Web Development

Lecture 7 – CSS Positioning

CSS Positioning

- div and span
- Pseudoclasses
- Box Model
- Floating Boxes
- Negative Margins
- Position Types
- Scrolling Boxes
- Stacking Order

The and tags

These are general purpose tags which you can use to apply styles to particular sections of text.

 sets up a block of text with a newline at the start and end.

 does not introduce a line break and can be used for inline sections of text.

Example of a tag in use

```
<p>
Call me on <span class="phone">12345</span> now.
</p>
```

I can then set up a style sheet which has something like this in:

```
span.phone
      {
      font-family: Arial;
      font-size: 12;
      line-height: 12;
      }
```

Pseudoclasses

These are usually used with the <a> element, which is used to set up links to other pages.

You can set up a stylesheet that says something like this:

```
a:link {color: red; }
a:visited {color: blue; }
a:hover {color: black; }
a:active {color: green; }
```

This enables you to change the colour of a hyperlink according to whether or not it has been clicked.

Pseudoelements

These allow you to define styles for part of a section of text.

:first-line

allows you to define the style of the first line of a section of text.

:first-letter

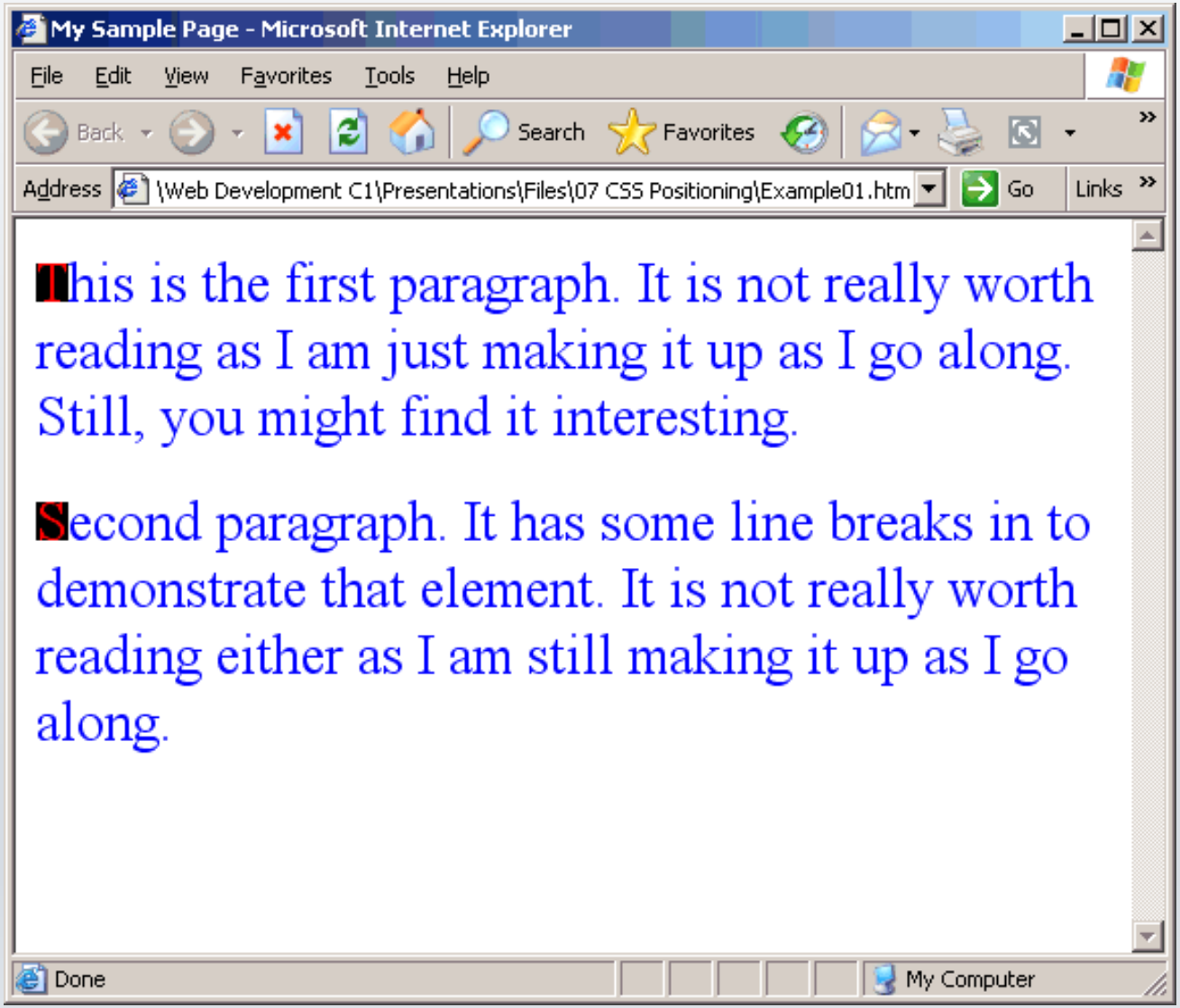
allows you to define the style of the first letter of a section of text.

Example of a stylesheet using pseudoelements.

```
p:first-letter
      {
      color: red;
      background: black;
      }

p
      {
      background: white;
      color: blue;
      font-size: 28px;
      font-family: times;
      }
```

Would give you this...



Using Stylesheets for Page Layout

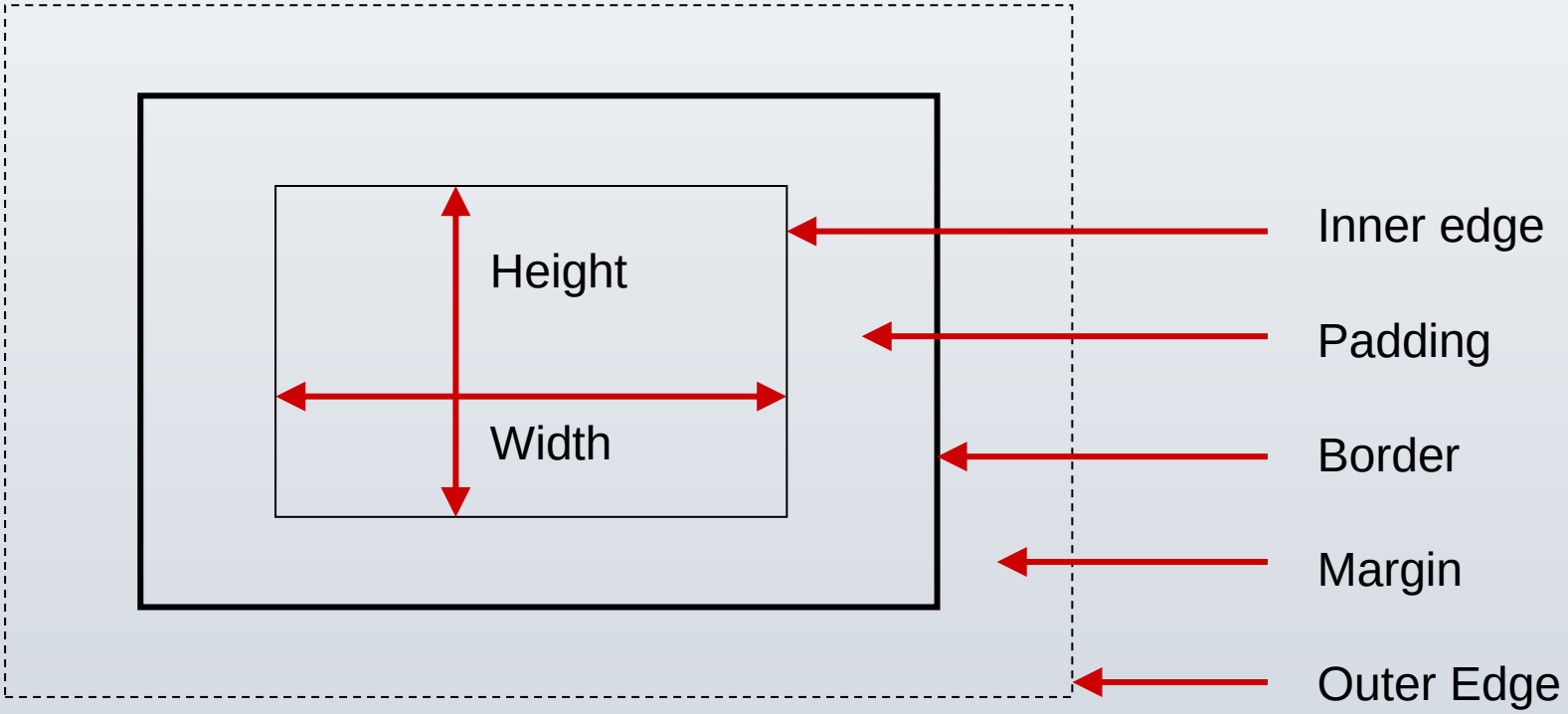
Stylesheets can be used to control page layout, not just the appearance of the text.

Each element on a webpage (image,paragraph, div) can be thought of as being inside a box.

You can control how that element is displayed by specifying measurements relating to the box.

Each box has a layout that looks like this.

Box Model



Some versions of Internet Explorer regard the height and width as applying to the whole box, not just the content area.

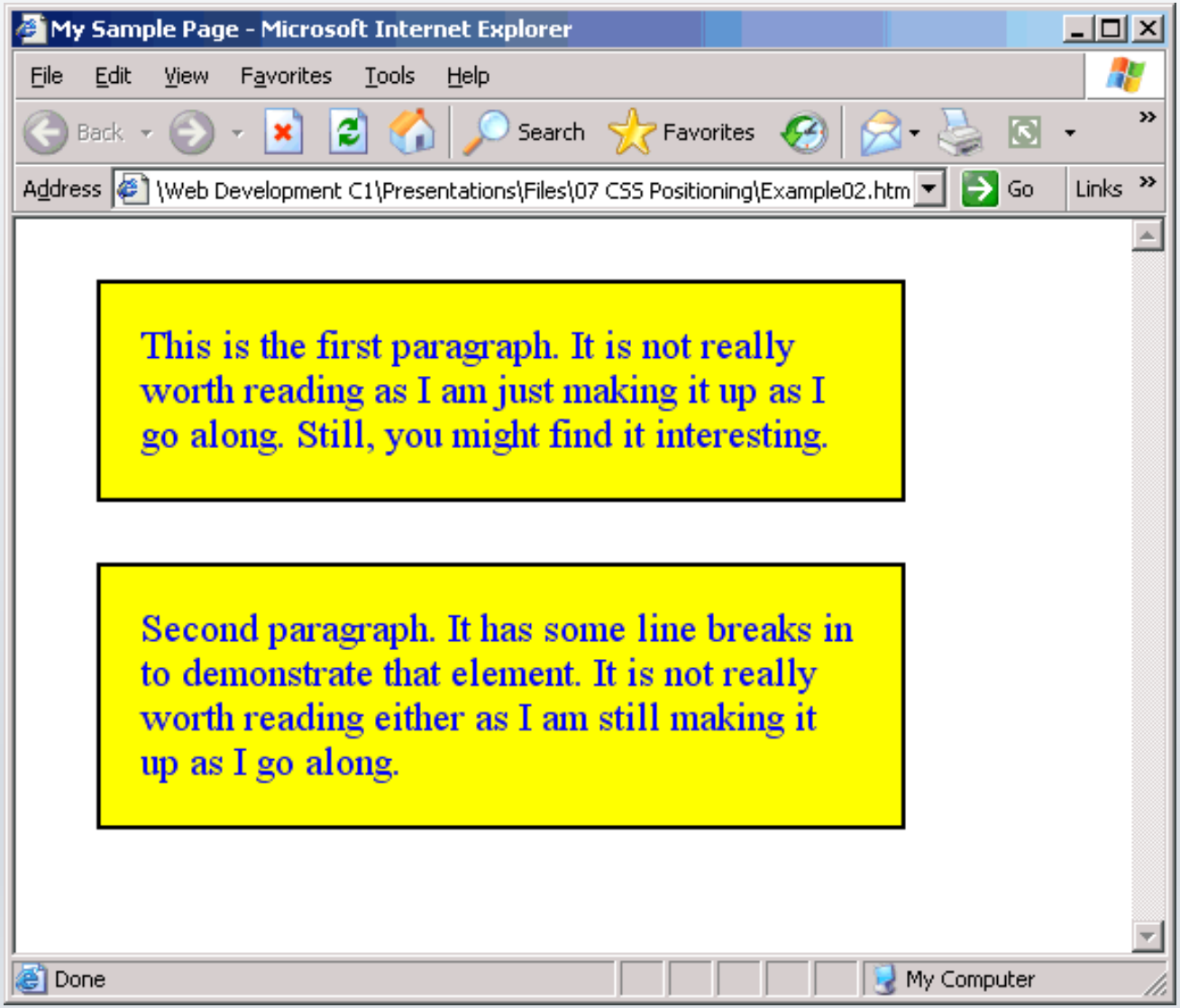
Formatting a paragraph

```
p
  {
   width: 400px;
   padding: 20px;
   margin: 30px;

   border-color: black;
   border-style: solid;
   border-width: 2px;

   background: yellow;
   color: blue;
   font-size: 20px;
   font-family: times;
  }
```

will give this



Box-related Properties

height
width

max-height
max-width
min-height
min-width

margin-right
margin-left
margin-top
margin-bottom
margin

border-top-style
border-right-style
border-bottom-style
border-left-style
border-style

border-top-width
border-right-width
border-bottom-width
border-left-width
border-width

border-top-color
border-right-color
border-bottom-color
border-left-color
border-color

border-top
border-right
border-bottom
border-left
border

padding-top
padding-right
padding-bottom
padding-left
padding

Floating

```
<body>
  <img id="image1" src="pic3.jpeg" alt="My Image">
  <img id="image2" src="pic4.jpeg" alt="My Image">
</body
```

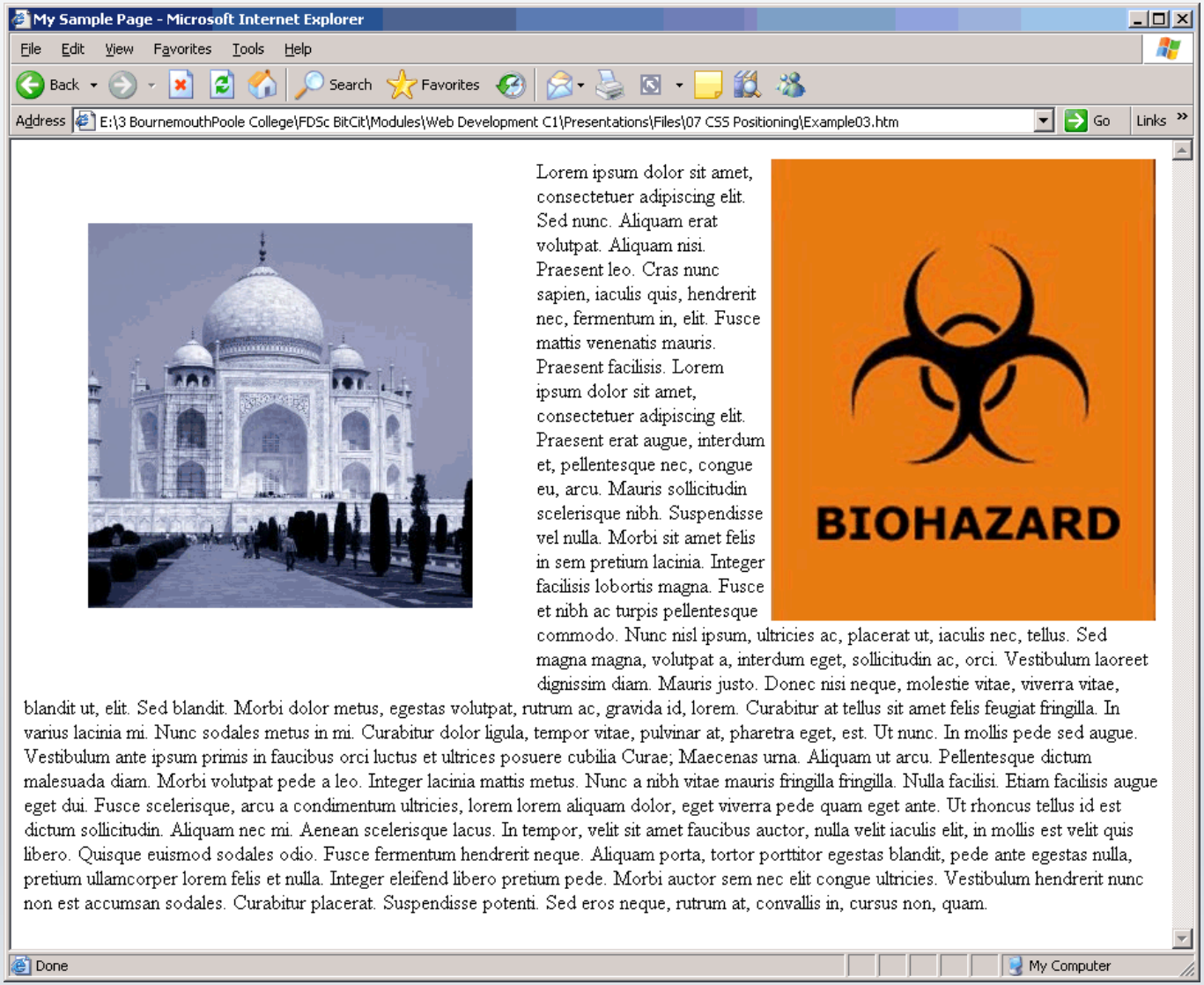
```
img#image1
{
float: right;
}

img#image2
{
float: left;
margin: 50px;
}
```

This will cause the first image to position itself on the right hand side of the page, and the second on the left.

The surrounding text will flow around them.

Will give this



Not just images

```
span.note
        {
          float: right;
          width: 200px;
          margin: 20px;
          background-color: blue;
          font-weight: bold;
        }

p
        {
          border: solid 2px red;
          padding: 30px;
        }
```

Note abbreviated use of border property.

When applied to this webpage

```
<html>
<body>
 <p>
   <span class="note">
      I am going to sit over here for a while.  Do not
      pay any attention to me.
   </span>
   Hello there how are you. Lorem ipsum dolor sit amet,
   consectetuer adipiscing elit. Aenean ornare diam eget
   nisl. Sed lacus. Cras vehicula tincidunt massa.
   Vestibulum enim. In hendrerit justo ut felis. Vivamus
   elementum
 </p>
</body>
</html>
```

Note how the element is **inside** the <p> element.

Will give this

Hello there how are you. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean ornare diam eget nisl. Sed lacus. Cras vehicula tincidunt massa. Vestibulum enim. In hendrerit justo ut felis. Vivamus

I am going to sit over here for a while. Do not pay any attention to me.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean ornare diam eget nisl. Sed lacus. Cras vehicula tincidunt massa. Vestibulum enim. In hendrerit justo ut felis. Vivamus

Layout rules.

Floated elements will generally stay within their parent elements.

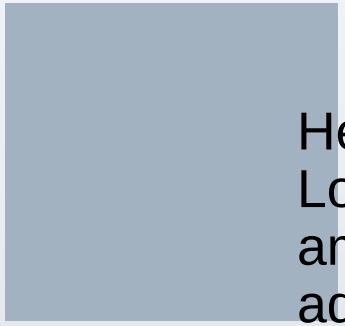
For example, the span element in the previous example is a child of the paragraph element because it is inside it.

Floated elements will generally not overlap.

This behaviour can be changed by specifying negative margins.

Negative Margin

```
img
  {
    float: left;
    margin: -10px;
  }
```



Hello there how are you. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean

Sed lacus. Cras vehicula tincidunt massa. Vestibulum enim. In hendrerit justo ut felis. Vivamus

Positioning – The Data

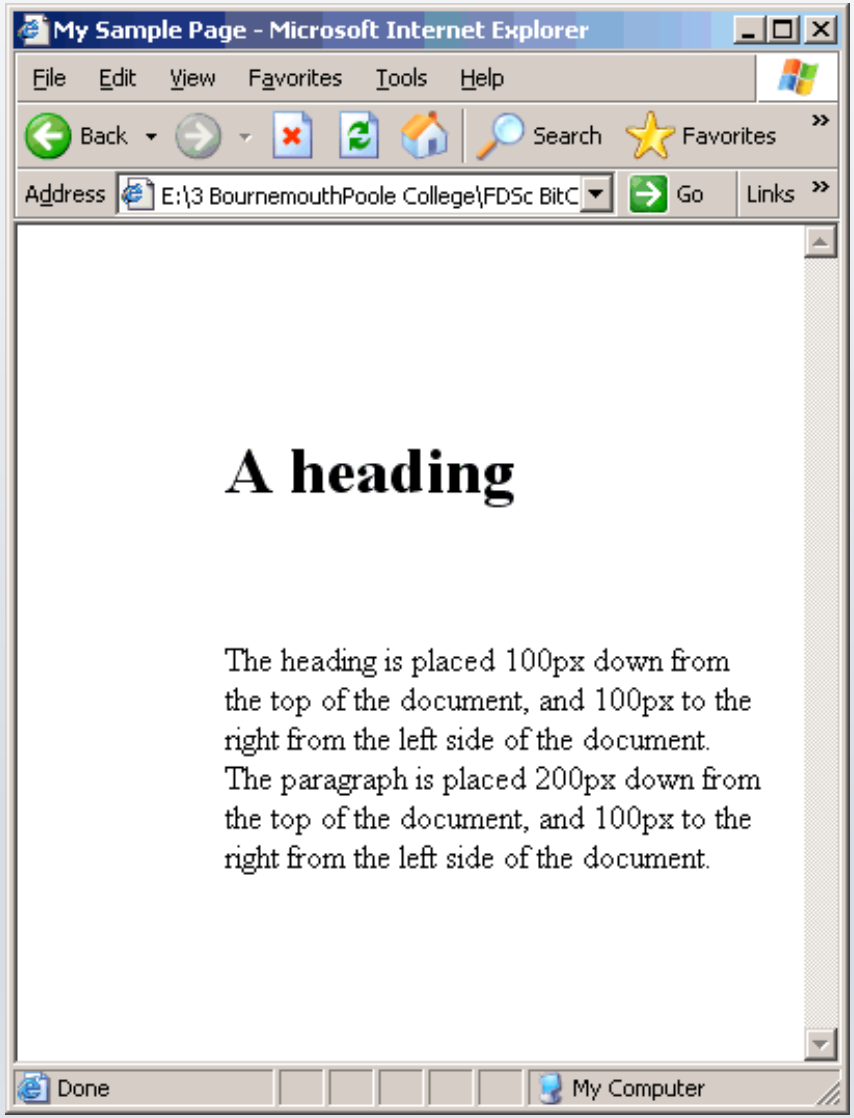
```
<html>
  <body>
    <h1>A heading</h1>
    <p>
    The heading is placed 100px
down from the top of the document,
and 100px to the right from the
left side of the document.
The paragraph is placed 200px
down from the top of the document,
and 100px to the right from the
left side of the document.
    </p>
  </body>
</html>
```

Positioning - Stylesheet

```
h1
  {
    position: absolute;
    top: 100px;
    left: 100px
  }

p
  {
    position: absolute;
    top: 200px;
    left: 100px
  }
```

will give this



Types of Positioning

static – Element boxes are placed in the document flow in the order that they appear.

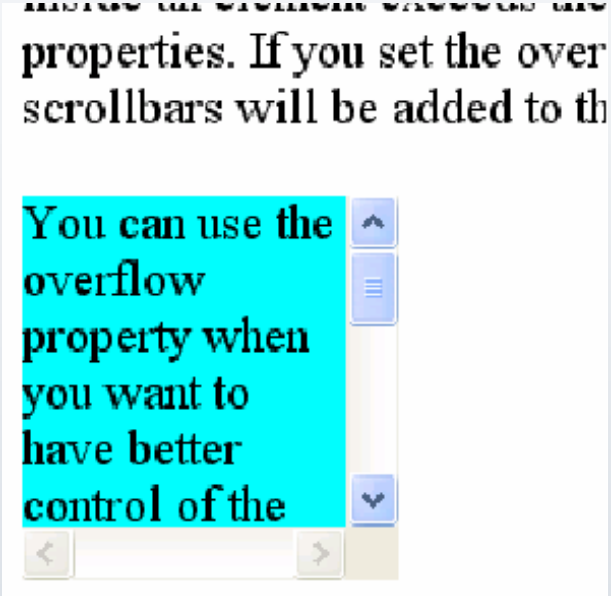
relative – Element boxes can be placed in a new position relative to their original place in the document flow.

absolute – Element boxes are positioned relative to their parent element. They are removed from the document flow and have no effect on surrounding elements.

fixed – Positioned relative to the browser window.

What if the text is too big to fit into the box?

```
div
   {
     background-color: #00FFFF;
     width:150px;
     height:150px;
     overflow: scroll;
   }
```



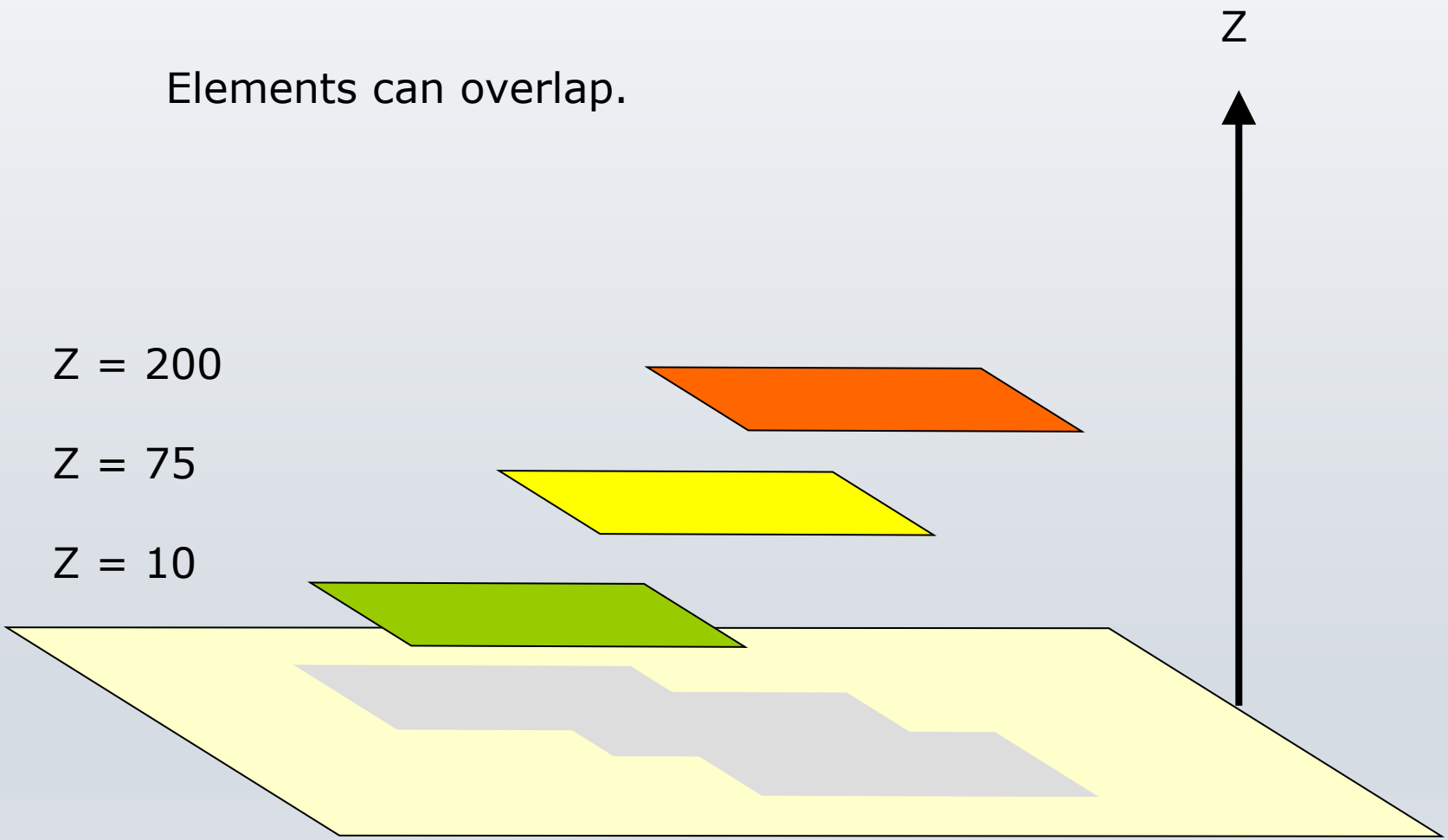
overflow: scroll; will put scroll bars bottom and side (whether they are needed or not)

overflow: auto; will put scroll bars on the side only when they are needed.

It is intended that this property should replace the non-standard iframes element that Microsoft introduced in Internet Explorer.

Stacking Order

Elements can overlap.



The html file

```
<html>
  <body>

    <p id="para1">
    This is paragraph one.  It contains some text which is absolutely
    fascinating and I would advise everyone to read it.
    </p>

    <div id="para2">
    This is paragraph two.  It is also worth a look as it contains some
    really interesting text which everyone can read.
    </div>

    <img id="image1" src="pic3.jpg" alt="biohazard"></img>

  </body>
</html>
```

The Stylesheet

```
p#para1
{
 position:absolute;
 width: 100px;
 background-color: yellow;
 left:0px;
 top:0px;
 z-index: 10;
}

div#para2
{
 position:absolute;
 width: 100px;
 background-color: olive;
 left: 50px;
 top: 50px;
 z-index: 50;
}
```

```
img#image1
{
 position:absolute;
 width: 100px;
 left: 100px;
 top: 100px;
 z-index: 100;
}
```

By changing the Z-index values...