Web typography and CSS font styling

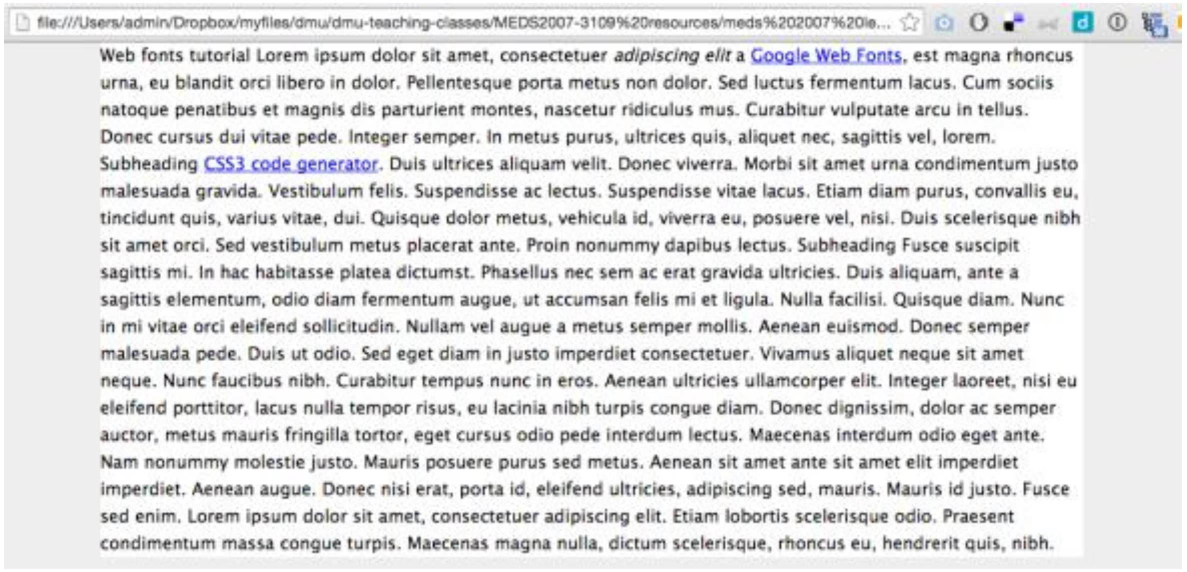
Author: Dave Everitt
Updated: 29 Jan 2017
Files: https://github.com/MEDS2007/web-typography-tutorial

In this exercise you will gain an understanding of typography for the web, and explore the various font controls available in CSS.

1. Download the files from Blackboard

1.1 Drag the folder 'webtypography' to the icon for *Atom*. It contains the 'index.html' and 'styles.css' files you'll be working on. Atom will open up the contents of the folder.

1.2 Drag 'index.html' to a web browser. You'll see a page of plain, unformatted text:


file:///Users/admin/Dropbox/myfiles/dmu/dmu-teaching-classes/MEDS2007-3109%20resources/meds%202007%20le...

Web fonts tutorial Lorem ipsum dolor sit amet, consectetuer *adipiscing elit* a Google Web Fonts, est magna rhoncus urna, eu blandit orci libero in dolor. Pellentesque porta metus non dolor. Sed luctus fermentum lacus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur vulputate arcu in tellus. Donec cursus dui vitae pede. Integer semper. In metus purus, ultrices quis, aliquet nec, sagittis vel, lorem. Subheading CSS3 code generator. Duis ultrices aliquam velit. Donec viverra. Morbi sit amet urna condimentum justo malesuada gravida. Vestibulum felis. Suspendisse ac lectus. Suspendisse vitae lacus. Etiam diam purus, convallis eu, tincidunt quis, varius vitae, dui. Quisque dolor metus, vehicula id, viverra eu, posuere vel, nisi. Duis scelerisque nibh sit amet orci. Sed vestibulum metus placerat ante. Proin nonummy dapibus lectus. Subheading Fusce suscipit sagittis mi. In hac habitasse platea dictumst. Phasellus nec sem ac erat gravida ultricies. Duis aliquam, ante a sagittis elementum, odio diam fermentum augue, ut accumsan felis mi et ligula. Nulla facilisi. Quisque diam. Nunc in mi vitae orci eleifend sollicitudin. Nullam vel augue a metus semper mollis. Aenean euismod. Donec semper malesuada pede. Duis ut odio. Sed eget diam in justo imperdiet consectetuer. Vivamus aliquet neque sit amet neque. Nunc faucibus nibh. Curabitur tempus nunc in eros. Aenean ultricies ullamcorper elit. Integer laoreet, nisi eu eleifend porttitor, lacus nulla tempor risus, eu lacinia nibh turpis congue diam. Donec dignissim, dolor ac semper auctor, metus mauris fringilla tortor, eget cursus odio pede interdum lectus. Maecenas interdum odio eget ante. Nam nonummy molestie justo. Mauris posuere purus sed metus. Aenean sit amet ante sit amet elit imperdiet imperdiet. Aenean augue. Donec nisi erat, porta id, eleifend ultricies, adipiscing sed, mauris. Mauris id justo. Fusce sed enim. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis scelerisque odio. Praesent condimentum massa congue turpis. Maecenas magna nulla, dictum scelerisque, rhoncus eu, hendrerit quis, nibh.

2. Look at the CSS code

2.1 In 'styles.css', look at the simple CSS styles for the html **`body`** tag, and for **`main`** tag containing the text. This exercise only concerns the **typography**, not the page layout, so you can ignore most of these.

Styles for the **`body`** tag affect the whole page, unless other styles further down override (replace) them. You can ignore most of these.

However, the value for the **`font-family`** property contains a series of names listing Windows and Mac versions of a popular font, and a final generic 'fallback' (**`sans-serif`**) in case neither is available on the computer of a visitor to your website:

```
"Lucida Sans Unicode", "Lucida Grande", sans-serif
```

When found on the website visitor's computer, a font from the above list will 'cascade' via the CSS throughout the rest of the HTML document, unless there is another **`font-family`** declaration further down in the CSS file.

The basic CSS styles for the heading and paragraph tags (**`p`**, **`h1`**, **`h2`**) contained in the **`main`** tag are also set up, but the styles will have *no effect* at this point because the text itself in the HTML file is not yet *marked up* with any HTML paragraph or heading tags (you'll do this in a moment).

NOTE: The **`serif`** and **`sans-serif`** values of the CSS **`font-family`** property are the two *generic* font styles—so if you specify one or more *sans-serif* fonts, like 'Arial Narrow' or 'Futura' and these are *not installed* on the user's computer, the generic *sans-serif* at the end of your list will fall back the user's default sans-serif font.

3. Understanding basic CSS selectors

The CSS rules already set up will affect all HTML **`h1`**, **`h2`** and **`p`** tags on the whole page. For this typography exercise that's fine, but on a real website you may also have (say) **`h2`** tags or paragraphs inside another HTML element, for instance a **`nav`**, **`aside`** or **`blockquote`** tag. You would then need to use *more specific* CSS selectors.

For instance, if **`h2`** and **`p`** tags are contained somewhere else on the page in an **`aside`** tag, you could target them like this and change their colour:

```
aside h2 {
  color: #335
}

aside p {
  color: #555;
}
```
NOTE: `aside` tags are sometimes used for 'pull quotes'—boxes that extract and highlight a bit of the main text.

You will use this technique later, to change the style of paragraphs in a `blockquote` tag.

So, when writing CSS secectors, **be as specific as necessary but no more**—i.e. you never need to put `body` before `aside h1`.

4. Mark up the headings with HTML tags

4.1 The text in 'index.html' consists of long chunks and three short lines. In a moment, you'll mark up the short lines as headings, and the long chunks as paragraphs. All the text in the HTML code is indented to show that it is *contained* in the `main` tag.

NOTE: ***Indentation*** is used throughout computer programming—in HTML code it shows how tags are *contained* inside other tags, and in CSS it shows a set of *style rules* inside CSS *declaration block*.

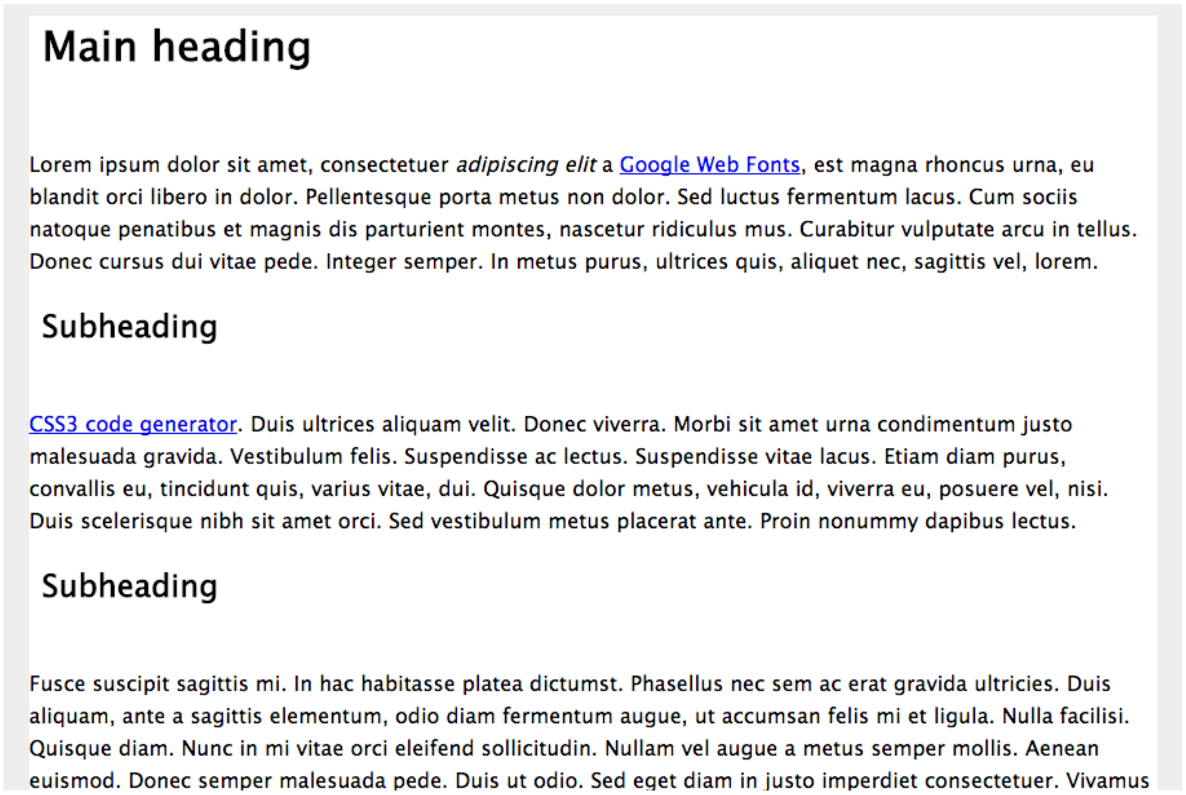
4.2 Add `<h1>` to the beginning of the first line immediately before the text, or the heading will have a space before it. Then, at the end of the line **close the tag** with `</h1>`:

```
<h1>Main heading</h1>
```

4.3 Similarly, wrap `<h2>` tags around the two 'subheading' lines:

```
<h2>Subheading</h2>
```

4.4 Reload the page in your browser (keyboard shortcut: Cmd-R). You should see the unformatted headings, complete with their basic CSS styles:



Main heading

Lorem ipsum dolor sit amet, consectetuer *adipiscing elit* a Google Web Fonts, est magna rhoncus urna, eu blandit orci libero in dolor. Pellentesque porta metus non dolor. Sed luctus fermentum lacus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur vulputate arcu in tellus. Donec cursus dui vitae pede. Integer semper. In metus purus, ultrices quis, aliquet nec, sagittis vel, lorem.

Subheading

CSS3 code generator. Duis ultrices aliquam velit. Donec viverra. Morbi sit amet urna condimentum justo malesuada gravida. Vestibulum felis. Suspendisse ac lectus. Suspendisse vitae lacus. Etiam diam purus, convallis eu, tincidunt quis, varius vitae, dui. Quisque dolor metus, vehicula id, viverra eu, posuere vel, nisi. Duis scelerisque nibh sit amet orci. Sed vestibulum metus placerat ante. Proin nonummy dapibus lectus.

Subheading

Fusce suscipit sagittis mi. In hac habitasse platea dictumst. Phasellus nec sem ac erat gravida ultricies. Duis aliquam, ante a sagittis elementum, odio diam fermentum augue, ut accumsan felis mi et ligula. Nulla facilisi. Quisque diam. Nunc in mi vitae orci eleifend sollicitudin. Nullam vel augue a metus semper mollis. Aenean euismod. Donec semper malesuada pede. Duis ut odio. Sed eget diam in justo imperdiet consectetuer. Vivamus

Oops! The top of the `main` tag has been pushed down from the top of the browser window so the body background colour is showing at the top of the page. Browsers apply **default settings** for margins, and the default margin is causing the problem. Let's fix it.

4.5 In the `h1` CSS rule, click after the semicolon, hit return for a new line, and type in another CSS rule to set all the `h1` margins to zero:

```
h1 {
  padding: 0 10px 1em;
  margin: 0;
}
```

Reload the page—the margin (space) above the `h1` heading will have gone.

5. Mark up the text with paragraph tags

5.1 In the HTML file, find the first long chunk of text (the one that starts 'Lorem ipsum dolor sit amet ') and wrap it in paragraph tags:

```
<p>Lorem ipsum dolor sit amet, consectetuer <em>adipiscing elit</em>
a <a href="https://www.google.com/fonts/">Google Web Fonts</a>, est
magna rhoncus urna, eu blandit orci libero in dolor. Pellentesque
porta metus non dolor. Sed luctus fermentum lacus. Cum sociis natoque
penatibus et magnis dis parturient montes, nascetur ridiculus mus.
Curabitur vulputate arcu in tellus. Donec cursus dui vitae pede.
Integer semper. In metus purus, ultrices quis, aliquet nec, sagittis
vel, lorem.</p>
```

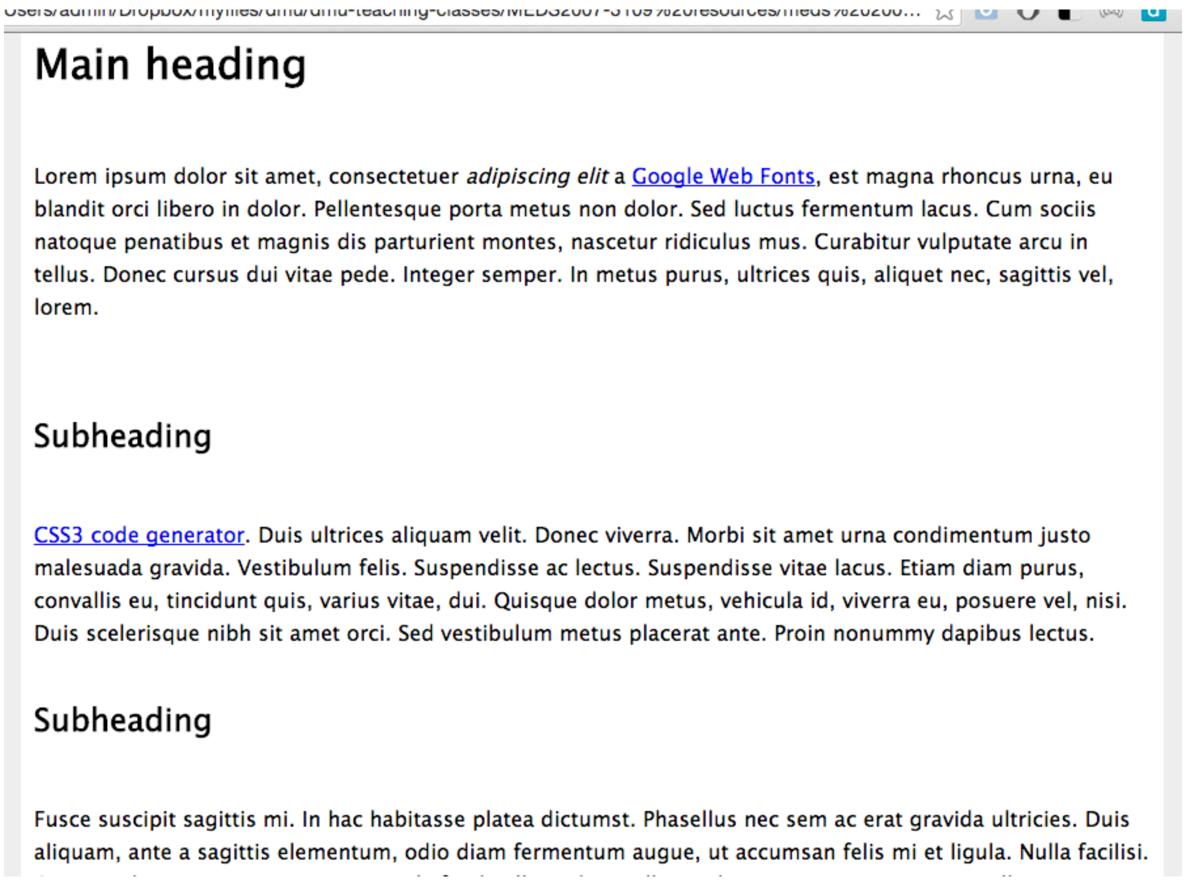
Do the same for the other chunks of text—because each tag describes what it contains—the page now is made up of **well-formed HTML**!

So there is **one level 1 heading** (Google likes *meaningful* level 1 headings that describe the page contents), then some **paragraphs**, divided by **level 2 headings**.

If the paragraphs of text had logical subsections inside the `h2` headings, we could *nest* (and style) level 3 (`h3`) headings to further divide them under each `h2` heading, keeping the structure of the text in order.

NOTE: This is one of the principles of what is known as *well-formed HTML*, which is good practice for usability, accessibility and machine-readability, and also for the *semantic web*. It also creates *valid* HTML coding.

Reload the page in your browser—the paragraphs now have the left-right CSS padding:



Main heading

Lorem ipsum dolor sit amet, consectetuer *adipiscing elit* a Google Web Fonts, est magna rhoncus urna, eu blandit orci libero in dolor. Pellentesque porta metus non dolor. Sed luctus fermentum lacus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur vulputate arcu in tellus. Donec cursus dui vitae pede. Integer semper. In metus purus, ultrices quis, aliquet nec, sagittis vel, lorem.

Subheading

CSS3 code generator. Duis ultrices aliquam velit. Donec viverra. Morbi sit amet urna condimentum justo malesuada gravida. Vestibulum felis. Suspendisse ac lectus. Suspendisse vitae lacus. Etiam diam purus, convallis eu, tincidunt quis, varius vitae, dui. Quisque dolor metus, vehicula id, viverra eu, posuere vel, nisi. Duis scelerisque nibh sit amet orci. Sed vestibulum metus placerat ante. Proin nonummy dapibus lectus.

Subheading

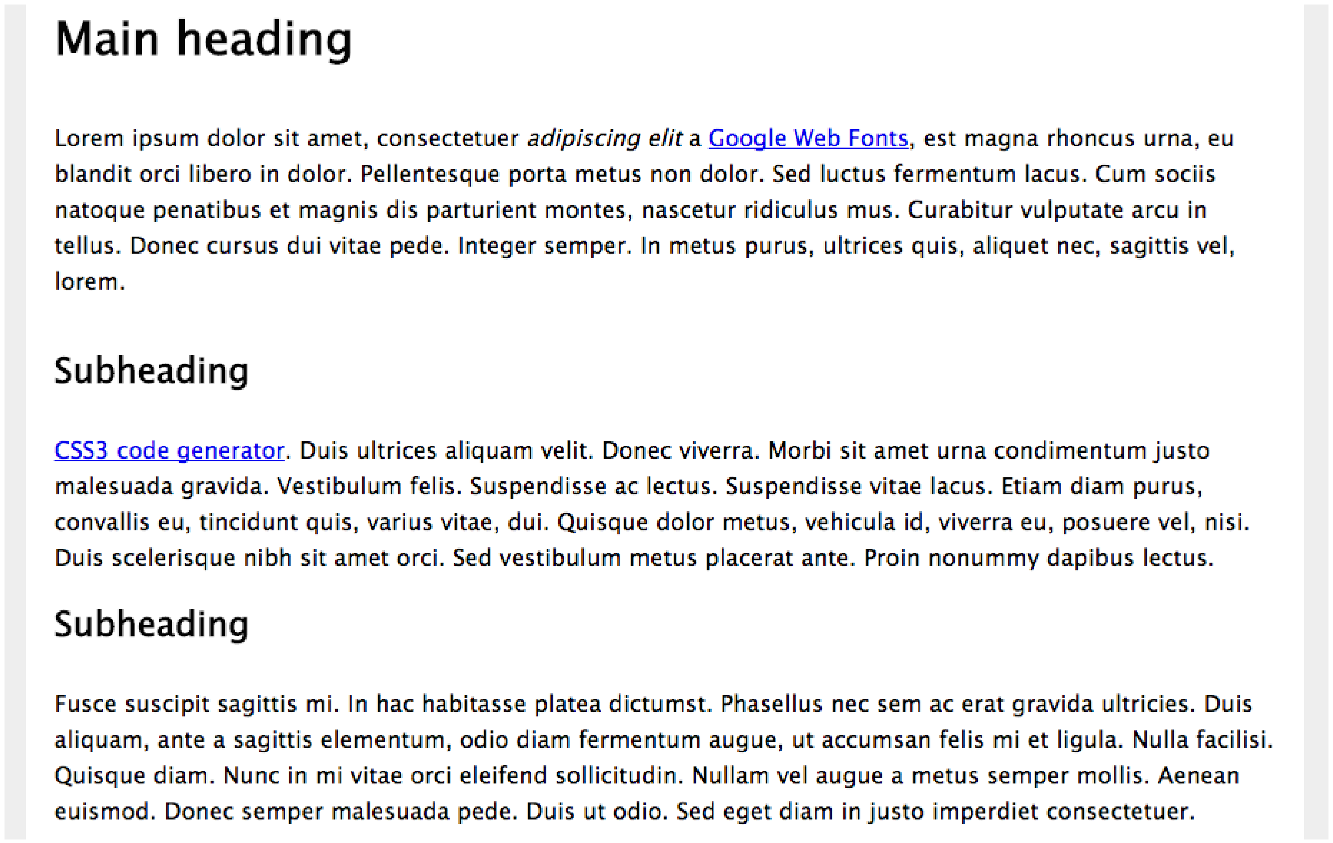
Fusce suscipit sagittis mi. In hac habitasse platea dictumst. Phasellus nec sem ac erat gravida ultricies. Duis aliquam, ante a sagittis elementum, odio diam fermentum augue, ut accumsan felis mi et ligula. Nulla facilisi.

5.2 However, the gaps between the headings and the text are too large. You can fix that by 'flattening' the margins for the level 2 headings and paragraphs to zero, just as we did for the level 1 heading (you can copy and paste the `margin: 0;` CSS rule from the `h1` selector—triple-click to copy the whole line AND the carriage return). Your CSS code for the `p` and `h2` selectors should now look like this:

```css
h2 {
  padding: 0 10px 1em;
  margin: 0;
}

p {
  padding: 0 10px 1em;
  margin: 0;
}
```

5.3 Reload the page in your browser—it should now appear as in this screen shot:



Main heading

Lorem ipsum dolor sit amet, consectetuer *adipiscing elit* a Google Web Fonts, est magna rhoncus urna, eu blandit orci libero in dolor. Pellentesque porta metus non dolor. Sed luctus fermentum lacus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur vulputate arcu in tellus. Donec cursus dui vitae pede. Integer semper. In metus purus, ultrices quis, aliquet nec, sagittis vel, lorem.

Subheading

CSS3 code generator. Duis ultrices aliquam velit. Donec viverra. Morbi sit amet urna condimentum justo malesuada gravida. Vestibulum felis. Suspendisse ac lectus. Suspendisse vitae lacus. Etiam diam purus, convallis eu, tincidunt quis, varius vitae, dui. Quisque dolor metus, vehicula id, viverra eu, posuere vel, nisi. Duis scelerisque nibh sit amet orci. Sed vestibulum metus placerat ante. Proin nonummy dapibus lectus.

Subheading

Fusce suscipit sagittis mi. In hac habitasse platea dictumst. Phasellus nec sem ac erat gravida ultricies. Duis aliquam, ante a sagittis elementum, odio diam fermentum augue, ut accumsan felis mi et ligula. Nulla facilisi. Quisque diam. Nunc in mi vitae orci eleifend sollicitudin. Nullam vel augue a metus semper mollis. Aenean euismod. Donec semper malesuada pede. Duis ut odio. Sed eget diam in justo imperdiet consectetuer.

5.4 Now fix the padding and apply a font-size rule to the headings:

```
h1 {
  padding: 0 10px 0.5em;
  margin: 0;
  font-size: 2em;
}

h2 {
  padding: 0 10px 0.25em;
  margin: 0;
  font-size: 1.25em;
}
```

Reload to see the level 2 headings shrink a little. CSS can be used to make heading (and most other) tags appear *just how you like*.

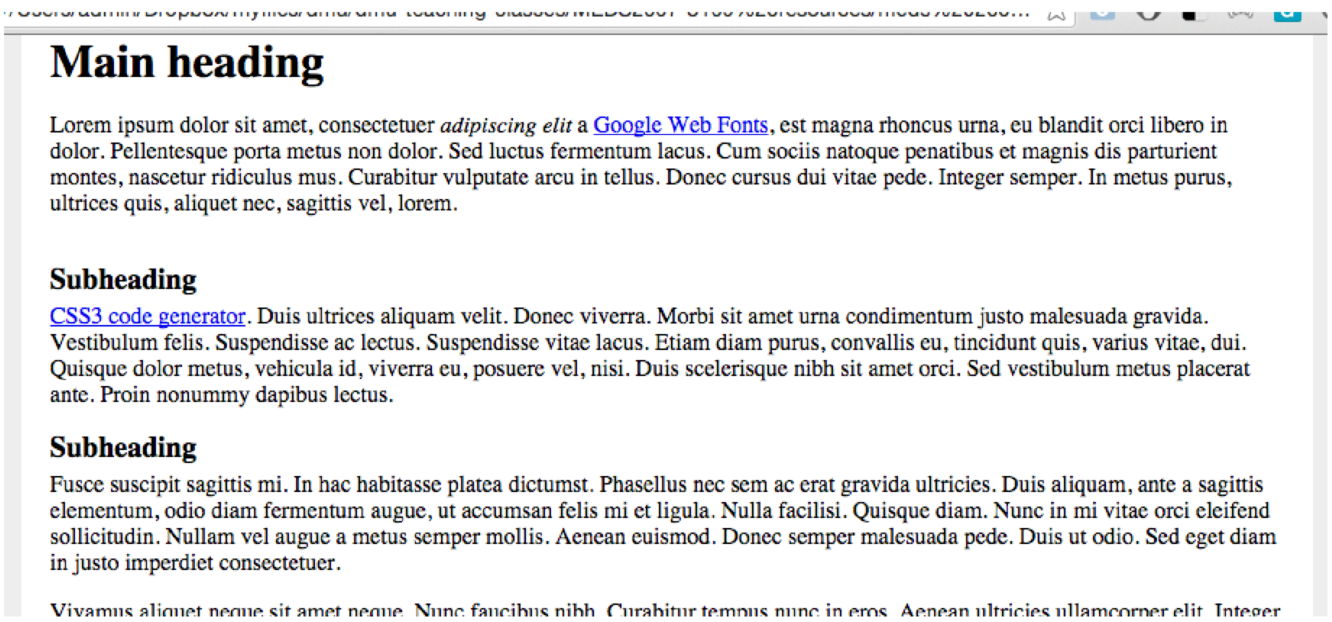
Now you can begin some serious styling.

6. Sans-serif or serif?

6.1 To illustrate how the CSS cascade works, add another CSS **`font-family`** rule below the existing one in the **`body`** tag rule block—this overrides the **`font-family`** rule above:

```
body {
  margin: 0;
  padding: 0;
  color: #000;
  background: #eee;
  font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
  font-family: serif;
}
```

Reload the page. Not only has this overridden the previous sans-serif fonts, it has *cascaded* to the *whole page*:



Main heading

Lorem ipsum dolor sit amet, consectetuer *adipiscing elit* a Google Web Fonts, est magna rhoncus urna, eu blandit orci libero in dolor. Pellentesque porta metus non dolor. Sed luctus fermentum lacus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur vulputate arcu in tellus. Donec cursus dui vitae pede. Integer semper. In metus purus, ultrices quis, aliquet nec, sagittis vel, lorem.

Subheading

CSS3 code generator. Duis ultrices aliquam velit. Donec viverra. Morbi sit amet urna condimentum justo malesuada gravida. Vestibulum felis. Suspendisse ac lectus. Suspendisse vitae lacus. Etiam diam purus, convallis eu, tincidunt quis, varius vitae, dui. Quisque dolor metus, vehicula id, viverra eu, posuere vel, nisi. Duis scelerisque nibh sit amet orci. Sed vestibulum metus placerat ante. Proin nonummy dapibus lectus.

Subheading

Fusce suscipit sagittis mi. In hac habitasse platea dictumst. Phasellus nec sem ac erat gravida ultricies. Duis aliquam, ante a sagittis elementum, odio diam fermentum augue, ut accumsan felis mi et ligula. Nulla facilisi. Quisque diam. Nunc in mi vitae orci eleifend sollicitudin. Nullam vel augue a metus semper mollis. Aenean euismod. Donec semper malesuada pede. Duis ut odio. Sed eget diam in justo imperdiet consectetuer.

Vivamus aliquet neque sit amet neque. Nunc faucibus nibh. Curabitur tempus nunc in eros. Aenean ultricies ullamcorper elit. Integer

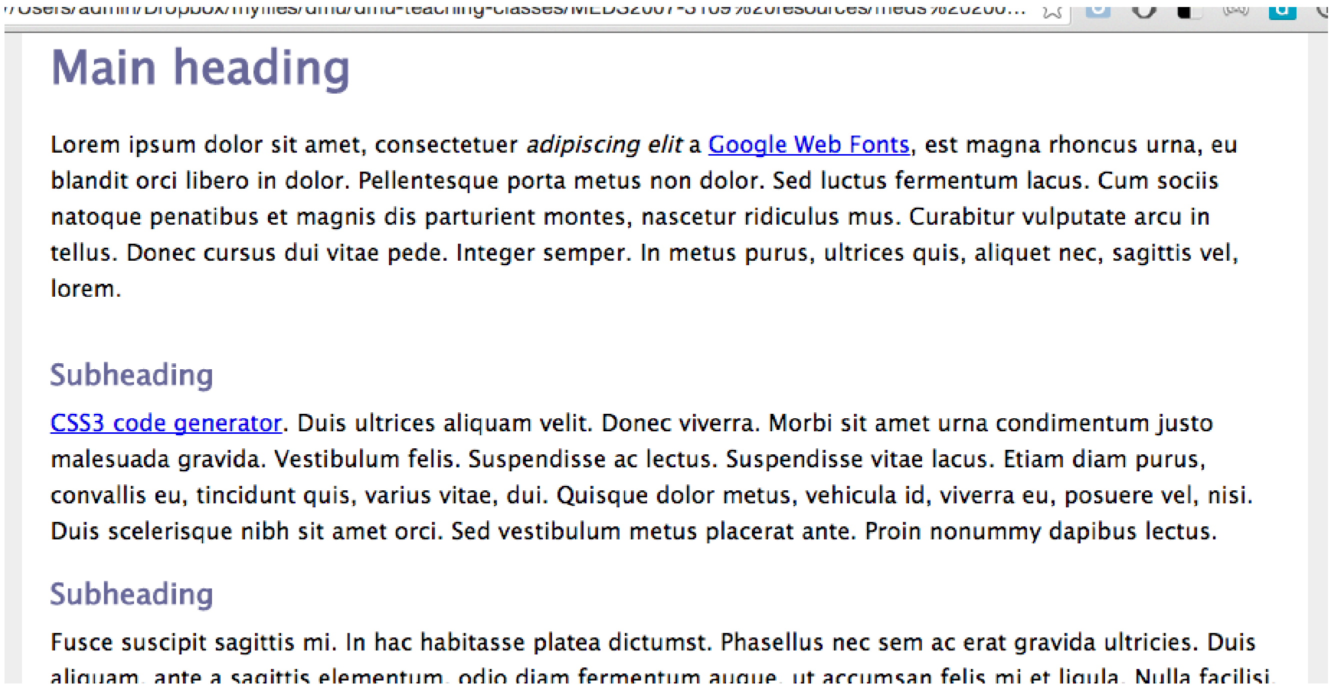
6.2 Remove the **`font-family: serif;`** rule you just added (triple-click it and hit delete). Now reload—your page should appear as before.

Next, change the colour of the **`h1`** and **`h2`** styles by adding **`color: #669;`** (mixing red/green/blue) at the bottom of their rule blocks:

```
h1 {
  padding: 0 10px 0.5em;
  margin: 0;
  font-size: 2em;
  color: #669;
}

h2 {
  padding: 0 10px 0.25em;
  margin: 0;
  font-size: 1.25em;
  color: #669;
}
```

Reload—your headings should now be a subtle shade of blue:



Main heading

Lorem ipsum dolor sit amet, consectetuer *adipiscing elit* a Google Web Fonts, est magna rhoncus urna, eu blandit orci libero in dolor. Pellentesque porta metus non dolor. Sed luctus fermentum lacus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur vulputate arcu in tellus. Donec cursus dui vitae pede. Integer semper. In metus purus, ultrices quis, aliquet nec, sagittis vel, lorem.

Subheading

CSS3 code generator. Duis ultrices aliquam velit. Donec viverra. Morbi sit amet urna condimentum justo malesuada gravida. Vestibulum felis. Suspendisse ac lectus. Suspendisse vitae lacus. Etiam diam purus, convallis eu, tincidunt quis, varius vitae, dui. Quisque dolor metus, vehicula id, viverra eu, posuere vel, nisi. Duis scelerisque nibh sit amet orci. Sed vestibulum metus placerat ante. Proin nonummy dapibus lectus.

Subheading

Fusce suscipit sagittis mi. In hac habitasse platea dictumst. Phasellus nec sem ac erat gravida ultricies. Duis aliquam, ante a sagittis elementum, odio diam fermentum augue, ut accumsan felis mi et ligula. Nulla facilisi.

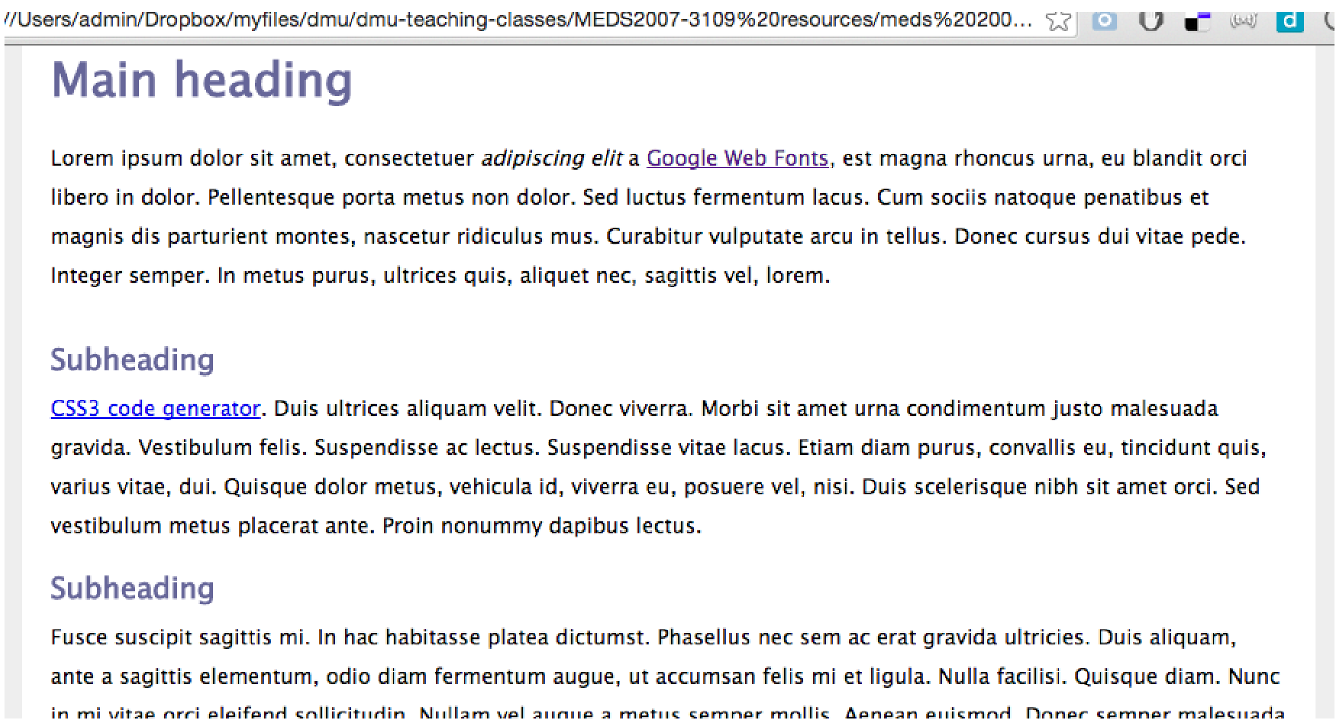
7. Readability and line-height

7.1 If the paragraph font size looks a bit too large by default, you can reduce it with `font-size`, then increase the line spacing with `line-height`:

```
p {
  padding: 0 10px 1em;
  margin: 0;
  font-size: 0.9em;
  line-height: 1.8em;
}
```

NOTE: As well as `em`s as a unit of measurement (think of an `em` as the height needed for a line of text), you can also use percentage (`%`), pixels (`px`) and other units for `font-size` and `line-height` properties. However, it is easier to stick to `em` or `%`.

Reload. Your page should now look like this—the text is now easier to read:



//Users/admin/Dropbox/myfiles/dmu/dmu-teaching-classes/MEDS2007-3109%20resources/meds%20200...

Main heading

Lorem ipsum dolor sit amet, consectetuer *adipiscing elit* a Google Web Fonts, est magna rhoncus urna, eu blandit orci libero in dolor. Pellentesque porta metus non dolor. Sed luctus fermentum lacus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur vulputate arcu in tellus. Donec cursus dui vitae pede. Integer semper. In metus purus, ultrices quis, aliquet nec, sagittis vel, lorem.

Subheading

CSS3 code generator. Duis ultrices aliquam velit. Donec viverra. Morbi sit amet urna condimentum justo malesuada gravida. Vestibulum felis. Suspendisse ac lectus. Suspendisse vitae lacus. Etiam diam purus, convallis eu, tincidunt quis, varius vitae, dui. Quisque dolor metus, vehicula id, viverra eu, posuere vel, nisi. Duis scelerisque nibh sit amet orci. Sed vestibulum metus placerat ante. Proin nonummy dapibus lectus.

Subheading

Fusce suscipit sagittis mi. In hac habitasse platea dictumst. Phasellus nec sem ac erat gravida ultricies. Duis aliquam, ante a sagittis elementum, odio diam fermentum augue, ut accumsan felis mi et ligula. Nulla facilisi. Quisque diam. Nunc in mi vitae orci eleifend sollicitudin. Nullam vel augue a metus semper mollis. Aenean euismod. Donec semper malesuada

It still looks very basic, but you have changed and added quite a few CSS style rules, each of which can be adjusted separately as you like.

8. A bit more style

8.1 The letter in the headings look a bit cramped, so add some letter-spacing:

```
h1 {
  padding: 0 10px 0.5em;
  margin: 0;
  font-size: 2em;
  color: #669;
  letter-spacing: 0.05em;
}

h2 {
  padding: 0 10px 0.25em;
  margin: 0;
  font-size: 1.25em;
  color: #669;
  letter-spacing: 0.1em;
}
```

Again, as with the line-height property, it's best to use **`em`**s for the letter spacing property . Then, if the browser font size is enlarged or reduced by you later, or by the user in the browser, the **`line-height`** and **`letter-spacing`** remains in proportion.

8.2 Reload, and enlarge or decrease browser view: ctrl or cmd (Windows/Mac) + (plus sign) and ctrl or cmd - (minus sign).

You can see that the **`letter-spacing`** and **`line-height`** stay in relative proportion to the altered font size because the **`em`** unit resizes browser text size *overall*.

9. Semantic markup

9.1 There are a few more HTML tags useful for marking up text *semantically*. Semantic markup is important, and indicates the *actual meaning* of content contained within HTML tags, e.g. instead of using bold **`<b>`** and italic **`<i>`** tags, it is now good practice to use **`<strong>`** (approximately: 'loud') and **`<em>`** (for 'emphasis'), as they give *semantic meaning* to the enclosed text. Screen-reading software will also read this semantic markup appropriately.

The **`<em>`** tag already wraps a few words in the text. Now also add the **`<strong>`** tag around the first three words in the first paragraph:

```
<p><strong>Lorem ipsum dolor</strong> sit amet, consectetuer
<em>adipiscing elit</em> a…
```

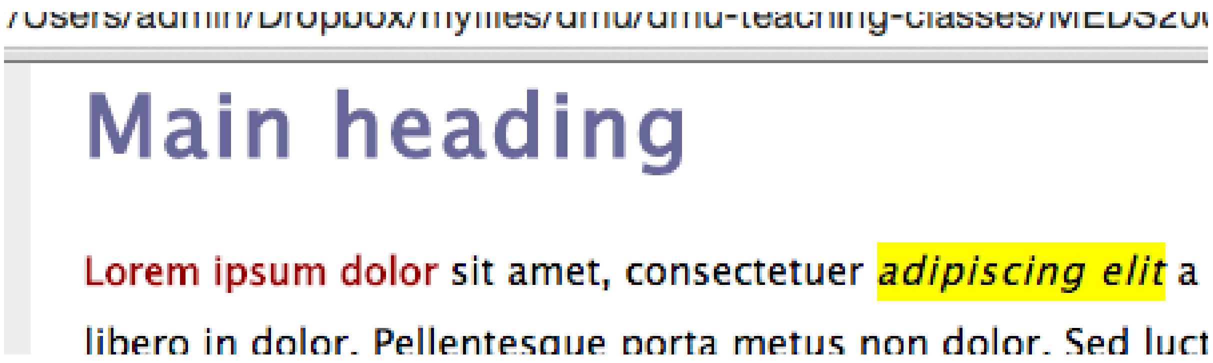
Reload the page in the browser. The default result is what you'd expect, although some fonts are more obviously 'bold'.

9.2 Now alter the default appearance of the `<strong>` and `<em>` tags (which can be used around any text) by adding two new CSS style blocks that target them *only* when they appear *inside paragraph tags*:

```
p strong {
  color: #900;
}

p em {
  letter-spacing: 0.05em;
  background: #ff0;
}
```

Now reload—your first paragraph should appear something like this:



Main heading

Lorem ipsum dolor sit amet, consectetuer *adipiscing elit* a

libero in dolor. Pellentesque porta metus non dolor. Sed luct

9.3 It's quite a common practice to style the first paragraph differently, for instance, as a summary. You can target a single paragraph by giving it a CSS *class*.

First, add a CSS class *attribute* to the paragraph just after the `<h1>` heading:

```
<h1>Main heading</h1>

  <p class="firstpara"><strong>Lorem ipsum dolor</strong>...
```

NOTE: classes can be named how you like—but you **can't use spaces**, or **start with a number**—it's best to use brief words that describe the purpose of the style, so you remember. However, don't name a style (for example) 'yellow-text' if it's just for highlighting—you might change your mind about the colour, so 'highlighted' would be a better name.

Add a CSS rule block to target the paragraph with a class attribute of `.firstpara`:

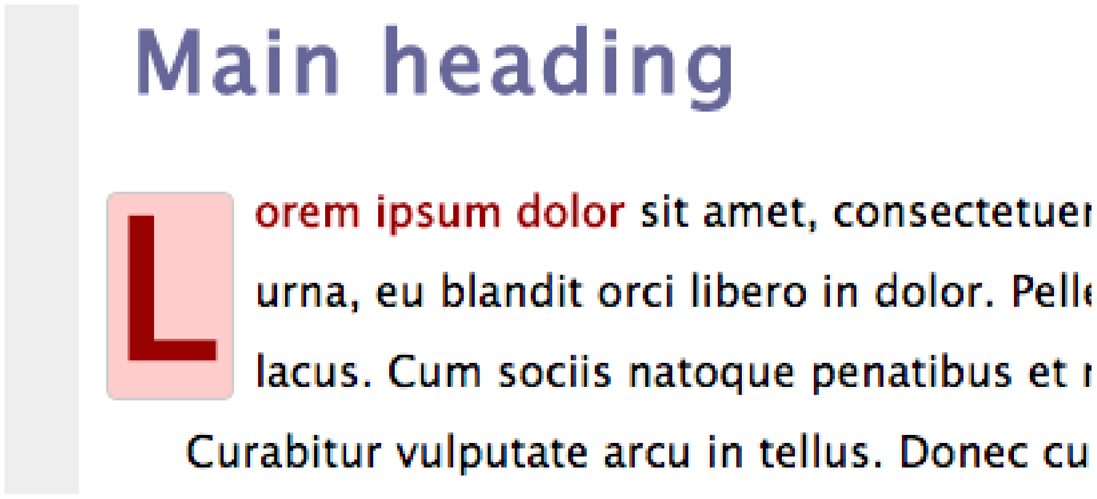
```
.firstpara {
  font-size: 1.1em;
  padding-left: 30px;
  padding-right: 30px;
  text-align: justify;
}
```

9.4 Now you'll create a drop capital letter (extra large first character) in the opening paragraph. Add a new CSS block with a *pseudo-element* selector `::first-letter` this targets (obviously) only the first letter in the `<p>` tag with the `.firstpara` class:

```
.firstpara::first-letter {
  padding: 2px 6px 2px 2px;
  border: 1px solid #ccc;
  border-radius: 4px;
  background: #fcc;
  float: left;
  font-size: 4.5em;
  margin: 0.1em 0.1em 0 -30px;
  line-height: 1em;
}
```

NOTE: In CSS, 'pseudo-elements' are prefixed with `::` (a double colon).

Reload the page in the browser:


Main heading

Lorem ipsum dolor sit amet, consectetuer
urna, eu blandit orci libero in dolor. Pell
lacus. Cum sociis natoque penatibus et
Curabitur vulputate arcu in tellus. Donec cu

There are quite a few style rules that make it appear like this. Try commenting out some lines (in VSCode or Atom, hit **cmd-/**) or change the values to see what they do.

10. Changing the default link colours

Browsers set an underline and a three default colours for link tags (**<a …>**). Unstyled links usually appear blue, then purple for links you've visited, and red when you actually click on them.

10.1 It's easy to change these colours and the underline with CSS. Add the following four style blocks after the closing curly bracket of the style block for the **<p>** tag:

```
p a:link {
  color: #555;
}
p a:visited {
  color: #888;
}
p a:hover {
  background: #eee;
  text-decoration: none;
}
p a:active {
  background: #fcc;
}
```

When you reload, **hover** over, **click** or **visit** a link, you'll see these styles in action.

More advanced styling

11. Styling a quotation with a blockquote tag and pseudo-element

11.1 Wrap the 4th paragraph (starting "Vivamus aliquet neque") in a `<blockquote>` tag, including a *class attribute* in the opening tag like this (text is abbreviated):

```
<blockquote class="open-quote">
  <p>Vivamus aliquet neque sit […] posuere purus sed metus.</p>

  <p>Maecenas porttitor venenatis […] Aliquam sodales.</p>
</blockquote>
```

11.2 Then in the CSS, add the following styles for the `blockquote` tag:

```
blockquote {
  position: relative;
  width: 70%;
  margin: 0 auto 1em;
  padding: 1em 1em 0 4.8em;
  background: #eee;
  font-size: 0.95em;
  border: 2px solid #ccc;
  border-radius: 12px;
}
```

The `::before` pseudo-element can be used to add an opening quotation mark to the `blockquote` tag. Since pseudo-elements can be styled, it's possible to do quite a bit with this. Since the parent `blockquote` element has `position: relative`, the quotation mark can be placed anywhere, using `position: absolute` . Add the following styles after the closing bracket of the style block above:

```
.open-quote::before {
  position: absolute;
  content: '"';
  top: -16px;
  left: -26px;
  font-family: arial, helvetica, verdana, sans-serif;
  font-size: 20em;
  line-height: 1em;
  color: #99c;
}
```

The `content` property contains the actual Unicode **curly opening quote**, in single straight ‘computer code font’ quotes (*not* curly).

NOTE: you can use curly quotes and other refined typographic elements like proper hyphens and ellipses (three dots) if the **charset** (character set) is set to **UTF-8** in the HTML file (as it is in the code for these examples).

Quotes in the Lucida font look a bit odd at large sizes, so the `font-family` has been changed to Arial, Helvetica… etc.

Negative margins allow the quote mark to be placed outside the containing box. The other settings are tweaks to get the size and position just right for that font—not guaranteed on everyone’s computer, but good enough.

IN PROGRESS

This tutorial is in progress and unfinished, but you can see the end result in the files named **index-complete.html** and **styles-complete.css** for the finished example.

The files for all three stages are downloadable from GitHub, and any updates will be added there, so check back later.

Adding a background image to an element—e.g. all level 2 headings

See the finished styles for the heading tags.

More: semantic HTML markup

As well as **blockquote** (for long quotations), other HTML tags lend semantic meaning to your text; **cite** (for the title of a work like an article, play or book), **abbr** (abbreviations), **dfn** (for a definition), all of which can be styled as you wish.

For instance, the **abbr** tag is often given a dotted underline and used with an HTML 'title' attribute to give a rollover 'tooltip' explaining what the abbreviation stands for.

More: CSS font control

If you want fonts outside the normal 'web fonts' (i.e. that aren't guaranteed to be on other computers), you can use Google web fonts as in the finished example.

Other useful CSS font rules (the bar | separates the possible values, so use only one):

```
font-style: normal|italic;
font-weight: normal|bold;
```

For paragraphs, this sets an indent for the first line, e.g. as sometimes seen in books:

```
text-indent: length|percent;
```

Useful links to take you further

If you follow up only one of these links, make it this one:
6 Ways To Improve Your Web Typography

If you'd like to know more, take a look at these:
Beautiful Web Typography: 7 tips on de-sucking the web
Typetester: compare various settings for screen type
The Elements of Typographic Style Applied to the Web
Setting Type on the Web to a Baseline Grid