

Rules for Successful Site/Web Page Design

- The design of Web page is extremely important and this tutorial gives some useful pointers and 'rules' to help you to design your site or web page effectively. Design issues should be on the forefront of your mind throughout the coding process and little additions to certain codes can make design a lot more uniform.
- When designing sites or web pages you should aim for uniform style. Style in terms of web page is really to do with colour, font, and images. The idea is that visitors still know they're on your site. When they leave the first page. Also, a particular style can set your site out from others on the same topic.
- Every aspect of your page/site should reflect the goals that led you to create it in the first place. The text and images should communicate this message and so should the way you fit these elements together.
- Beware of clashing colours. Your background colour and text colour should not be too close to each other or visitors will not be able to read the text.
- General rule about fonts: only use up to maximum of three in any page or a maximum of four for the whole site. Otherwise you get a hotch-potch of visual messages and no uniformity.
- Try not to use images you've "collected" off the internet – they are rarely of the same style, size and technique and make a site look very amateur.
- Make sure a site is well organized and that there is **uniformity of implementation**. Sounds complicated but in fact it only refers to the way you show things on your web site. If you use particular graphic to represent a link or a text hyperlink to get a particular page, use that same style throughout.
- Remember site maintenance is a must. Don't code a site, upload it and expect people to keep coming back. The beauty of the internet is its immediacy – visitors expect up-to-date information and add-ons to sites and stop coming back if they don't get them. This is especially important if you have a site with links to others. Just because you don't maintain your site, doesn't mean they aren't. A broken link on your site does not look good and disappoints visitors. Check them regularly and tell people you've checked them.
- And through all coding and designing – have fun. The internet is giving you an opportunity to get published in a way that has not been possible before.

Introduction to HTML Programming

Hypertext Markup Language (HTML) – The standard text-formatting language since 1989 for documents in interconnected computing network known as the World Wide Web. HTML documents are text files that contain two parts: content that is meant to be rendered on a computer screen; and markup or tags, encoded information that directs the text format on the screen and is generally hidden from the user.

Tags – a command inserted in the document that specifies how the document, or portion of the document, should be formatted. Tags are used by all format specifications that store documents as text files. It is case-insensitive which you can type them however you want to – in CAPITAL, small letters or BoTH. Tags are also embedded where you may put them where you want an effect or action to occur.

Elements – it made up of three parts: a start tag, content, and an edge tag. Most tags possess 'closing tags' such as </BODY> which mark the place where the effect of the 'opening' tag should stop.

Attributes – a language construct that programmers use to add additional information to code elements to extend their functionalities.

HTML Document Structures

Divided into two major portions

- **Head**
 - o Contains information about the document such as its title and "meta" information.
 - o Head information is not displayed in the browser windows when page is loaded.
 - o Mainly used for search engines with page descriptions.
 - o Contents of the Head section are run first by the browser.
- **Body**
 - o Contains the actual contents of the document (the part shown in the browser window).

Note: The most important tags in HTML are tags that define headings, paragraphs and line breaks.

Head Tags

The **HEAD** elements contains information about a document, such as its title, keywords that may be useful to search engines, and other data that is not content that you want visitors to see. They are therefore not displayed.

The main tags you can use between the <HEAD> and </HEAD> tags are:

```
<TITLE> ~ </TITLE>
<META> (this tag does NOT require a closing tag)
```

The <TITLE> ~ </TITLE> tags contain the title of the document. This is not the title that is displayed on the web page, but one that is put on the top of the browser. For instance if you were put the following between the <TITLE> ~ </TITLE> tags, in Internet Explorer and Netscape Navigator (as well in other browser) you will see the title displayed at the very top.

```
<HTML>
<HEAD>
<TITLE> Welcome to HTML Basics! </TITLE>
</HEAD>
<BODY>
    Hello World!
</BODY>
</HTML>
```

The <META> tag has two main attributes: NAME="" and CONTENT="". Here's how they would look for HTML Basics:

```
<META NAME="keywords" CONTENT="HTML Programming">
```

So, putting the whole thing together looks something like this:

```
<HTML>
<HEAD>
    <TITLE> Welcome to HTML Basics! </TITLE>
    <META NAME="keywords" CONTENT="HTML Programming">
</HEAD>
<BODY>
    Hello World!
</BODY>
</HTML>
```


Body Tags

The BODY of a document contains the content of the Web page. That content can be enhanced by colouring or placing an image behind.

The **<BODY>** tag is where information relating to WHAT is shown on a Web page and the **PRESENTATION** and **POSITIONING** of that material, is stored.

The **<BACKGROUND>** allows you to use a graphic/image file for the *background* of the web page. For example, the graphic used on every page of this site is put into the code like this:

```
<HTML>
<HEAD>
  <TITLE>Welcome to HTML Basics!</TITLE>
  <META NAME="keywords" CONTENT="HTML Programming">
</HEAD>
<BODY BACKGROUND="images/back.gif" BGPARTIES="fixed">

</BODY>
</HTML>
```

The **BGPARTIES** attribute tells the browser that you do not want the image to be moved when the visitor scrolls the page up and down.

The **<BGCOLOR>** allows you to colour the background instead of putting a tiled image there. It has the benefit of being a great deal quicker to do, as the browser does not have to download an image to do it.

There are three ways of specifying colour. The following examples show how to create a red background:

1. `<BODY BGCOLOR="red">`
2. `<BODY BGCOLOR="#FF0000">`
3. `<BODY BGCOLOR=rgb(255,0,0)>`

The first uses simple or literal word commands (of which there are about 140 you can use to specify basic colours).

The second uses *hexadecimal* code to specify a colour. For instance, "#FF0000" is the same as "red". The advantage to using hexadecimal code is that there are 16,777,216 colours you can use in a browser.

The third uses RGB values (red, green and blue colour combinations), which accomplishes the same results as a hexadecimal code.

Example of setting a background colours:

```
<HTML>
<HEAD>
  <TITLE> Welcome to HTML Basics! </TITLE>
  <META NAME="keywords" CONTENT="HTML Programming">
</HEAD>
<BODY BGCOLOR="red">
</BODY>
</HTML>
```

Hyperlinks

Hyperlinks are used to navigate around a page, site or from one site to another. They are known as 'anchors' and use a specific tag: `<A attribute>~`

The **<A>** tag means 'anchor'. An anchor is a piece of text or some other object (such as an image) which marks the beginning and/or the end of a hyperlink. Hyperlinks must be closed using ``.

The `<A>` tag is used with two main attributes: `HREF=""` and `NAME=""`.

Therefore an anchor is set up like this:

```
<A HREF=" ? ">linkage text</A>
```

The **linkage text** in this instance (which can be "click here" or whatever is appropriate) becomes what is clickable to take you to wherever the link is aimed at. The question mark can either be a URL (Uniform Resource Locator or web address) both on the same site or another. It could also link a visitor to a graphic, video, sound, etc. file. For instance, the following code would create a hyperlink which would take you to "www.yahoo.com."

Click here to go to 'www.yahoo.com'!

Example:

```
<HTML>
<HEAD>
  <TITLE> Welcome to HTML Basics! </TITLE>
  <META NAME="keywords" CONTENT="HTML Programming">
</HEAD>
<BODY>
  <A HREF="http://www.yahoo.com">Click here to go to www.yahoo.com!</A>
</BODY>
</HTML>
```

Table

Not only can tables be used to present information in a chart (such as numerical data), they can be used invisibly to control the structure of your web pages.

The <TABLE> tag is remarkably easy to use, once you've got the hang of it. It begins with the <TABLE> tag, which has several attributes: ALIGN, WIDTH, BGCOLOR, BACKGROUND, CELLSPACING, CELLPADDING and BORDER.

```
<HTML>
<HEAD>
  <TITLE> Welcome to HTML Basics! </TITLE>
  <META NAME="keywords" CONTENT="HTML Programming">
</HEAD>
<BODY>
  <TABLE ALIGN="center" WIDTH=200 BORDER=3 CELLPADDING=1 CELLSPACING=1>
    <TR><TD WIDTH=50%>A</TD><TD WIDTH=50%>B</TD></TR>
    <TR><TD WIDTH=50%>C</TD><TD WIDTH=50%>D</TD></TR>
    <TR><TD WIDTH=50%>E</TD><TD WIDTH=50%>F</TD></TR>
  </TABLE>
</BODY>
</HTML>
```

Text Tags and Line Breaks

When placing text on Web page, you have to remember to 'say what you need to say in few words possible'. It is well known fact that it is more difficult to read text on a monitor than from paper, and therefore visitors to your site are less inclined to read through huge amount of text. Once you have done that, it is good to break up text and emphasize it in different ways.

Basic HTML Tags

Tag	Description
<html>	Defines an HTML document
<body>	Defines the document's body
<h1> to <h6>	Defines header 1 to header 6
<p>	Defines a paragraph
 	Inserts a single line break
<hr>	Defines a horizontal line
<!-->	Defines a comment

Text Formatting Tags

Tag	Description
	Defines bold text
<big>	Defines big text
	Defines emphasized text
<i>	Defines italic text
<small>	Defines small text
	Defines strong text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
	Defines deleted text

Computer Output Tags

Tag	Description
<code>	Defines computer code text
<kbd>	Defines keyboard text
<samp>	Defines sample computer codes
<tt>	Defines teletype text
<var>	Defines a variable
<pre>	Defines preformatted text

Citations, Quotations, and Definition Tags

Tag	Description
<abbr>	Defines an abbreviation
<acronym>	Defines an acronym
<address>	Defines an address element
<bdo>	Defines the text direction
<blockquote>	Defines a long quotation
<q>	Defines a short quotation
<cite>	Defines a citation
<dfn>	Defines a definition term

Headings

Headings are defined with the <h1> to <h6> tags. <h1> defines the largest heading. <h6> defines the smallest heading.

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
<h4>This is a heading</h4>
<h5>This is a heading</h5>
<h6>This is a heading</h6>
```

HTML automatically adds an extra blank line before and after a heading.

Paragraphs

Paragraphs are defined with the <p> tag.

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

HTML automatically adds an extra blank line before and after a paragraph.

Line Breaks

The
 tag is used when you want to end a line, but don't want to start a new paragraph. The
 tag forces a line break wherever you place it.

```
<p>This <br> is a paragraph with line breaks</p>
```

The
 tag is an empty tag. It has no closing tag.

Comments in HTML

The comment tag is used to insert a comment in the HTML source code. A comment will be ignored by the browser. You can use comments to explain your code, which can help you when you edit the source code at a later date.

```
<!-- This is a comment... -->
```

Note that you need an exclamation point after the opening bracket, but not before the closing bracket.

Basic Notes – Useful Tips

When you write HTML text, you can never be sure how the text is displayed in another browser. Some people have large computer displays, some have small. The text will be reformatted every time the user resizes his window. Never try to format the text in your editor by adding empty lines and spaces to the text.

HTML will truncate the spaces in your text. Any number of spaces count as one. Some extra information: In HTML a new line counts as one space.

Using empty paragraphs `<p>` to insert blank lines is a bad habit. Use the `
` tag instead. (But don't use the `
` tag to create lists. Wait until you have learned about HTML lists.)

You might have noticed that paragraphs can be written without the closing tag `</p>`. Don't rely on it. The next version of HTML will not allow you to skip ANY closing tags.

HTML automatically adds an extra blank line before and after some elements, like before and after a paragraph, and before and after a heading.

Use a horizontal rule (the `<hr>` tag), to separate the sections in our tutorials.

Character Entities

Some characters have a special meaning in HTML, like the less than sign (`<`) that defines the start of an HTML tag. If we want the browser to actually display these characters we must insert character entities in the HTML source.

A character entity has three parts: an ampersand (`&`), an entity name or a `#` and an entity number, and finally a semicolon (`;`).

To display a less than sign in an HTML document we must write: `<` or `<`;

The advantage of using a name instead of a number is that a name is easier to remember. The disadvantage is that not all browsers support the newest entity names, while the support for entity numbers is very good in almost all browsers.

Note that the entities are case sensitive.

Non-breaking Space

The most common character entity in HTML is the non-breaking space.

Normally HTML will truncate spaces in your text. If you write 10 spaces in your text HTML will remove 9 of them. To add spaces to your text, use the ` ` character entity.

The Most Common Character Entities:

Result	Description	Entity Name
	Non-breaking space	<code>&nbsp;</code> ;
<code><</code>	Less than	<code>&lt;</code> ;
<code>></code>	Greater than	<code>&gt;</code> ;
<code>&</code>	Ampersand	<code>&amp;</code> ;
<code>"</code>	Quotation mark	<code>&quot;</code> ;
<code>'</code>	Apostrophe	<code>&apos;</code> <i>does not work in IE</i>

Some Other Commonly Used Character Entities

Result	Description	Entity Name
<code>¢</code>	Cent	<code>&cent;</code> ;
<code>£</code>	Pound	<code>&pound;</code> ;
<code>¥</code>	Yen	<code>&yen;</code> ;
<code>€</code>	Euro	<code>&euro;</code> ;
<code>§</code>	Section	<code>&sect;</code> ;
<code>©</code>	Copyright	<code>&copy;</code> ;
<code>®</code>	Registered	<code>&reg;</code> ;
<code>×</code>	Multiplication	<code>&times;</code> ;
<code>÷</code>	Division	<code>&divide;</code> ;

The Anchor Tag and the Href Attributes (Hyperlink)

HTML uses the `<a>` (anchor) tag to create a link to another document.

An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc.

The syntax of creating an anchor:

```
<a href="url">Text to be displayed</a>
```


The <a> tag is used to create an anchor to link from, the href attribute is used to address the document to link to, and the words between the open and close of the anchor tag will be displayed as a hyperlink. This anchor defines a link to <http://www.yahoo.com>:

```
<a href="http://www.yahoo.com/">Visit Yahoo Web Site!</a>
```

The line above will look like this in a browser:

[Visit Yahoo Web Site!](http://www.yahoo.com/)

The Target Attribute

With the target attribute, you can define **where** the linked document will be opened.

The line below will open the document in a new browser window:

```
<a href="http://www.yahoo.com/" target="_blank">Visit Yahoo Web Site!</a>
```

The Anchor Tag and the Name Attribute

The name attribute is used to create a named anchor. When using named anchors we can create links that can jump directly into a specific section on a page, instead of letting the user scroll around to find what he/she is looking for.

Below is the syntax of named anchor:

```
<a name="label">Text to be displayed</a>
```

You should notice that a named anchor is not displayed in a special way.

To link directly to the "tips" section, add a # sign and the name of the anchor to the end of a URL, like this:

```
<a href="http://www.w3schools.com/html_links.asp#tips">Jump to the Useful Tips Section</a>
```

A hyperlink to the Useful Tips Section from WITHIN the file "html_links.asp" will look like this:

```
<a href="#tips">Jump to the Useful Tips Section</a>
```

Basic Notes – Useful Tips

Always add a trailing slash to subfolder references. If you link like this:

```
href="http://www.w3schools.com/html",
```

you will generate two HTTP requests to the server, because the server will add a slash to the address and create a new request like this:

```
href="http://www.w3schools.com/html/"
```

Named anchors are often used to create "table of contents" at the beginning of a large document. Each chapter within the document is given a named anchor, and links to each of these anchors are put at the top of the document.

If a browser cannot find a named anchor that has been specified, it goes to the top of the document. No error occurs.

Tag	Description
<a>	Defines an anchor

Frames

With frames, you can display more than one HTML document in the same browser window. Each HTML document is called a frame, and each frame is independent of the others.

The disadvantages of using frames are:

- The web developer must keep track of more HTML documents
- It is difficult to print the entire page

The Frameset Tag

- The <frameset> tag defines how to divide the window into frames
- Each frameset defines a set of rows **or** columns
- The values of the rows/columns indicate the amount of screen area each row/column will occupy
- The <frame> tag defines what HTML document to put into each frame

In the example below we have a frameset with two columns. The first column is set to 25% of the width of the browser window. The second column is set to 75% of the width of the browser window. The HTML document "frame_a.htm" is put into the first column, and the HTML document "frame_b.htm" is put into the second column:

```
<frameset cols="25%,75%">
  <frame src="frame_a.htm">
  <frame src="frame_b.htm">
</frameset>
```

Note: The frameset column size value can also be set in pixels (cols="200,500"), and one of the columns can be set to use the remaining space (cols="25%,").

Basic Notes - Useful Tips

If a frame has visible borders, the user can resize it by dragging the border. To prevent a user from doing this, you can add noresize="noresize" to the <frame> tag.

Add the <noframes> tag for browsers that do not support frames.

Important: You cannot use the <body></body> tags together with the <frameset></frameset> tags. However, if you add a <noframes> tag containing some text for browsers that do not support frames, you will have to enclose the text in <body></body> tags! See how it is done in the first example below.

Frame Tags

Tag	Description
<frameset>	Defines a set of frames
<frame>	Defines a sub window (a frame)
<noframes>	Defines a noframe section for browsers that do not handle frames
<iframe>	Defines an inline sub window (frame)

Tables (Continued)

Tables are defined with the <table> tag. A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). The letters td stands for "table data," which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

How it looks in a browser:

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

Tables and the Borders Attribute

If you do not specify a border attribute the table will be displayed without any borders. Sometimes this can be useful, but most of the time, you want the borders to show.

To display a table with borders, you will have to use the border attribute:


```

<table border="1">
<tr>
<td>Row 1, cell 1</td>
<td>Row 1, cell 2</td>
</tr>
</table>

```

Headings in a Table

```

<table border="1">
<tr>
<th>Heading</th>
<th>Another Heading</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>

```

How it looks in a browser:

Heading	Another Heading
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

Empty Cells in a Table

Table cells with no content are not displayed very well in most browsers.

```

<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td></td>
</tr>
</table>

```

How it looks in a browser:

row 1, cell 1	row 1, cell 2
row 2, cell 1	

Note that the borders around the empty table cell are missing (Mozilla Firefox displays the border).

To avoid this, add a non-breaking space () to empty data cells, to make the borders visible:

```

<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>&nbsp;</td>
</tr>
</table>

```

How it looks in a browser:

row 1, cell 1	row 1, cell 2
row 2, cell 1	

Basic Notes – Useful Tips

The `<thead>`, `<tbody>` and `<tfoot>` elements are seldom used, because of bad browser support. Expect this to change in future versions of XHTML.

Tag	Description
<code><table></code>	Defines a table
<code><th></code>	Defines a table header
<code><tr></code>	Defines a table row
<code><td></code>	Defines a table cell
<code><caption></code>	Defines a table caption
<code><colgroup></code>	Defines group of table columns
<code><col></code>	Defines the attribute values for one or more columns in a table
<code><thead></code>	Defines a table head
<code><tbody></code>	Defines table body
<code><tfoot></code>	Defines a table footer

HTML List

HTML supports ordered, unordered and definition lists.

Unordered List

An unordered list is a list of items. The list items are marked with bullets (typically small black circles).

An unordered list starts with the `` tag. Each list item starts with the `` tag.

```
<ul>
<li>Coffee</li>
<li>Milk</li>
</ul>
```

Here is how it looks in a browser:

- Coffee
- Milk

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

Ordered Lists

An ordered list is also a list of items. The list items are marked with numbers.

An ordered list starts with the `` tag. Each list item starts with the `` tag.

```
<ol>
<li>Coffee</li>
<li>Milk</li>
</ol>
```

Here is how it looks in a browser:

1. Coffee
2. Milk

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

Definition Lists

A definition list is **not** a list of items. This is a list of terms and explanation of the terms.

A definition list starts with the `<dl>` tag. Each definition-list term starts with the `<dt>` tag. Each definition-list definition starts with the `<dd>` tag.

```
<dl>
<dt>Coffee</dt>
```



```

<dd>Black hot drink</dd>
<dt>Milk</dt>
<dd>White cold drink</dd>
</dl>

```

Here is how it looks in a browser:

```

Coffee
    Black hot drink
Milk
    White cold drink

```

Inside a definition-list definition (the <dd> tag) you can put paragraphs, line breaks, images, links, other lists, etc.

List Tags

Tag	Description
	Defines an ordered list
	Defines an unordered list
	Defines a list item
<dl>	Defines a definition list
<dt>	Defines a definition term
<dd>	Defines a definition description

HTML Forms and Input

HTML Forms are used to select different kinds of user input.

Forms

A form is an area that can contain form elements.

Form elements are elements that allow the user to enter information (like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.) in a form.

A form is defined with the <form> tag.

```

<form>
  <input>
  <input>
</form>

```

Input

The most used form tag is the <input> tag. The type of input is specified with the type attribute. The most commonly used input types are explained below.

Text Fields

Text fields are used when you want the user to type letters, numbers, etc. in a form.

```

<form>
  First name:
  <input type="text" name="firstname">
  <br>
  Last name:
  <input type="text" name="lastname">
</form>

```

How it looks in a browser:

```

First name: 
Last name: 

```

Note that the form itself is not visible. Also note that in most browsers, the width of the text field is 20 characters by default.

Radio Buttons

Radio Buttons are used when you want the user to select one of a limited number of choices.

```
<form>
<input type="radio" name="sex" value="male"> Male
<br>
<input type="radio" name="sex" value="female"> Female
</form>
```

How it looks in a browser:

☐ Male
☐ Female

Note that only one option can be chosen.

Checkboxes

Checkboxes are used when you want the user to select one or more options of a limited number of choices.

```
<form>
I have a bike:
<input type="checkbox" name="vehicle" value="Bike" />
<br />
I have a car:
<input type="checkbox" name="vehicle" value="Car" />
<br />
I have an airplane:
<input type="checkbox" name="vehicle" value="Airplane" />
</form>
```

How it looks in a browser:

I have a bike: ☐
I have a car: ☐
I have an airplane: ☐

The Form's Action Attribute and the Submit Button

When the user clicks on the "Submit" button, the content of the form is sent to another file. The form's action attribute defines the name of the file to send the content to. The file defined in the action attribute usually does something with the received input.

```
<form name="input" action="html_form_action.asp" method="get">
Username:
<input type="text" name="user">
<input type="submit" value="Submit">
</form>
```

How it looks in a browser:

Username:

If you type some characters in the text field above, and click the "Submit" button, you will send your input to a page called "html_form_action.asp". That page will show you the received input.

Form Tags

Tag	Description
<form>	Defines a form for user input
<input>	Defines an input field
<textarea>	Defines a text-area (a multi-line text input control)
<label>	Defines a label to a control
<fieldset>	Defines a field set
<legend>	Defines a caption for a fieldset
<select>	Defines a selectable list (a drop-down box)
<optgroup>	Defines an option group
<option>	Defines an option in the drop-down box
<button>	Defines a push button

