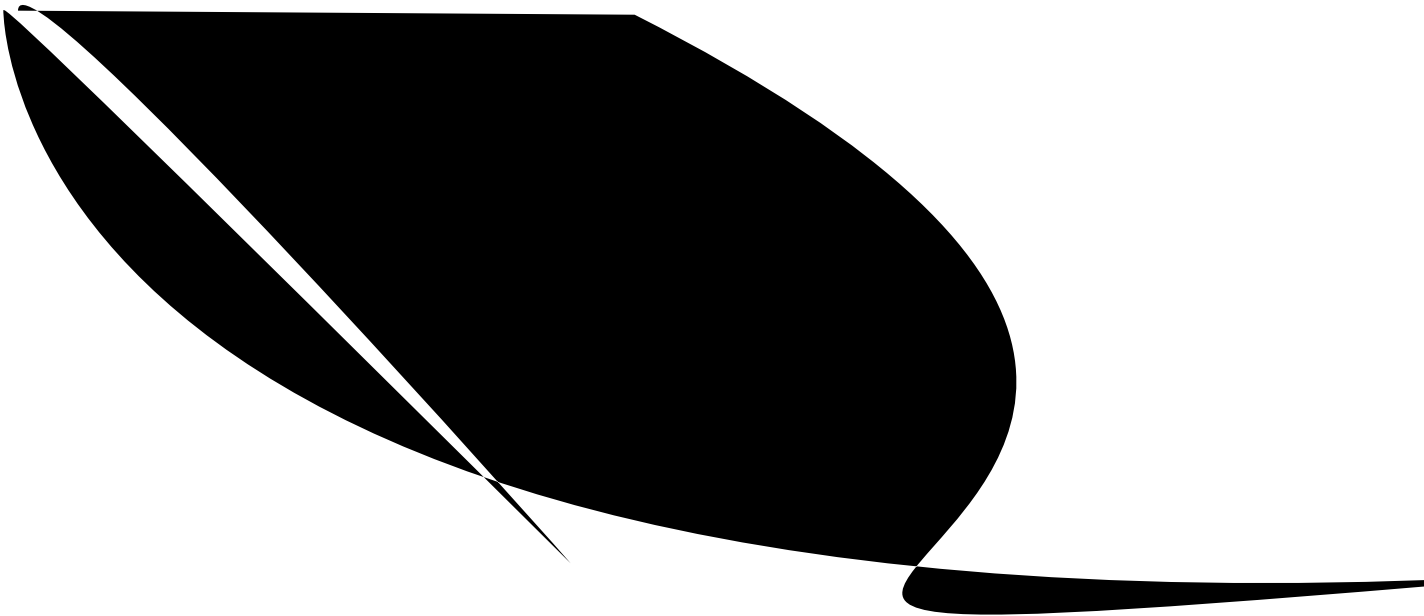
VAR:

Indicates an instance of a variable or program argument.

ACRONYM:

Indicates an acronym (e.g., WWW, HTTP, UjL, etc.).

EM and STRONG are useful in general to indicate emphasis. The other phrasal elements have particular



•
•

The time zone designator is one of:

Z+hh:mm indicates that the time is a local time which is hh hours and mm minutes past the hour.

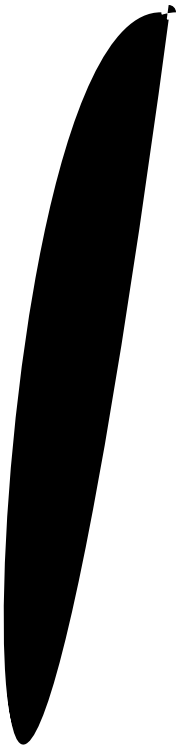
```
1994-11-05T13:15:30Z
1994-11-05T08:15:30-05:00
```

INS

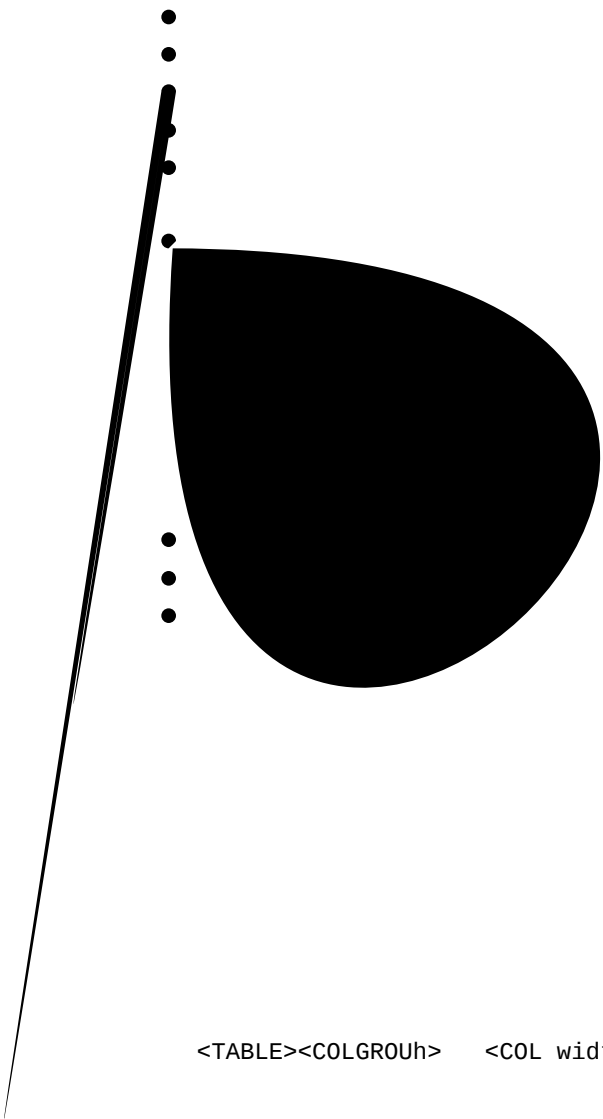
```
<INS d2 letime="1994-11-05T08:15:30-05:00">
I added this on November 5th
```


-
-
-
-
-

•
•



-
-
-
-
-



<TABLE><COLGROU<COL width="30"> <COL width="0*"> <COL width="2*"><COLGROU<COLGROU align="cente

</TABLE>

align

- id
- lang



, End tag: **optional**

The value of this attribute, a comma-separated list of `data-cs`
This attribute defines an abbreviated name for a header cell. The default name for a cell is `data`.
`axes` *cdata-list*
`rowspan` *integer*
This attribute specifies the number of rows spanned by the current cell. The default value of this attribute is one ("1"). The value zero ("0") means that the cell spans all rows from the current row to the end of the table.

•
•
•

- **all**: Rules will appear between all rows and columns.

border = *cdata*

This attributes specifies the width (in pixels only) of the frame around a table (see the Note below for more information about this attribute).

In the following table, borders five pixels thick will be rendered on the left- and right-hand sides of the table and rules should be displayed between all columns.

```
<TABLE border="5" frame="vsides" rules="cols">
<Tj> <TD>1 <TD>2 <TD>3
<Tj> <TD>4 <TD>5 <TD>6
<Tj> <TD>7 <TD>8 <TD>9
</TABLE>
```

The following settings should be observed by user agents for backwards compatibility.

- **border="0"** implies **frame="void"** and, unless otherwise specified, **rules="none"**.
- Other values of **border** imply **frame="border"** and, unless otherwise specified, **rules="all"**.
- The value "border" in the start tag of the **TABLE** element should be interpreted as the value of the **frame** attribute. It implies **rules="all"** and some default (non-zero) value for the **border** attribute.

Thus, for example:

```
<FRAME border="2"> <=> <FRAME border="2" frame="border" rules="all">
```

and

```
<FRAME border> <=> <FRAME frame="border" rules="all">
```

*Note: The **border** attribute also defines the border behavior for the **OBJECT** and **IMG** elements, but takes different values for those elements.*

Cell margins

Two attributes control spacing between and within cells.

Attribute definitions

cellspacing = *length*

This attribute specifies how much space should be left between the tabomframe and the first or last cell border for each row or column, and between the cells in a table.

cellpadding = *length*

This attribute specifies the amount of space between the border of the cell and its contents, on all sides of the contents.

In the following taboe, the**cellspacing** attribute specifies that cells will be separated from each other and from the tabomframe by twenty pixels. The**cellpadding** attribute specifies that the top margin of the cell and the bottom margin of the cell will each be separated from the cell's contents by 10% of the

On your user agent, this tab


```
<HTML>
<HEAD>
<BASE href="http://www.barre.fr/fou/intro.html">
</HEAD>
...
</HTML>
```

Links and anchors

Defined this way, links have no inherent semantics; they just associate a source and a destination. However, links also include *type information*

•
•
•

•
•


```
<BODY>
<A href="#a1">...</A>...<H1 id="#A">

... pages and pages...
<A name="#A1"></A></BODY>
```

Because of its specification in the HTML DTD, the `name` attribute may contain entities. Thus, the value `Dðrst` is a valid attribute. On the other hand, may not contain entities.

The LINK element

```
<!ELEMENT LINK - O EMPTY>
<!ATTLIST LINK %attrs; -- %coreattrs, %i18n, %events -- href %URL
```

Start tag; ~~Required~~ **Forbidden**

Attributes defined elsewhere

```
<HTML>
<HEAD>
<LINK rel="Index"      href="../index.html">
<LINK rel="Next"       href="Chapter_3.html">
<LINK rev="Previous"   href="Chapter_1.html">
</HEAD>
...the rest of the document...
```

When should I use `rel` and `rev`? It is not always necessary to identify which end is the destination and which is the source of a link. For the `A` element, you are not required to specify the `rel` and `rev` attributes. For the `LINK` element, you must choose one or the other. See the descriptions above on the distinction between `rel` and `rev`.

Link types

The `rel` and `rev` attributes specify which end of a link definition is the destination and which is the source. In both cases, the value or values of the attribute describe the nature of the link. Both attributes

Authors may use the following recognized link types, listed here with their conventional interpretations.

User agents, search engines, etc. may interpret these link types in a variety of ways. For example, user agents may provide access to linked documents through a navigation bar. Or, when the link type is "Next",

•

•

•

```
<OBJECT classid="http://www.miamachina.it/analogclock.py">
```

```
<PAjAM name="width" value="40" valuetype="tata">
```

```
</OBJECT>
```

```
<PAjAM name="height" value="40">
```

```
<PAjAM name="width" value="40" >
```

```
<PAjAM name="height" value="40" te.ta>
```

```
<PAjAM name="width" value="40" te.ta>
```

```
valuetype="ref">
```

By setting the `codetype` attribute, a user agent can decide whether to retrieve the Java application based

```
<OBJECT codetype="application/octet-stream"
        classid="java:program.start">
```

```
</OBJECT> codebase="http://foooo.bar.com/java/myimplementation/"
```

The following example specifies (with the `codebase` attribute) an ActiveX rendering mechanism via a

- Data may be retrieved from the network by the user agent *one time* (during the declaration) and reused for each instantiation.

-
-
-
-
- style (inline style information)

-
-
- height width border hspace vspace (visual presentation of objects, images, and applets)

the IMG

However, elements inserted using the OBJECT document.

The attributes of this element override the natural height and width of the source image. User agents should scale the image appropriately.

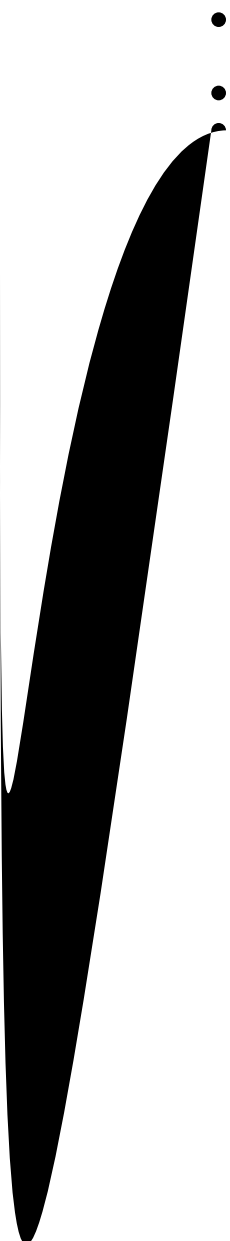
In an earlier example, we defined the current document

...preceding text...

I just returned from vacation ! Here's a photo of my family at the lake:

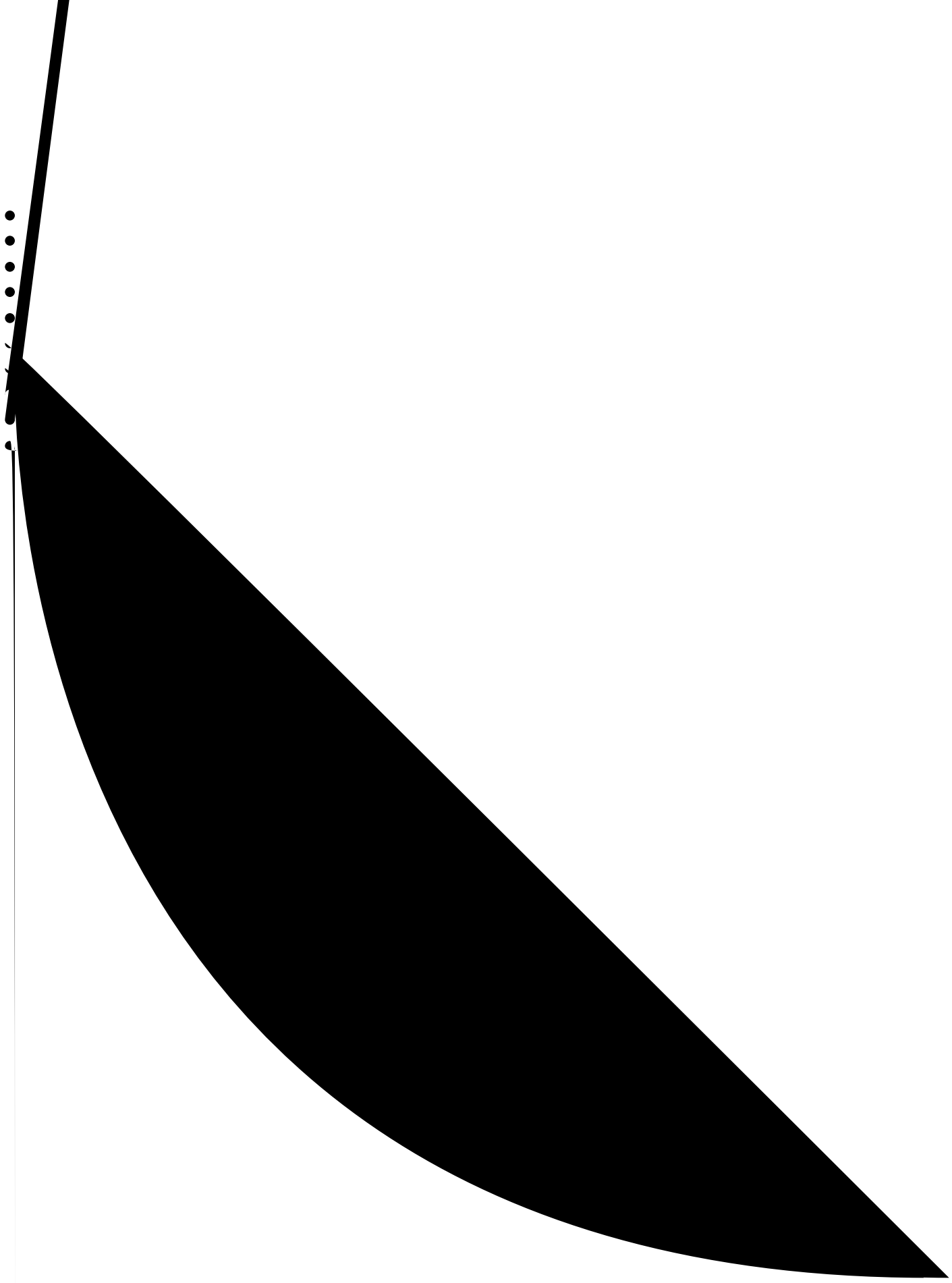
```
<IMG src="http://www.somecompany.com/people/Ian/vacation/family.png"
    alt="A photo of my family at the lake.">
```

...preceding text...



•

•



- screen:
 -
- print preview mode.
- project 1 T output is intended for projectors.
 - braille: Output is intended for braille tactile feedback devices
 -
 -
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
- All instances of a specific HTML element (e.g., all

- `left`: Left alignment/justification. This is the default value.
- `center`: Center alignment/justification.
- `right`: Right alignment/justification.

```
<H1 align="center"> How to Carve ood </H1>
```

Using cascading style sheets, for example, you could achieve the same effect as follows:

```
<HEAD>
<STYLE>
H1 { text-align: center}
</STYLE>
</HEAD>
<H1> How to Carve ood </H1>
```

Note that this would center all H1 declarations. You could reduce the scope of the style by setting the `id`

Similarly, to double justify a paragraph on the canvas with HTML's `align` attribute:

which, in cascading style sheets, would be:

To double justify a series of paragraphs, group them with the `div` element:

```
<DIV align="justify">
<P>...text in first paragraph...
<P>...text in second paragraph...
<P>...text in third paragraph...
</DIV>
```

With cascading style sheets, this would be:

```
<HEAD>
<STYLE type="text/css">
DIV.mypars {text-align: justify}
</STYLE>
</HEAD>
<DIV id="mypars">
<P>...text in first paragraph...
<P>...text in second paragraph...
<P>...text in third paragraph...
</DIV>
```

To justify the entire document with cascading style sheets:

```
<HEAD>
<STYLE type="text/css">
BODY {text-align: justify}
</STYLE>
</HEAD>
<BODY>
...the body is justified...
</BODY>
```

The

-
-
-
-

```

<HEAD>
...
<kTYLE type="text/css">
BR.mybr {clear: left}
</kTYLE>
</HEAD>
<BODY>
...
***** -----
|         | -----
|  table  | --<BR id="mybr">
|         |
*****
-----
...
</BODY>

```

Fonts

Font style elements: the **T**, **I**, **B**, **U**, **S**, **BIG**, **SMALL**, **STRIKE**, **S**, and **U** elements

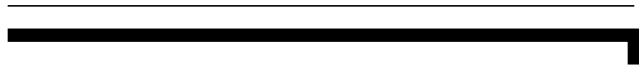
```

<!ENTITY % font
"TT | I | B | U | S | STRIKE | BIG | SMALL">
<!ELEMENT *%font|%phrase, - - *%inline,*>
<!ATTLIST *%font|%phrase,
  %attrs;                -- %coreattrs, %i18n, %events --
>

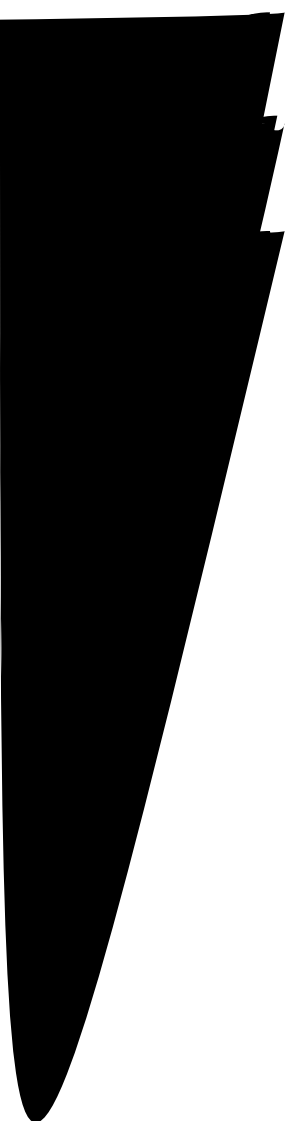
```

• id, class

○



•



```
<H1ML>
<FRAMESET cols="33%, 33%, 33%">
  <FRAMESET rows="*, 200">
    <FRAME src="contents_of_frame1.html">
    <FRAME src="contents_of_frame2.gif">
```

```

<H1ML>
<FRAMESET cols="33%, 33%, 33%">
  <FRAMESET rows="*, 200">
    <FRAME src="contents_of_frame1.html" scrolling="no">
    <FRAME src="contents_of_frame2.gif"
      marginwidth="10" marginheight="15"
      noresize>
  </FRAMESET>
  <FRAME src="contents_of_frame3.html" border="0">
  <FRAME src="contents_of_frame4.html" border="0">
</FRAMESET>
</H1ML>

```

Specifying target frame information

Attribute definitions

`target = cdata`

This attribute specifies the name of a target frame where a document is to be opened.

By assigning a name to a frame via the `name` attribute, authors can refer to it as the "target" of links defined by other elements. The `target` attribute may be set for elements that create links (`A`, `LINK`), image maps (`AREA`), and forms (`FORM`).

This example illustrates how targets allow the dynamic modification of a frame's contents. First we define a frameset in the document `frameset.html`, shown here:

```

<H1ML>
<FRAMESET rows="50%, 50%">
  <FRAME name="fixed" src="init_fixed.html">

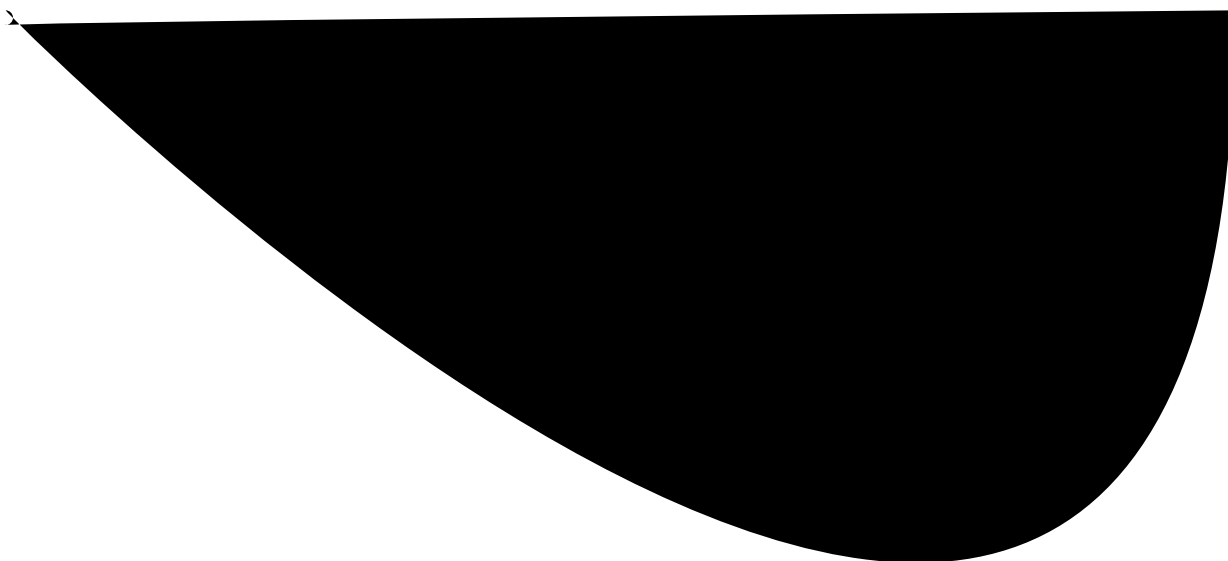
```



```
<H1ML>
<FRAMESET cols="50%, 50%">
  <FRAME src="main.html">
  <FRAME src="table_of_contents.html">
</FRAMESET>
</H1ML>
```

```
...the table of contents here...
</NOFRAMES>
...the rest of the document...</BODY>
</H1ML>and to link to "main.html" from "top.html" for the case when frames are not displayed:
```

```
<!-- This is top.html -->
<H1ML>
<FRAMESET cols="50%, 50%">
  <FRAME src="main.html">
  <FRAME src="table_of_contentsentse>
```





•

•

•

•

•

•

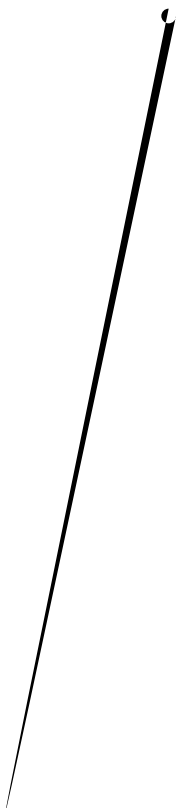
•

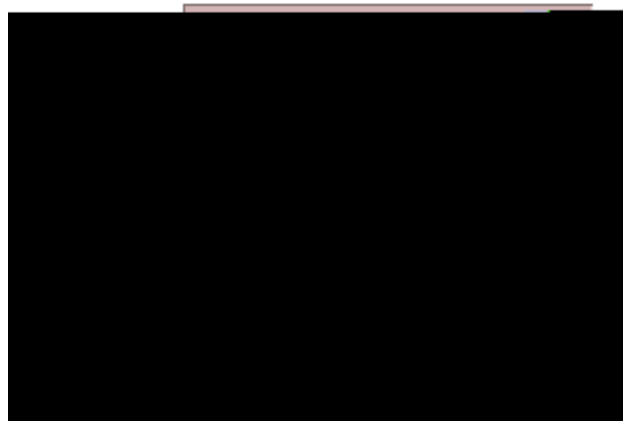
•

•

•

•





In the section on the LABEL element, we discuss marking up labels such as "First name".

•

```
<BUTTON>  
<IMG src="foo.gif" usemap="...">  
</BUTTON>
```

both cause controls to regain their initial values, but the

The

the specification in the section on intrinsic events.

INPUT element whose type is "reset". They
element allows richer presentation.

later in

OPTION elements

SELECT Attribute definitions

ktart tag:

38ION

required

Attribute definitions

name =

Attributes defined elsewhere



•

LABEL for To associate a label with another control implicitly, make the control the contents of the LABEL
 case, the LABEL
 after the associated control. INPUT elements. Notice that the im
 association prevents us from being able to layout the label and its associated control in a table (s
 previous example)ã

```
<FORM action="..." method="post">
<LABEL>
    First Name
</LABEL>
<LABEL>
    <INPUT type="text" name="lastname">
    Last Name
</LABEL>
</FORM>
```

The **LEGEND** elements

```
<!ELEMENT FIELDSET ON-ONLY whitespace space-is-allowed >The %i18n, %events --
<!-- ATTLIST LEGEND -- fieldset legend --
%attrs; -- %coreattrs, %i18n, %events --
align %LAlign; #IMPLIED -- relative to fieldset --
>Start tag:
```

- `bottom`: The legend is below the fieldset.

- `left`: The legend is to the left of the fieldset.
- `right`: The legend is to the right of the fieldset.

Attributes defined elsewhere

- `id`, `class` (document-wide identifiers)
- `lang` (language information), `dir` (text direction)
- `title` (element titles)
- `style` (inline style information)
- `accesskey`
- `align`
- `onclick`

```

...more medical history...
</FIELDkET>
<FIELDkET>
<LEGEND align="top">Current Medication</LEGEND>

<INPUT name="medication_now"

```

If you are currently taking medication, please indicate it in the space below:

```
<TEXTAREA name="current_medication"
```

```
</TEXTAREA>
```

Note that in this example, we might improve the presentation of the form by aligning elements within each (with style sheets), adding color and font information (with style sheets), adding scripting (say, to only open the "current medication" text area if the user indicates he or she is currently on

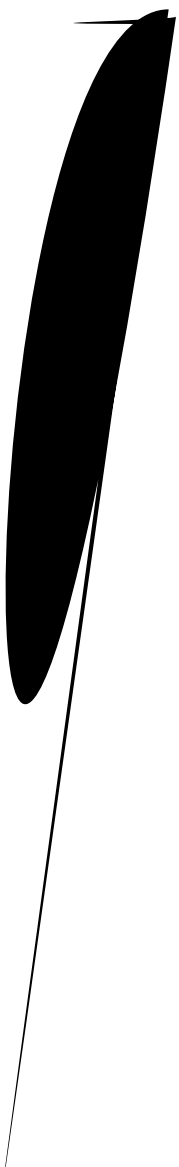
Giving focus to an element

Active elements in HTML documents must receive *focus*

Similarly, users must give a

- Designate the element with a pointing device.
- Navigate from one element to the next with the keyboard. The document's author may define a that specifies the order in which elements will receive focus if the user navigates the
- Select an element by a series of keyboard actions known as an (sometimes called "keyboard shortcut" or "keyboard accelerator").

Attribute definitions



= url This attribute specifies the location of an external script.

The `SCRIPT` element places a script within a document. This element may appear any number of times in the `HEAD` or `BODY` of an HTML document. The script may be defined with the `src` attribute or as an external file. If the `src` attribute is not set, user agents must interpret the contents of the element as the script. If the

Specifying the scripting language

Documents that contain neither a default scripting language declaration nor a local one for a `SCRIPT` element are incorrect. User agents may still try to interpret the script but are not required to.

declaration in the `HEAD`

META

Here's a more interesting window handler:

```
<SCRIPT type="text/tcl">
```


The following sections discuss issues about lack of support for scripting that authors should also consider

The **NOSCRIPT** Element

<!ELEMENT **NOSCRIPT** - - (%block)>

required

required

%HTMLspecial;

<!ENTITY % font

```
<!ATTLIST (INS|DEL)
  %attrs
  cite      %URL      #IMPLIED -- info on reason for change --
  -- %coreattrs, %i18n, %events --
```

```
color      CDATA      #IMPLIED -- #RRGGBB in hex, e.g. red: "#FF0000" -- Tj T* ( face CD
```

<!ENTITY % blocklevel

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


```
<!ELEMENT LINK - O EMPTY>
<!ATTLIST LINK
```

```
<!ENTITY % Length "CDATA" -- nn for pixels or nn% for percentage length -->
```

```
<!ENTITY % IAlign "(top|middle|bottom|left|right)" -- center? -->
```

```

<!--LIST APPLET
  codebase      %URL      #IMPLIED  -- optional base URL for applet --
  archive       CDATA     #IMPLIED  -- comma separated archive list --
  code          CDATA     #IMPLIED  -- applet class file --
  object        CDATA     #IMPLIED  -- serialized applet file --
  alt           CDATA     #IMPLIED  -- description for text only browsers --
  name          CDATA     #IMPLIED  -- allows applets to find each other --
  width         %Pixels   #REQUIRED -- suggested width in pixels --
  height        %Pixels   #REQUIRED -- suggested height in pixels --
  align         %Align    #IMPLIED  -- vertical or horizontal alignment --
  hspace        %Pixels   #IMPLIED  -- suggested horizontal gutter --
  vspace        %Pixels   #IMPLIED  -- suggested vertical gutter --
>

```

```

<!--ELEMENT HR - 0 EMPTY>

```

```
<!ELEMENT P - 0 (%inline)*>
<!ATTLIST P
```

```
    %align;
```

```
    -- align, text alignment --
```


<!ELEMENT THEAD - 0 (TR+)>
<!ELEMENT TFOOT - 0 (TR+)>

<!ELEMENT COLGROUP - 0 (col*)>

<!--

COL elements define the alignment properties for cells in a given column or spanned columns. The WIDTH attribute specifies the width of the columns, e.g.

width=64 width in screen pixels
width=0.5* relative width of 0.5

-->

```
<!ATLLIST COL                   -- column groups and properties --  
%attrs;                        -- %coreattrs, %i18n, %events --  
span           NUMBER        1        -- number of column  
width          CDATA        #IMPLIED -- column width specification --
```

<!ELEMENT FRAME - 0 EMPTY>

<!ATTLIS1 IFRAME

<!--

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

<!ATTLIS1 HEAD

<!--===== Document kstructure =====-->

<!ELEMENT HTML O O (%html.content)>

•

•

•


```

'is implied by' as ISOtech suggests -->
<!ENTITY uArr      CDATA "&#8657;" -- upwards double arrow, u+21D1 ISOamsa -->
<!ENTITY rArr      CDATA "&#8658;" -- rightwards double arrow, u+21D2 ISOtech -->
<!-- Unicode does not say this is the 'implies' character but does not have another
character with this function so ? rArr can be used for 'implies' as ISOtech suggests -->
<!ENTITY dArr      CDATA "&#8659;" -- downwards double arrow, u+21D3 ISOamsa -->
<!ENTITY hArr      CDATA "&#8660;" -- left right double arrow, u+21D4 ISOamsa -->

<!-- Mathematical Operators -->
<!ENTITY forall    CDATA "&#8704;" -- for all, u+2200 ISOtech -->
<!ENTITY part      CDATA "&#8706;" -- partial differential, u+2202 ISOtech -->
<!ENTITY exist     CDATA "&#8707;" -- there exists, u+2203 ISOtech -->
<!ENTITY empty     CDATA "&#8709;" -- empty set, =null set, =diameter, u+2205 ISOamso -->
<!ENTITY nabla     CDATA "&#8711;" -- nabla, =backward difference, u+2207 ISOtech -->
<!ENTITY isin      CDATA "&#8712;" -- element of, u+2208 ISOtech -->
<!ENTITY notin     CDATA "&#8713;" -- not an element of, u+2209 ISOtech -->
<!ENTITY ni        CDATA "&#8715;" -- contains as member, u+220B ISOtech -->
<!-- should there be a more memorable name than 'ni'? -->
<!ENTITY prod      CDATA "&#8719;" -- n-ary product, =product sign, u+220F ISOamsb -->
<!-- prod is NOT the same character as u+03A3 ISO'greek capunal letter pu' though the same
glyph might be used for both -->
<!ENTITY sum       CDATA "&#8721;" -- n-ary sumation, u+2211 ISOamsb -->
<!-- sum is NOT the same character as u+03A3 'greek caputal letter sigma' though the same
glyph might be used for both -->
<!ENTITY minus     CDATA "&#8722;" -- minus sign, u+2212 ISOtech -->
<!ENTITY lowast     CDATA "&#8727;" -- asterisk operator, u+2217 ISOtech -->
<!ENTITY radic     CDATA " 30;" -- square root, =radical sign, u+221A ISOtech -->
<!ENTITY prop      CDATA "033;" -- proportional to, u+221D ISOtech -->
<!ENTITY infin     CDATA "&#8734;" -- infinity, u+221E -->

```

```

<!ENTITY loz      CDATA "&#9674;" -- lozenge, u+25CA ISOpub -->

<!-- Miscellaneous kymbols -->
<!ENTITY spades   CDATA "&#9824;" -- black spade suit, u+2660 ISOpub -->
<!-- black here seems to mean filled as opposed to hollow -->
<!ENTITY clubs    CDATA "&#9827;" -- black club suit, =shamrock, u+2663 ISOpub -->
<!ENTITY hearts   CDATA "&#9829;" -- black heart suit, =valentine, u+2665 ISOpub -->
<!ENTITY diams    CDATA "&#9830;" -- black diamond suit, u+2666 ISOpub -->

```

Named entities for markup-significant and internationalization characters

The named character entities in this section are for escaping markup-significant characters (these are the same as those in HTML 2.0 and 3.2), for denoting spaces and dashes. Other characters in this section bidirectional text for C

Entities have also been added for the remaining characters occurring in CP-1252 which do not occur in the HTMLlat1 or HTMLsymbol entity sets. These all occur in the 128 to 159 range within the cp-1252 charset. These entities p

To support these entities, user agents may ses sport full [ISO10646] or use other means. Display of glyphs for these characters may be obtained by being able to display the relevant [ISO10646] characters or by other means, such as

The list of characters

```

<!-- Special characters for HTML -->

<!-- Character entity set. Typical invocation:
<!ENTITY % HTMLspecial PmBLIC      "-//W3C//ENTITIES kpecial//EN//HTML">
%HTMLspecial; -->

<!-- Portions s s51 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.-->

<!-- Relevant ISO entity set is given unless names are newly introduced.
New names (i.e., not in ISO 8879 list) do not clash with any
existing ISO 8879 entity names. ISO 10646 character numbers      are given for each character, in hex. CDTTÄ values
-->

<!-- C0 Controls and Basic Latin -->
<!ENTITY quot     CDATA "&#34;"      -- quotation mark, =apl quote, u+0022 ISOnum -->
<!ENTITY amp      CDATA "&#38;"      -- ampersand, u+0026 ISOnum -->
<!ENTITY lt       CDATA "&#60;"      -- less-than sign, u+003C ISOnum -->
<!ENTITY gt       CDATA "&#62;"      -- greater-than sign, u+003E ISOnum -->

<!-- Latin Extended-A -->
<!ENTITY OElig    CDATA "&#338;"    -- latin capital ligature oe, u+0152 ISOlat2 -->

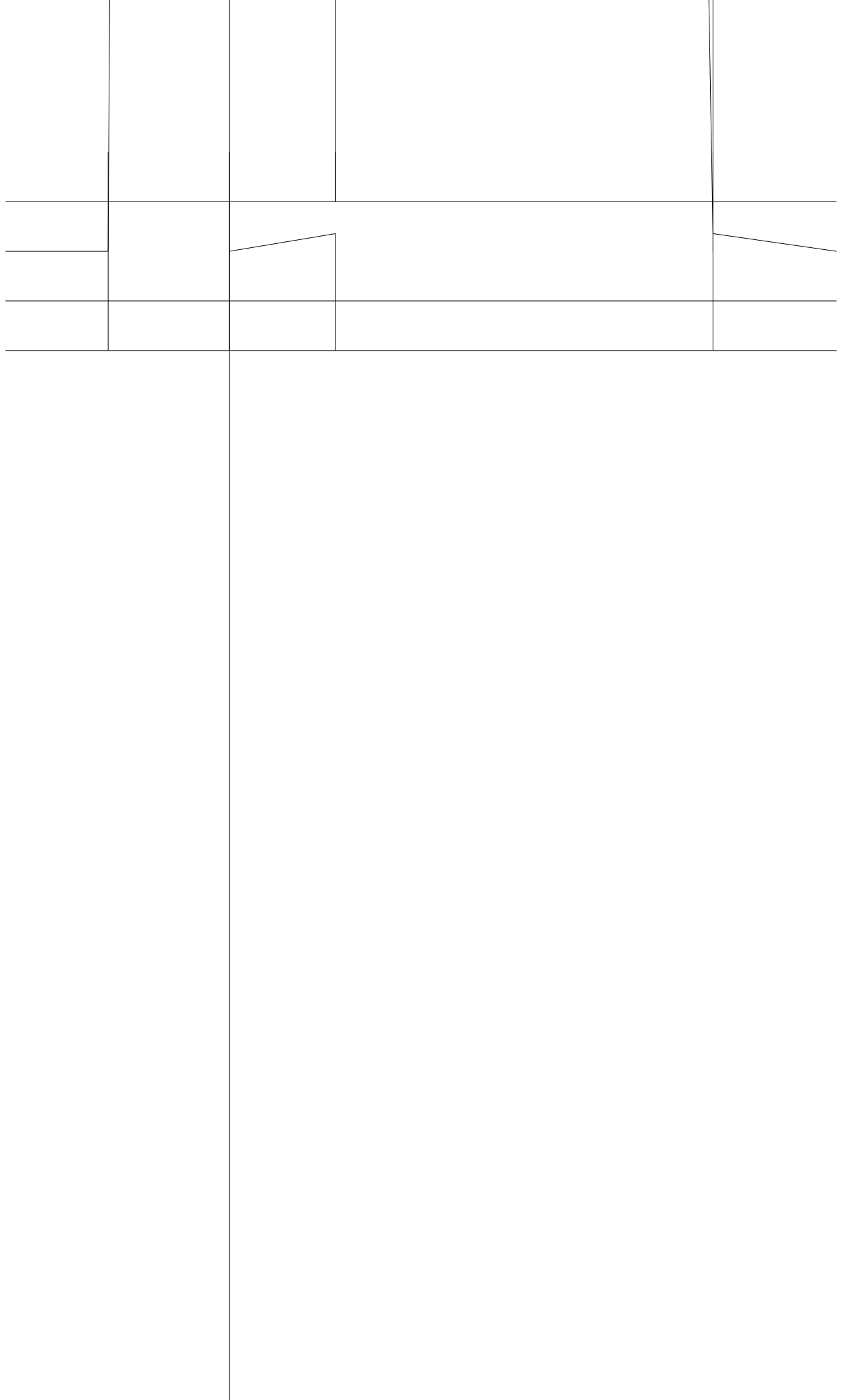
```

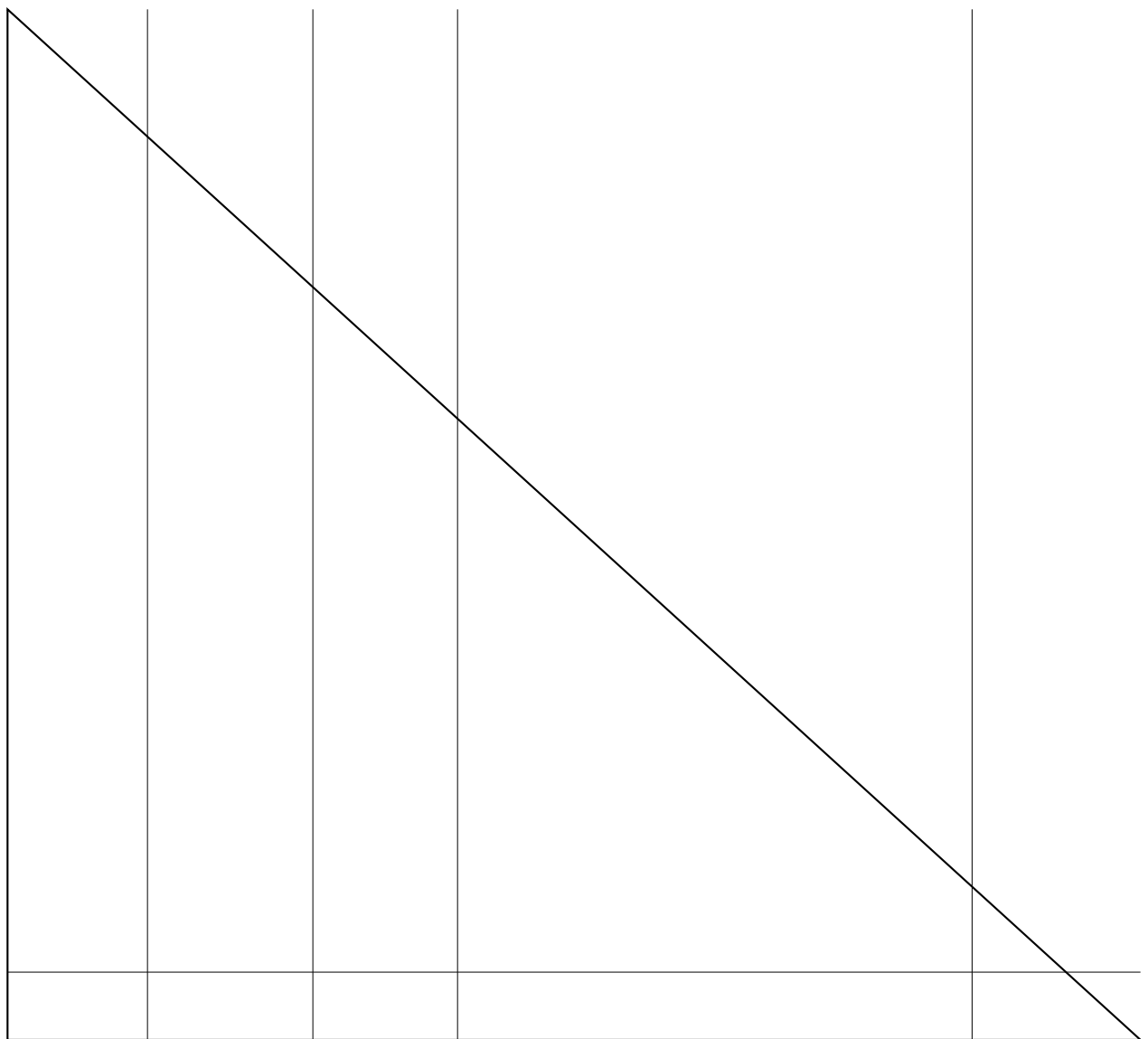
```
<!ENTITY oelig CDATA "&#339;" -- latin small ligature oe, u+0153 ISOlat2 -->
<!-- ligature is a misnomer, this is a separate character in some languages --><!ENTITY Scaron DAÄÄ "&#352;" -- latin

<!-- Spacing Modifier Letters -->
<!ENTITY circ DDÄÄ "&#710;" -- modifier letter circumflex accent, u+02C6 ISOpub --><!ENTITY tilde CDÄÄ "&#732;"
<!ENTITY ensp DAÄTA "&#8194;" -- enT6pace, u+2002 ISOpub --><!ENTITY emsp DDATA "&#8195;" -- emT6pace, u+2003 IS
```


•
•

|





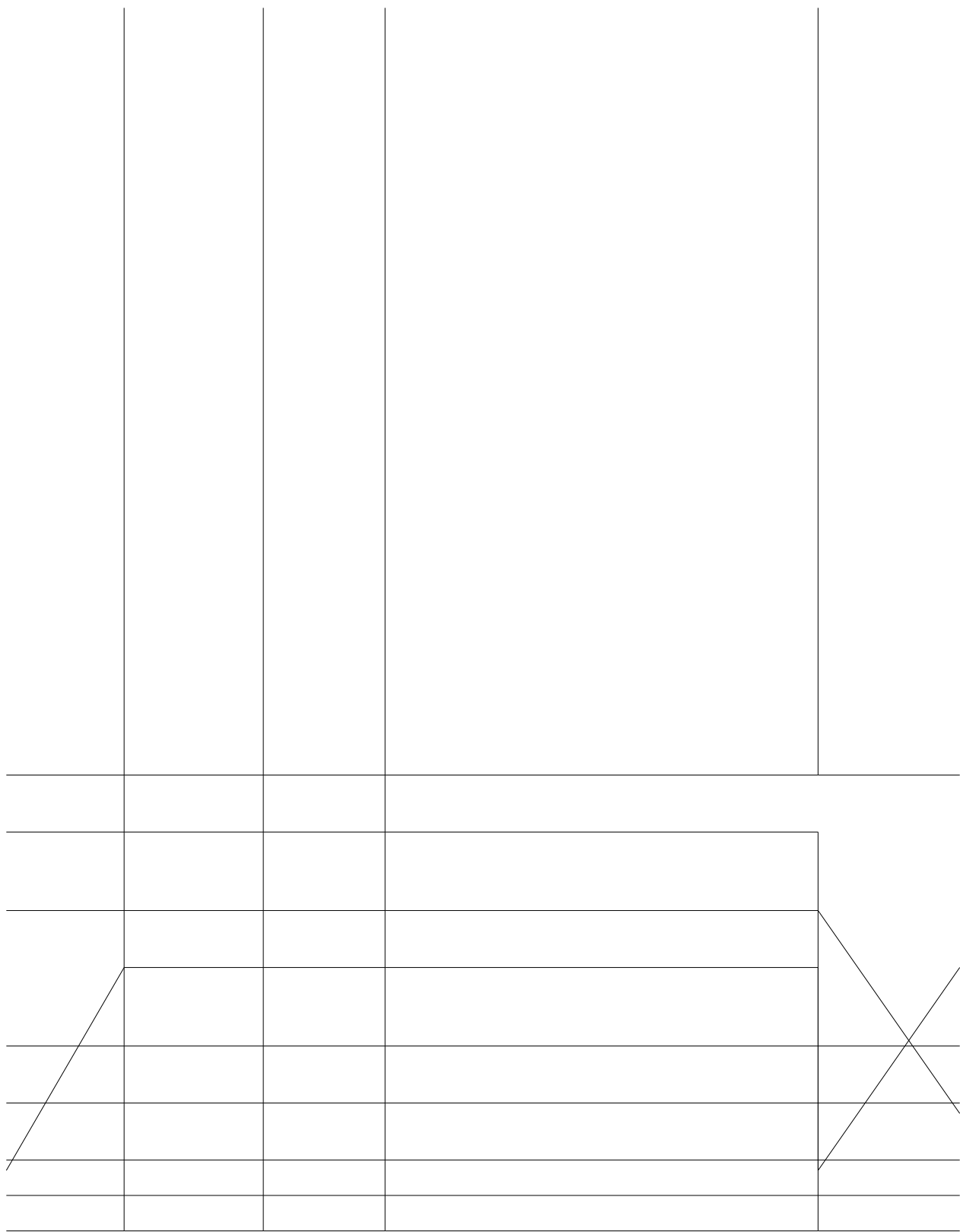
[illegible]

--	--	--	--	--

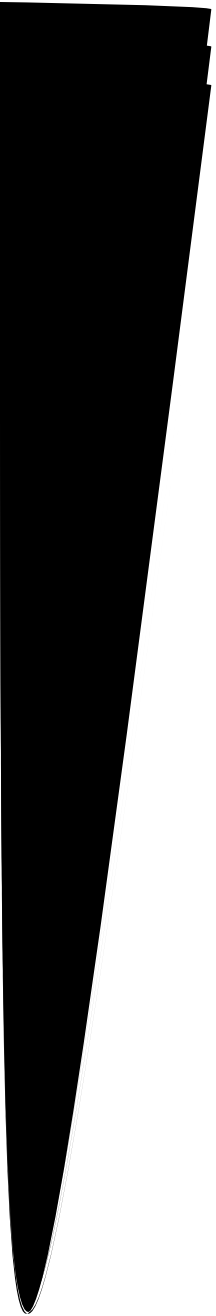
--	--	--

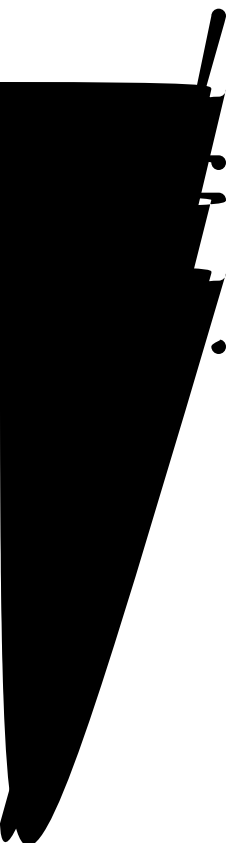
--	--	--	--	--

--	--	--



[illegible]

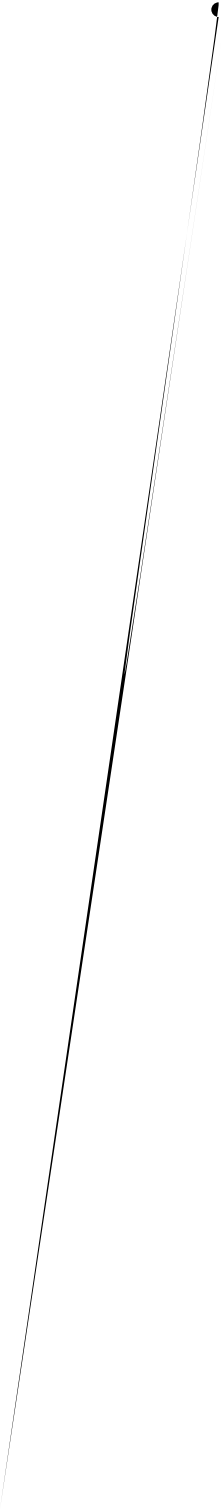




•

•

•



The next example uses JavaScript to set the coordinates for a client-side image map:

This example sets the size of an image based upon document properties:

```
<IMG src="bar.gif" width='&{document.banner.width/2};' height='50%'/>
```

You can set the URL for a link or image by script:

```
<SCRIPT>  
  function manufacturer(widget) {
```

This last example shows how SGML CDATA attributes can be quoted using single or double quote

