



Higham and Rushden

Registered Charity Number 1164952

Introduction to HTML

(creating web pages)

Open Notepad (in Start menu under Windows Accessories).

In the document that opens, type the following:

```
<!DOCTYPE html>  
<html>  
<body>  
<p>This is my first web page</p>  
</body>  
</html>
```

Now go to File/Save as and give the file an appropriate name such as mywebpage.html, ensuring that you overwrite the default .txt file ending to .htm or .html.

Open File Explorer, navigate to the folder where you saved that file, and double-click it.

What just happened?

If everything went according to plan, the file should have opened in your default web browser (Edge, Internet Explorer, Mozilla Firefox, Safari, Opera or Google Chrome). You have created a web page.

During the rest of this presentation you will be creating increasingly detailed web pages, and it is important that you pay attention and type exactly what is shown on the slide.

From here on please remember that you will need to open your web page in 2 different ways, as follows:

- To edit the content find the file in File Explorer, right-click and choose Open with Notepad.
- To view your web page double-click in File Explorer to open in your default browser.

We have now defined some fundamental elements of a web page.

The first line on the top, `<!DOCTYPE html>`, is a document type declaration and it lets the browser know which flavour of HTML you're using. It's important to stick this in - if you don't, browsers will assume you don't really know what you're doing and can sometimes act in a very peculiar way.

`<html>` is the opening **tag** that kicks things off and tells the browser that everything between that and the `</html>` closing tag is an HTML document. The stuff between the first `<body>` tag and the second (but note the difference) `</body>` tag is the main content of the document that will appear in the browser window.

Not all tags have closing tags like this (`<html></html>`): some tags, which do not wrap around content will close themselves. The line-break tag for example, looks like this : `
` - a line break doesn't hold any content so the tag sits by itself. All you need to remember is that all tags with content between them should be closed, in the format of opening tag , content, closing tag. You might come across "self-closing" tags, whereby a br tag, for example, will look like "`
`" instead of simply "`
`". Some developers prefer one way, some prefer the other, but both work equally well.

All HTML pages should have a page title.

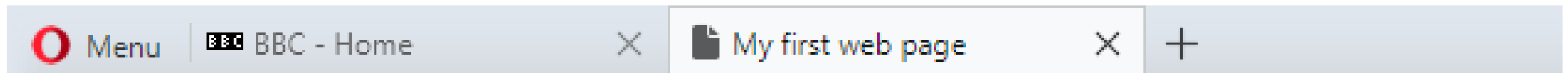
To add a title to your page, open it again in Notepad and change your code so that it looks like this:

```
<!DOCTYPE html>
<html>
<head>
  <title>My first web page</title>
</head>
<body>
<p> This is my first web page</p>
</body>
</html>
```

We have added two new elements here, that start with the head tag and the title tag (and notice how both of these close).

The head element (which starts with the <head> opening tag and ends with the </head> closing tag) appears before the body element (starting with <body> and ending with </body>) and contains information about the page. The information in the head element does not appear in the browser window.

If you look at this document in the browser (save the document again, go back to the web browser and reload the page), you will see that “My first web page” will appear on a tab or the title bar of the window (not the actual canvas area).



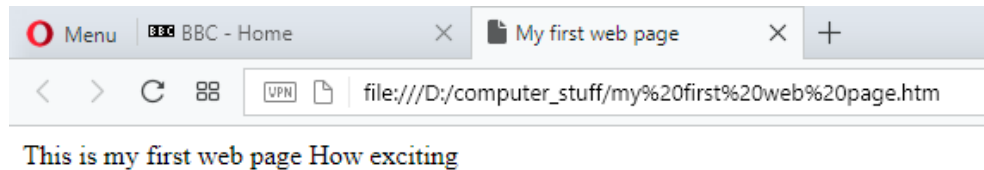
The text that you put in between the title tags has become the title of the document.

Now that you have the basic structure of an HTML document, you can mess around with the content a bit.

Go back to your document in Notepad and add another line to your page:

Save it again and look at it in your browser.

You might have expected your document to appear as you typed it, on two lines, but instead you should see something like this:



```
<!DOCTYPE html>
<html>
<head>
  <title>My first web page</title>
</head>
<body>
  <p> This is my first web page</p>
    How exciting
</body>
</html>
```

This is because web browsers don't usually take any notice of what line your code is on. They also don't take any notice of spaces (you would get the same result if you typed "This is my first web page How exciting").

If you want text to appear on different lines or, rather, if you intend there to be two distinct blocks of text (because, remember, HTML is about meaning, not presentation), you need to explicitly state that.

Change your two lines of content so that they look like this:

```
<p>This is my first web page</p>
<p>How exciting</p>
```

Look at the results of this: save your document and open it in your browser. The two lines will now appear on two lines because the browser recognizes them as separate paragraphs. The <p> tag is used for paragraphs, and this is a good demonstration of how each paragraph <p> tag needs to be closed by a closing </p> tag.

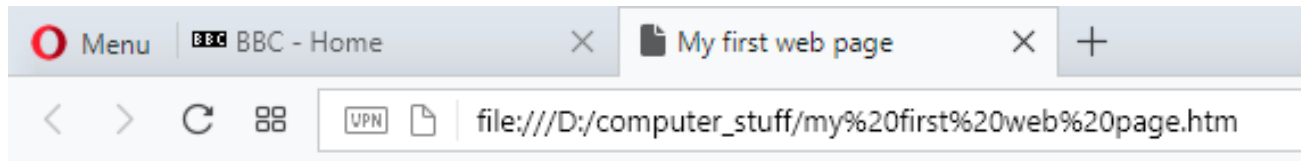
Emphasis

You can emphasise text in a paragraph using `em` (emphasis) and `strong` (strong importance).

`<p>Yes, that really is exciting.`

`Warning: But please don't get over-excited.</p>`

Traditionally, browsers will display `em` in italics and `strong` in bold by default but they are not the same as the old `<i>` and `` tags which have their origins in italic and bold. Remember - HTML isn't for presentation.



Line breaks

The line-break tag can also be used to separate lines like this:

`<p>This is my first web page
`

`How exciting</p>`

There's no content involved in breaking lines so there is no closing tag.

It could be tempting to over-use line breaks and `
` shouldn't be used if two blocks of text are intended to be separate from one another (because if that's what you want to do you probably want the `<p>` and `</p>` tags).

Headings

The <p> and </p> tags are just the start of text formatting.

If you have documents with genuine headings, then there are HTML tags specifically designed just for them.

They are h1, h2, h3, h4, h5 and h6, h1 being the emperor of headings and h6 being the lowest serf.

Change your code to the following:

Now save your document and open or refresh it in your browser.

Note that the h1 tag is only used once, as the main heading of the page. h2 to h6, however, can be used as often as desired, but they should always be used in order, as they were intended. For example, an h4 should be a sub-heading of an h3, which should be a sub-heading of an h2.

```
<!DOCTYPE html>
<html>
<head>
  <title>My first web page</title>
</head>
<body>
  <h1>My first web page</h1>
  <h2>What this is</h2>
  <p>A simple page put together using HTML</p>
  <h2>Why this is</h2>
  <p>To learn HTML</p>
</body>
</html>
```

Lists

Unordered lists and ordered lists work the same way, except that the former is used for non-sequential lists with list items usually preceded by bullets and the latter is for sequential lists, which are normally represented by incremental numbers.

The `ul` tag is used to define unordered lists and the `ol` tag is used to define ordered lists. Inside the lists, the `li` tag is used to define each list item. Let's have a look at that in practice. Change your document again by using the code on the right and have a look at it in your browser to see that this code creates a list preceded by bullet points. If you prefer a numbered list simply change `` and `` to `` and ``.

```
<!DOCTYPE html>
<html>
<head>
  <title>My first web page</title>
</head>
<body>
  <h1>My first web page</h1>
  <h2>What this is</h2>
  <p>A simple page put together using HTML</p>
  <h2>Why this is</h2>
  <ul>
    <li>To learn HTML</li>
    <li>To show off</li>
    <li>Because I feel like it.</li>
  </ul>
</body>
</html>
```


Lists can also be included in lists to form a structured hierarchy of items.

Replace your list code with the code on the right and have a look at the result.

You should now have a list within a list. You could put another list within that, and another within that, and so on and so forth ad nauseam.

```
<ul>
  <li>To learn HTML</li>
  <li>
    To show off
    <ol>
      <li>To my boss</li>
      <li>To my friends</li>
      <li>To my cat</li>
    </ol>
  </li>
  <li>Because I want to.</li>
</ul>
```

Links

So far you've been making a stand-alone web page, which is all very well and good, but what makes the Internet so special is that it all links together.

The "H" and "T" in "HTML" stand for "hypertext", which basically means a system of linked text.

An anchor tag (a) is used to define a link, but you also need to add something to the anchor tag — the destination of the link.

Use the code on the right to create a simple hyperlink. Save the document and open or refresh it in your browser, and the last line of text should now be in blue and underlined. If you click on it, you should also be taken to the external website if you are online.

The destination of the link is defined in the href attribute of the tag.

```
<!DOCTYPE html>
<html>
<head>
  <title>My first web page</title>
</head>
<body>
  <h1>My first web page</h1>
  <h2>What this is</h2>
  <p>A simple page put together using HTML</p>
  <h2>Why this is</h2>
  <p>To learn HTML</p>
  <h2>Where to find the tutorial</h2>
  <p><a href="http://www.htmldog.com">HTML Dog</a></p>
</body>
</html>
```

Images

Things might seem a little bland and boring with all of this text formatting. Of course, the web is not just about text, it is a multi-media extravaganza and the most common form of sparkle is the image.

The `img` tag is used to put an image in an HTML document and it looks like this:

```

```

The `src` attribute tells the browser where to find the image.

The `width` and `height` attributes are necessary because if they are excluded, the browser will tend to calculate the size as the image loads, instead of when the page loads, which means that the layout of the document may jump around while the page is loading.

The `alt` attribute is the alternative description. This is an accessibility consideration, providing meaningful information for users who are unable to see the image (if they are visually impaired, for example).

Note that, like the `
` tag, because the `` element does not enclose any content, no closing tag is required.

The construction of images for the web is a little outside of the remit of this tutorial, but it is worth noting a few things...

The most commonly used file formats used for images are JPEGs, GIFs, and PNGs. They are compressed formats, and have very different uses.

A JPEG uses a mathematical algorithm to compress the image and will distort the original slightly. The lower the compression, the higher the file size, but the clearer the image. JPEGs are typically used for images such as photographs.

A GIF can have no more than 256 colours, but they maintain the colours of the original image. The lower the number of colours you have in the image, the lower the file size will be. GIFs also allow any pixel in the image to be transparent. GIFs are typically used for images with solid colours, such as icons or logos.

A PNG (pronounced “ping”) replicates colours, much like a GIF, but allows 16 million colours as well as alpha transparency (that is, an area could be 50% transparent). PNGs are typically used for versatile images in more complex designs BUT they are not fully supported by some older browsers.

The web is forever getting faster and faster but you obviously want your web pages to download as quickly as possible. Using super-high resolution images isn't doing your, or your user's, bandwidth (or patience) any favours. Image compression is a great tool and you need to strike a balance between image quality and image size. Most modern image manipulation programs allow you to compress images and the best way to figure out for yourself what works best is trial and error.

To see real-life examples of HTML, open any web page in your browser, right-click, and select “View Source”(or equivalent wording, according to the browser you are using). An excellent page to use is the HTML Dog website, the source of much of the information in this tutorial and a highly recommended resource if you want to learn more about HTML and CSS. If you look at the source of <http://htmldog.com/guides/html/beginner/forms/> you will see that the top of the page looks like this:

```
1 <!DOCTYPE html>
2
3 <html dir="ltr" lang="en">
4 <head>
5   <meta charset="utf-8">
6   <title>Forms | HTML Dog</title>
7
8 <!--[if !(lt IE 9)]><!-->
9
10   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
11
12   <link href="/rAF/arcticFox.css" rel="stylesheet">
13
14
15
16   <link href="https://fonts.googleapis.com/css?family=Lato:300,400,700|Source+Code+Pro:300,400,500" rel="stylesheet">
17
18 <!--<![endif]>-->
19
20 <!--[if lt IE 9]><script src="/rAF/html5shiv.js"></script><link href="/rAF/altBasic.css" rel="stylesheet"><![endif]>-->
21
22
23
24 <link rel="icon" type="image/png" href="/favicon-32x32.png" sizes="32x32">
25 <link rel="icon" type="image/png" href="/android-chrome-192x192.png" sizes="192x192">
26 <link rel="icon" type="image/png" href="/favicon-96x96.png" sizes="96x96">
27 <link rel="icon" type="image/png" href="/favicon-16x16.png" sizes="16x16">
28 <link rel="manifest" href="/manifest.json">
29 <link rel="mask-icon" href="/safari-pinned-tab.svg">
30 <meta name="msapplication-TileColor" content="#2d89ef">
31 <meta name="msapplication-TileImage" content="/mstile-144x144.png">
32 <meta name="theme-color" content="#0088ee">
33
34 </head>
35
36
37
38
39
40 <body>
```

You can see that it starts with the `<!DOCTYPE html>` declaration followed by the `<head>` `</head>` section.

Further down the page is a line exactly like those we have already seen:

`<p>Forms` are used to collect data inputted by a user. They can be used as an interface for a web application, for example, or to send data across the web.`</p>`

Everything we have done so far will appear in your browser in the default font used by your browser, rather than the font used when you created it in Notepad. In order to change the font we need to style the html and the modern way to do this is to use cascading style sheets, normally shortened to CSS. This is outwith the scope of this tutorial, but your browser will almost certainly still recognise the old-fashioned way of changing individual bits of text. If you insert the code below between the <body> tags you should see how it works.

```
<p><font color="red" face="Verdana, Geneva, sans-serif" size="100%">Red text!</p>
```

Obviously you can change any of the values in parentheses, so please feel free to play around.

Other tags you can use are for strikethrough, for bold text, for italics, for superscript and for subscript. Don't forget that the text you want to format goes between the opening tag and the closing tag (with the/).

If you do not close a formatting tag then that format will continue to the end of the page.

Acceptable colour names (but when coding don't forget to spell it "color") can be found at:
https://www.w3schools.com/colors/colors_names.asp

Should you ever decide to create your own website, you may find it easier to use website creation software. These are often called WYSIWYG programs (What-You-See-Is-What-You-Get) and translate plain text into html. They can be standalone products or built into website builders (some of which are called CMS – content management systems).

Clearly this is much easier than learning html coding (and CSS!) but the process is slightly more restrictive and the finished website can be much larger in size which may be a problem if your hosting plan is small. Wix (www.wix.com) is one such online website builder which offers free hosting in return for adverts on your site and may be worth investigating if you are thinking of creating a small website for a group to which you belong, or even if you just want to publish some photographs for the benefit of extended family.

Finally let me shamelessly plug some of my own websites, as follows:

- <https://jimandmaggie.co.uk/> for some photos of Scotland and slideshows of our US road trips
- <https://woodworkprojects.co.uk/> for some fantastic Japanese saws
- <https://thebestidea.website/> for my e-book bout some of America's less well-known National Parks.

THE END

**A pdf version of this presentation is available at:
<https://www.highamandrushdenu3a.org.uk/groups/tutorials.html>**