

WEB DEVELOPMENT

Lecture 02

Introductory XHTML

Basic XHTML Elements

- **Document Structure**

- Text

- Links

- Lists

- Generic Elements

Document Structure Elements

<html> The root element of the document

<head> Header

<title> Document title

<meta> Meta data
(information about the document)

<body> The body of the document

Basic Document Structure

```
<html>  
  <head>  
    <title>My Sample Page</title>  
  </head>  
  
  <body>  
    Content of web page  
  </body>  
</html>
```

The <head> element

must contain a <title> element

may contain a:

<meta> element	which provides extra information about the document
<link> element	which links the document to another document – usually a stylesheet.
<base> element	which serves as reference for all pathnames and links in the document.

The <meta> Element

There are two types:

name: Which hold extra descriptive information about the document.

Can make up names or use standard ones supplied by search engines.

http-equiv: Which hold information that will affect the way the readers browser will handle the document.

eg Can specify how the document should be displayed, or how often it should be refreshed.

Example of <head> elements

```
<head>
```

```
<title>This is my document</title>
```

```
<base href="http://www.mysite.com/pages/"></base>
```

```
<link rel="stylesheet" href="/pathname/stylesheet.css"  
type="text/css"></link>
```

```
<meta name="author" content="Kevin Wilson"></meta>
```

```
<meta name="copyright" content="Wilson plc"></meta>
```

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8"></meta>
```

```
<meta http-equiv="refresh" content="10; url=http://www.bbc.co.uk"></meta>
```

```
</head>
```

Short hand for empty elements

Empty elements, like this one:

```
<br></br>
```

can be written like this:

```
<br />
```

to save space. Another example is this:

```
<meta name="author" content="Kevin Wilson"></meta>
```

Although there are two attributes, there is nothing between the open and close tag, and so it could be written as follows:

```
<meta name="author" content="Kevin Wilson" />
```


The <body> element

The body elements contain the main content of the page – all the text, tables and frames, if there are any.

This is a good place to introduce the idea of CSS – Cascading Style Sheets – and deprecated elements and attributes.

In the early days of HTML, you would often have seen something like this in a web page:

```
<body text="red" link="blue" bgcolor="black">  
  This is the contents of a web page  
</body>
```

What do you think this did?
Why is it wrong to do it now?

Deprecated Elements and Attributes

Any attribute or element that mixes up the appearance of the page with the content of the page is now deprecated.

It's use is strongly discouraged.

It is now considered good practice to store all information relating to the appearance of the page in a separate file called a Stylesheet.

For example, the example above would be done this way:

stylesheet.css

```
body
{
color: red;
background-color: black;
}
```

```
a:link
{
color: blue;
}
```

Note how the stylesheet contains a set of rules which tell the browser how to draw each part of the page.

The page is linked to the stylesheet by means of the `<link>` element in the `<head>` section at the top of the page.

There should be nothing in the web page to do with the appearance of the data in the browser.

Basic XHTML Elements

- Document Structure

- **Text**

- Links

- Lists

- Generic Elements

Text Elements

<code><h1>...<h6></code>	Headings
<code><p></code>	Paragraphs
<code><pre></code>	Preformatted Text
<code>
</code>	Line Break
<code><hr></code>	Horizontal Rule

There are others, but they have been mostly superseded by the generic elements `<div>` and ``.

Character Entity References

- `<h#>` Browsers usually show them in bold with decreasing size, but can override this by using CSS.
- `<p>` These are block elements, so they always start on a new line. They cannot contain other block elements – only text and inline elements. Usually shown left justified, but can override this by using CSS.
- `<pre>` Browsers will cut out any white spaces in a `<p>` element. Use this one if you want the text to look exactly like the text that you typed.

```
<body>
  <h1>This is heading level 1</h1>

  <p>This is the first paragraph. It is not really worth reading as
I am just making it up as I go along. Still, you might find it
interesting.</p>

  <p>This is the second paragraph. It has some line<br /> breaks in
to demonstrate that element. It is not<br />really worth reading
either as I am still making it up as I go along.</p>

  <hr />

  <h3>This is heading level 3</h3>

  <p>This is          another          paragraph      with
lots of          white spaces in</p>

  <pre>This is          another          paragraph      with
lots of          white spaces in  but this time it
is in pre tags</pre>
</body>
```




textstyle.css

```
body
{
background-image: url(feathertexture.bmp);
}
```

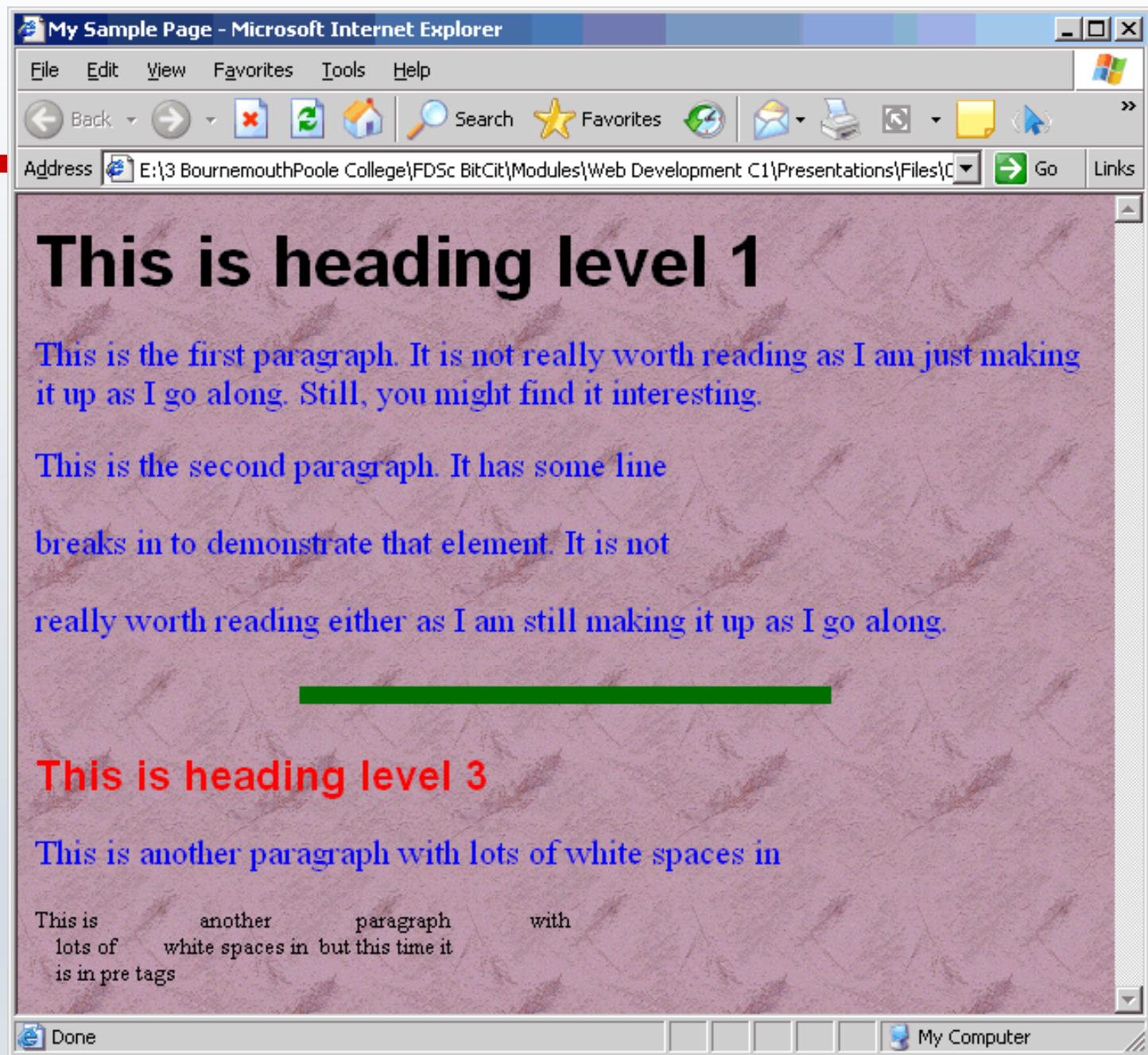
```
h1
{
font-family: Arial;
font-size: 40px;
}
```

```
h3
{
font-family: Arial;
font-size: 24px;
color: red;
}
```

```
p
{
font-size: 20px;
color: blue;
}
```

```
hr
{
height: 10px;
width: 50%;
color: green;
background-color: green;
}
```

```
pre
{
font-family: "Times New Roman", serif;
}
```



Character Entity References

There are some characters which cannot be used directly in HTML code.

If you want to display them, you must use a character reference.

This is known as *escaping* the character.

You can either use an abbreviation, or a numeric code.

The number is the ascii (or unicode) code for that character, and so this method can be used to display any character that doesn't appear on the keyboard.

Character Entity References

(space)	 	
&	&	&
<	<	<
>	>	>
'		'
“	&lddquo;	“
”	”	”
™	™	™
£	£	£
¥	¥	¥
©	©	©
®	®	®

Basic XHTML Elements

- Document Structure
- Text
- **Links**
- Lists
- Generic Elements

Link elements

`<a>` anchor. Identifies text or an image that serves as a hypertext link to another document.

Uses an href attribute to refer to the linked document.

You make text clickable:

I am `linking` to you.

or an image:

``

Absolute and Relative URLs

It is assumed that all of your webpages will be in the same folder.

If they are not, you will have to specify their location either using an absolute URL:

`href="http://www.mysite.com/web/index.htm"`

or a path that is relative to the current document (or to the base element if it is present):

`href="chap1/page1.htm"` **or**

`href="../index.htm"`

Linking Within a Document

You can link to a particular point within a page (called a fragment).

You need to be able to label a particular point within a page. Two main ways:

Use `<a>` element with a `name` attribute instead of `href`

```
<a name="section1">Section 1</a>
```

Add the `id` attribute to *any* HTML element

```
<h1 id="section1">Section 1</h1>
```

The name or id must be unique within the document.

Linking to the Fragment

Use the hash symbol:

```
<a href="#section1">Goto Section 1</a>
```

Linking to a fragment in another page

```
<a href="anotherpage.html#section2">Goto Section 2</a>
```

To open a page in a new popup window use the target attribute to give the popup window a name – then you can get multiple links to open a page in the same window.

```
<a href="http://www.bbc.co.uk/" target="display">click here</a>
```