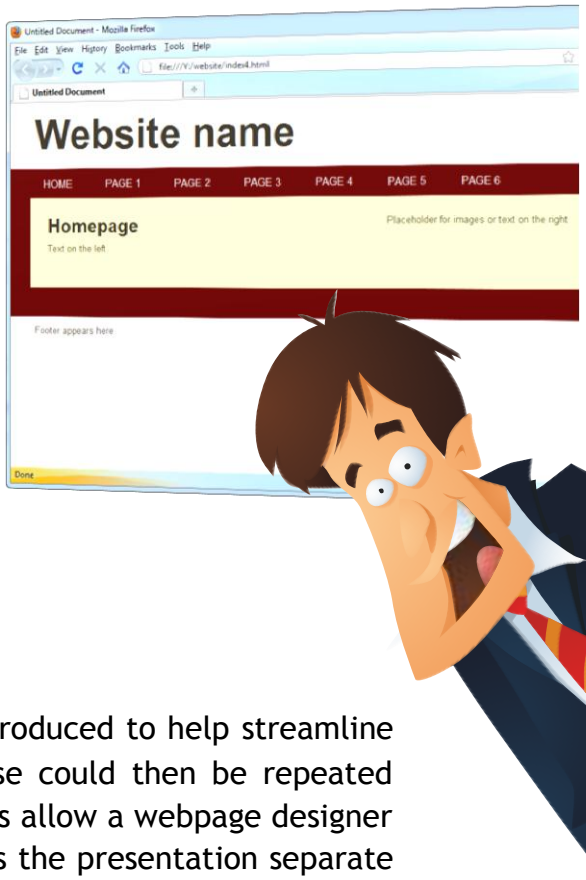
A CSS + HTML powered website

Introduction



This guide will help you create your own webpage coded in html and styled using CSS (Cascading Style Sheets). Modern web design software can handle all of the CSS for you, but it is good practice to delve into the code as then you can start to appreciate how it all works.

This guide provides you with basic, step-by-step instructions how to build a starter page. You can then customise it further.

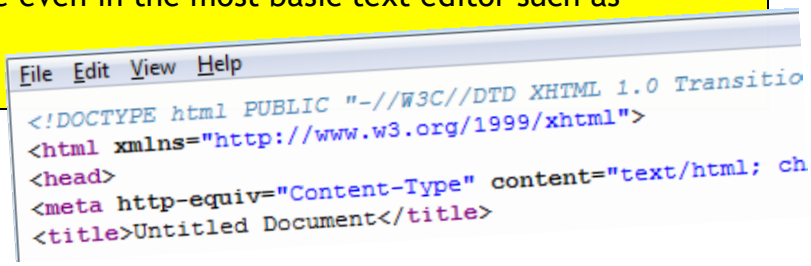
What is CSS?

CSS stands for **C**ascading **S**tyle **S**heets. They were introduced to help streamline websites by allowing designers to define styles. These could then be repeated across an entire website with multiple pages. The styles allow a webpage designer control over layout, fonts, colours, navigation – it keeps the presentation separate from the actual content.

What program should I be using?

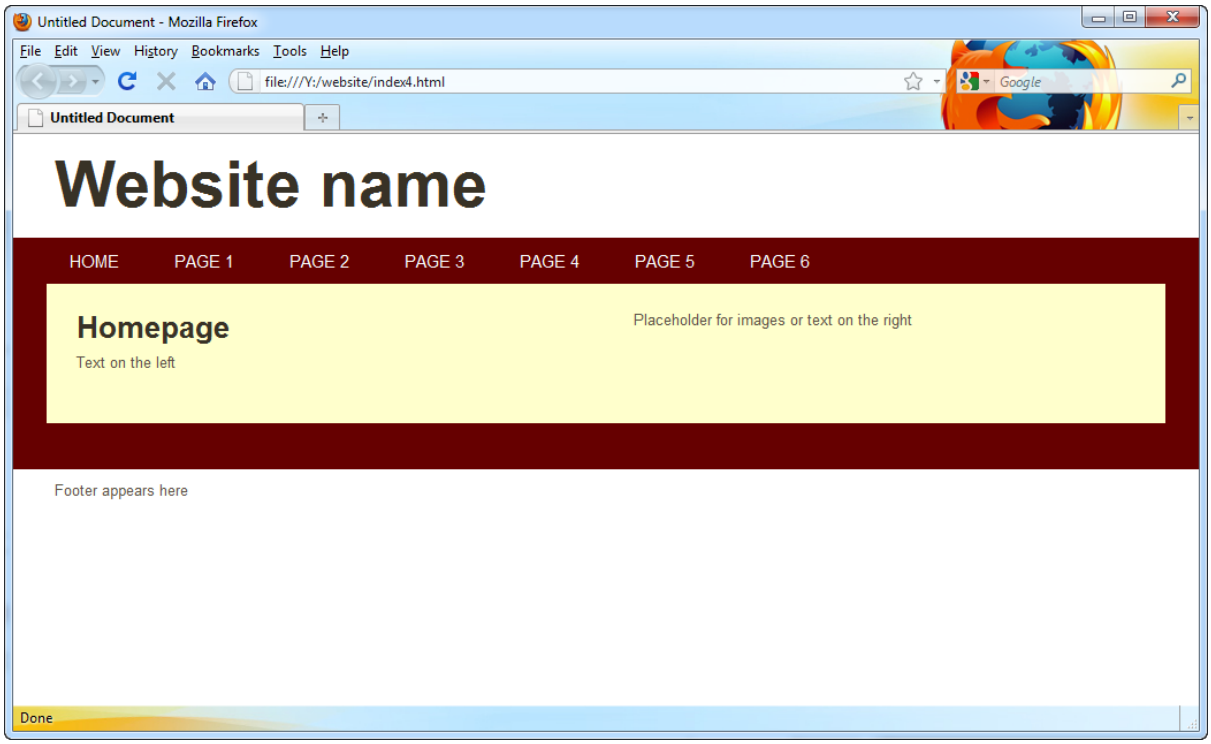
Adobe® Dreamweaver® CS5
Version 11.0 Build 4964

The screenshots shown here are from Dreamweaver CS5, but you can use any program you like – you can code even in the most basic text editor such as Notepad.

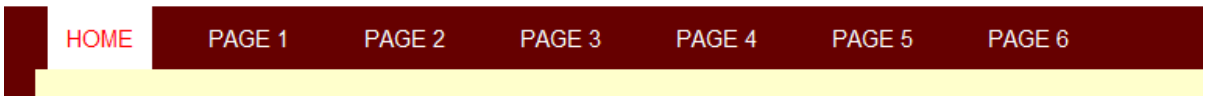


What are we creating exactly?

You are going to create a webpage like this – which you can then customise and develop into your own masterpiece.



The navigation bar near the top uses CSS to provide hover-over effects:

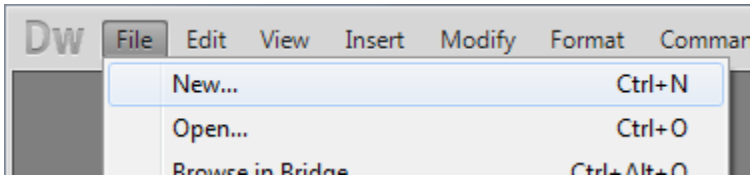


As you develop the page, you can customise the colours, presentation and everything else – this is the beauty of using CSS to design a website. It keeps the content and the design neatly separated.

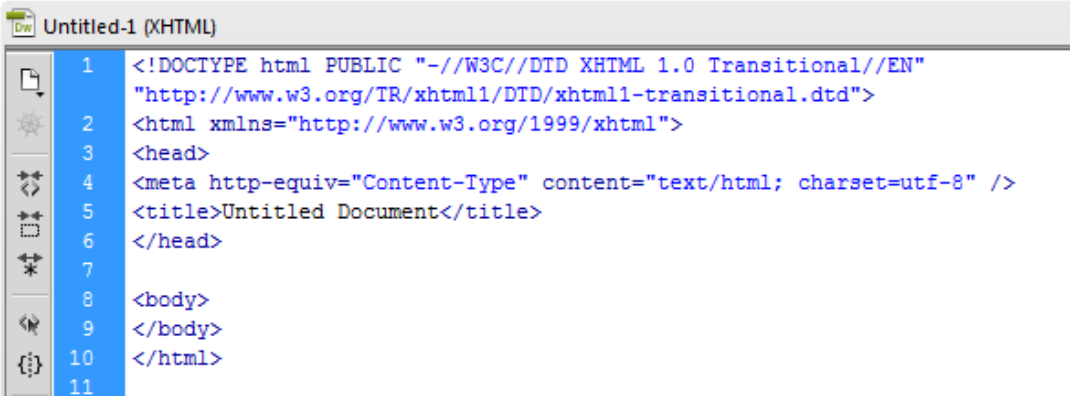
This basic page has a CSS header, a block of dark colour across the whole page including the navigation bar. There is also a section for the main content within the dark box - here you will setup two columns to help organise your content. There is also a footer section at the bottom of the page.

1. Load your program of choice and make sure you can edit code directly.

   For example, if you are using Dreamweaver, load the program, select 'File' and 'New'. Click on the option to create a basic .html file.



2. Dreamweaver automatically creates the core .html code that forms a webpage: [*If you're using an older version it might look slightly different*]



You need to read through the next page – it helps explain all the complex code.

You don't have to do anything though – just read it and try to understand.

At least give it a try?!

3. The key parts of a webpage are shown in this code. It looks really complex, but when you look carefully it really isn't too bad.

Remember a webpage is built up in sections. Each of these sections, known as **HTML tags**, are a bit like brackets. You open a tag, provide the information and then close the tag.

The **DOCTYPE note** is to tell the web browser what type of html code is being used.

The **<html> tag** contains the actual webpage.

The **<head> tag** is used to setup the page - things like the title, CSS style information and additional code that needs used in the page.

The **<body> tag** is used for the main page content.

DOCTYPE note

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

<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Untitled Document</title>
</head>

<body>
</body>

</html>
```

<html> tag

<head> tag

<body> tag

Might be worth having a lie down or something after that code.

All you really need to know is that the code is the main structure behind the webpage.

4. We now need to setup the CSS code for the starter website. This is more complex code, but this time you need to actually add it yourself. It is the best way to start becoming familiar with the way CSS code is structured.

5. Add (**type - don't copy and paste**) the following text underneath the:

```
<title>Untitled document</title>
```

```
<style type="text/css">
body {
     font-family: Geneva, Arial, Verdana, sans-serif;
     font-size: 13px;
     background: #fff;
     margin:0;
}
```

- The **<style> tag** is the way we can add CSS code directly to a webpage.
- The **body { }** section is where we define how the main 'body' of the webpage will look - we set the default font, the font size and background colour.

6. Next add the following CSS code. You can copy and paste it this time.

```
p {
    margin: 0 0 18px;
     line-height: 18px;
     font-size: 13px;
}

a {
     color:#00F;
     font-weight: normal;
     text-decoration: underline;
     }

a:hover {
     color:#F00;
     }

h1 {
     font-weight: bold;
     font-size: 26px;
     color: #383226;
     margin: 0 0 6px;
}

h2 {
     font-weight: bold;
     font-size: 56px;
     color: #383226;
     margin: 0 0 6px;
}
```

Here we have defined how default paragraph will look - the **p { … } style**

The **a { … } style** refers to hyperlinks. The colour used for links, the colour shown when you hover over the link.

h1 { … } and **h2 { … } styles** define how we want headings to be used on our page.

Look at the CSS code. If we use the h1 style it will set the font size 26px. What size will text using h2 style be?

Now look at the webpage mock up - which part do you reckon uses the h2 style?

7. So far you have defined the CSS styles for the standard parts of a webpage:

   - **p { … }** = paragraphs
   - **a { … }** = hyperlink behaviour
   - **h1 { … } and h2 { … }** = header text

8. We now need to setup the layout of the page. This is where you can define different sections such as a place for text or an image at the top of a page, a navigation bar and a section for all the main content to appear.

   We are going to define our own custom styles to use on the page. Add this code to your webpage.

```
/* Layout */

div.wrapper {
     width: 978px;
     overflow: hidden;
     margin: 0 auto;
}

div.top {
     width: 952px;
     height: 70px;
     padding: 10px;
     overflow: hidden;
}

div.main {
     background:#600;
     overflow: hidden;
     padding: 0 0 40px;
}

div.navigation {
     clear: both;
     background:#600;
     overflow: hidden;
     padding: 0;
}

div.main div.wrapper {
     padding: 0 20px 26px 10px;
     width: 942px;
     background:#FFC;
}
```

```
div.column {
     width: 442px;
     padding: 22px 0 0;
     overflow: hidden;
}

div.left {
     float: left;
     padding-left: 16px;
}

div.right {
     float: right;
}


div.footer {
     padding: 10px;
     overflow: hidden;
}
```

Look through the code!

- What do you think the ‘`div.wrapper`’ style is going to be used for?
- If ‘`div.top`’ is the custom header - how high will it be?
- What do you think ‘`div.navigation`’ refers to?
- How might ‘`div.column`’, ‘`div.left`’ and ‘`div.right`’ be linked?

They will be used for the main content of the page - column style has setup a standard width. The left and right styles are used for the different sides.	The code says `height: 70px;` so it will be 70 pixels high.
It defines the section where the navigation bar will appear. The CSS code has set a background colour with the code `#600;`	It is used to ‘wrap’ around sections on the page to set the width and how the section will be displayed.

9. Finally (in terms of the complex CSS) we need to add the custom styles to setup our navigation bar.

   Add the following code:

```
/* Navigation bar */

ul#nav {
     padding: 0;
     margin: 0 auto;
     width: 980px;
     list-style: none;
     list-style-image: none;
}

ul#nav li {
     display: block;
     list-style: none;
     list-style-image: none;
     float: left;
     margin: 0;
     padding: 0 12px;
}

ul#nav li a {
     overflow: hidden;
     height: 40px;
     display: block;
     padding: 0 12px;
     line-height: 40px;
     float: left;
     margin: 0;
     font-size: 15px;
     color: #fff;
     text-decoration: none;
     text-transform:uppercase;
}

ul#nav li a:hover {
     background: #fff;
     color: #f00;
}

</style>
```

10. This code takes a standard bullet list (the `li` code) and customises it to create a horizontal navigation bar. This shows the power of CSS. To find out what it actually does, we need to experiment with it on our actual page.

11. Make sure you have added the code to the correct place. All CSS code needs to be within the **<head>** tag of your webpage.

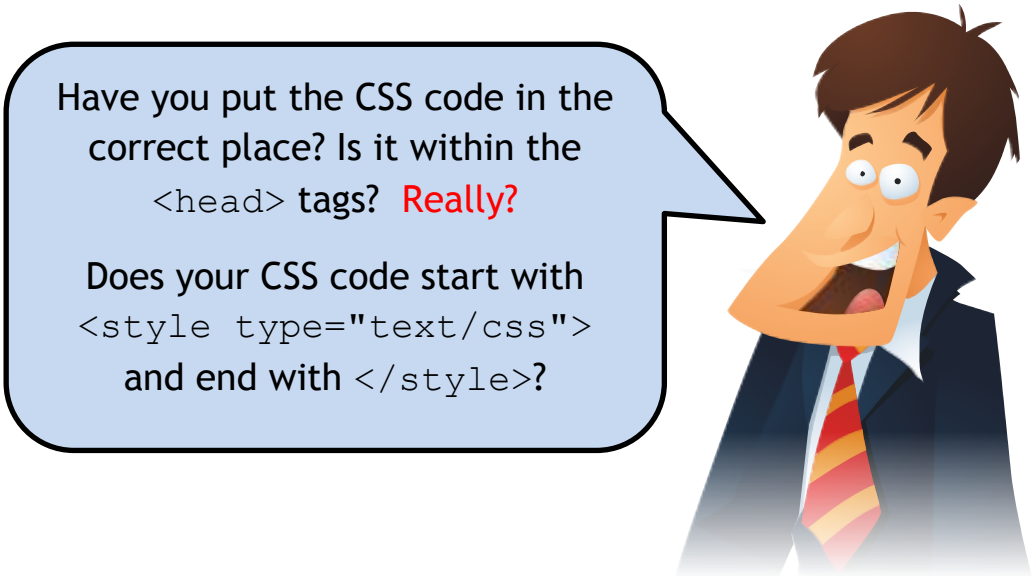
    If it is in the correct place, you should see the final lines of the CSS code, ending with </style> followed by the </head>. The slashes are what html uses to close a tag.

```
    color: #f00;
}

</style>
</head>

<body>
</body>

</html>
```



12. The main content can now be added to the **<body>** section. Here we will make use of all the custom CSS styles we have created.

    Add this code below the **<body>** tag:

```
<div class="wrapper">
        <div class="top">
                <h2>Website name</h2>
        </div>
    </div>
```

```
<body>
    <div class="wrapper">
        <div class="top">
                <h2>Website name</h2>
        </div>
    </div>

</body>

</html>
```

13. Save your webpage and view it in your browser. If you are using Dreamweaver, you can use the **'File'** > **'Preview in browser'** option.

    You should see a short webpage with a heading:

    Not much to look at yet, but we're nearly there.

14. Next we can add the navigation html code. Without the CSS this would appear as a vertical bullet point list.

    The **<a href** code refers to the hyperlinks for your navigation bar. At the moment they are just blank, but we'll need to change them to actual page links eventually.

```
<div class="navigation">
     <ul id="nav">
          <li><a href="#">Home</a></li>
          <li><a href="#">Page 1</a></li>
          <li><a href="#">Page 2</a></li>
          <li><a href="#">Page 3</a></li>
          <li><a href="#">Page 4</a></li>
       <li><a href="#">Page 5</a></li>
          <li><a href="#">Page 6</a></li>
     </ul>
</div>
```

Look through the code!

- It is actually really simple – nothing about font sizes, colours.
- As you've done the work to design the CSS all you need to do is apply these styles to our content.
- See if you can customise your links…

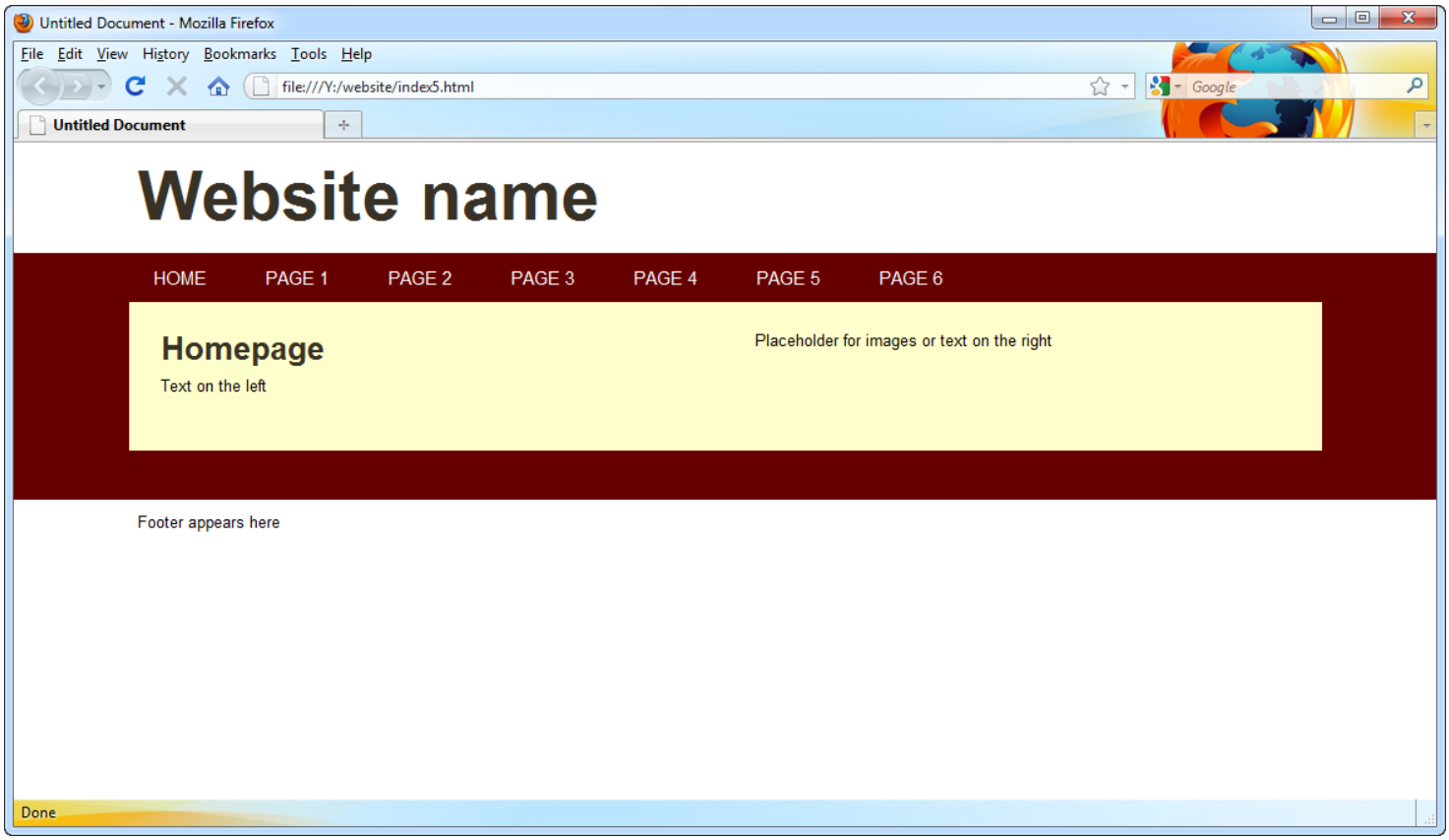
15. Save your webpage and preview the page in your browser again. This time you should see your header text and your navigation bar:

16. Finally, we need to add the code for our main content and 'footer':

```
<div class="main">
          <div class="wrapper">
               <div class="left column">
                    <h1>Homepage</h1>
                    <p>Text on the left</p>
               </div>
               <div class="right column">
              <p>Placeholder for images or text on the
              right</p>
               </div>
          </div>
     </div>

     <div class="wrapper">
          <div class="footer">
          Footer appears here
          </div>
     </div>
```

17. Preview your page and you should see something very similar to this:



If your page doesn't look like this you need to go back and check your html code. You may have pasted or typed code into the wrong place.

The code at the end of your page should look like this:

```
            <p>Placeholder for images or text on the right</p>
            </div>
        </div>
    </div>

    <div class="wrapper">
        <div class="footer">
        Footer appears here
        </div>
    </div>
</body>

</html>
```

See how each time a tag is opened (e.g. `<div … >`) it is also closed?

At the end of the page, the `</body>` and `</html>` tags are closed.

Now over to you…

You have been given a basic page structure. You now need to play around and experiment. Can you change the colours used on the page? Can you add working links in the navigation bar? Can you customise the design of the page further?

Think about some of these issues and suggestions:

- Save your current work before you experiment too much – experiment with a new copy of the page.
- The best sites don't actually have the CSS code within each page, they link to a separate file. Research 'external style sheets' and see if you can…
- For your own website, you need to create pages before you can link to them. You could create one page you're happy with and then duplicate it.
- This is very much about you taking total ownership of this work – you've been given a start, but now be confident to adapt it, change it and create your own genius piece of web design.