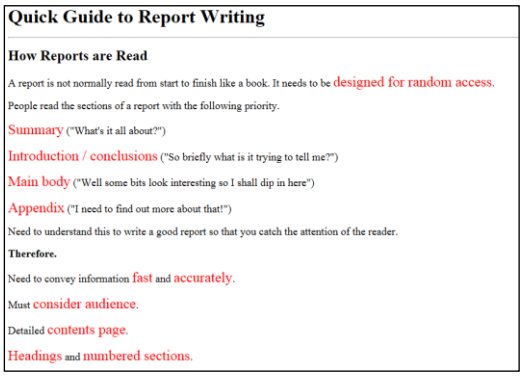
Styles, Style Sheets, the Box Model and Liquid Layout

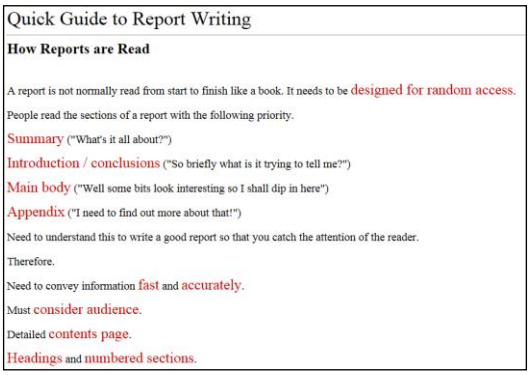
This session will guide you through examples of how styles and Cascading Style Sheets (CSS) may be used in your Web pages to simplify maintenance of your pages / site. We shall also take a look at how the box model may be used to control the positioning of content on a page.

Introduction to Styles

To begin with take a look at the following two documents...



Bad



Better

So what is the difference between the two documents?

The contents of both documents are pretty much identical. However there are significant differences in the way that the documents are organised internally.

Overview of HTML

A simple HTML document has the following format...

```
<html>

<head>
    <title>Sample HTML Document</title>
</head>

<body>

Hello this is some text <u>underlined</u> <b>bold</b> and <i>italics</i>

k/body>

</html>
```

The Font Tag

If I wanted to make a section of text a different colour one way we could do this is to use the font tag...

```
<font color="red">
    Hello this is some text <u>underlined</u> <b>bold</b> and <i>italics</i>
</font>
```

Before proceeding to the next section, look at the documents again but this time view the source HTML they are based on.

The Bad Example – Embedded Styling

The bad example has been created without the use of styles and is fairly typical of how many amateur web pages are created...

```
<body>

<p align="left"><font size="6"><b>Quick Guide to Report Writing</b></font></p>

<hr>

<p><font size="5"><b>How Reports are Read</b></font></p>

<p><font size="4">A report is not normally read from start to
finish like a book. It needs to be </font><font color="#FF0000"
size="5">designed for random access</font><font size="4">.</font></p>

<p><font size="4">People read the sections of a report with the
following priority.</font></p>

<p><font color="#FF0000" size="5">Summary</font><font size="4">
(&quot;What's it all about?&quot;)</font></p>

<p><font color="#FF0000" size="5">Introduction / conclusions</font><font
size="4"> (&quot;So briefly what is it trying to tell me?&quot;)</font></p>
```

For example when the text is “styled” we are using the “deprecated” or outdated HTML tag called <font>

This means that the HTML for the red text...

Summary ("What's it all about?")

Introduction / conclusions ("So briefly what is it trying to tell me?")

Main body ("Well some bits look interesting so I shall dip in here")

Appendix ("I need to find out more about that!")

Is constructed like so...

```
<p><font color="#FF0000" size="5">Summary</font><font size="4">
(&quot;What's it all about?&quot;)</font></p>

<p><font color="#FF0000" size="5">Introduction / conclusions</font><font
size="4"> (&quot;So briefly what is it trying to tell me?&quot;)</font></p>

<p><font color="#FF0000" size="5">Main body</font><font size="4">
(&quot;Well some bits look interesting so I shall dip in
here&quot;)</font></p>

<p><font color="#FF0000" size="5">Appendix</font><font size="4">
(&quot;I need to find out more about that!&quot;)</font></p>
```

What we are doing here is formatting each instance of the red text with its own <font> tag.

In this document there are 63 instances of red text applied throughout the document.

The problem comes when we want to change the text of the red sections to blue or some other colour. To do that we will need to make 63 changes to the entire document.

The Better Example – Internal Style Sheet

For this document we have modified the internal structure of the document and created an internal style sheet.

If you view the HTML for your page you will see the internal style sheet:

```
<style type="text/css">
<!--
.MainHeading {
    font-family: "Times New Roman", Times, serif;
    font-size: 24pt;
}
.SectionLevel1 {
    font-family: "Times New Roman", Times, serif;
    font-size: 18pt;
    font-weight: bold;
}
.BodyHighlight {
    font-family: "Times New Roman", Times, serif;
    font-size: 18pt;
    color: #CC0000;
}
.BodyText {
    font-family: "Times New Roman", Times, serif;
    font-size: 14pt;
    line-height: 30pt;
}
.ExampleSections {
    font-family: "Times New Roman", Times, serif;
    font-size: 14pt;
    font-style: italic;
}
-->
</style>
```

This style sheet defines the five styles that we are using within the document.

So the code defines a style Class called .BodyHighlight.

```
.BodyHighlight {
    font-family: "Times New Roman", Times, serif;
    font-size: 18pt;
    color: #CC0000;
}
```

A style is defined once in the style sheet, then, when we want to use it we mark up the HTML using suitable tags.

Marking up the HTML

Once a style is defined in a style sheet then there needs to be some mechanism for applying the style to the HTML. This process is called styling your pages.

In general there are two sorts of styles that may be applied, they are inline and block.

Inline styles when applied to the text do not add a new paragraph to the marked up section.

For example the text

A report is not normally read from start to finish like a book. It needs to be designed for random access.

Uses the tag to apply the style like so...

```
<span class="BodyText">A report is not normally read from start to finish like a book. It needs to be </span>
  <span class="BodyHighlight">designed for random access</span>
  <span class="BodyText">.<br />
<span class="BodyHighlight">designed for random access</span>
```

Div versus Span

Applying a span keeps the flow of the text. For example when viewed in the browser the formatted text would look something like this.

k. It needs to be designed for random access .

Block styles when applied create a new paragraph and would not be suitable for use in this section of text.

If the second line of mark-up is changed from a to a like so...

```
<span class="BodyText">A report is not normally read from start to finish like a book. It needs to be </span>
  <div class="BodyHighlight">designed for random access</div>
  <span class="BodyText">.<br />
<span class="BodyText">A report is not normally read from start to finish like a book. It needs to be </span>
<div class="BodyHighlight">designed for random access</div>
```

We are now applying block level formatting. Resulting in the text looking like this...

A report is not normally read from start to finish like a book. It needs to be

designed for random access

.

A report is not normally read from start to finish like a book. It needs to be

designed for random access

Not quite the effect we are after.

Inline formatting is normally used if we want to format words within a sentence. Block formatting is used if we want to format an entire block of our page for example an entire paragraph.

The Best Example - External Style Sheets

The third example we have already mentioned in storing our styles is to make use of an external style sheet.

In the example above we have created a set of styles which may be used in this document only.

The external CSS is a separate file to the HTML. The CSS file contains all of the styles and the HTML contains the unformatted content along with a line of HTML telling the page where to get its formatting.

The external style sheet is a separate file to the HTML document. In the case of an internal style sheet it only applies to a single page. A single external style sheet may be applied to multiple pages.

The big advantage of an external style sheet is that a single change here impacts the look of the whole site. Internal style sheets are useful if you want to apply changes to individual pages only.

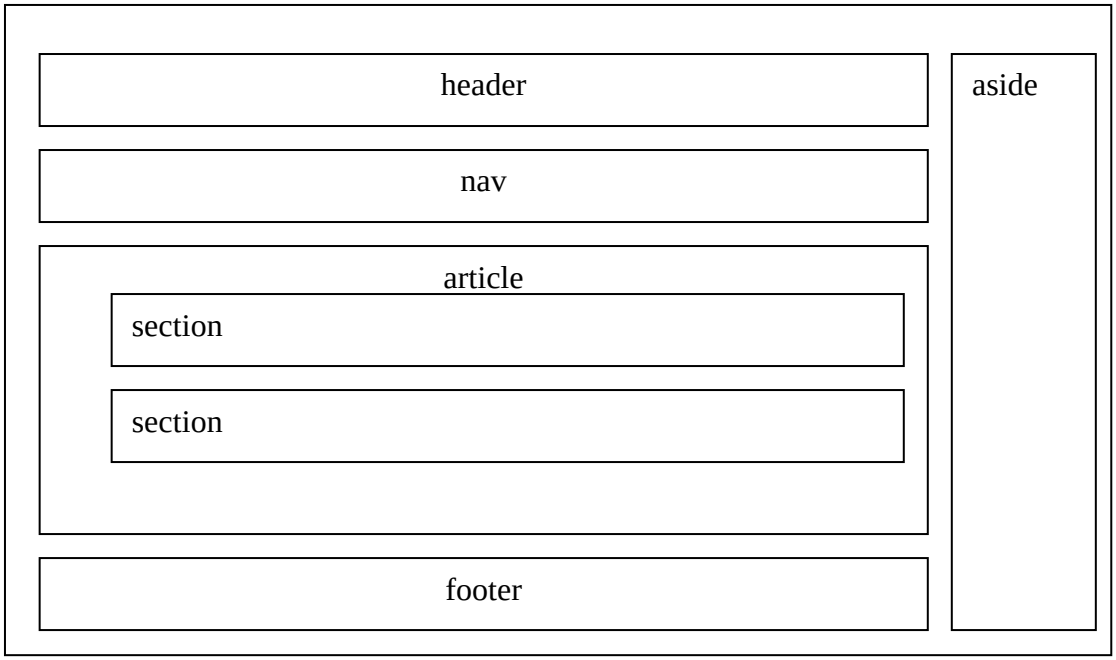
Layout using Styles and the Box Model

As well as allowing us to format specific sections of text using and we also have the option of laying out the positioning of text and graphics using styles.

HTML 5

HTML 5 gives us a set of new tags which are helpful in deciding on the names we may use to format the document.

For example...



Notice that the names of the sections describe what they do not where they go.

header	Content that appears on every page of the site typically at the top
footer	Content that appears on every page of the site typically at the bottom
nav	The navigation elements for the page
article	The main document for this page
section	A sub section in the main document
aside	A section on the page with additional information (often advertising)

Creating the Layout

We shall set up a new HTML document with the following unformatted content...

This is the header

This is the nav

This is the article

This is a section

This is the footer of our page

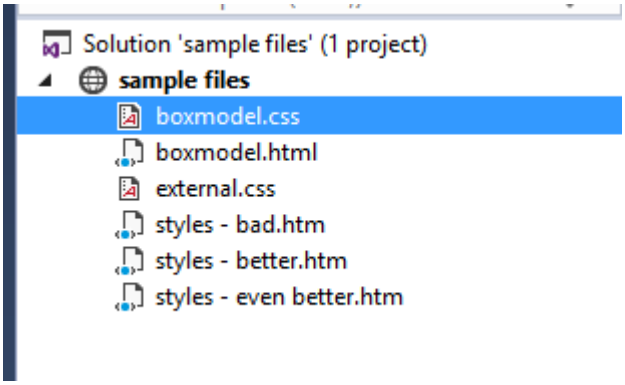
This is the aside

Here is the HTML for the page.

```
<!DOCTYPE html>
<html>
<head>
<title>Untitled Document</title>
<link href="boxmodel.css" rel="stylesheet" type="text/css" />
</head>

<body>
<p>This is the header</p>
<p>This is the nav</p>
<p>This is the article</p>
<p>This is a section</p>
<p>This is the footer of our page</p>
<p>This is the aside</p>
</body>
</html>
```

The next step is to create an external style sheet to store the styles...



We are not going to format every style in this example we shall concentrate on the header style for these examples.

The first thing to notice is the flow of the text...

This is the header

This is the nav

This is the article

This is a section

This is the footer of our page

This is the aside

In formatting a specific section of text we have several options to consider in its positioning.

- Static - We may leave the text in the flow of the text
- Absolute - We may remove the text from the flow and specify exact coordinates with respect to the parent container
- Fixed - We may remove the text from the flow and specify exact coordinates with respect to the browser window
- Relative - We may position the text relative to its position in the flow

Re-Defining Tags

One thing that we can do in applying styling to HTML is to re-define existing tags.

What this means is that we can apply formatting to the page whenever a tag appears.

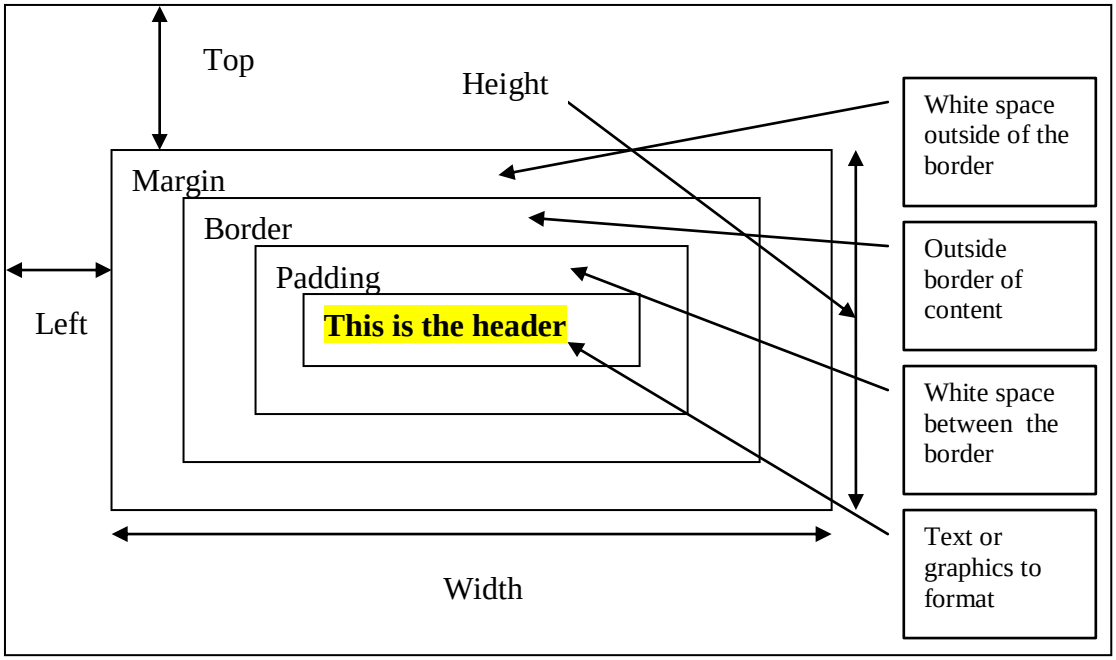
This means the first thing we need to do is mark-up the document o state where the formatting we want will apply, like so…

```
<!DOCTYPE html>
<html>
<head>
<title>Untitled Document</title>
<link href="boxmodel.css" rel="stylesheet" type="text/css" />
</head>

<body>
    <header>
        <p>This is the header</p>
    </header>

<p>This is the nav</p>
<p>This is the article</p>
<p>This is a section</p>
<p>This is the footer of our page</p>
<p>This is the aside</p>
</body>
</html>
```

In order to understand the box model we need to think of the text as having a box drawn round it...



As well as the boxes within the main box the main box also has a top, left, height and width property.

These are often determined with reference to the parent container which may or may not be the main browser window!

The first thing we will apply is a border so that we can see where the edges of the styling are applied.

boxmodel.html* boxmodel.css*

```
@charset "utf-8";
/* CSS Document */

header {
    border:solid;
}
```

Which creates the following effect…

This is the header

This is the nav

This is the article

This is a section

This is the footer of our page

This is the aside

Adjusting the Padding

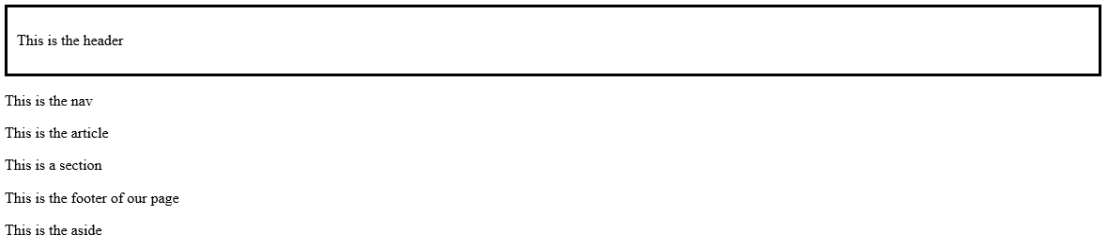
If we change the padding we may change how the text sits in relation to the border...

boxmodel.html boxmodel.css*

```
@charset "utf-8";
/* CSS Document */

header {
    border:solid;
    padding: 10px;
}
```

Resulting in 10 pixels (px) of space between the text and the border…



Adjusting the Margin

If we increase the margin we increase the spacing beyond the border...

boxmodel.html boxmodel.css*

```
@charset "utf-8";
/* CSS Document */

header {
    border:solid;
    padding: 10px;
    margin: 40px;
}
```

Like so...

Changing Positioning

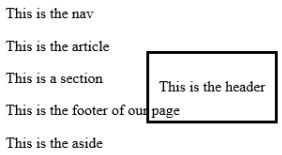
We may also change the way that the style is positioned and take it out of the flow of the text by changing the position to absolute...

boxmodel.html boxmodel.css

```
@charset "utf-8";
/* CSS Document */

header {
    border:solid;
    padding: 10px;
    margin: 40px;
    position: absolute;
    top: 5%;
    left: 10%;
}
```

Producing the not very attractive...



Notice so far we have been setting the units of measurement as “px” pixels or dots.

It is also worth considering if % or em produce better results.

- A percentage value mans that the box is calculated as a percentage of its parent.
- An em value means that the box size is calculated based on the size of the specified font.

Percentage values work well in many cases however an em value will scale better if large fonts are used.

In this next example if we set the top and left of the box to 5% the box will scale even if we change the size of the browser window (or view the same page on a mobile phone!)

Adjusting Width and Height

The following changes…

boxmodel.html boxmodel.css*

```
@charset "utf-8";
/* CSS Document */

header {
    border:solid;
    padding: 10px;
    margin: 40px;
    position: absolute;
    top: 5%;
    left: 10%;
    width:80%;
    height:90%;
}
```

Give us…