

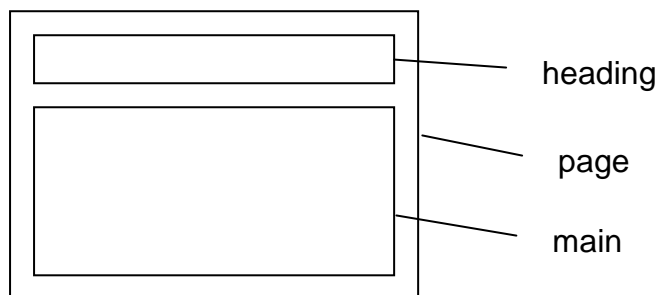
Web page design, <div>s and stylesheets

Question

How do we get round the conflicting requirements of good (visual) web page design, accessibility and separation of content and style?

Answer

Good visual design makes use of a structural approach to content. Layout on the page will be constrained to various areas, appropriately arranged. This structural approach is easy to reproduce in XHTML, without writing any content. The trick is to match the design's structure to a set of <div> tags in the page. E.g. the following simple design:



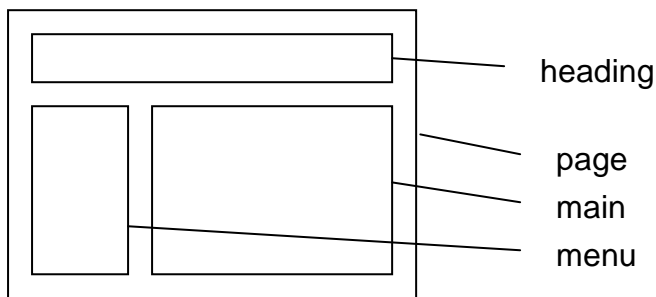
Can be matched to the following xhtml structure:

```
<?xml ...  
...  
</head>  
<body>  
<div id="page">  
  <div id="heading">  
    ...  
  </div>  
  <div id="main">  
    ...  
  </div>  
</div>  
</body>  
</html>
```

The diagram shows the mapping between the XHTML code and the visual layout. A large curly bracket on the right groups the `<div id="page">` block, which contains the `<div id="heading">` and `<div id="main">` blocks. Lines connect the labels 'heading', 'main', and 'page' to their respective code blocks.

Now we have a basic structure into which we place our web page content. We should only put content in the appropriate <div> elements. The content should not use any styling features (e.g. , or <small>) or attributes (e.g. align=center).. All this is dealt with in your style sheet. Your content should not contain any images which are purely decorative either. However, images which provide information will be included as part of your content.

Once the content is in place you can think about styling. Let's use a more complicated example:



Its structure in XHTML would be:

```
<?xml ...  
...  
</head>  
<body>  
<div id="page">  
  <div id="heading">  
    ...  
  </div>  
  <div id="menu">  
    ...  
  </div>  
  <div id="main">  
    ...  
  </div>  
</div>  
</body>  
</html>
```

The code is annotated with curly braces on the right side. A large brace groups the opening <div id="page">, the three inner div blocks, and the closing </div>. A smaller brace groups the <div id="heading"> block. Another brace groups the <div id="menu"> block. A third brace groups the <div id="main"> block. Labels 'heading', 'menu', 'main', and 'page' are placed to the right of their respective braces.

Without a style sheet, all the content you put in this structure would appear one after the other down the web page. The style sheet's job is to make the design happen.

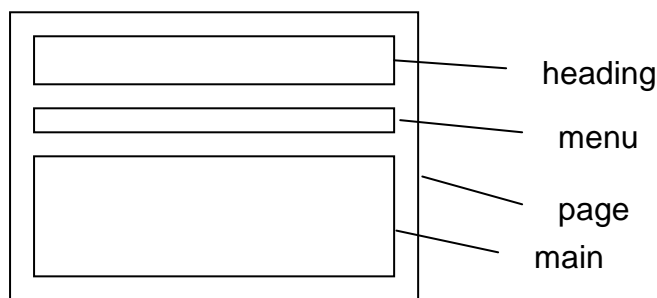
Think about the tasks the style sheet has to perform. It not only formats text and images and backgrounds, it also needs to perform the layout of the sections on the web page. The simplest set of styles to achieve the layout above will be:

```
#page {
    width: 100% /* set the page width so that */
                /* things go side by side */
}

#menu {
    width: 15em;
    float: left; /* allows the following section */
                /* to wrap along the right hand side */
}

#main {
    margin-left: 16em; /* makes sure the main content */
                      /* stays aligned at the left */
}
```

There are alternative designs which would have exactly the same XHTML structure. For example:



The style sheet for this is simpler in one respect, as the menu does not have to float to the left and the main section doesn't need a left margin. However, the menu is more complicated, as it needs to be displayed across the page.

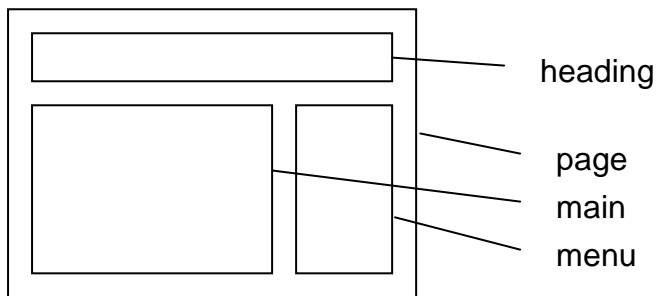
The CSS for this is quite simple:

```
#menu li {
    display: inline;
    list-style-type: none;
    padding-right: 2em;
}
```

With more effort the menu can be made more stylish.

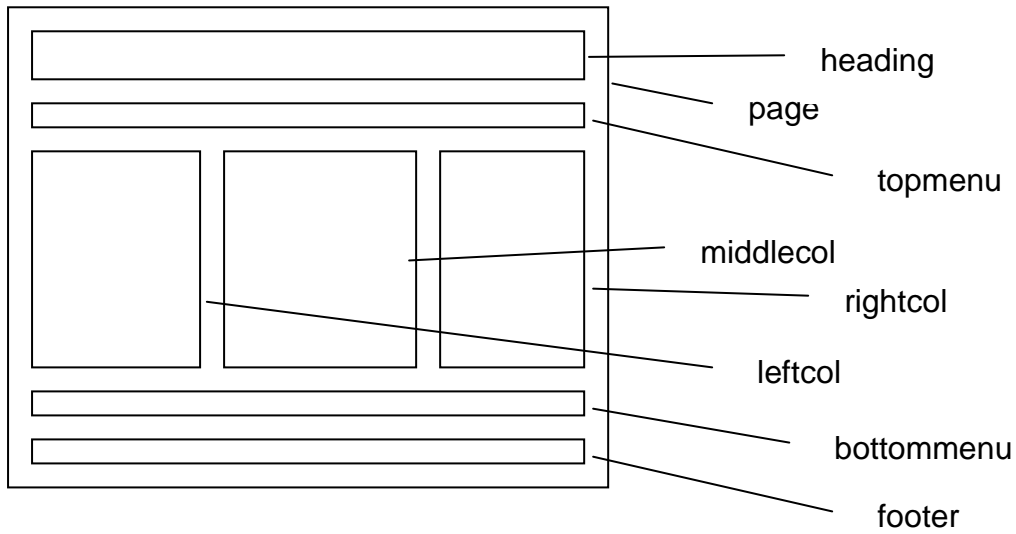
Another example would be with the menu at the right: The only change to the style sheet is to use `float: right;` rather than `float: left;` and to set the margin on the right for `#main`.

The design looks like this, but the XHTML structure is unchanged.



More sophisticated example

Look at this visual design:



In XHTML this looks like:

```
<?xml ...  
...  
</head>  
<body>  
<div id="page">  
  <div id="heading"> } heading  
  ...  
</div>  
  <div id="topmenu"> } topmenu  
  ...  
</div>  
  <div id="leftcol"> } leftcol  
  ...  
</div>  
  <div id="middlecol"> } middlecol  
  ...  
</div>  
  <div id="rightcol"> } rightcol  
  ...  
</div>  
  <div id="bottommenu"> } bottommenu  
  ...  
</div>  
  <div id="footer"> } footer  
  ...  
</div>  
</div>  
</body>  
</html>
```

page

The style sheet for this is not too complicated. Everything flows from top to bottom apart from the three columns so we can just do this:

```
#page {
    width: 100% /* set the page width so that */
                /* things go side by side */
}

#leftcol {
    width: 33%; /* a third of the page */
    float: left; /* allows the following section */
                /* to wrap along the right hand side */
}
#middlecol {
    width: 33%; /* the middle third */
    float: left; /* makes sure the main content */
                /* stays aligned at the left */
}
#bottommenu {
    clear: both;
}
```

Try this out. Its bound to have bugs. Try and fix them.