the centre for computing history

Build Your Own Website

HyperText Markup Language (HTML) was invented in the early 90's by Tim Berners-Lee, and today is used on pretty much every single webpage on the internet. It is not a programming language, but rather a markup language. As it separates the content of a page from the annotations which describe it, providing structure.

The webpages as we know it are split up into three parts: Structure and Content, Presentation, and Behaviour. Different languages cover different areas, and structure and content is covered by HTML.

For this worksheet, we're going to be using Notepad++ as our file editor. You can download Notepad++ here https://notepad-plus-plus.org/. Notepad++ is completely free, and can be used for pretty much any programming language. If you're using a different operating system, feel free to use any text editor you can find, they should all work with this guide!

First open notepad++ on the desktop. Once it's open, it should automatically bring up a new document. If not go to File > New or click on the new icon just underneath it.

Now just type this into the text editor:

This is a really awesome first webpage!

Then go to File > Save or just hit the little floppy disk save icon, call the file whatever you want, then post-fix it with .html which is the extension for html files. The operating system should recognise it immediately. Just double click on it on the desktop and there you go, your first webpage in an internet browser! It may simply be a sentence, but it is content which if uploaded to the internet, anyone can view.

Tags are what we use to give our content some meaning. By default, web browsers have a lot of different tags which you can use to apply meaning, but there are some tags which must be used. Tags are structured with <> and the bit in brackets define what sort of tag it is. They also always come in pairs, so for every starting tag you have, there must also be the same ending tag, which is the exact same just it starts with </ instead of just <.

The tag names are not case sensitive, do you can written <BODY>, <body> or <BoDy> and any variation between.

Going back to the text editor type in the following:

```
<!DOCTYPE html>
<html>
   <head>
      <title>The best webpage!</title>
   </head>
   <body>
      <p>This is a really awesome first webpage!</p>
   </body>
</html>
```

The <!DOCTYPE html> is unique in that it's not really a tag, but is really just telling the browser what version of HTML you are using, so it knows what it's doing. The basic !DOCTYPE html is used for the latest HTML5 standard.

The html tag is the what wraps everything, and tells the internet browser "this is the webpage" and everything between the opening <html> and </html> is the document.

Things which go in the <head> tags are for information about the page. For instance what the title of the page is, what files are used to style the page etc. An example of this is the <title> tag used above, the content of the tag appears in the browser tab, check it out!

Within the <body> tags are where everything that appears in the web browser goes, and is used purely for content and structure. You can only have one <html>, <title> and <body> tag per page.

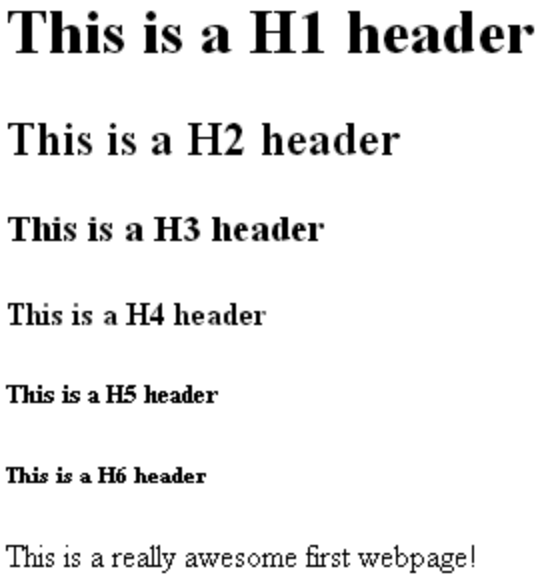
So here in the <body> tag we can start adding some stuff. Currently we are using the <p> tag around our text which defines a paragraph.

Start Tag Content End Tag

<title>This is content</title>

Headers

Let's add some more content, starting with a page header. Headers by default come in six flavours, each declining in size from 1 to 6. The image gives some size references for the default sizes, and a paragraph tag at the end for reference.



Decide on what you want the header for the webpage to be, and choose a size for it. In between the <body> tags and above the <p> tag, make a new line and put <h1> tags where the number is the size you want, and the content is what you want the header to be.

You can have as many of these tags on a page as you want, and you don't have to use them at all, it's entirely up to you. You could do a page of entirely h1 tags if you really wanted to. It is however good convention to use h1 only once on the page as a main heading, and h2-h6 as many times are you want throughout the page.

One thing to remember, is that if you want all the content to be on the same line, place it within the same tags. By default, HTML Elements will always be placed on a new line as if it were a new paragraph.

Links and Attributes

Links are an important part of any webpage, how else would you get around? For this we use the anchor tag <a>. You can put the <a> tag around anything, and whatever you put it around, if you click it you will be taken to the link defined in the attribute of the <a> tag.

Attributes are placed in between the starting tags like this <a href="www.google.co.uk"> and just gives the tag a bit more information to add a bit of interactivity and features to the page.

For links we use the href attribute in an <a> anchor tag, so decide on the page you want to link to and place it around some text like this:

```
<a href="www.bbc.co.uk">Keep up with the news here!</a>
```

Refresh the page and it should be coloured different and be a link. Click on it to find out if it worked!

Images

Images are an integral part of any website, from displaying a logo to your favourite cat. For images we use the <img> tag like this:

```
<img src="cat.jpg" alt="Cat on a bed">
```

First thing to note is there is no closing img tag. This is because the img tag does not enclose any content so it doesn't need a closing tag.

The src attribute is a lot like the href attribute in that it links to another file, in this case an image to load. Files are referenced local to the html file. So if the image is in the same folder as the html file, just put the file name + file extension there. Otherwise you'll need to reference the folder it is in as well, like if it was in an "images" folder, you would reference it like this:

```
<img src="images/cat.jpg" alt="Cat on a bed">
```

The alt attribute gives a description of the image, for those who can't see the image whether from it not loading, or being visually impaired. It's good practice to do it like this.

Lists

Organising content on a webpage can be a bit difficult, and the list tags help a lot with that.

It comes in two varieties: an ordered list which is <ol> and an unordered list which is <ul>. Structurally they are the same, with the difference being in an order list each item is number incrementally, useful for putting items in an order. is unordered, and the start of each item is just a bullet point.

Each item in a list has to have a <li> tag wrapped around it, which stands for "List Item".

So if you wanted a basic shopping list, you could use this example

```
<h1>My Shopping List</h1>
<ul>
    <li>Carrots</li>
    <li>Pizza</li>
    <li>Bread</li>
    <li>Milk</li>
    <li>Chocolate</li>
</ul>
```

Try changing the <ul> tags to <ol> and see what changes.

Tables

In the past tables have been used to layout pages, which is ultimately a misuse of how it should be used. Tables should be used to structure data which needs to be in a table, such as average rainfall per month, basic science test results etc.

A table is split into 3 main tags. First is the <table> tag, which acts as the root of the table for which everything relating to the table is held within. Then each row of the table is described with <tr> (standing for table row). So if you're table has 4 rows then you will need 4 <tr> tags. For each bit of data within the row, you will need as many <td> tags as well which stands for table data. Something else we can to help describe the data, is add table headings using the <th> tag. This goes on its own row above the data, and is used in replacement of the <td> tag.

So let's look at a quick example, here's the table we want on our webpage:

First Name	Second Name	Age	Height
John	Toms	19	182cm
Sophie	Dale	21	165cm
Helen	Clyne	17	160cm
Sam	Docherty	24	163cm

Have a go at trying to describe that with HTML by yourself, and to check or if you get stuck, see the example below:

```
<table border="1">
   <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Age</th>
      <th>Height</th>
   </tr>
   <tr>
      <td>John</td>
      <td>Toms</td>
      <td>19</td>
      <td>185cm</td>
   </tr>
   <tr>
      <td>Sophie</td>
      <td>Dale</td>
      <td>21</td>
      <td>178cm</td>
   </tr>
   <tr>
      <td>Helen</td>
      <td>Clyne</td>
      <td>17</td>
      <td>160cm</td>
   </tr>
   <tr>
      <td>Sam</td>
      <td>Docherty</td>
      <td>24</td>
      <td>163cm</td>
   </tr>
</table>
```

As you can see, this is pretty long-winded, but it results in a nicely formatted table with all the data in it. You'll also see that in the <table> tag at the start, it has a border="1" attribute. This is so we can see the tables boundaries on the page, otherwise the web browser doesn't draw them.

Put it together

So there we have the basics of HTML down, and the best thing to do before carrying on is to put it all together onto one page. Pretty much every single web page out there is based upon these few elements. There are a few tags missed out such as the tag and . We unfortunately don't have time to explain them in this short lesson, but if there's spare time at the end, ask the person teaching you to briefly explain it. Create a page about yourself or something you're interested in, write some paragraphs about it. Add some images, a table or a list if you need it. Make sure you also give it a big header as well. When you're done get ready to move onto styling it, and making it look nice.

Styling

The next biggest thing of any webpage other than the content is how it looks. For this we use CSS (Cascading Style Sheets). There are several ways of implementing CSS into a webpage, but we will be putting it in a separate file from the html. So to get started, in Notepad++ create a new file, then save it in the same folder as your html file, and call it "style.css".

Now, before we start styling we need to tell the html page where the style sheet is, so in the <head> tag either above or below your <title> tag (it doesn't matter, as long as it's between the <head> tags, type this:

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

And there, the style sheet should be set up and ready to go.

Changing the background

To start off we're going to change the background colour of the whole page. Make sure you've got the style.css file open, and type the following.

```
body {
  background-color: skyblue;
}
```

So we start off a CSS statement by saying what we want to style, this is called a selector. In this case the html <body> tag as it fills the whole page, a bit later on we will talk about other kinds of selectors.

Then between the curly brackets are the declarations, with in this example each declaration getting its own line to make it easier to read.

There are two parts to any declaration, a property which is on the left of the colon, and the value for that property on the right of the colon. At the end of every declaration there is a semi-colon to denote that the declaration is finished.

In the case of what we just wrote, we're setting the background colour (Note that American spelling is used, it will not work if you use colour, only color) to a default of skyblue. This means that the background colour of our selector (<body>) is being set to all the same colour.

More Styling

Now, going back to your webpage you built, choose any tag which you want to change the colour of. It can be a header (<h1>, <h2> etc), a paragraph <p> or whatever you want. Now try and make your own style! It's basic setup is like this:

```
*The tag you've chosen without the < and >* {
  color: *your colour*;
}
```

So in the top selector bit, if you want to change the colour of all the <p> tags on your webpage, just type p by itself.

For your colour, you can try typing something like red, green, yellow etc and see what the browser already knows for colours.

Alternatively, you can try using a HEX value for the colours, starting with a # and then 6 hexadecimal digits after (where the first two digits are red, second two are green and the last two are blue), so it can look like #90F266. The higher the number, the more of that colour is added to the mix.

And lastly you can do it as an RGB number, by typing rgb(125, 45, 200), where each number can go up to 255.

Try playing around with the colours, and see what you can make. If you get stuck with this here's an example of styling a <p> tag and a <h1>.

```
p {
  color: blue;
}
h1 {
  color: # 50b848;
}
```

Big Text

You can change the size of a font by using the font-size property.

```
h1 {
  font-size: 60px;
}
```

Now here, you can see we're using a number followed by px. The px stands for pixels, so in this case, the font in the <h1> tags is going to be 60 pixels tall from the top of a capital letter, to the bottom of a lowercase letter. You can make the number as small as you want, or as big as you want, it's up to you.

You can try other things too, for instance, try replacing the px with a % sign and see what happens.

Alignment

Currently all your content is on the left side of the screen, so let's fix that. A lot of modern websites have all the content in the center of the screen.

```
h1 {
  text-align: center;
}
```

This will put your header in the middle of the page. If you want absolutely everything to be centered, there is a universal selector which you can use, which is * . Place * instead of the h1 and see that everything is now centered.

Conclusion

So there you have it, a basic website which you can put whatever you want on, and start to make it look like what you want as well. Now this is only a really simple start to get you interested and inspired into creating websites in a hands on way where you can ask us questions too.

If you want to take making websites further, try out www.codecademy.com to learn some more advanced topics such as JavaScript and jQuery.