

HyperText Markup Language

Contents

- [1 Introduction](#)
 - [1.1 Before we start](#)
 - [1.2 A simple document](#)
- [2 Head and Body](#)
 - [2.1 The HEAD element](#)
 - [2.2 The TITLE element](#)
 - [2.3 The BODY element](#)
- [3 Paragraphs and Headings](#)
 - [3.1 Paragraphs](#)
 - [3.2 Headings](#)
 - [3.3 Example](#)
- [4 Sundial](#)
 - [4.1 History](#)
 - [4.1.1 Installation of standard sundials](#)
- [5 Text Formatting](#)
 - [5.1 Emphasis](#)
 - [5.2 Preformatted text](#)
 - [5.3 Special Characters](#)
 - [5.4 Abbreviations](#)
 - [5.5 Discouraged Formatting](#)
 - [5.6 Cascading Style Sheets](#)
- [6 Hyperlinks](#)
 - [6.1 Absolute vs. Relative](#)
 - [6.2 Linking to a location within a page with Anchor](#)
 - [6.3 Target Links](#)
 - [6.4 Special targets](#)
 - [6.5 Hyperlinks on images](#)
- [7 Images](#)
 - [7.1 Placement](#)
 - [7.2 Alternative text and tooltip](#)
 - [7.3 Width and height](#)
 - [7.4 All attributes](#)
- [8 Lists](#)
 - [8.1 Ordered Lists](#)
 - [8.2 Unordered Lists](#)
 - [8.3 Definition Lists](#)
 - [8.4 Nested Lists](#)

HTML

- [8.5 Note on format](#)
- [8.6 Example](#)
- [9 Tables](#)
 - [9.1 Minimal tables](#)
 - [9.2 Caption and headings](#)
 - [9.3 Borders](#)
 - [9.4 Height and Width](#)
 - [9.5 Cell Spacing and Cell Padding](#)
 - [9.6 Alignment of table cells](#)
 - [9.7 Row span and column span](#)
 - [9.8 Background colour and images](#)
 - [9.9 Column groups](#)
 - [9.10 Summary](#)
- [10 Quotations](#)
 - [10.1 Inline quotations](#)
 - [10.2 Block quotations](#)
- [11 Comments](#)
- [12 Forms](#)
 - [12.1 Formatting with CSS](#)
 - [12.1.1 The HTML](#)
 - [12.1.2 The CSS](#)
 - [12.2 References](#)
- [13 CSS](#)
 - [13.1 What does CSS do?](#)
 - [13.2 How to add a CSS style sheet](#)
- [14 Validating HTML](#)
- [15 Conditional Comments](#)
 - [15.1 Syntax](#)
 - [15.2 Use with CSS](#)
 - [15.3 External links](#)
- [16 Proscribed Techniques](#)
- [17 Frames](#)
 - [17.1 Framing](#)
 - [17.2 Inline frames](#)
- [18 Layers](#)
- [19 Music](#)
- [20 Browsers' extensions](#)
- [21 Conditional Comments](#)
 - [21.1 Syntax](#)
 - [21.2 Use with CSS](#)
 - [21.3 External links](#)
- [22 Appendices](#)

HTML

- [23 Tag List](#)
- [24 Standard Attributes List](#)
 - [24.1 Attributes](#)
 - [24.1.1 class](#)
 - [24.1.2 code](#)
 - [24.1.3 codebase](#)
 - [24.1.4 dir](#)
 - [24.1.5 height](#)
 - [24.1.6 id](#)
 - [24.1.7 lang](#)
 - [24.1.8 style](#)
 - [24.1.9 title](#)
 - [24.1.10 width](#)
 - [24.2 More attributes](#)
 - [24.2.1 accesskey](#)
 - [24.2.2 tabindex](#)
- [25 Glossary](#)
 - [25.1 B](#)
 - [25.2 E](#)
 - [25.3 I](#)
 - [25.4 T](#)

Introduction

HTML is a markup language used in most of the pages of the World Wide Web. HTML files are text files that, unlike completely plain text, contain additional formatting markup—sequences of characters telling web browsers what parts of text should be bold, where the headings are, or where tables, table rows and table cells start and end. HTML may be displayed by a visual web browser, a browser that reads the text of the page to the user, a Braille reader that converts pages to a braille format, email client, or a wireless device like a cellular phone.

Before we start

To author and test HTML pages, you will need an editor and a web browser. HTML can be edited in plain text editors, including those that highlight HTML markup with colors to make it easier to read. There are also WYSIWYG (what you see is what you get) editors of HTML, and complex WYSIWYG editors with website project management and development environments

(It is also a good idea to learn XHTML, Javascript, ASP.NET and/or PHP to make high quality websites, though.)

Plain text editors include Notepad (or Notepad++) for Microsoft® Windows, TextEdit for Mac, or Vim, Emacs and others for Linux.

Commercial HTML editors include Adobe Contribute CS5 and Dreamweaver CS5 (both Win/Mac), and Microsoft's Visual Web Developer (Win). There are also free HTML editors out there including Evrsoft First Page (Win), Mozilla KompoZer (Win/Mac/Lin) and Quanta Plus (Lin). It is usually better to gain a basic knowledge of HTML using a code-based HTML editor before delving into the WYSIWYG editors (all of the previous in this paragraph).

To preview your documents, you'll need a web browser. To make your documents look good to the greatest number of readers, test the documents in several browsers. Each browser has slightly different rendering, and most have their quirks, resulting in certain sequences of correctly written HTML rendered incorrectly.

Microsoft Internet Explorer is the most widely used browser, as of November 2010 having a 46% market share. Other common browsers include Google Chrome, Mozilla Firefox, Safari, and Opera. To make sure that your documents are readable in a text only environment, you can use Lynx.

A simple document

Let's start with a simple document. Write this code in your editor (or copy-and-paste it) and save it as "index.html" or "index.htm". The file must be saved with the exact extension, or will not be rendered correctly.

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

HTML

```
</head>
<body>
<p>The text of the document goes here.</p>
</body>
</html>
```

Now open the document in your browser and see the result. From the above example, we can deduce certain essentials of a HTML document:

- The first line of a valid HTML document must state which version of HTML the document uses. This example uses the strict variant of HTML version 4.01.
- The HTML document begins with a `<html>` tag and ends with its counterpart, the `</html>` tag.
- Within the `<html></html>` tags, there are two main pairs of tags, `<head></head>` and `<body></body>`.
- Within the `<head></head>` tags, there are the `<title></title>` tags which enclose the textual title to be shown in the title bar of the web browser.
- Within the `<body></body>` is a paragraph marked by a `<p></p>` tag pair.
- Most tags must be written in pairs between which the effects of the tag will be applied.
 - `This text is emphasized` – *This text is emphasized*
 - This text includes `<code>computer code</code>` – This text includes computer code
 - `This text is emphasized and has <code>computer code</code>` – *This text is emphasized and has computer code*
- HTML tag pairs must be aligned to encapsulate other tag pairs, for example:
 - `<code>This text is both code and emphasized</code>` – *This text is both code and emphasized*
 - **A mistake:** – `<code>This mark up is erroneous</code>`

The HTML Tag:

The `<html>` and `</html>` tags are used to mark the beginning and end of an HTML document. This tag does not have any effect on appearance of document.

This tag is used to make browsers and other programs know that this is an HTML document.

Attributes of `<html>` tag: *dir attribute*- This tag specifies that in which manner the browser will present text within the entire document.

This tag can have values either ltr(left to right) or rtl(right to left).

By default this is set to ltr. Generally rtl is used for languages like Chinese, Hebrew, Urdu etc.

Example: `<html dir="ltr">`

lang attribute- lang attribute generally specifies which language is being used within the document.

HTML

Special types of codes are used to specify different languages:

en - english

fr - french

de - german

it - italian

nl - dutch

el - greek

es - spanish

pt - portugese

ar - arabic

he - hebrew

ru - russian

zh - chinese

ja - japanese

hi - hindi

Example: `<html lang="en">`

Head and Body

An HTML file is divided into two basic sections: the *header* and the *body*. Each of these sections are demarcated by their respective tags. Thus, the essential structure of an HTML document looks like this :

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

The HEAD element

All data in the header section of an HTML document is considered "meta-data", meaning "data about data". The information in this section is not normally displayed directly. Instead elements such as style affect the appearance of other elements in the document.

The head element must contain a title element. This element is used to set the title of the HTML document, which is commonly displayed by the web browser in the title bar of the window. Here is an example of the use of the title element:

```
<head>
  <title>This is the Title</title>
</head>
```

There can only be one title in the header section.

There may be any number of the following elements inside the head element:

style

Used to embed style rules in a document. In web sites where consistent look across multiple web pages is required, style should be specified in a separate stylesheet file, linked using the link element.

link

Used to link the page to various external files, including a style sheet, or the location of the RSS feed for the page. The type of link is set using the rel attribute. The type attribute specifies the [MIME type](#) of the document found at the location given by the href attribute. This allows the browser to ignore links to MIME types it does not support. Examples:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

```
<link rel="alternate" type="application/rss+xml" href="rss.aspx" title="RSS 2.0">
```

HTML

script

Used to link to an external Javascript file or to embed Javascript in the page. Linking to an external file is the preferred technique in real web pages though many examples embedded the script for simplicity.

meta

Used to set additional meta-data properties for the HTML document, such as related keywords, etc. Examples:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
<meta name="keywords" content="web, HTML, markup, hypertext">
```

object

Embeds a generic object. This element is not commonly used in the header section. It is normally used in the body section.

There may also be a single base element. This element sets the *base URI* for resolving relative URIs. It is rarely necessary to use this element.

The TITLE element

The title element contains your document title and identifies its contents in a global context.

The title is typically displayed in the title bar at the top of the browser's window.

It is also displayed on the bookmark list of the browser.

Title is also used to identify your page for search engines.

This tag is always placed in the head section i.e. between `<head>` and `</head>`.

Example:

```
<html> <head> <title>This is Title</title> </head> </html>
```

The BODY element

Unlike the head element, any plain text placed between the `<body>` tags will be displayed on the web page by the browser.

What to avoid. The following example is better avoided:

```
<body text='black' link='red' alink='pink' vlink='blue'  
      bgcolor='#DDDDDD' background='wallpaper.gif">
```

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

The `text`, `link`, `alink`, `vlink`, `bgcolor` and `background` attributes have all been *deprecated* in HTML 4. This means that they should not be used in new documents. They have been superseded by the CSS rules given below. The values from the previous example have been used as examples in these rules.

HTML

text
body { color:*black* }

bgcolor
body { background-color:#*DDDDDD* }

background
body { background-image: url(*wallpaper.gif*) }

link
a:link { color:*red* }

alink
a:active { color:*pink* } (an active link is a link that is being clicked or has the keyboard focus)

vlink
a:visited { color:*blue* }

hover (not an html attribute)
a:hover { color:*green* } ('hover' is the style of a link when the mouse pointer is over it)

Paragraphs and Headings

The bulk of a web page consists of paragraphs, structured with the use of headings. The two kinds of elements are described in the following.

Paragraphs

The `p` element is used to split text into paragraphs.

```
<p>An introductory paragraph.</p>
<p>Another introductory paragraph.</p>
```

Headings

There are six levels of headings. The most important heading(s) in a document should be level one. Sub-headings should be level two. Sub-sub-headings should be level three, etc. Do not skip levels. If the default sizes do not suit your document, use CSS to change them. Headings should be used to effectively outline your content. By doing so, users can find information more quickly (i.e. wikipedia), and some search engines use headings to help rank page content.

```
<h1>This is Level 1</h1>
```

This is Level 1

```
<h3>This is Level 3</h3>
```

This is Level 3

```
<h5>This is Level 5</h5>
```

This is Level 5

Example

This example will be used in the next section where we see how to change the appearance of a document.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
  <head>
    <title>Sundial</title>
  </head>
  <body>
    <h1>Sundial</h1>
    <p>From Wikipedia, the free encyclopedia.</p>
    <p>A sundial measures time by the position of the sun. The most commonly seen designs, such as the
```

HTML

'ordinary' or standard garden sundial, cast a shadow on a flat surface marked with the hours of the day. As the position of the sun changes, the time indicated by the shadow changes. However, sundials can be designed for any surface where a fixed object casts a predictable shadow.

</p>

<p>Most sundial designs indicate apparent solar time. Minor design variations can measure standard and daylight saving time, as well.

</p>

<h2>History</h2>

<p>Sundials in the form of obelisks (3500 BC) and shadow clocks (1500 BC) are known from ancient Egypt, and were developed further by other cultures, including the Chinese, Greek, and Roman cultures. A type of sundial without gnomon is described in the old Old Testament (Isaiah 38:2).

</p>

<p>The mathematician and astronomer Theodosius of Bithynia (ca. 160 BC-ca. 100 BC) is said to have invented a universal sundial that could be used anywhere on Earth. The French astronomer Oronce Finé constructed a sundial of ivory in 1524. The Italian astronomer Giovanni Padovani published a treatise on the sundial in 1570, in which he included instructions for the manufacture and laying out of mural (vertical) and horizontal sundials. Giuseppe Biancani's Constructio instrumenti ad horologia solaria discusses how to make a perfect sundial, with accompanying illustrations.

</p>

<h3>Installation of standard sundials</h3>

<p>Many ornamental sundials are designed to be used at 45 degrees north. By tilting such a sundial, it may be installed so that it will keep time. However, some mass-produced garden sundials are inaccurate because of poor design and cannot be corrected. A sundial designed for one latitude can be adjusted for use at another latitude by tilting its base so that its style or gnomon is parallel to the Earth's axis of rotation, so that it points at the north celestial pole in the northern hemisphere, or the south celestial pole in the southern hemisphere.

</p>

<p>A local standard time zone is nominally 15 degrees wide, but may be modified to follow geographic and political boundaries. A sundial can be rotated around its style or gnomon (which must remain pointed at the celestial pole) to adjust to the local time zone. In most cases, a rotation in the range of 7.5 degrees east to 23 degrees west suffices.

</p>

<p>To correct for daylight saving time, a face needs two sets of numerals or a correction table. An informal standard is to have numerals in hot colors for summer, and in cool colors for winter. Rotating the sundial will not work well because most sundials do not have equal hour angles.

</p>

<p>Ordinary sundials do not correct apparent solar time to clock time. There is a 15 minute variation through the year, known as the equation of time, because the Earth's orbit is slightly elliptical and its axis is tilted relative to the plane of its orbit. A quality sundial will include a permanently-mounted table or graph giving this correction for at least each month of the year. Some more-complex sundials have curved hour-lines, curved gnomons or other arrangements to directly display the clock time.

</p>

</body>

</html>

Sundial

From Wikipedia, the free encyclopedia.

A sundial measures time by the position of the sun. The most commonly seen designs, such as the 'ordinary' or standard garden sundial, cast a shadow on a flat surface marked with the hours of the day. As the position of the sun changes, the time indicated by the shadow changes. However, sundials can be designed for any surface where a fixed object casts a predictable shadow.

Most sundial designs indicate apparent solar time. Minor design variations can measure standard and daylight saving time, as well.

History

Sundials in the form of obelisks (3500 BC) and shadow clocks (1500 BC) are known from ancient Egypt, and were developed further by other cultures, including the Chinese, Greek, and Roman cultures. A type of sundial without gnomon is described in the old Old Testament (Isaiah 38:2).

The mathematician and astronomer Theodosius of Bithynia (ca. 160 BC-ca. 100 BC) is said to have invented a universal sundial that could be used anywhere on Earth. The French astronomer Oronce Finé constructed a sundial of ivory in 1524. The Italian astronomer Giovanni Padovani published a treatise on the sundial in 1570, in which he included instructions for the manufacture and laying out of mural (vertical) and horizontal sundials. Giuseppe Biancani's *Constructio instrumenti ad horologia solaria* discusses how to make a perfect sundial, with accompanying illustrations.

Installation of standard sundials

Many ornamental sundials are designed to be used at 45 degrees north. By tilting such a sundial, it may be installed so that it will keep time. However, some mass-produced garden sundials are inaccurate because of poor design and cannot be corrected. A sundial designed for one latitude can be adjusted for use at another latitude by tilting its base so that its style or gnomon is parallel to the Earth's axis of rotation, so that it points at the north celestial pole in the northern hemisphere, or the south celestial pole in the southern hemisphere.

A local standard time zone is nominally 15 degrees wide, but may be modified to follow geographic and political boundaries. A sundial can be rotated around its style or gnomon (which must remain pointed at the celestial pole) to adjust to the local time zone. In most cases, a rotation in the range of 7.5 degrees east to 23 degrees west suffices.

To correct for daylight saving time, a face needs two sets of numerals or a correction table. An informal standard is to have numerals in hot colors for summer, and in cool colors for winter. Rotating the sundial will not work well because most sundials do not have equal hour angles.

Ordinary sundials do not correct apparent solar time to clock time. There is a 15 minute variation through the year, known as the equation of time, because the Earth's orbit is slightly elliptical and its axis is tilted relative to the plane of its orbit. A quality sundial will include a permanently-mounted table or graph giving this correction for at least each month of the year. Some more-complex sundials have curved hour-lines, curved gnomons or other arrangements to directly display the clock time.

Text Formatting

The **Text Formatting** elements give logical structure to phrases in your **HTML** document. This structure is normally presented to the user by changing the appearance of the text.

We have seen in the [Introduction](#) to this book how we can *emphasize* text by using `` tags. Graphical browsers normally present emphasized text in italics. Some [Screen readers](#), utilities which read the page to the user, may speak emphasized words with a different inflection.

A common mistake is to confuse the appearance of the element in your preferred browser for the meaning of the element. This is one reason why you should use more than one browser to test your pages. Ideally you would use a graphical browser, a text-only browser and a screen reader.

You can change the default presentation for any element using [Cascading Style Sheets](#). For example if you wanted all emphasized text to appear in red normal text you would use the following CSS rule:

```
em { font-style:normal; color:red; }
```

In this section, we will explore a few basic ways in which you can markup the logical structure of your document.

Emphasis

HTML has elements for two degrees of emphasis:

- The `em` element for emphasized text, usually rendered in italics.
- The `strong` element for strongly emphasized text, usually rendered in bold.

An example of emphasized text:

It is essential not only to guess but actually `observe` the results.

An example rendering:

It is essential not only to guess but actually *observe* the results.

An example of strongly emphasized text:

Let us now focus on `structural markup`.

An example rendering:

Let us now focus on **structural markup**.

HTML

Preformatted text

Preformatted text is rendered using fixed-width font, and without condensing multiple spaces into one, which results in preserved spacing. Newlines are rendered as newlines, unlike outside preformatted text. HTML markup in the preformatted text is still interpreted by browsers though, meaning that "`a`" will still be rendered as "a".

To create preformatted text, start it with `<pre>` and end it with `</pre>`.

An example:

```
<pre>
```

```
| No. | Person |
|-----|-----|
| 1.  | Bill Newton |
| 2.  | Magaret Clapton |
```

```
</pre>
```

The resulting rendering:

```
| No. | Person |
|-----|-----|
| 1.  | Bill Newton |
| 2.  | Magaret Clapton |
```

Omitting the preformatting tags will cause the same text to appear all in one line:

```
,-----| No. | Person | |-----| | 1. | Bill Newton | | 2. | Magaret Clapton |
```

Special Characters

To insert non-standard characters or characters that hold special meaning in HTML, a HTML character reference is required. For example, to input the ampersand, "&", "&" needs to be typed. Characters can also be inserted by their Ascii or Unicode number code.

Name Code	Number Code	Glyph	Description
"	"	"	double quotation mark
&	&	&	ampersand
⁄	/	/	slash
<	<	<	less-than sign
>	>	>	greater-than sign
–	–	–	en dash
—	—	—	em dash
 	 		nonbreaking space (invisible)
¡	¡	¡	inverted exclamation
¢	¢	¢	cent sign
£	£	£	pound sterling
¤	¤	¤	general currency sign
¥	¥	¥	yen sign
¦ or &brkbar;	¦	‡	broken vertical bar
§	§	§	section sign
¨ or ¨	¨	¨	umlaut
©	©	©	copyright
ª	ª	ª	feminine ordinal
«	«	«	left angle quote
¬	¬	¬	not sign
­	­		soft hyphen
®	®	®	registered trademark
¯ or &hibar;	¯	ˉ	macron accent
°	°	°	degree sign
±	±	±	plus or minus
²	²	²	superscript two
³	³	³	superscript three
´	´	´	acute accent
µ	µ	µ	micro sign

HTML

¶	¶	¶	paragraph sign
·	·	·	middle dot
¸	¸	¸	cedilla
¹	¹	¹	superscript one
º	º	º	masculine ordinal
»	»	»	right angle quote
¼	¼	¼	one-fourth
½	½	½	one-half
¾	¾	¾	three-fourths
¿	¿	¿	inverted question mark
‘		‘	left single quote
’		’	right single quote
‚		‚	single low-9 quote
“		“	left double quote
”		”	right double quote
„		„	double low-9 quote
†		†	dagger
‡		‡	double dagger
‰		‰	per mill sign
‹		‹	single left-pointing angle quote
›		›	single right-pointing angle quote
♠		♠	black spade suit
♣		♣	black club suit
♥		♥	black heart suit
♦		♦	black diamond suit
‾		—	overline, = spacing overscore
←		←	leftward arrow
↑		↑	upward arrow
→		→	rightward arrow
↓		↓	downward arrow
™		™	trademark sign
×	×	×	multiplication sign
÷	÷	÷	division sign
À	À	À	uppercase A, grave accent
Á	Á	Á	uppercase A, acute accent
Â	Â	Â	uppercase A, circumflex accent
Ã	Ã	Ã	uppercase A, tilde

HTML

Ä	Ä	Ä	uppercase A, umlaut
Å	Å	Å	uppercase A, ring
Æ	Æ	Æ	uppercase AE
Ç	Ç	Ç	uppercase C, cedilla
È	È	È	uppercase E, grave accent
É	É	É	uppercase E, acute accent
Ê	Ê	Ê	uppercase E, circumflex accent
Ë	Ë	Ë	uppercase E, umlaut
Ì	Ì	Ì	uppercase I, grave accent
Í	Í	Í	uppercase I, acute accent
Î	Î	Î	uppercase I, circumflex accent
Ï	Ï	Ï	uppercase I, umlaut
Ð	Ð	Ð	uppercase Eth, Icelandic
Ñ	Ñ	Ñ	uppercase N, tilde
Ò	Ò	Ò	uppercase O, grave accent
Ó	Ó	Ó	uppercase O, acute accent
Ô	Ô	Ô	uppercase O, circumflex accent
Õ	Õ	Õ	uppercase O, tilde
Ö	Ö	Ö	uppercase O, umlaut
Ø	Ø	Ø	uppercase O, slash
Ù	Ù	Ù	uppercase U, grave accent
Ú	Ú	Ú	uppercase U, acute accent
Û	Û	Û	uppercase U, circumflex accent
Ü	Ü	Ü	uppercase U, umlaut
Ý	Ý	Ý	uppercase Y, acute accent
Þ	Þ	Þ	uppercase THORN, Icelandic
ß	ß	ß	lowercase sharps, German
à	à	à	lowercase a, grave accent
á	á	á	lowercase a, acute accent
â	â	â	lowercase a, circumflex accent
ã	ã	ã	lowercase a, tilde
ä	ä	ä	lowercase a, umlaut
å	å	å	lowercase a, ring
æ	æ	æ	lowercase ae
ç	ç	ç	lowercase c, cedilla
è	è	è	lowercase e, grave accent

HTML

é	é	é	lowercase e, acute accent
ê	ê	ê	lowercase e, circumflex accent
ë	ë	ë	lowercase e, umlaut
ì	ì	ì	lowercase i, grave accent
í	í	í	lowercase i, acute accent
î	î	î	lowercase i, circumflex accent
ï	ï	ï	lowercase i, umlaut
ð	ð	ð	lowercase eth, Icelandic
ñ	ñ	ñ	lowercase n, tilde
ò	ò	ò	lowercase o, grave accent
ó	ó	ó	lowercase o, acute accent
ô	ô	ô	lowercase o, circumflex accent
õ	õ	õ	lowercase o, tilde
ö	ö	ö	lowercase o, umlaut
ø	ø	ø	lowercase o, slash
ù	ù	ù	lowercase u, grave accent
ú	ú	ú	lowercase u, acute accent
û	û	û	lowercase u, circumflex accent
ü	ü	ü	lowercase u, umlaut
ý	ý	ý	lowercase y, acute accent
þ	þ	þ	lowercase thorn, Icelandic
ÿ	ÿ	ÿ	lowercase y, umlaut

Abbreviations

Another useful element is abbr. This can be used to provide a definition for an abbreviation, e.g.

```
<abbr title="HyperText Markup Language">HTML</abbr>
```

Will be displayed as: HTML

When you will hover over HTML, you see HyperText Markup Language

Graphical browsers normally show abbreviations with a dotted underline. The title appears as a tooltip. Screen readers may read the title at the user's request.

Unfortunately, Internet Explorer version 6 and lower do not support abbr. It does however support the related element acronym which has resulted in this element commonly being used for all abbreviations not just acronyms.

An acronym is a special abbreviation in which letters from several words are pronounced to form a new word (e.g. radar - RAdio Detection And Ranging). The letters in HTML are pronounced separately, technically making it a different sort of abbreviation known as an initialism.

Discouraged Formatting

HTML supports various formatting elements whose use is discouraged in favor of the use of cascading style sheets (CSS). Let us have a short overview of the discouraged formatting, so that you know what it is when you see it in some web page, and know how to replace it with CSS formatting. Some of the discouraged elements are merely discouraged, others are deprecated in addition. HTML formatting is generally used with older browsers that do not support CSS.

Element	Effect	Example	CSS Alternative	Note
b	boldface	bold	font-weight: bold;	
i	italics	<i>italics</i>	font-style: italics;	
u	underlined	<u>underlined</u>	text-decoration: underline	<i>deprecated</i>
tt	typewriter face	typewriter face	font-family: monospace	
s	strikethrough	strikethrough	text-decoration: line-through	<i>deprecated</i>
strikethrough	strikethrough	strikethrough	text-decoration: line-through	<i>deprecated</i>
big	big font	big	font-size: larger	
small	small font	small	font-size: smaller	
font	font size	size=1	font-size:(value)	<i>deprecated</i>
center	center a block		<div align="center">	<i>deprecated</i>

Cascading Style Sheets

The use of style elements such as `` for **bold** or `<i>` for *italic* is straight-forward but unfortunately it couples the presentation layer with the content layer. By using Cascading Style Sheets, the HTML author can decouple these two distinctly different parts so that a properly marked-up document may be rendered in various ways while the document itself remains unchanged. Thus, for example, if the publisher would like to change cited references in a document to appear as **bold** text as they were previously *italic*, they simply need to update the style sheet and not go through each document changing `` to `<i>` and vice-versa. Cascading Style Sheets also allow the reader to make these choices, overriding those of the publisher.

Continuing with the above example, let's say that the publisher has correctly marked up all their documents by surround references to cited material (such as the name of a book) in the documents with the `<cite>` tag:

```
<cite>The Great Gatsby</cite>
```

Then to make all cited references bold, one would put something like the following in the style sheet:

```
cite { font-weight: bold; }
```

Later someone tells you that references really need to be italic. Before CSS, you would have to hunt through all your documents, changing the `` and `` to `<i>` and `</i>` (but being careful *not* to change words that are in bold that are not cited references).

But with CSS, it's as simple as changing one line in the style sheet to

```
cite { font-style: italic; }
```

Hyperlinks

Hyperlinks are the basis of navigation of the internet. They are used for everything from moving around various bookmarks in the same page, to downloading applications and jumping to web pages on other web servers. Let us start with a quick example:

To learn more, see `Wikipedia`.

Will be displayed as: To learn more, see [Wikipedia](http://en.wikipedia.org/wiki/Main_Page).

Absolute vs. Relative

Before we get into creating a hyperlink (or link for short), we need to discuss the difference between an Absolute URL and a Relative URL. First, the Absolute URL can be used to direct the browser to any location. For example, an absolute URL might be :

<http://www.google.co.za/>

However, when there is a need to create links to multiple objects in the same directory tree as the web page, it is a tiring procedure to repeatedly type out the entire URL of each object being linked to. It also requires more work should the webpage move to a new location. This is where Relative URL's come in. They point to a path relative to the current directory of the web page. For example:

```
home.html
./home.html
../home.html
```

This is a relative URL pointing to a HTML file called home.html which resides in the same directory (folder) as the current web page containing the link. Likewise:

```
images/top_banner.jpg
```

This is another relative URL pointing to a subdirectory called images which contains an image file called "top_banner.jpg".

Linking to a location within a page with Anchor

Sometimes specifying a link to a page isn't enough. You might want to link to a specific place within a document. The book analogue of references of this type would be saying "Third paragraph on page 32" as opposed to just saying "page 32". The anchor tag (`<a>`) can be used for this too. Let's assume that you want a link from document **a.html** to a specific location in a document **b.html**. Then you start by naming the interesting location in **b.html**. This is done by adding a `` (where *some_name* is a string of your choice) tag at the interesting place in **b.html**. Now that location can be referenced to with `` from document **a.html**.

HTML

Target Links

Now we are ready to create a hyperlink. Here is the basic syntax :

```
<a href="URL location" target="target">Alias</a>;
```

In the above syntax "URL location" is either the absolute or relative path of the object being linked to. "target" is an optional attribute which specifies where the object being linked to is to be opened / displayed. For example :

```
<a href="http://www.google.co.za" target=0>Google Search Engine</a>
```

The above example uses an Absolute URL of <http://www.google.co.za>, and specifies a target of "0" (which would cause the URL to be opened in a new browser window). If frames are used, the name of a frame could be specified as the target for the URL to open in. This is useful if you have one frame containing a navigation bar with important links, and then another frame which is used to display the linked pages/objects.

Special targets

_blank

A new blank window is opened to load the linked document into. The location in the address bar (if shown in the new window) gives the hyperlink location of the new resource requested by the user's clicking on the hyperlink.

_self

The current frame that contains the document and the link to be clicked on is used to load the linked document; if the link is part of a document that occupies a whole window then the new document is loaded into the whole window, but in the case of a frame, the linked document is loaded into the current frame. The location won't be shown in the address bar unless the linked document was loaded into the main window as opposed to a child frame of a frameset.

_parent

The linked document is loaded into the parent frame of the one containing the link to be clicked on; this is only important in nested framesets. If window W contains frameset F consisting of a child frame A and also a child frame B that is itself a frameset FF with "grandchildren" frames C and D (giving us Window W with three visible panes A, C and D), then clicking a hyperlink in the page in frame D with a target=_parent will load the linked document into D's parent frame, that is, into frame B, so replacing frameset FF that was previously defined as the content of frame B. Documents C and D that were the frames of this frameset FF in B will be entirely replaced and this will leave only frame A and the new document from the hyperlink left in frame B, all inside the main frameset F in window W. The location is only shown in the address bar of the window if the parent frame happened to be the window itself.

_top

The linked document is loaded into the window, replacing all files currently displayed in the window in whatever frames they may be found in. The location at the top of the window, in the address/location bar is seen to point to the linked document once the hyperlink is clicked.

Hyperlinks on images

An example:

```
<a href="http://en.wikipedia.org/wiki/HTML">  
</a>
```

Example rendering:

```
25 </head>  
26 <body text="#000000"  
    bgcolor="#FFFFFF">  
27 <table width="1000"  
28     <tr>  
29         <td width="200"  
30         </td>  
31         <td valign="top"  
32             <div align="c  
33             </div>  
34             <p class="Bod  
35             <h1 class="He  
36             <p class="Cap  
Entertainment</a>  
37         | <a href=
```

As you can see, placing hyperlinks on images is in complete analogy to placing them on text. Instead of putting text inside the a element, you place there an image.

Images

Let us start with a quick minimum example:

```

```

And let us also have a look at more extended example:

```

```

Images are mostly stored not directly in the HTML source, but rather in an external file. To place an image into an HTML source, use the `img` tag with the `src` attribute containing the URL of the image file. To support browsers that cannot render images, you can provide the `alt` attribute with a textual description of the image. To provide a tooltip to the image, use the `title` attribute.

The space before the `/>` in the examples is there on purpose. Some older browsers behave strangely if the space is omitted.

Placement

Per default, images are placed *inline* to its surroundings. To place the image as a *block* or *float* instead, you can use [Cascading Style Sheets](#).

Alternative text and tooltip

The HTML specification requires that all images have an `alt` attribute. This is commonly known as *alt text*. Images can be split in to two categories:

- those that add to the content of the page;
- those that are purely decorative, e.g. spacers, fancy borders and bullets.

Decorative images should have empty alt text, i.e. `alt=""`. Images used as bullets may use an asterisk, `alt="*"`.

All other images should have meaningful alt text. Alt text is not a description of the image, use the `title` attribute for short descriptions or `longdesc` to link to long descriptions. Alt text is something that will be read instead of the image. For example,

```
 makes the best widgets in the world.
```

The alt text should be the company's name not the ever popular 'Our logo', which would give the sentence 'Our logo makes the best widgets in the world.' when read in a text only browser.

The `alt` attribute stands for *alternate* which non-graphic-capable browsers (such as Lynx) may use to better enable its user to understand the purpose of the image. Many browsers, such as Internet Explorer, incorrectly use the `alt` attribute' tag to produce image tooltips. However, the `title` attribute should actually be used for this. Other browsers such as Mozilla Firefox display tooltips only when `title` is used.

Width and height

To have an image appear smaller than it is in the external file, use the attributes `width` and `height`.

In addition, you can help web browsers layout your web pages more quickly by always supplying the `width` and `height` attributes. Make sure you give the correct values or your images will be distorted.

All attributes

Above, we have covered some of the most important attributes of the `img` element. What follows is a list of all attributes that are allowed in the `img` element as specified in the HTML 4 DTD:

Attributes of IMG element

Attribute	Description
<code>src</code>	the URL of the image that should be displayed by the browser.
<code>alt</code>	a short textual replacement for the image that may be used by the browser instead of showing the image, e.g. for blind users, users without a graphic-capable browser or users who have disabled image loading.
<code>longdesc</code>	a URL that provides a <i>long description</i> of the image.
<code>name</code>	<i>for backwards compatibility, use <code>id</code> in new documents</i> - the name of the image which may be used by scripting languages.
<code>id</code>	the unique identifier for the image which may be used by scripting languages.
<code>class</code>	assigns one or more CSS classes to this image to control styling
<code>lang</code>	language information
<code>dir</code>	text direction
<code>title</code>	title of the image, which some browsers display in a hover box (tooltip), especially when the user holds his/her mouse steady over the image for a short time.
<code>style</code>	in-line CSS information.
<code>ismap</code>	(not yet written)
<code>usemap</code>	(not yet written)
<code>align</code>	<i>deprecated, use CSS in new documents</i> - alignment of the image i.e. <i>bottom</i> , <i>middle</i> , <i>top</i> , <i>left</i> or <i>right</i> .
<code>width</code>	the width of the image, for bitmap images this should be given as a plain number, e.g. <i>200</i> , which would result in a width of 200 pixels
<code>height</code>	height of the image, see <i>width</i> for usage
<code>border</code>	<i>deprecated, use CSS in new documents</i> - border width when image is linking to something, usually set to <i>0</i>
<code>hspace</code>	<i>deprecated, use CSS in new documents</i>
<code>vspace</code>	<i>deprecated, use CSS in new documents</i>

Lists

In HTML, there are three kinds of lists, each appropriate for a different kind of information. A bullet list, meant for elements that have no order or the order is unimportant, is created using the `` and `` opening and closing tag. A numbered list, meant for elements whose order matters, such as a sequence of steps to perform, is created using the `` and `` opening and closing tag. Finally, there are definitions lists, created with `<dl>` and `</dl>` tags.

Ordered Lists

Ordered lists provide a list of items, each of which are preceded by an incremental number starting from 1.

Sample HTML:

```
<ol>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ol>
```

Example rendering:

1. First item
2. Second item
3. Third item

Unordered Lists

Unordered lists display a list of items, each of which is prefixed by a bullet.

Sample HTML:

```
<ul>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ul>
```

Example rendering:

- First item
- Second item
- Third item

HTML

Definition Lists

Definition lists display a list of bolded terms, followed by the definition on a new line and prefixed by a tab (by default).

Sample HTML:

```
<dl>
  <dt>Term 1</dt>
  <dd>Definition of Term 1</dd>
  <dt>Term 2</dt>
  <dd>Definition of Term 2</dd>
</dl>
```

Example rendering:

Term 1
Definition of Term 1
Term 2
Definition of Term 2

If two terms share the same definition they can be grouped together like so:

```
<dl>
  <dt>Cascading Style Sheets</dt>
  <dt>CSS</dt>
  <dd>Definition of Cascading Style Sheets (aka CSS)</dd>
  <dt>Term 2</dt>
  <dd>Definition of Term 2</dd>
</dl>
```

Example rendering:

Cascading Style Sheets CSS
Definition of Cascading Style Sheets (aka CSS)
Term 2
Definition of Term 2

If a term has two alternative definitions use a separate dd element for each definition, e.g.

```
<dl>
  <dt>Mouse</dt>
  <dd>Small mammal</dd>
  <dd>Input device for a computer</dd>
</dl>
```

Example rendering:

Mouse

HTML

Small mammal
Input device for a computer

Longer definitions can be broken up into paragraphs by nesting p elements within the dd element.

```
<dl>
  <dt>Term 2</dt>
  <dd>
    <p>First paragraph of the definition.</p>
    <p>Second paragraph of the definition.</p>
  </dd>
</dl>
```

Example rendering:

Term 2

First paragraph of the definition.
Second paragraph of the definition.

Nested Lists

Lists can be nested. An example:

```
<ul>
  <li>Lists
    <ul>
      <li>Numbered Lists</li>
      <li>Unnumbered Lists</li>
    </ul>
  </li>
</ul>
```

Example rendering:

- Lists
 - Numbered Lists
 - Unnumbered Lists

When nesting, nested list elements should be within a parent *list item* element.

An example of *incorrect nesting*:

```
<ul>
  <li>Lists</li>
  <ul>
    <li>Numbered Lists</li>
    <li>Unnumbered Lists</li>
  </ul>
</ul>
```

HTML

```
</ul>
```

A further example of *incorrect nesting*, with two consecutive UL elements:

```
<ul>
  <li>
    Outer list item
    <ul>
      <ul>
        <li>
          Inner list item within two consecutive UL elements
        </li>
      </ul>
    </ul>
  </li>
</ul>
```

Note on format

The above descriptions of each of the three list types refer to the default rendering of the corresponding HTML code by the web browser. However, by using CSS, you are able to change the formatting of the lists. For example, with CSS you are able to make the lists horizontal as opposed to the vertical.

Example

An example of using lists to mark up a recipe for pancakes.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
  <head>
    <title>Pancakes</title>
  </head>
  <body>
    <h1>A Recipe for Pancakes</h1>
    <p>From <a href="http://en.wikibooks.org/wiki/Cookbook:Pancake">Wikibooks Cookbook</a>.</p>
    <p>These pancakes make a good breakfast for a family.
      They go well with real maple syrup.
      They are healthy too (as long as you don't over do the syrup)
      since whole wheat flour contributes to your fiber intake.
    </p>
    <h2>Ingredients</h2>
    <ul>
      <li>1 cup whole wheat flour</li>
      <li>1 tablespoon common granulated sugar</li>
      <li>2 teaspoons baking powder</li>
      <li>1/4 teaspoon salt</li>
      <li>1 egg</li>
      <li>1 cup milk</li>
      <li>2 tablespoons oil</li>
      <li>additional oil for frying</li>
```

HTML

```
</ul>
<h2>Procedure</h2>
<ol>
  <li>Oil a frying pan.</li>
  <li>Mix the dry ingredients in one bowl.</li>
  <li>In another bowl, scramble the egg, then add the other wet ingredients.
    This includes the 2 tablespoons of oil.</li>
  <li>Mix the dry and wet ingredients together,
    well enough to eliminate dry spots but no more.</li>
  <li>Heat the frying pan to medium temperature.
    The pan is hot enough when a drop of water dances around
    rather than simply boiling away.</li>
  <li>Pour a pancake, about 4 inches in diameter using about a 1/4 cup of batter.</li>
  <li>The pancake will bubble. When the bubbling settles down and
    the edges are slightly dry, flip the pancake.</li>
  <li>When the pancake looks done, remove it and start another one.</li>
</ol>
<h2>Toppings</h2>
<p>Traditionally, pancakes are topped with butter and maple syrup.
  Other toppings can include strawberries, applesauce, cinnamon, or sugar.
</p>
</body>
</html>
```

Tables

Tables are used for presenting tabular data and abused for laying out pages. They can be inserted anywhere on the page, even within other tables. We will be looking at creating a basic table and then adding lots of tags to it so we can see just what the outcome will be. Experimentation is the name of the game here. The tags most useful when creating tables are `<table>` - table, `<tr>` - table row, `<td>` - table data, and `<th>` - table heading.

Minimal tables

First let us have a look at quick example:

```
<table>
<tr><th>Food</th><th>Price</th></tr>
<tr><td>Bread</td><td>$2.99</td></tr>
<tr><td>Milk</td><td>$1.40</td></tr>
</table>
```

Every table begins with a `<table>` tag and ends with a `</table>` tag. In the table tag, you can define the attributes of the table, as you will see later.

The table contains rows, each begins with the `<tr>` table row tag and optionally ends with the `</tr>` tag. Rows must be inside tables.

The rows contain cells, each begins with the `<td>` table data tag and optionally ends with the `</td>` tag. Cells must be inside rows.

If you put a table cell outside a row, or if you forget to close a cell, or row, or table it will have unpredictable results. Text intended to be in the table may appear at an unexpected position, outside the table. At worst, the entire contents of the table will not be displayed.

For example, in IE and Firefox:

- A cell outside a row is treated as in a separate row at the vertical position concerned
- All content outside cells, whether in a row or not, is put before the whole table, in the order in which it occurs; IE puts each item on a new line; Firefox does not, but puts in some cases a blank space between items.

If the optional `</td>` and `</tr>` are not put, the above refers to content before the first row, and in rows before the first element only. Note that `</table>` is required; if it is forgotten all following content is considered part of the last cell of the last row, even further tables.

Task - Create a table

1. Open your default.htm and save it as table.htm in the appropriate folder
2. Create this HTML code in the body of the document

```
<table>
<tr><th>Food</th><th>Price</th></tr>
<tr><td>Bread</td><td>$2.99</td></tr>
```

HTML

```
<tr><td>Milk</td><td>$1.40</td></tr>
</table>
```

1. Save the file and view it in the browser.

The result is:

Food Price

Bread \$2.99

Milk \$1.40

It doesn't look too much like a table yet, but we'll add more soon.

Note: This table is made up of two rows (check out the two `<tr>` tags) and within each row there are two data entries (the two `<td>` tags)

You might compare a table with a spreadsheet. This one has two rows and two columns making 4 cells containing data. (2 rows x 2 columns = 4 cells)

Caption and headings

Let us start with a quick example:

```
<table>
<caption>Formulas and Results</caption>
<tr><th>Formula</th><th>Result</th></tr>
<tr><td>1 + 1</td><td>2</td></tr>
<tr><td>3 * 5</td><td>15</td></tr>
</table>
```

Captions are useful for defining or describing the content of the table. They are optional.

To add a caption to a table, enter the caption element after the opening table tag, with the text of the caption inside the element, as shown in the following.

```
<table>
<caption>Formulas and Results</caption>
...
</table>
```

Captions are usually displayed outside the border of the table, at the top. The exact appearance and placement of captions is subject to CSS styling.

Table **headings** are a way of defining the contents of the table columns. They are usually only used in the first `<tr>`, table row.

Instead of using a `<td>` for the cell, we use a `<th>`.

By default the text in table headings is displayed bold and centered.

The Syntax is: `<tr><th>text</th><th>text</th></tr>`

HTML

Task - Table Caption and Headings

1. Open your table.htm file
2. Add your own caption to the table
3. View the result
4. Add the table headings ITEMS and \$ PRICE for the table
5. View the result

Borders

A border around a table is optional: sometimes they help to define the table, and sometimes the table looks better without them.

However having borders turned on while you are creating the table is a very good idea since it makes tables much easier to work with. You can get rid of the border once the table is completed.

The border of this table is 1 pixel wide.

The border on this table is 5 pixels wide.
--

The default value is 0 (i.e. borderless).

Border is an attribute of the table tag. The syntax is:

`<table border=X>` where X is the border size in pixels.

You can also specify a border colour, although this is an Internet Explorer tag only. The syntax is:

`<table bordercolor="#000000">`

Note that it is not recommended to specify the border colour using HTML - it is much better to use [CSS](#) for this purpose.

Task - Create a border around a table

1. Open your table.htm file.
2. In the `<table>` tag, add `border=2`
i.e. `<table border=2>`.
3. Save the file and view it.
4. Change the size of the border (i.e., try 0, 10, and try a crazy number).
5. View the results as you go.

Did you spot that only the outside border gets bigger?

Height and Width

A table, by default, will be as large as the data that is entered into it.

HTML

We can change the overall height and width of the table to get it the size we want.

It is possible to give the size in absolute pixels, or as a relative percentage of the screen size.

The syntax is: `<table height=??? width=???` where ??? is the size in pixels or percentage.

It is also possible to control the dimensions of individual table cells or rows.

e.g. `<tr height=80> <td width="50%">`

It is possible to mix absolute and relative heights and widths.

Note that you can do the same thing with CSS by changing the padding.

Task - Define the size of a table

1. Open your table.htm file.
2. In the `<table border=2>` tag, we will add the height and width
e.g. `<table border=2 height=200 width=300>`
3. Save the file and then view it. Resize the browser window, and watch what happens - the table size stays the same.
4. Experiment changing the measurements and view the file again.
5. Now replace the pixels measurements with percentages
e.g. `<table border=2 height="40%" width="50%">`
6. Save the file and then view it. Resize the browser window, and watch what happens - this time the table changes size as the window size changes.

Cell Spacing and Cell Padding

Cell Spacing is the number of pixels between the table cells.

Cell Padding is the pixel space inside the cells. i.e. the distance between the information and the sides of the table cells.

Both these options are attributes of the `<table>` tag

e.g. `<table border=1 cellspacing=0 cellpadding=0>`

Note: The default for both is 2

Task - Cell Spacing and Padding

1. Open your table.htm file. Make sure that your table has a large height and width set (e.g. 300x200) - if not then you won't be able to see the effect of cellpadding and cellspacing.
2. Experiment with changing the size of the table border, cellspacing and cellpadding. Try different combinations of 0, 1, 5, 10, etc.
3. View the result each time

Alignment of table cells

The default alignment of the contents of table cells is left and vertically centered.

HTML

If you want to change the alignment of cells, it has to be done individually for each cell. The align command is included in the <td> tag. You can also change the alignment of an entire row by specifying the alignment in the <tr> tag

Horizontal alignment

Syntax:

<td align="position"> where position is left, center, or right
or

<tr align="position"> where position is left, center, or right

Vertical alignment

Syntax:

<td valign="position"> where position is top, middle or bottom
or

<tr valign="position"> where position is top, middle or bottom

You can also include align and valign commands in the table row tag and in the table tag.

Note: Including align="left" or align="right" in the table tag does NOT align the contents of the table. Instead it aligns the whole table on the page. i.e., it makes the text outside the table wrap around the table.

Task - Alignment of table cells

1. Open your table.htm file
2. Change the alignment of the table cells so that they look like:

bread	\$2.99
Milk	\$1.40

or

bread	\$2.99
Milk	\$1.40

1. Experiment with other vertical and horizontal alignments.
2. View the result each time

Row span and column span

Every row must have the same number of table datas, occasionally table datas have to span more than one column or row. In this case the tags colspan and/or rowspan are used - where they are set to a number.

			<i><-- This row has three table datas</i>
			<i><-- This row has two. The first uses colspan="2"</i>
			<i><-- This row has three table datas, but one spans two rows because it uses rowspan="2"</i>
			<i><-- This row has only two table datas, because its first is being taken up.</i>

Syntax:

- `<td colspan=X>` where X is the number of columns that the cell spans across.
- `<td rowspan=X>` where X is the number of rows that the cell spans across.

Task - Row span and column span

1. Open your table.htm file.
2. Experiment with making one table cell span across multiple rows.
3. Experiment with making one table cell span across multiple columns.
4. View the result each time.

Background colour and images

It is possible to give each table cell, (or row, or table) a different background colour.

Syntax:

```
<td bgcolor="colour">  
<tr bgcolor="colour">  
<table bgcolor="colour">
```

where colour is a colour name or hexadecimal code.

HTML

Note: table background colours only display in version 3 browsers and above, and they may not print correctly.

Note: it is not recommended to specify background colours using HTML - it is much better to use [Cascading Style Sheets](#) for this purpose.

A **background image** is another modification of the appearance of a cell, row, or a complete table. Again these only display in version 3 browsers and above, and they may not print correctly.

Syntax:

```
<td background="filename">
<tr background="filename">
<table background="filename">
```

where filename is the path and filename of the background image.

Note: it is not recommended to specify background images using HTML - it is much better to use [CSS](#) for this purpose.

Task - Background colour and images

1. Open your table.htm file.
2. Experiment with changing the background colour of a table cell, a table row, and the table itself.
3. Add a background image to a table cell, a table row, and the table itself.
4. View the result each time.

Column groups

To specify a given format for a table column, you can use the `<col>` and `<colgroup>` tags. These tags are located at the top of the table, and specify the default format for the given column.

With the `<col>` tag, the first instance indicates the formatting for the first column, the second for the second column, and so on. `<colgroup>` works similarly, but also includes the `span` tag to cover multiple columns.

```
<colgroup> <col span="3" style="background-color:red"> <col style="background-color:yellow">
<col span="2" style="background-color:green"> </colgroup>
```

Project Completion

Jan Feb Mar Apr May Jun
3% 17% 40% 55% 86% 100%

Note:

Chrome, Firefox, and Safari only supports the `span` and `width` elements of these tags.

Summary

In this module you learn how to:

- Create and customise HTML tables,
- Control their dimensions and appearance,
- Add a caption to a table,
- Control the alignment of the table contents,
- Various attributes of the table tags.

Quotations

There are two kinds of quotations supported by HTML--inline ones and block quotations.

Inline quotations

An example:

```
<q>An inline quotation of significant length  
(say 25 words, for example) goes here...</q>.
```

Will be displayed as:

An inline quotation of significant length (say 25 words, for example) goes here....

Block quotations

An example:

```
<blockquote>Lorem ipsum dolor sit amet, consectetur adipisicing elit,  
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.  
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris  
nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in  
reprehenderit in voluptate velit esse cillum dolore eu fugiat  
nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt  
in culpa qui officia deserunt mollit anim id est laborum.  
</blockquote>
```

Example rendering:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Comments

HTML offers the possibility to insert comments into the page. To place a comment in your HTML code, start it with `<!--` and close it with `-->`. An example:

```
<p>The first paragraph.</p>
<!-- This comment spans one line, and will not be displayed to the browser. -->
<p>The second paragraph.</p>
<!--
  This comment spans multiple lines,
  and will also not be displayed to the browser.
-->
<p>The third paragraph.</p>
```

Unlike notes in office suites, comments are completely ignored by browsers, so the readers of the page have no idea of their presence. They can be viewed in the source of the web page though.

You should avoid **nested comments**, as these can cause troubles to many browsers. An example:

```
<p>The second paragraph.</p>
<!--
  <!--
    Nested comments should better be avoided.
  -->
-->
<p>The third paragraph.</p>
```


Forms

HTML forms are an easy way to gather data from the end user. Processing them requires a server-side scripting language such as [PHP](#) or (in some cases when limited interaction is to be provided within a single page) a client-side scripting language such as [JavaScript](#).

Here is a simple form:

```
<form id="" action="" method="post">
<fieldset>
<legend>Personal details</legend>

<label for="first">First name</label>
<input type="text" name="first" id="first"><br />

<label for="family">Family name</label>
<input type="text" name="family" id="family"><br />

<input type="submit" name="personal">

</fieldset>
</form>
```

Here's an explanation -

id

The name of the form or control.

action

The URL of a server-side script which can process the data.

method

The method used to send the information. Two methods are supported, POST and GET.

POST is the preferred method except for simple searches which generally use GET. Use with server-side languages.

fieldset

Form controls are normally contained in a fieldset element. Complex forms may have multiple fieldsets. Fieldsets can contain other fieldsets.

legend

Each fieldset begins with a legend element. The content of the element is used as a title placed in the border of the fieldset.

input type="" name="" id=""

various types of input controls. Supported types are - submit, text, password, checkbox, radio, reset, file, hidden, image and button. The name Attribute is used by the server to identify which piece of data was entered in a given box on the form. The id attribute is used to match an input with its label. The name and id attributes normally have identical values for text inputs but different values for checkbox and radio inputs.

label for=""

A label for a single form control. The value of the for attribute must match the id attribute of a

HTML

form control in the same form.

select

There is also a SELECT element for drop down lists and a TEXTAREA element for multi-line text input.

This simple example uses `
` tags to force newlines between the different controls. A real-world form would use more structured markup to layout the controls neatly.

Formatting with CSS

The HTML

The HTML for this form is amazingly simple and you do not have to create hundreds of divs all aligned left and right; this will cause a lot of frustration and a lot of debugging as various browsers still interpret CSS code differently.

```
<form>
```

```
  <label for="name">Name</label>
  <input id="name" name="name"><br />
```

```
  <label for="address">Address</label>
  <input id="address" name="address">
```

```
</form>
```

The CSS

The CSS for this code is a little bit more interesting:

```
label,input {
  float: left;
  width: 150px;
  display: block;
  margin-bottom: 10px;
}
```

```
label {
  width: 75px;
  text-align: right;
  padding-right: 20px;
}
```

```
br {
  clear: left;
}
```

Let's explain the code

HTML

```
label,input {  
    float: left;  
    width: 150px;  
    display: block;  
    margin-bottom: 10px;  
}
```

The CSS for the label has four sections to it:

1. float: the float command is used to establish that the label is floated to the left hand side of the form
2. width: this defines how big the label must be, keeping all the labels at a fixed width keeps everything in a nice ordered line.
3. display: the element will be displayed as a block-level element, with a line break before and after the element
4. margin-bottom: by adding a margin to the bottom of this label it insures that labels are positioned nicely one under another with a nice padding between each

```
label {  
    width: 75px;  
    text-align: right;  
    padding-right: 20px;  
}
```

1. width: again this is to define a fixed width giving everything a nice defined unity.
2. Text-align: align the text right keeps everything away from the left aligned labels again keeping things in unison.
3. Padding-right: this means that there is a nice padding on the right keeping things once again fine tuned.

```
br {  
    clear: left;  
}
```

1. clear: this is the most important part without the clear:left nothing will align properly this basically makes everything within each element sequence align underneath each other and to the left.

For more details, see the [HyperText Markup Language/Tag List/form](#) section of this book.

CSS

So far we have seen how to divide text into paragraphs and to create section headings. Whilst HTML allows you to define the structure of your documents it gives very limited control over their appearance. [Cascading Style Sheets](#) (CSS) is a language that describes the presentation of documents. You can use CSS to alter the appearance of your HTML documents.

This section gives an introduction to styling HTML with CSS. CSS itself is covered in the companion wikibook [Cascading Style Sheets](#).

What does CSS do?

The example page from the [Paragraphs and Headings section](#) would look something like this without CSS.

Sundial

From Wikipedia, the free encyclopedia.

A sundial measures time by the position of the sun. The most commonly seen designs, such as the 'ordinary' or standard garden sundial, cast a shadow on a flat surface marked with the hours of the day. As the position of the sun changes, the time indicated by the shadow changes. However, sundials can be designed for any surface where a fixed object casts a predictable shadow.

Most sundial designs indicate apparent solar time. Minor design variations can measure standard and daylight saving time, as well.

History

Sundials in the form of obelisks (3500 BC) and shadow clocks (1500 BC) are known from ancient Egypt, and were developed further by other cultures, including the Chinese, Greek, and Roman cultures. A type of sundial without gnomon is described in the old Old Testament (Isaiah 38:2). The mathematician and astronomer Theodosius of Bithynia (ca. 160 BC-ca. 100 BC) is said to have invented a universal sundial that could be used anywhere on Earth. The French astronomer Oronce Finé constructed a sundial of ivory in 1524. The Italian astronomer Giovanni Padovani published a treatise on the sundial in 1570, in which he included instructions for the manufacture and laying out of mural (vertical) and horizontal sundials. Giuseppe Biancani's *Constructio instrumenti ad horologia solaris* discusses how to make a perfect sundial, with accompanying illustrations.

Installation of standard sundials

Many ornamental sundials are designed to be used at 45 degrees north. By tilting such a sundial, it may be installed so that it will keep time. However, some mass-produced garden sundials are inaccurate because of poor design and cannot be corrected. A sundial designed for one latitude can be adjusted for use at another latitude by tilting its base so that its style or gnomon is parallel to the Earth's axis of rotation, so that it points at the north celestial pole in the northern hemisphere, or the south celestial pole in the southern hemisphere.

A local standard time zone is nominally 15 degrees wide, but may be modified to follow geographic and political boundaries. A sundial can be rotated around its style or gnomon (which must remain pointed at the celestial pole) to adjust to the local time zone. In most cases, a rotation in the range

HTML

of 7.5 degrees east to 23 degrees west suffices.

To correct for daylight saving time, a face needs two sets of numerals or a correction table. An informal standard is to have numerals in hot colors for summer, and in cool colors for winter. Rotating the sundial will not work well because most sundials do not have equal hour angles. Ordinary sundials do not correct apparent solar time to clock time. There is a 15 minute variation through the year, known as the equation of time, because the Earth's orbit is slightly elliptical and its axis is tilted relative to the plane of its orbit. A quality sundial will include a permanently-mounted table or graph giving this correction for at least each month of the year. Some more-complex sundials have curved hour-lines, curved gnomons or other arrangements to directly display the clock time.

By adding a style sheet the appearance could be changed to:

Sundial

From Wikipedia, the free encyclopedia.

A sundial measures time by the position of the sun. The most commonly seen designs, such as the 'ordinary' or standard garden sundial, cast a shadow on a flat surface marked with the hours of the day. As the position of the sun changes, the time indicated by the shadow changes. However, sundials can be designed for any surface where a fixed object casts a predictable shadow.

Most sundial designs indicate apparent solar time. Minor design variations can measure standard and daylight saving time, as well.

History

Sundials in the form of obelisks (3500 BC) and shadow clocks (1500 BC) are known from ancient Egypt, and were developed further by other cultures, including the Chinese, Greek, and Roman cultures. A type of sundial without gnomon is described in the old Old Testament (Isaiah 38:2).

The mathematician and astronomer Theodosius of Bithynia (ca. 160 BC-ca. 100 BC) is said to have invented a universal sundial that could be used anywhere on Earth. The French astronomer Oronce Finé constructed a sundial of ivory in 1524. The Italian astronomer Giovanni Padovani published a treatise on the sundial in 1570, in which he included instructions for the manufacture and laying out of mural (vertical) and horizontal sundials. Giuseppe Biancani's *Constructio instrumenti ad horologia solaris* discusses how to make a perfect sundial, with accompanying illustrations.

Installation of standard sundials

Many ornamental sundials are designed to be used at 45 degrees north. By tilting such a sundial, it may be installed so that it will keep time. However, some mass-produced garden sundials are inaccurate because of poor design and cannot be corrected. A sundial designed for one latitude can be adjusted for use at another latitude by tilting its base so that its style or gnomon is parallel to the Earth's axis of rotation, so that it points at the north celestial pole in the northern hemisphere, or the south celestial pole in the southern hemisphere.

A local standard time zone is nominally 15 degrees wide, but may be modified to follow geographic and political boundaries. A sundial can be rotated around its style or gnomon (which must remain pointed at the celestial pole) to adjust to the local time zone. In most cases, a rotation in the range of 7.5 degrees east to 23 degrees west suffices.

To correct for daylight saving time, a face needs two sets of numerals or a correction table. An informal standard is to have numerals in hot colors for summer, and in cool colors for winter.

HTML

Rotating the sundial will not work well because most sundials do not have equal hour angles. Ordinary sundials do not correct apparent solar time to clock time. There is a 15 minute variation through the year, known as the equation of time, because the Earth's orbit is slightly elliptical and its axis is tilted relative to the plane of its orbit. A quality sundial will include a permanently-mounted table or graph giving this correction for at least each month of the year. Some more-complex sundials have curved hour-lines, curved gnomons or other arrangements to directly display the clock time.

How to add a CSS style sheet

CSS is normally kept in a separate file from the HTML document. This allows a style sheet to be shared by several HTML documents. The link element is used to apply the rules from a CSS style sheet to a document. The basic syntax is:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

where *style.css* is the URL for the style sheet.

The CSS file for the example above contains the following:

```
body {
  background:#ffc;
  color:#000;
  font-family:cursive
}
```

```
h1 {
  color:red;
  text-align:center;
  font-size:1.2em;
  font-weight:bold;
  margin:0
}
```

```
h2 {
  text-align:center;
  font-size:1em;
  font-weight:bold;
  margin:1em 0 0
}
```

```
p {
  text-indent:2em;
  text-align:justify;
  margin:0
}
```

Save this as *sundial.css*.

The HTML document from the [previous section](#) with a link element added on the fifth line is given below. Save this as *sundial2.htm* in the same directory.

HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
  <head>
    <title>Sundial</title>
    <link rel="stylesheet" type="text/css" href="sundial.css">
  </head>
  <body>
    <h1>Sundial</h1>
    <p>From Wikipedia, the free encyclopedia.</p>
    <p>A sundial measures time by the position of the sun. The most commonly seen designs, such as the 'ordinary' or standard garden sundial, cast a shadow on a flat surface marked with the hours of the day. As the position of the sun changes, the time indicated by the shadow changes. However, sundials can be designed for any surface where a fixed object casts a predictable shadow.</p>
    <p>Most sundial designs indicate apparent solar time. Minor design variations can measure standard and daylight saving time, as well.</p>
    <h2>History</h2>
    <p>Sundials in the form of obelisks (3500 BC) and shadow clocks (1500 BC) are known from ancient Egypt, and were developed further by other cultures, including the Chinese, Greek, and Roman cultures. A type of sundial without gnomon is described in the old Old Testament (Isaiah 38:2).</p>
    <p>The mathematician and astronomer Theodosius of Bithynia (ca. 160 BC-ca. 100 BC) is said to have invented a universal sundial that could be used anywhere on Earth. The French astronomer Oronce Finé constructed a sundial of ivory in 1524. The Italian astronomer Giovanni Padovani published a treatise on the sundial in 1570, in which he included instructions for the manufacture and laying out of mural (vertical) and horizontal sundials. Giuseppe Biancani's Constructio instrumenti ad horologia solaria discusses how to make a perfect sundial, with accompanying illustrations.</p>
    <h2>Installation of standard sundials</h2>
    <p>Many ornamental sundials are designed to be used at 45 degrees north. By tilting such a sundial, it may be installed so that it will keep time. However, some mass-produced garden sundials are inaccurate because of poor design and cannot be corrected. A sundial designed for one latitude can be adjusted for use at another latitude by tilting its base so that its style or gnomon is parallel to the Earth's axis of rotation, so that it points at the north celestial pole in the northern hemisphere, or the south celestial pole in the southern hemisphere.</p>
    <p>A local standard time zone is nominally 15 degrees wide, but may be modified to follow geographic and political boundaries. A sundial can be rotated around its style or gnomon (which must remain pointed at the celestial pole) to adjust to the local time zone. In most cases, a rotation in the range of 7.5 degrees east to 23 degrees west suffices.</p>
    <p>To correct for daylight saving time, a face needs two sets of numerals or a correction table. An informal standard is to have numerals in hot colors for summer, and in cool colors for winter. Rotating the sundial will not work well because most sundials do not have equal hour angles.</p>
    <p>Ordinary sundials do not correct apparent solar time to clock time. There is a 15 minute variation through the year, known as the equation of time, because the Earth's orbit is slightly elliptical and its axis is tilted relative to the plane of its orbit. A quality sundial will include a permanently-mounted table or graph giving this correction for at least each month of the year. Some more-complex sundials have curved hour-lines, curved gnomons or
```

HTML

other arrangements to directly display the clock time.

```
</p>  
</body>  
</html>
```

Open *sundial2.htm* with your web browser and you should see a page with a pale yellow background.

Validating HTML

There are fixed rules that define which tags may be used in an HTML document, where they can be placed. As your documents get larger it can be difficult to be sure that everything is correct. There are automated tools that will check your HTML for you. These tools are known as validators. Several validators are free to use, including

- [Web Design Group HTML Validator](#)
- [The W3C Markup Validation Service](#)

Try uploading the index.html or index.htm file you created in the previous section to one of the validators listed above. Alternately both validators have options that let you enter HTML directly so you could cut and paste the example from this page into the validator.

There is also an HTML-validating Firefox extension that can validate HTML using either HTML Tidy or the SGML Parser (what the w3 validator is based on). It is available [here](#) for all platforms.

It is good practice to validate each HTML document you create. Note that many visual design tools will let you create invalid webpages so it is important to check pages produced in these packages as well.

If the HTML document is valid it means that the webpage will display exactly as you programmed it to on stable W3C compliant browsers. In the case of text browsers such as Lynx, the text will format correctly so that it can be read easily by the consumer. Knowing HTML also means that you can edit the pages created using WYSIWYG programs manually, as these will without fail throw in unnecessary coding which clogs up and slows down the loading of your page.

Conditional Comments

Conditional comments are a proprietary extension to Microsoft Internet Explorer for Windows (IE/win) version 5.0 and later. They are not available in Internet Explorer for Mac (IE/mac). They are a very useful way of handling the CSS bugs in the various versions of Internet Explorer.

Syntax

An ordinary (X)HTML comment looks like this:

```
<!-- This text will be ignored by the browser. -->
```

Conditional comments add additional syntax to comments. The simplest example is:

```
<!--[if IE]> This text will be shown by IE/win ver. 5.0 and higher. <![endif]-->
```

Browsers that don't understand the conditional comment syntax will process this as a normal comment, i.e. the content of the comment will be ignored.

Specific versions of IE/win can be targeted by changing the expression after the if. For example to target any version of IE/win with a major version of 5 use:

```
<!--[if IE 5]> The major version number of this browser is 5. <![endif]-->
```

The text will display in IE/win versions 5.0 and 5.5.

To target a specific version number, e.g. 5.0, the syntax is slightly quirky.

- `<!--[if IE 5.0]> You are using IE/win 5.0. <![endif]-->`
- `<!--[if IE 5.5000]> You are using IE/win 5.5. <![endif]-->`
- `<!--[if IE 6.0]> You are using IE/win 6.0. <![endif]-->`

Inequalities can be used in the expression by placing an operator **before** the IE. The operators are:

lt

less than (but at least version 5.0 which is the lowest version supporting conditional comments)

lte

less than or equal (but at least version 5.0 which is the lowest version supporting conditional comments)

gt

greater than

gte

greater than or equals

Example:

```
<!--[if gte IE 6]> This text will be shown by IE/win ver. 6.0 and higher. <![endif]-->
```

All the expressions can be negated by prefixing with !, e.g.

```
<!--[if !gte IE 6]> This text will be shown by versions of IE/win ver. below 6 that support conditional comments. <![endif]-->
```

HTML

`<!--[if !IE]>` This text will be not be shown by any version of IE/win that understands conditional comments. It won't be shown by any other browsers either because they will treat this as a normal comment. `<![endif]-->`

The second example may seem pointless but with a small tweak you can arrange to hide text from IE/win version 5 and above.

`<!--[if !IE]>-->` This text will be not be shown by any version of IE/win that understands conditional comments. It will be shown by other browsers because they will treat this as text sandwiched between two normal comments. `<!--<![endif]-->`

The following HTML document is a working example.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
  <head>
    <title>Conditional comments</title>
  </head>
  <body>
    <!--[if !IE]>-->
      <p>This is page is not being viewed with Internet Explorer for Windows version 5.0 or higher.</p>
    <!--<![endif]-->
    <!--[if IE]>
      <p>This is page is being viewed with Internet Explorer for Windows version 5.0 or higher.</p>
    <![endif]-->
  </body>
</html>
```

Use with CSS

Conditional comments can be used to pass additional stylesheets to IE/win. These stylesheets can provide fixes to layout bugs in IE/win. The basic idea is:

```
<head>
  <title>Conditional comments</title>
  <link rel="stylesheet" type="text/css" href="style.css">
  <!--[if IE 5]>
  <link rel="stylesheet" type="text/css" href="bugFixForIE5x.css">
  <![endif]-->
</head>
```

Proscribed Techniques

Frames

Frames allow a section of the webpage to be constantly visible in a column or section, by having multiple pages shown in separate subwindows. Usage of frames has been deprecated in HTML 4.0, and may only be used in Transitional or Frameset mode. Frames have been replaced by a combination of Cascading Style Sheets and AJAX scripting, as they keep a more suitable method of keeping or updating content on the screen.

Framing

Frames must begin with the `<frameset>` tag, and end with the `</frameset>` tag. The frameset has optional parameters to determine the number of rows and columns along with their sizes (e.g. `rows="2"`). A frameset tag may be nested within another tag, to allow complex frame layouts.

Within a frameset element, you can either specify a new frameset, or add the `<frame>` element. The frame element has the attributes `src` and `name` to specify the location of the document, and the `name` used for interconnection with other links.

Since some web browsers cannot display frames you must specify alternate content for these browsers in a `noframes` element within the frameset.

Inline frames

An inline frame begins with the `<iframe>` tag, and ends with the `</iframe>` tag. As with the frame element, you can apply the `src` and `name` attributes

Layers

Layers are not a part of HTML 4.01 or XHTML. They were a proprietary element created by Netscape. You can achieve the same effect using [CSS's](#) `z-index` property.

Confusingly, Dreamweaver has a layers feature which is functionally similar but is based on `div` elements styled with CSS and optionally animated with Javascript.

Music

Users should have control over the background sound or music; therefore, media player controls should always be visible. Don't try to script your own controls since they might not work in all environments.

One valid way to do an autostarting (background) music player is to use Windows Media Player ActiveX for IE. Other browsers like Firefox, Opera and Safari can use a generic solution which doesn't dictate what plugin should be used. Plugin can be, for example, QuickTime (Windows, Mac) or MPlayer (GNU/Linux). If a suitable player doesn't exist, then fall back to a plain link.

-Copy the following code to your site

-Replace all four occurrences of "clip.mp3" with correct filename

Other widely supported file formats are midi and wav, if you use those remember to change the type attribute accordingly:

audio/midi for midi files

audio/x-wav for wav files

```
<!--[if IE]>
<object id="player" classid="clsid:6BF52A52-394A-11d3-B153-00C04F79FAA6" width="288" height="69">
  <param name="URL" value="clip.mp3">
</object>
<![endif]-->
<!--[if!IE]> <-->
<object id="player" type="audio/mpeg" data="clip.mp3" width="288" height="69">
  <param name="src" value="clip.mp3">
  <a href="clip.mp3">Play background music artist - title [MP3]</a>
</object>
<!--> <![endif]-->
```

Browsers' extensions

Conditional Comments

Conditional comments are a proprietary extension to Microsoft Internet Explorer for Windows (IE/win) version 5.0 and later. They are not available in Internet Explorer for Mac (IE/mac). They are a very useful way of handling the CSS bugs in the various versions of Internet Explorer.

Syntax

An ordinary (X)HTML comment looks like this:

```
<!-- This text will be ignored by the browser. -->
```

Conditional comments add additional syntax to comments. The simplest example is:

```
<!--[if IE]> This text will be shown by IE/win ver. 5.0 and higher. <![endif]-->
```

Browsers that don't understand the conditional comment syntax will process this as a normal comment, i.e. the content of the comment will be ignored.

Specific versions of IE/win can be targeted by changing the expression after the if. For example to target any version of IE/win with a major version of 5 use:

```
<!--[if IE 5]> The major version number of this browser is 5. <![endif]-->
```

The text will display in IE/win versions 5.0 and 5.5.

To target a specific version number, e.g. 5.0, the syntax is slightly quirky.

- `<!--[if IE 5.0]> You are using IE/win 5.0. <![endif]-->`
- `<!--[if IE 5.5000]> You are using IE/win 5.5. <![endif]-->`
- `<!--[if IE 6.0]> You are using IE/win 6.0. <![endif]-->`

Inequalities can be used in the expression by placing an operator **before** the IE. The operators are:

lt

less than (but at least version 5.0 which is the lowest version supporting conditional comments)

lte

less than or equal (but at least version 5.0 which is the lowest version supporting conditional comments)

gt

greater than

gte

greater than or equals

Example:

```
<!--[if gte IE 6]> This text will be shown by IE/win ver. 6.0 and higher. <![endif]-->
```

HTML

All the expressions can be negated by prefixing with !, e.g.

`<!--[if !gte IE 6]>` This text will be shown by versions of IE/win ver. below 6 that support conditional comments. `<![endif]-->`

`<!--[if !IE]>` This text will be not be shown by any version of IE/win that understands conditional comments. It won't be shown by any other browsers either because they will treat this as a normal comment. `<![endif]-->`

The second example may seem pointless but with a small tweak you can arrange to hide text from IE/win version 5 and above.

`<!--[if !IE]>-->` This text will be not be shown by any version of IE/win that understands conditional comments. It will be shown by other browsers because they will treat this as text sandwiched between two normal comments. `<!--<![endif]-->`

The following HTML document is a working example.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
  <head>
    <title>Conditional comments</title>
  </head>
  <body>
    <!--[if !IE]>-->
      <p>This is page is not being viewed with Internet Explorer for Windows version 5.0 or higher.</p>
    <!--<![endif]-->
    <!--[if IE]>
      <p>This is page is being viewed with Internet Explorer for Windows version 5.0 or higher.</p>
    <![endif]-->
  </body>
</html>
```

Use with CSS

Conditional comments can be used to pass additional stylesheets to IE/win. These stylesheets can provide fixes to layout bugs in IE/win. The basic idea is:

```
<head>
  <title>Conditional comments</title>
  <link rel="stylesheet" type="text/css" href="style.css">
  <!--[if IE 5]>
  <link rel="stylesheet" type="text/css" href="bugFixForIE5x.css">
  <![endif]-->
</head>
```

Appendices

Tag List

The following is a list of all elements in HTML 4, in alphabetical order. Click on a element for its description. XHTML 1.0 has the same elements but the attributes differ slightly.

- [a](#)
- [abbr](#)
- [acronym](#) – in most instances use abbr instead
- [address](#)
- ([applet](#) – deprecated, use object instead.)
- [area](#)
- [b](#) – use strong or the CSS property font-weight set to the value bold instead.
- [base](#)
- ([basefont](#) – deprecated, set the CSS property font on the body element instead.)
- [bdo](#)
- [bgsound](#) Used for inserting background sounds.
- [big](#) – the CSS property font-size set to the value larger or a percentage greater than 100% may be more appropriate.
- [blink](#) used to make the text blink (Deprecated).
- [blockquote](#)
- [body](#) Identifies the main content of a Web Page.
- [br](#) – use the p element for paragraphs. Use the CSS properties margin and padding to increase or decrease the space between paragraphs. Consider using structured elements such as lists or tables instead.
- [button](#)
- [caption](#)
- ([center](#) – deprecated, use a div element instead and set the CSS property text-align to the value center.)
- [cite](#)
- [code](#)
- [col](#)
- [colgroup](#)
- [dd](#)
- [del](#)
- [dfn](#)
- ([dir](#) – deprecated, use ul.)
- [div](#)
- [dl](#)
- [dt](#)

HTML

- [em](#)
- [fieldset](#)
- ([font](#) – deprecated, use the CSS property font. For finer control use the CSS properties font-style, font-variant, font-weight, font-size, line-height and font-family.)
- [form](#) Creates a form.
- [frame](#) Specifies information for one frame.
- [frameset](#) – avoid using frames if possible.
- [head](#) Contains information about a Web Page.
- [hr](#)
- [html](#)
- [h1](#)
- [h2](#)
- [h3](#)
- [h4](#)
- [h5](#)
- [h6](#)
- [i](#) – use em or the CSS property font-style set to the value italic instead.
- [iframe](#)
- [img](#)
- [input](#)
- [ins](#)
- ([isindex](#) – deprecated, use input.)
- [kbd](#)
- [label](#)
- [legend](#)
- [li](#)
- [link](#)
- [map](#)
- ([menu](#) – deprecated, use ul.)
- [meta](#)
- **no**[br](#) is a proprietary element type supported by some web browsers. It is used to prevent automatic wrapping of lines.
- [noframes](#)
- [noscript](#) Specifies what should be done if there is no javascript found on the browser.
- [object](#)
- [ol](#)
- [optgroup](#)
- [option](#)
- [p](#)
- [param](#)
- [pre](#)
- [q](#)

HTML

- ([s](#) – deprecated, use `del` to indicate deleted text. If the text isn't 'deleted' use the CSS property `text-decoration` set to the value `line-through`.)
- [samp](#)
- [script](#)
- [select](#)
- [small](#) – the CSS property `font-size` set to the value `smaller` or a percentage less than 100% may be more appropriate.
- [span](#)
- ([strike](#) – deprecated, use `del` to indicate deleted text. If the text isn't 'deleted' use the CSS property `text-decoration` set to the value `line-through`.)
- [strong](#)
- [style](#)
- [sub](#)
- [sup](#)
- [table](#)
- [tbody](#)
- [td](#)
- [textarea](#)
- [tfoot](#)
- [th](#)
- [thead](#)
- [title](#)
- [tr](#)
- [tt](#)
- ([u](#) – deprecated, use the CSS property `text-decoration` set to the value `underline` instead.)
- [ul](#)
- [var](#)

Standard Attributes List

Below is a list of all attributes which are available for most elements in HTML.

Attributes

class

This attribute allows you to designate an element to be a member of a given class. Multiple elements can be assigned to the same class (eg. `<p class="foo"> ... </p> <p class="foo"> ... </p>`), as well as a single element belonging to multiple classes (eg. `<p class="foo bar"> ... </p>`).

code

codebase

dir

With this attribute you can define which direction the text is written for a given element, either ltr for left-to-right or rtl for right-to-left.

height

Setting this attribute with a numerical value defines the height of an element in pixels (eg. `<div height="150"> ... </div>`)

id

This attribute allows you to define a unique identifier for each element. This would be useful for hyperlinks that link to a specific section of a page or when styling using a style sheet.

lang

With this attribute you can specify a language that is used for an element.

style

This attribute allows you to apply specific styling to a given element.

HTML

title

With this attribute you can define what will be displayed when a user hovers the element. It is not available for [base](#), [head](#), [html](#), [meta](#), [param](#), [script](#), [style](#), and [title](#).

width

Setting this attribute with a numerical value defines the width of an element in pixels (eg. `<div width="230"> ... </div>`)

More attributes

accesskey

The accesskey attribute defines a keyboard shortcut for a hyperlink or form element. The combination of keys need to activate the shortcut varies from browser to browser. In Microsoft Internet Explorer the user must press `Alt+accesskey`. If the shortcut is for a link the user must then press Enter to follow the link. The choice of `Alt+accesskey` means that access keys can clash with shortcuts built-in to the browser.

It is quite common to use numbers for the access keys since these don't clash with any major browser's built-in shortcuts, e. g.

- 1 = Home Page
- 0 = List of access keys on this website.

```
<div id="navigation">
  <h2>Navigation</h2>
  <ul>
    <li><a accesskey="1" href="/">Home page</a></li>
    <li><a accesskey="2" href="/about">About</a></li>
    <li><a accesskey="0" href="/accesskeys">Access keys</a></li>
  </ul>
</div>
```

There is no standard way to let users know the access keys that are available on the page.

tabindex

Glossary

This is a glossary of the book.

Contents [Top](#) - [0-9](#) [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#)
: [Y](#) [Z](#)

B

block element
TODO

E

element
A part of a document starting with an opening tag and ending with a closing tag, such as "`<p>keyword is important</p>`".

I

inline element
TODO

T

tag
The opening and closing sequence of characters of an element, such as "`<p>`" or "`</p>`". To be distinguished from *element*.