

HTML Basics

What is HTML?

HTML stands for Hyper Text Markup Language. All web pages are delivered to the user's browser as HTML code (though there may be many other technologies involved, such as CSS, JavaScript, flash and PHP, to name a few). You can view the HTML code of a web page by right-clicking in the page and choosing 'View Source'.

HTML code consists of plain, unformatted text containing **markup tags** such as `<p>` and `<h1>`. These tags define the **structure** of the page content, which tells the Web browser how to display the page.

An HTML file must be saved with an `htm` or `html` file extension to be viewable in a web browser.

Web authoring programs such as Dreamweaver can save a lot of work writing HTML, however an HTML file can be created using a simple text editor like Notepad. There are also lots of free HTML editors. HTML-kit (www.html-kit.com) is a good one, but if you google for 'free html editors' you can find plenty more.

HTML tags

HTML tags are used to mark up the different **elements** of a web page, such as headings, paragraphs and links. All HTML tags are enclosed by the two characters `<` and `>`, called angle brackets.

Most tags are written in pairs: the first tag in the pair is called the **start**, or **opening** tag, the second tag is called the **end**, or **closing** tag. The content of the element (which can be text and/or other elements) goes between the opening and closing tags. This, for example, defines a paragraph:

```
<p>A paragraph</p>
```

Some HTML tags do not have a closing tag. In this case, the single tag is closed by adding a space and a `/` before the closing `>`, thus:

```
<br />
```

(NB. this applies only to XHTML - single tags in HTML 4 close without the space or slash, thus: `
`)

HTML attributes

Attributes can be added to HTML tags to provide additional information for the element. For example, for the `` tag, which defines an image, the `src` attribute tells the browser where the target image is located.

Attributes must be specified in the opening tag of an HTML element, and are written in **name/value** pairs, with the value enclosed in quotes, like so: `name="value"`.

You can put as many attributes into an element as you like, provided that the attribute is allowed for that element. The attributes can be specified in any order.

For example:

```
<p align="center" title="My Paragraph">A paragraph</p>
```

Naming conventions

When writing html documents:

- all tags and attribute names must be written in **lowercase**, ie `<h1>` rather than `<H1>` (attribute values can be written in uppercase - see the example above)
- all files should be named in **lowercase**
- file names should only contain **letters**, **numbers**, and **hyphens** or **underscores** - spaces are not allowed. This also applies to any other files included in the page, such as image files.

Required Tags

These tags must be included in any HTML document in order for it to be valid HTML. They define the basic structure of a web page.

<!DOCTYPE>

This goes at the very start of an html document; it tells the browser which version of html the document is written in. For **XHTML 1.0 Transitional**, the doctype is:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

<html>

This element tells the browser that this is an HTML document. The start tag follows the doctype, and the end tag goes right at the end of the document.

<head>

The head element can contain information about the document. The browser does not display this information to the user (apart from the contents of the <title> tag). This element goes immediately after the <html> start tag.

<title>

This element defines the title of the document. It displays in the title bar at the top of the browser.

<body>

The body element contains all the contents of the document (text, images, colours etc). Only content within this element will display in the browser window. The body element goes just after the head element.

Note that each document must have one head and one body, in that order; everything else in the page must be contained within those two tags.

Example:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <title>My Home Page</title>
  </head>
  <body>
    Contents of my home page...
  </body>
</html>
```

Common Formatting Tags

Now you know how to build the basic structure of a web page, you can try out some of the tags which are most commonly used to add content to pages. These tags can be placed anywhere within the `<body>` tag. See www.sjhwebdesigns.com/resources/htmltutorial/commonformattingtags/ for examples.

`<h1>` to `<h6>`

The `<h1>` to `<h6>` tags define headings. `<h1>` is the largest heading; `<h6>` is the smallest heading.

`<p>`

The `<p>` tag defines a paragraph. A browser will usually add a small amount of space before and after a paragraph.

`
`

The `
` tag inserts a single line break.

The `
` tag is an empty tag - it has no end tag, so `
</br>` is wrong.

``

The `img` element defines an image.

The `src` attribute tells the browser where to find the image. The `alt` attribute defines alternate text to display if the image can't be shown.

Like `
`, `` is a single tag, with no closing tag.

``

Renders as emphasized text - usually *italicised*.

``

Renders as strong emphasized text - usually **bold**.

`<!-- ... -->`

The `<!-- ... -->` tag adds a comment in the html code. Everything inside the comment tags will be ignored by the browser. You can use comments to explain your code, which can help you when you edit the source code later.

Nesting & Element Types

Nesting tags

You can put HTML elements within other elements, like this:

```
<p><strong>A well coded paragraph</strong></p>
```

Note that the tags are closed in the reverse order of the opening tags. This means that the tags are correctly nested, that is, the inner element is contained by the outer element, without overlap; by contrast, here is the wrong way:

```
<p><strong>A badly coded paragraph</p></strong>
```

However, you can't just put any tag inside any other tag! For example, we can have `` inside `<p>` as above, but not `<p>` inside ``! To understand what can go inside what, you need to understand the difference between...

Block and Inline Elements

HTML elements come in two distinct types: **block** and **inline**.

Block elements:

- take a whole line to themselves, and usually have a vertical margin above and below
- can contain other block elements as well as inline elements *
- Block elements include `<h1>`, `<p>`, ``, ``.

* Alas, there are a couple of exceptions to this rule: paragraphs and headings can only contain inline elements. This might seem like it's getting a tad complicated, but it's fairly logical really - a paragraph within a heading, or a heading within another heading, doesn't really make much sense.

Inline elements:

- do not take up a whole line - only the amount of space the element content occupies
- can only contain other inline elements
- Examples of inline elements: `<a>`, ``, ``.

Strictly speaking, inline elements should be placed within other block elements, not directly inside the `<body>` tag. The most important thing to remember, though, is not to nest block inside inline.

Don't pay too much mind to the difference in **appearance** between block and inline elements; this can easily be changed using CSS.

Presentational HTML

There are several attributes within html that allow you to control the appearance of your page, such as fonts, colours and layout. All of these attributes are now **deprecated** (meaning that their use is discouraged by the [World Wide Web Consortium](#)); it is better to use Cascading Style Sheets for presentational effects as they offer far more precise control, increase access to users with disabilities by separating **content** from **presentation**, and can be achieved with much less coding.

However, the [basic EDCC assessment](#) requires making a couple of minor presentational changes to a page. So, in order that we may complete this first assessment before starting into CSS, I grudgingly offer to you this small collection of presentational attributes:

To change the background colour of a web page:

This can be achieved by adding the `bgcolor` attribute to the `<body>` tag. Values can be named colours, eg red, green, brown. For a full list of colour names, see www.w3schools.com/html/html_colornames.asp. However, you can have a much wider range of colours at your disposal if you use [hexadecimal colour codes](#).

For example, `<body bgcolor="#ff0000">` sets the page background to a brilliant red. Yikes!

To change the text colour of a web page:

Add the `text` attribute to the `<body>` tag. Allowed values are the same as for the `bgcolor` attribute. For example, `<body bgcolor="#ff0000" text="#00ff00">` sets the text colour to bright green, which is positively headache-inducing over the red background. Please choose a more tasteful (and legible) colour combination.

To align text within a heading or paragraph:

Add the `align` attribute to the relevant tag. Allowed values are **left** (which is the default value), **right**, **center** (note the American spelling) and **justify** (all full lines will be aligned to the left and right edges of the page).