

4. Introduction to HTML: Part 1

HTML is a *markup language* for identifying the elements of a page so a browser can render that page on a computer screen.

Document *presentation* is generally separated from *structure*.

Text is marked up into *elements*, delineated by tags (keywords in `<...>`'s).

The keywords are case insensitive.

E.g.,

```
<body>  
...  
</body>
```

or

```
<BODY>  
...  
</BODY>
```

Create and edit a page using your favorite editor.

(Or use FrontPage.)

An HTML file has as its extension either `.html` or (for old Windows/DOS) `.htm`.

Your *home page* is the first in your site a user sees.

The default file sought by a browser is named `index.html`.

Common Tags

Comments: `<!-- ... -->`

An HTML file is separated into

- a header element (not generally rendered):

```
<head>
...
</head>
```

- a body (with the page contents):

```
<body>
...
</body>
```

The `title` element is in the header:

```
<title> ... </title>
```

It

- appears on the bar at the top of the browser window,
- appears as the text identifying the page if it's added to a favorites list, and
- is used by search engines for cataloguing.

Various elements let you place text in the body

The paragraph element

```
<p> ... </p>
```

sets its content apart from the surrounding material by several vertical spaces above and below.

Tags that do not enclose text don't have matching closing tags – e.g., `
` (line break), but also `</br>` (without a matching opening tag).

Example:

```
<html>
<!-- A simple example -->

<head>
<title>The Hello World example</title>
</head>

<body>

<p>Hello world!</p>

</body>
</html>
```

In the bar at the top of the browser:

The Hello World example – Microsoft Internet Explorer

In the window:

Hello world!

Headers vary text size based on their levels, h1 (largest) to h6 (smallest).

A `center` element centers text horizontally.

Example

```
<center>  
<h1>Level 1 Header</h1>  
<h3>Level 3 Header</h3>  
<h6>Level 6 Header</h6>  
</center>
```

Rendering:

Level 1 Header

Level 3 Header

Level 6 Header

Text Styling

Some *styles*:

Underline: `<u> ... </u>`

Emphasis (italic): ` ... `

Strong (bold): ` ... `

Styles that've been deprecated (they conflate content and presentation):

Bold: ` ... ` – use `strong`

Italic: `<i> ... </i>` – use `em`

Elements can be nested.

The closing tags should appear in the reverse of the order in which the corresponding starting tags appear.

(This doesn't always matter to a browser.)

Example

```
<p>
XXX<em>XXX<u>XXX</u>XXX</em>XXX
</p>
```

Rendering:



The `align` attribute has possible values `left`, `center`, `right`.

Syntax:

`<tag attribute = "value" > ... </tag>`

Example:

```
<h2 align = "center"><u>A Heading</u></h2>
```

Rendering:



Linking

HTML's most important function is to create hyperlinks to documents possibly on other servers.

Both text and images can act as anchors to link to other pages.

Links are marked using the `a` (*anchor*) element; the address is the value of the `href` attribute:

```
<a href = "address" > ... </a>
```

The address specifies the protocol (`http`, `ftp`, or `mailto`) and the URL.

The URL can be an address relative to the current directory.

Files with extension `.html` default to `http`.

Clicking the anchor element causes the browser to engage in the specified protocol.

Generally, anchors are rendered as underlined by default.

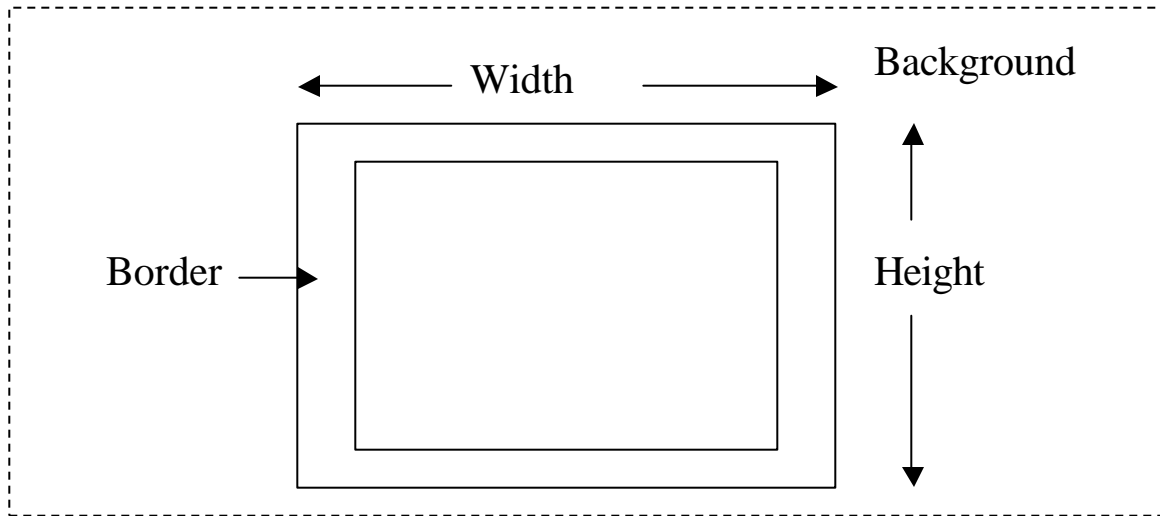
Example:

```
<p>XXX<a href = "http://www.eng.ncat.edu">CCC</a></p>  
<p>YYY<a href = "mailto:esterlin@ncat.edu">DDD</a></p>
```

XXXCCC

YYYDDD

Images



The background is specified by an attribute added to the BODY tag.

`background = "filename"` – the browser tiles the image across and down.

`bgcolor = "color"`, where *color* is either

- a predefined color name (see Appendix E) or
- a hexadecimal coding of an RGB color giving the strength of each component (ranging 00-FF)

`#RRGGBB`

Black is `#000000` and white is `#FFFFFF`.

To include an image, use the `` tag (with no closing tag).

Attributes:

`border, height, width`: in pixels

`src`: the file containing the image

`alt`: text to display if a browser isn't viewing images.

To speed loading, always include the `height` and `width` attributes.

Example:

```
<center>
<img src = "pic1.gif" border = "2" height = "150"
      width = "200" alt = "Another picture">
</center>
```

Images can be used as anchors – e.g.,

```
<a href = "main.html">
<img src = "main.jpg" width = "50" height = "50"
      border = "0" alt = "Main Page">
</a><br>
```

Formatting Text with

Attributes of the font element:

color

size: the size of the text

face: the font of the text (e.g., Times, Arial, Courier, Helvetica)

For size, "+x" (resp., "-x") makes the text *x* font point sizes larger (resp., smaller) than the surrounding text.

Example

```
<p><font size = "+2" face = "Courier">  
XXX<strong>XXX</strong>XXX</font>  
<font size = "-2" face = "Arial">  
CCC<u>CCC</u>CCC</font>  
</p>
```

XXXXXXXXXX cccccc

Special Characters

Special characters are inserted with code of the form `&code;`, where `code` is either

- an abbreviation for the character (e.g., `amp` for ampersand, `copy` for copyright) or
- a hex code (e.g., `#38` for ampersand).

Example

```
<p>&copy; Wiley &#38; Sons</p>
```

© Wiley & Sons

A `del` element strikes through text with a horizontal line (as in text revision).

Example

```
<p><del>XXX</del></p>
```

~~XXX~~

Subscripting and superscripting are achieved with the `sub` and `sup` elements.

Example

```
<p>There are 10<sup>14</sup> germs  
in a pond of H<sub>2</sub>O.</p>
```

There are 10^{14} germs in a pond of H_2O .

Horizontal Rules

The `hr` element (no closing tag) is a horizontal line occupying its own line.

Attributes:

`size`: height in pixels

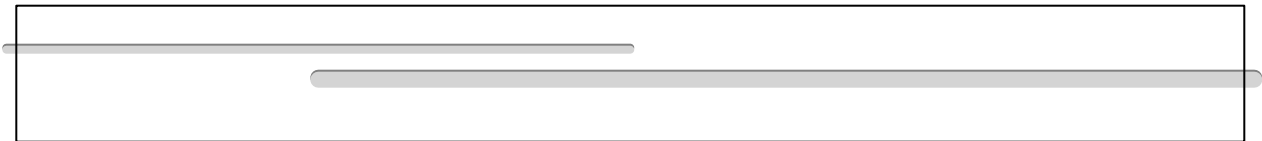
`width`: the extent across the screen, either in pixels or as a percentage of the screen width

`align`

`noshade` (a valueless attribute): turns off the default shading effect; displays a solid-color bar.

Example

```
<hr align = "left" width = "50%" size = "4">  
<hr align = "right" width = "75%" size = "8">
```



Unordered Lists

The unordered list element, `ul`, creates a list of bulleted elements, each enclosed in

```
<li> ... </li>
```

(list item) tags.

Example

```
<h2>Some Common Trees</h2>
<ul>
  <li>Elm</li>
  <li>Oak</li>
  <li>Ash</li>
</ul>
```

Some Common Trees

- Elm
- Oak
- Ash

A nested list is indented one level and has a different bullet type.

Example

```
<h3>Shopping List</h3>
<ul>
  <li>Grocery Store</li>
    <ul>
      <li>Bread</li>
      <li>Milk</li>
    </ul>
  <li>Office Supply Store</li>
    <ul>
      <li>Pencils</li>
      <li>Paper</li>
    </ul>
</ul>
```

Shopping List

- Grocery Store
 - Bread
 - Milk
- Office Supply Store
 - Pencils
 - Paper

Ordered Lists

The ordered list element, `ol`, creates a list of sequentially numbered elements, each in

```
<li> ... </li>
```

tags.

The default numbering is 1, 2,

The `type` attribute in the `` tag changes the sequence type:

```
type = "1": 1, 2, ...
```

```
type = "I": I, II, ...
```

```
type = "i": i, ii, ...
```

```
type = "A": A, B, ...
```

```
type = "a": a, b, ...
```

Example

```
<h3>Some Common Trees</h3>
<ol type = "A">
  <li>Elm</li>
  <li>Oak</li>
  <li>Ash</li>
</ol>
```

Some Common Trees

- A. Elm
- B. Oak
- C. Ash