More on font subsetting

Idea from Bram Stein's Webfont Handbook

I've discussed font subsetting regarding ebooks. This post will review how do we load multiple fonts and how we subset fonts. THen we'll seek to answer additional questions:

- Can you control how browsers synthetize bold and italics?
- Can you control font timeouts?
- How do you know what characters to use in a subset?
- Can you use multiple subsets of the same font? Pros? Cons?

Loading multiple font-faces

This is what my normal font loading CSS looks like. I use 5 `@font-face` commands to load the fonts. One for the monospace font used in code block examples and 4 for our content font: regular, bold, italics and bold italics.

```
/* Monospaced font for code samples */
@font-face {
  font-family: "notomono";
  src:  url("../fonts/notomono-regular.woff2") format("woff2"),
        url("../fonts/notomono-regular.woff") format("woff"),
        url("../fonts/notomono-regular.ttf") format("truetype");
  font-weight: normal;
  font-style: normal;
}
/* Regular font */
@font-face {
  font-family: "notosans";
  src:  url("../fonts/notosans-regular.woff2") format("woff2"),
        url("../fonts/notosans-regular.woff") format("woff"),
        url("../fonts/notosans-regular.ttf") format("truetype");
  font-weight: normal;
```

```
  font-style: normal;
}
/* Bold font */
@font-face {
  font-family: "notosans";
  src:  url("../fonts/notosans-bold.woff2") format("woff2"),
        url("../fonts/notosans-bold.woff") format("woff"),
        url("../fonts/notosans-bold.ttf") format("truetype");
  font-weight: 700;
  font-style: normal;
}
/* Italic Font */
@font-face {
  font-family: "notosans";
  src:  url("../fonts/notosans-italic.woff2") format("woff2"),
        url("../fonts/notosans-italic.woff") format("woff"),
        url("../fonts/notosans-italic.ttf") format("truetype");
  font-weight: normal;
  font-style: italic;
}
/* bold-italic font */
@font-face {
  font-family: "notosans";
  src:  url("../fonts/notosans-bolditalic.woff2") format("woff2"),
        url("../fonts/notosans-bolditalic.woff") format("woff"),
        url("../fonts/notosans-bolditalic.ttf") format("truetype");
  font-weight: 700;
  font-style: italic;
}
```

I use Font Face Observer to load fonts asynchronously. A newer API that I hadn't heard before is the CSS Font Loading API, currently an editor draft, to perform a similar task.

Creating a subset

Using the full fonts can be costly in terms of file size. Even at 40KB size per file (on average) we get 200KB of fonts that contain glyphs athat may never be used.

The simplest way to create a font subset remains Font Squirrel's Web Font Generator which has a section for subsetting, shown below

Figure 1: Font Squirrel's Web Font Generator Subsetting Section

If we know the Unicode ranges that we want to subset we can use fonttools, a collection of Python library for font manipulation.

Install it with Pip... If you've installed Python using Homebrew your Pip installer may be `pip2` or `pip3` depending on the Python version you have installed.

```
pip install fonttools
```

The tool we use to subset a font is `pyftsubset` that takkes the name of the font we want to subset, one or more Unicode glyphs (characters) that we want to include in the subset and the format of the subset file.

```
pyftsubset font.otf --unicodes="U+a,U+20,U+2e,U+44,U+45,U+4d,U+54,U+59,U+6
```

In the next section we'll explore a way to generate the subset based on the characters used in the page.

How do you know what characters to use in a subset?

`pyftsubset` assumes that you know the Unicode glyphs of the characters you want to subset. This is not always the case.

So we'll look at it in a different way... instead of entering the unicode characters to subset, we'll use a separate tool to query a page to see what characters the page is using and feed that data to `pyftsubset` to create the subsets.

Glyphanger is a Node utility that generates a combined list of every glyph used on a list of sample files or urls. I prefer to use a global installation of Glyph Hanger as it's not related to a project and benefits from being in my path.

```
npm install -g glyphhanger
```

The first thing we do is get a list of all the Unicode Glyps used in a given site. We do this by querying the site and adding the `--unicodes` flag.

```
glyphhanger http://www.example.com/ --unicodes
```

We can then feed the result of this command to `pyftsubset` to create the subset font. Or we can use the subset command directly from Glyphhanger by adding `--subset` parameter to point where the fonts you want to subset.

```
glyphhanger http://www.example.com/ --unicodes  --subset=fonts/*.ttf
```

Can you use multiple subsets of the same font?

We've figured out how to subset fonts there is one more question... Can we use more than 1 subset? In the example below we use two subsets. The first subset is for English using the standard Latin subset and the second one with the Extended Latin character sets to incorporate accents and other marks used in Latin scripts other than English.

```
@font-face {
  font-family: Source Sans Pro;
  src: url(latin.woff) format("woff");
  unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02C6,"woff"A, U+02DC
}
```

```
@font-face {
  font-family: Source Sans Pro
  src: url(latin-extended.woff) format("woff");
  unicode-range: U+0100-024F, U+1E00-1EFF, U+20A0-20AB, U+20AD-20CF, U+2C
}
"woff"
```

The idea is that for browsers that support the `unicode-range` rule and are displaying English content the first subset will be used. Whenever we use glyphs from the Extended Latin set the browser will load the second font subset and use them together when laying out the text. You may think that this is unnecessary; remember that Englissh borrows many words from other languages many of which use accents and other marks that are not used in English.