HTML & CSS
Brief Introduction and Exercises

Chris Sparks

HTML and CSS are useful to understand when working on digital editions, since they are very frequently published online, and hence often make use of generated HTML and CSS. It is good to understand what is going on.

A simple way of thinking about HTML and CSS is that HTML defines the *content* of a page – the text and images that go in it, whereas CSS defines it's *appearance*, including what fonts are used, what colours are employed, and where things appear on the page.

At this early stage, HTML and CSS also provide a way of playing and experimenting with a more forgiving XML-like language, and a very quick way to see results.

Exercises Part I: Writing HTML

In this exercise, you will create a simple HTML page. Open the page at http://jsfiddle.net/sparkyc84/wd288kcu/ – this gives you an online HTML development environment that lets you write code and quickly see results.

1. Add Paragraphs

- Type (or copy/paste) the following two paragraphs to the end of the HTML in the top left **HTML** section of the editor, after the `</ul>` but before the `</section>`:

> You do not need to know every detail of HTML to work with pages generated from TEI, but it will help if you are aware of some basic elements.
>
> A good way of familiarising yourself with HTML is to click 'View', and then 'Page Source' in your web browser.

- Once you have typed these paragraphs in, add `<p>` tag at the beginning of the first, before 'You'. Delete the closing `<p>` tag that is (un)helpfully added automatically.
- Now add a closing `</p>` tag after 'elements.'
- Next, add an opening `<p>` tag before 'A good' (and delete the automatically-added closing tag)
- And finally a closing `</p>` after 'web browser.'

Once you have added these paragraphs, click the **[> Run]** button in the top left of the browser. Check the bottom right panel to see whether it worked.

2. Add a Link

A key feature – in fact some would say the *defining* feature of HTML is the hyperlink. Here, we will make the word 'DEMM' link to the DEMM website. This will make use of the `<a>` (anchor) element and its 'href' attribute:

- First, delete the word 'DEMM' (we could surround the existing word with tags, but in this editor, it will be quicker to re-type the word)
- Next, add an opening `<a>` tag. Notice that a closing `</a>` tag is added automatically/
- After the first `<a`, type a space, then `href="`. Note that the closing quotation marks (") are added automatically.
- Inside the `href=""` tags, type `http://www.digitalmanuscripts.eu/` , so that it looks like this:
  href="http://digitalmanuscripts.eu"
- Next, add back in the word 'DEMM' between the `<a></a>` tags, so that it all looks like this:
  `<a href="http://www.digitalmanuscripts.eu">DEMM</a>`
- Click [> Run].

When you click on the DEMM link in the bottom-right preview pane, you should now be taken to the DEMM website.

3. Working with Lists

Add a list item, as follows:

- Type (or copy/paste) the following line after the last `</li>` but before the `</ul>`
  Like XML, HTML does not care about white space
- Surround it with `<li>` at the start, and `</li>` at the end of the line.
- Click [> Run] to see whether or not you have been successful.

The page currently contains a `<ul>` element. This describes an **u**nordered list, containing a series of `<li>` elements, each making up a separate list item. To make our list clearer, and so that we can refer to it later, we will make it into an **o**rdered **l**ist. To do this, all we have to do is the following:

- replace the opening `<ul>` element with an `<ol>` element
- This will immediately display an error, as the `<ol>` element isn't closed, and the old `</ol>` is still in place.
- Replace the closing `</ul>` with a closing `</ol>`, and the error should be resolved

Have a go, then click **[> Run]**

4. Being emphatic

`<em></em>` tags are used to surround text that should be given emphasis within a sentence. Generally, they are equivalent to italics in printed text. We'll edit the third list item so it reads as follows (note the italics):

It describes *content* rather than *appearance*

To do this:

- First, place the cursor before the word 'content' and type `<em>`
- The editor will (un)helpfully immediately insert a closing `</em>`. We'll now delete it (since we don't want it here), but this will create an error.
- Next, go the the end of the word 'content' and type `</em>`. This will close the `<em>` tag, and resolve the error we just created in the previous step.
- Repeat with the word 'appearance'
- Click [> Run] to check all this has worked

Exercises Part II: Writing CSS

Colours

We want emphasised text to be really striking – make it red using a CSS rule.

- In the empty CSS tab, type:

```
em {
  color: DarkMagenta;
}
```

Note that the editor will helpfully complete some of this for you automatically…

Click **[> Run]** and observe the result.

Fonts

Change the font for your paragraphs:

- Type the following at the end of the CSS file:

```
p {
    font-family: Helvetica, Verdana;
}
```

Click **[> Run]** and observe the result.

Give the first paragraph a pink background. This time we will select using the first paragraph using its unique ID: Type the following:

```
#first {
    background-color: pink;
}
```

- Click **[> Run]** and observe the result.

Borders

Make the . list items have a green dotted border:

- Type the following in the CSS tab:

```
li {
    border-width: 2px;
    border-style: dotted;
    border-color: green;
}
```

  - Click **[> Run]** and observe the result.

Layout

We'll add some more margins between paragraphs:

- Type the following in the CSS tab:

```
p {
    margin: 10px;
}
```

- Click **[> Run]** and observe the result.

If you finish this, try adding some more elements and text from the examples on this worksheet.

HTML – A Primer

The following lists some basic HTML elements. Elements are perhaps best described as "things" within your page, like paragraphs, links, lists or areas of emphasis within text. Some elements contain text and other elements, for example a paragraph might contain a link.

Elements normally have an opening **tag** that marks their start: `<p>`, and a closing tag that marks their end `</p>`. Some elements, like images, are self-contained and so are self-closing: `<img />`.

Elements can have **attributes**, which are specified as `attribute="[value]"`. Common attributes include `href=""` (link reference) `src=""` (image source) and `id=""` and `class=""` used for labelling elements so that styles can be applied to them.

The following lists some basic tags, to look up others, see: http://www.w3schools.com/tags/

Element	Description
`<h1></h1>`	defines a **level-one heading.** (See also h2-h6).
`<p></p>`	defines **paragraph** of text.
`<br />`	adds a **line break** within a paragraph. Inter-paragraph spacing is not applied and the text before and after the break is still treated as one block when applying things like font styles.
`<ol></ol>`	defines an **ordered list** of text, whose items are numbered sequentially (other markers can also be used).
`<ul></ul>`	defines an **unordered list** of text, items are simply marked out with bullet points.
`<li></li>`	defines an individual **list item** within either a <ul></ul> or <ol></ol> block.
`<em></em>`	demarcates text that should be given **emphasis** (italicised by default in most browsers).
`<strong></strong>`	demarcates text that is **important** (made bold by default in most browsers).
`<a href="[HREF]"></a>`	defines an **anchor**, which can take various forms. Most commonly, it is used with the href (hyperlink reference) attribute to the page specified by the [HREF] attribute.
`<img src="[SRC]" />`	defines an **image** at the address specified in the [SRC] attribute.

HTML: Some newer, "semantic" elements

The latest version of HTML defined some new elements. They include:

Element	Description
`<article></article>`	defines a self-contained composition in a page, for example a blog post.
`<nav></nav>`	defines a **navigation** element within a page, that contains primarily links to other pages or parts of the same page.
`<section></section>`	represents a thematic grouping of content (or **section**) within a page.
`<heading></heading>`	defines a section heading, which may itself contain headings and sub-headings.

And finally… not really a tag, but something that you might see in HTML:

`<!--[comment]-->`	defines a HTML comment – for text that is a sort of note-to-self for the programmer. Anything between the `<!--` and the closing `-->` is assumed to be a comment and ignored by the browser when it outputs the page to your screen.

CSS – A primer

A CSS file is made up of sets of selectors and declarations. Selectors, surprisingly, select an HTML element within the page to apply rules to. Declarations set out rules to apply to those elements. A typical block might look like this:

```
p {
    font-family: "Times New Roman";
    color: red;
}
```

Selectors

Selectors take three basic forms, outlined below.

`p`	Selects an HTML element. Replace p with any other HTML tag, like `a`, or `article` or `strong`.
`.class`	Selects any HTML element with the `class="[class]"` attribute set on it. It's the full-stop that is important here.
`#id`	Selects the single HTML element with the specified ID, set using `id="[id]"`

You can find more about selectors at: http://www.w3schools.com/css/css_selectors.asp

Declarations

Declarations are formed of sets of property: value pairs. Property defines the feature you are trying to manipulate (font style, background colour and so forth), whilst value defines what you are trying to set it to. The property:value pairs are separated by semicolons.

There are a huge range of declarations, but here are some basic examples:

Property	Example Values	Description

`font-family:`	`Verdana,`
	`Arial`
	`sans-serif`
		Change the font used. Not all computers have every font installed, so you can specify a comma-separated list of fonts, which the computer will try in order, using the first one it finds installed

`font-size:`	`2em`
	`1px`
	`large`
		The size of letters. Can be specified in a range of units.

`color:`	`#ffffff`
	`white`
	`pink`
	`green`
		Sets the **foreground** colour of the specified element. Colours can be specified by common names or using hexadecimal codes – sets of letters and numbers preceded by a hash sign. No-one has hexadecimal codes memorised, but there are plenty of tools available – just Google for "html color picker".

`background-color:`	`orange` `#ffcc11`	Sets the **background** colour of the specified element
`border-width:`	`1px`	Sets the width of the specified element's border.
`border-color:`	`Red`	Sets the colour of the specified element's border.

`border-style:`	`dashed`
	`dotted`
	`solid`
		Sets the type of line used in the border.

`padding:`	`12px`
	`3em`
		Sets the amount of space between an element's borders and its contents.

`margin:`	`10px`
	`2em`
		Sets the amount of space between an element and other (block-level) elements.

And finally… neither a selector nor a declaration, but something that you might see in CSS.

`/*`
`[comments]`
`*/`
	A multi-line comment. Everything between the `/*` and `*/` is assumed to be notes-to-self by the author, and ignored by the web-browser.

`// [comments]`	Single-line version of a comment. Everything after `//` is ignored by the browser.

For more on CSS, go to: http://www.w3schools.com/css/default.asp