Noto Fonts: Same family, multiple languages

One of the limitations of the World Wide Web is the lack of fonts for all the world languages. Even if we were to find fonts for all the languages we want to use it's unlikely that we'll find fonts that work well together.

Google has developed and released as open source a family of fonts that cover all the languages covered in the Unicode standard. They are not quite up to the inteded goal of full Unicoode support but they are getting close.

The idea is to have a set of fonts for Unicode languages that work well together and look good together.

This is a different problem than the one Variable fonts solve; variable fonts interpolate font attributes such as weight, slant, and custom attributes between axes meaning that we need fewer font files to represent the same group of characters. However, this doesn't include all the characters that we need for non western European languages; for that we still need multiple files.

Quick note about licenses

As we start diving into the Noto family we need to look at the license for the fonts. For a while Google used to license fonts under the Apache 2 license, just like their other software but as of 2015 they moved the license to the SIL Open Font License 1.1

Using the fonts

I'm working with web content so I'll concentrate in that aspect of font usage and not worry about local use or bundling them with an app.

Make sure that the fonts are available as web fonts when you declare them. When you declare the font in your CSS but don't load them as web fonts, the browser will immediately go to the next option in the stack.

Compress the fonts using either WOFF or WOFF2 depending on your browser support needs. The same suggestions outlined in Improving Font Performance:

Work to control font loading and Improving Font Performance: Subset fonts using Glyphhanger apply when working with multiple fonts of the same family.

I strongly recommend using font-display to improve performance and keep yourself from Flash of Invisible and Flash of Unstyled Text.

Serve your fonts from the same domain as your app. Some Noto fonts are available at Google Web Fonts Early Access but they may not be the latest version of Noto or it may only cover a subset of the language.

In addition, be aware that the web latency for large fonts, such as for Noto Sans CJK or when using multiple languages, can be large. Google Fonts uses `font-display: swap` to mitigate the problem but the solution may still not be ideal in all cases, particularly when working with slow devices or in areas with pooor connectivity.

See Zach Leat's Google Fonts Is Adding font-display 🎉, and this twitter thread for more in-depth analysis of how to best use Google Fonts.

Using Noto fonts in the CSS font-family property

Because fonts in the Noto family may overlap there are some considerations as written by the Google Fonts team

Put fonts for the languages/scripts you care about most at the very beginning.

This should go without saying but don't include fonts you don't need to improve Time to First Byte (TTFB), Time To Interactive (TTI), and other performance metrics.

It is recommended to retain "Noto Sans" in the list. Other Noto fonts usually do not cover Latin letters, digits or punctuation. If you're writing content that doesn't require any Latin language you can forego Noto Sans on your list.

Put "Noto Sans" before "Noto Sans CJK". Currently, the Latin characters in the CJK fonts are from Adobe's Source Sans Pro so they may not look the same as the other sans serif fonts in the document.

Wherever possible use "Noto Sans CJK {JP, KR, SC, TC}" rather than "Noto Sans {JP, KR, SC, TC}" (note the difference in the file names).

Each of the font families “Noto Sans CJK {JP, KR, SC, TC}” supports all four languages, but has a different default language. So depending on the primary language you’re targetting what font you should use.

Examples

These examples assume that you’ve already loaded using `@font-face` or, if you’re happy with the restrictions that Google Fonts may add to the fonts it delivers, through Google Fonts. It also assumes that the names on the @font-face declarations match the names we use in the `font-family` rules.

For a Japanese website:

```
font-family:  "Noto Sans",
              "Noto Sans CJK JP",
              sans-serif;
```

For a website targeting Hindi, and then Tamil users

```
font-family:  "Noto Sans Devanagari",
              "Noto Sans Tamil",
              "Noto Sans",
              sans-serif;
```

For an Arabic website that needs to use an UI font for UI elements, such as buttons and tabs, that have more strict vertical space:

```
font-family:  "Noto Naskh Arabic UI",
              "Noto Sans UI",
              sans-serif;
```

For a website targeting Armenian and Georgian users who prefer serif style:

```
font-family:  "Noto Serif Armenian",
              "Noto Serif Georgian",
              "Noto Serif",
```

```
                serif;
```

Why bother?

I like to think that fonts like Noto and other fonts that target specific non Latin (especially ethnic, endangered or minority) languages give people who wouldn’t normally have one a voice on the web. It allows them to communicate in their own language and helps make true the promise of a world wide web.