

## MPS115/MPS116 PRESENTATION LAB 8

### 1. ADVANCED HTML TEXT EDITORS

See if Notepad++ is installed on the computer; if not, install it from the Software Center. If this isn't possible, just use Notepad.

*Notepad++ is a simple and well-regarded text editor. You can have multiple tabs open at once, and it has keyboard shortcuts for launching your file in a browser (for example, Alt+Ctrl+Shift+X for Firefox and Alt+Ctrl+Shift+R for Chrome).*

Other more advanced software you could consider installing on your own device – Mac, Windows or Linux – include Visual Studio Code and Brackets; these are sophisticated editors that allow you to move between files in your site easily and much more, but we will not use them today. (I use Visual Studio Code for LaTeX, HTML and sometimes Python.)

*If you have any problems opening the source code for one of your HTML pages, try either right-clicking on your file and using 'Open with', or opening the software first (e.g. Notepad or Notepad++), then going to File, Open.*

### 2. CASCADING STYLE SHEETS

**2.1. Getting started.** Go to the course website, right-click on the page, select 'view source'<sup>1</sup> and copy and paste the HTML code into your text editor. Save it as `lab_8.html` and check that it opens in a browser. The page will have lost its formatting because it can no longer find a stylesheet. We will build one from scratch. Near the top of the HTML code is a line reading

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

Change the stylesheet URL to `href="lab_8.css"` and save the file. Now start a new text file – Notepad++ allows more than one tab open – and save it as `lab_8.css` in the same directory as your HTML file. Enter the code

```
body {  
    background-color: #ddd0c8;  
}
```

and save the file. Return to your browser, refresh the page and check that it has changed. You might have to do a 'hard refresh' to reload the CSS file, that is typically done with the shortcut `Ctrl+Shift+R`. This command has set the `background-color` property of the HTML `<body>` element to the hex value `ddd0c8`.

Now, make the browser background white by giving the `<html>` element the background colour of white. This will contrast the body of the page with the browser background.

---

<sup>1</sup>In some browsers this option may not be there. Try Firefox or Chrome.

*In general on web pages you should use a small number of contrasting and complementary colours, there are tools on the web for creating suitable colour schemes or palettes, but we will not look at that now.*

Let's find a picture to fill the gap on the page. You can use any search engine but we will illustrate how to check licensing with Google. Using Google's image search, search for pictures of pythons.

- (1) Change the 'Usage rights' under the Tools option to 'Creative Commons licenses' and search again.
- (2) Choose one of the pictures of a python, and check the licensing details by clicking on the image then 'License details'.
- (3) If it can be used freely, then right-click and save it in your folder as `mps115.jpg`.
- (4) Refresh your webpage to make sure the image appears. You might want to set the `width` attribute of the image in the HTML to make it an appropriate size.

*Every image that is created or photo that is taken has a copyright belonging to the author, and they can control how the picture is used. Any image that you use on a website must be free for use. Often, you will need to attribute the creator of the image. You can use the <small> tag for that.*

By looking at the stylesheet for the MPS115/MPS116 webpage found at [http://maths-skills.group.shef.ac.uk/css/course\\_pages.css](http://maths-skills.group.shef.ac.uk/css/course_pages.css), try to find out how to do the following by adding style specifications to the `body` selector in your `lab_8.css` file.

- (5) Change the text colour (`color`) of everything (except the links) to `darkblue`.
- (6) Limit the width of the body to 820 pixels.
- (7) Give the body a `double` border in black.
- (8) Pad the body of your document by 2em (that is, two font-widths).
- (9) Try to get the body of the document central on the page.

All of these effects are used on the course webpage, so keep experimenting until you find the right commands.

**2.2. Margins, borders and padding.** HTML builds up pages by means of *boxes*. That is, every HTML element that you put on a page comes with an invisible box around it. Every box has an associated *margin*, *border* and *padding*. Add the following to your CSS file.

```
h1 {
    margin: 0.1em;
    padding: 1em;
    border-width: 3px;
    border-color: #000;
    border-style: solid;
    background-color: #eee;
}
```

This will affect the main page heading. There should now be a black border of 3 pixels around the heading. Also there is a margin of 0.1 em outside of the border, and padding of 1 em inside the border. These margin, border

and padding specifications can be altered one side at a time by specifying `margin-left` or `padding-top`, for example.

- (10) Give the image a 2 pixel `outset` border in black. Give it 2 pixels of padding and change its background colour to `#eee`.
- (11) Try to centre the image with respect to the body. (Hint: the same method that you used to get the body centred works here, but you need to also change the image's `display` property to `block`.)

HTML elements come in two types: *inline* and *block*.

Block elements are things like paragraphs (`<p>`) and headings (`<h1>`, `<h2>`, etc.), which always start on a new line and take up the full width available.

Inline elements are things like images (`<img>`) and emphasised text (`<em>`) which display inline (like using single-dollars in L<sup>A</sup>T<sub>E</sub>X). The `display` property in CSS can switch this type, useful for centring images.

An image is centred by changing its margins to 'auto', which makes the browser calculate equal margins on left and right. On the other-hand, things inside block elements (such as text in a paragraph) are centred using `text-align: center`.

**2.3. Sectioning and styling.** If you look carefully at the HTML file you will see that it has the following schematic form, where I've ignored closing tags for clarity.

```

<html>
  <head>
    <title>
  <body>
    <header>
      <h1>Page title
    <main>
      <section>
        <h2>Course materials
        <table>
      <section>
        <h2>Extras
        <section>
          <h3>General materials
        <section>
          <h3>Python materials
        <section>
          <h3>LaTeX
        <section>
          <h3>HTML and web design
    <footer>
  
```

This uses HTML *semantic* elements like `<header>`, `<main>`, `<section>` and `<footer>` to give meaningful structure to the document; these elements do not immediately make any difference to how the document looks, however we can use CSS to style these different parts of the document.

Add the following to your CSS file. (You can speed this up with some copying and pasting!)

```
section {
```

```

border-width: 1px;
border-color: #000;
border-style: solid;
margin-top: 1em;
padding-left: 2em;
padding-right: 2em;
padding-top: 0.1em;
padding-bottom: 0.5em;
background-color: #fff;
}

```

Refresh your browser and see the difference this has made.

Now suppose we want to display the four subsections inside the Extras section differently, these are section elements so at first glance this would seem to not be obvious. However, we can use the fact that each is a `<section>` element within a `<section>` element. Try the following code which gives a section inside a section the additional properties to those above.

```

section section {
  margin-bottom: 1.5em;
  background-color: #eee;
  border-color: #ddd0c8;
}

```

Similarly we could apply this to an `<h2>` element inside a `<header>` element with the “`header h2`” selector.

**2.4. Classes.** Look at the HTML for the course website, inside the `<table>` element for the table of course materials you will see that the week numbers are inside *table data* or `<td>` elements, for instance you should see

```
<td class="week">2</td>
```

The name ‘week’ is one I made up and this class attribute allows us to style these table data elements differently to other table data elements. Add the following to your CSS to centre and embolden the week numbers. Note that the class name in the CSS starts with a full-stop.

```

.week {
  text-align: center;
  font-weight: bold;
}

```

In the HTML file find the subtitle that states the lecturers’ names. Notice that it is an `<h2>` element, but it has been given the class ‘subtitle’.

- (12) Give properties to the subtitle class in your CSS file to ensure that the lecturers names are centred and the `font-style` is `oblique` (i.e., slanted).
- (13) Change the selector from `.subtitle` to `header h2`. Does this have the same effect? Why?

**2.5. Tables.** Let’s style the table. There are various HTML elements in a table like the table header `<thead>`, the table body `<tbody>` and table rows `<tr>`, here we will style the table data entry `<td>` elements and the table header entry `<th>` elements. Put the following in your CSS file.

```
table {  
    border-collapse: collapse;  
    background-color: #eee;  
    margin-bottom: 1em;  
}  
  
td, th {  
    padding: 0.4em;  
    border-width: 0px;  
    border-bottom-width: 1px;  
    border-style: solid;  
    border-color: black;  
}  
  
th {  
    border-top-width: 2px;  
    border-bottom-width: 2px;  
}
```

Note that the selector “`td, th`” means that the properties are applied to *both* `<td>` and `<th>` elements.

Play around with this a little to see the differences you can make.

**2.6. Navigation elements.** Last week you created a menu (or *navigation bar*) at the top of your page using some plain HTML. What you probably didn’t realise was that the CSS file you added controlled how it looked. The `<nav>` element is another HTML semantic element.

Find last week’s lab attempt (or download the finished HTML and CSS files from the course webpage) and play around with the look of the menu. The page [http://www.w3schools.com/css/css\\_navbar.asp](http://www.w3schools.com/css/css_navbar.asp) will give much more on making navigation bars.

### 3. FURTHER READING

We have only covered the basics of CSS, but already these ideas can do a lot. If you have got this far, you can look at the HTML and CSS tutorials made by HTML Dog available at <http://www.htmldog.com/guides>.