Font formats for the web and converting from one to another

Using fonts on the web has a longer story than what many would think.

The @font-face at-rule has been around since the early days of the web. Initially there were no fonts on the web… the goal was to share documents, not to produce the polished documents that we see today.

In the beginning, web browsers relied on locally installed fonts both for performance (the modems of the time would take forever to download a font).

We could use any of the pre-defined fonts:

- Andale Mono
- Arial
- **Arial Black**
- Comic Sans MS
- Courier New
- Georgia
- **Impact**
- Times New Roman
- Trebuchet MS
- Verdana
- ▶📖⚙♥ⓘ●■? (Webdings)

There were several methods for embeding fonts that didnt' catch on like TrueDoc in 1996, an outline font standard developed by Bitstream, or Embedded OpenType (EOT) in 1997.

CSS 2.0, introduced in 1998, included support for @font-face in pretty much the same way as it does now but browsers were afraid of being seen as contributors to font piracy so they didn't implement it when it was first released.

It wasn't until 2008 that browsers (re)introduced @font-face with support for different formats:

- The beta version of Safari 4 implemented @font-face support for TrueType fonts
- The beta version of Firefox 3.1 implemented support for both TrueType and OpenType

Other browsers continued to support separate formats which eventually led to the "bulletproof @font-face syntax", dating back to 2009 in Paul Irish's Bulletproof @font-face Syntax

in 2009, Small Batch announced the Typekit project, a subscription-based 'middleman' service for designers and developers, which allowed safer embedding of fonts via the @font-face CSS declaration and some JavaScript.

We now have a variety of services like TypeKit (acquired by Adobe and now known as Adobe Fonts). The biggest one is Google Fonts

Things have improved in terms of common support. All modern browsers except IE and Opera Mini now support WOFF2 fonts.

The four most used formats for fonts are:

- TTF (True Type Fonts)
- OTF (Open Type Fonts)
- WOFF (Web Open Font Format 1.0)
- WOFF2 (Web Open Font Format 2.0)

What format you use depends on the browsers you must support.

Not all fonts are available in all formats and I've found out over time that I have to do manual conversions, usually from TTF or OTF to newer formats.

I will run through the process both to illustrate the tools and process I use to compress fonts from TTF to WOFF and WOFF2

Getting Started

I downloaded Open Sans from Google Fonts as the basis for the compression examples.

WOFF

To convert to WOFF, I use the [ttf2woff](#) utility, available as a Node package.

To install it, run the following command:

```
npm i -g ttf2woff
```

To conver the font to WOFF, use the following command, replacing both instances of the font-name with the font you're working with.

```
ttf2woff \
OpenSans-VariableFont_wdth,wght.ttf \
OpenSans-VariableFont_wdth,wght.woff
```

and then do the same thing with the italic version of the variable fonts:

```
ttf2woff \
OpenSans-Italic-VariableFont_wdth,wght.ttf \
OpenSans-Italic-VariableFont_wdth,wght.woff
```

If you're not working with a variable font, you can run the same command for each individual font.

```
ttf2woff OpenSans-Bold.ttf OpenSans-Bold.woff
ttf2woff OpenSans-BoldItalic.ttf OpenSans-BoldItalic.woff
ttf2woff OpenSans-Italic.ttf OpenSans-Italic.woff
ttf2woff OpenSans-Regular.ttf OpenSans-Regular.woff
```

WOFF2

WOFF2 is an evolution of the WOFF format that provides further savings in terms of file size.

To convert the font to WOFF2 I'm using Google's [WOFF2 Reference](#)

[Implementation](#)

Install the library via Homebrew or your platform-specific package manager with the following command:

```
brew brew install woff2
```

And run the compression tool with the following commands:

```
woff2_compress \
OpenSans-VariableFont_wdth,wght.ttf
woff2_compress \
OpenSans-Italic-VariableFont_wdth,wght.ttf
```

These are the results of compressing our regular variable font with the different formats:

Font	Format	Size
OpenSans-VariableFont_wdth,wght	TTF	530KB
OpenSans-VariableFont_wdth,wght	WOFF	345KB
OpenSans-VariableFont_wdth,wght	WOFF2	280KB

And these are the results for the italic variable font. Note how the values for all formats are slightly larger in this example.

Font	Format	Size
OpenSans-Italic-VariableFont_wdth,wght	TTF	580 KB
OpenSans-Italic-VariableFont_wdth,wght	WOFF	389KB
OpenSans-Italic-VariableFont_wdth,wght	WOFF2	310KB

If we also take into consideration that WOFF2 is supported across all browsers, except IE and Opera Mini, it is safe to say that, size-wise, WOFF 2 is the best format to use.

Subsetting

We converted the fonts to WOFF2 and did the best we could do without resorting to exotic compression tools like Brotli or Zopfli.

There is one more thing we can do to improve the file size of the fonts: subsetting.

SUbsetting is the process of taking a large font file and creating smaller font files, with fewer characters or OpenType features.

Subsetting to the latin character set

For this demo we'll just reduce the characters in the font to those necessary for Latin scripts using Glyphhanger to generate both WOFF and WOFF2 versions of the subset fonts.

```
glyphhanger --latin \
--subset=OpenSans-VariableFont.woff2  \
--formats=woff-zopfli,woff2
```

And the equivalent command for the italic variable font:

```
glyphhanger --latin \
--subset=OpenSans-Italic-VariableFont.woff2  \
--formats=woff-zopfli,woff2
```

Subsetting to ASCII

Perhaps the most aggressive subsetting available that doesn't match pages is subsetting to the US-ASCII character set. This will handle English and no other language since ASCII was created to represent characters in the English alphabet.

Running Glyphhanger to subset to the US_ASCII character set looks like this:

```
glyphhanger --US_ASCII \
```

```
    --subset=OpenSans-VariableFont.woff2  \
    --formats=woff-zopfli,woff2
```

The equivalent command for the Italic font:

```
glyphhanger --US_ASCII \
    --subset=OpenSans-Italic-VariableFont.woff2  \
    --formats=woff-zopfli,woff2
```

This produces the smallest font size by a large margin as shown in the following table

Font	Subset	Format	Size
OpenSans-VariableFont	Latin	WOFF + Zopfli	330KB
OpenSans-VariableFont	Latin	WOFF2	276KB
OpenSans-VariableFont	ASCII	WOFF + Zopfli	74KB
OpenSans-VariableFont	ASCII	WOFF2	63 KB
OpenSans-Italic-VariableFont	Latin	WOFF + Zopfli	373KB
OpenSans-Italic-VariableFont	Latin	WOFF2	310KB
OpenSans-Italic-VariableFont	ASCII	WOFF + Zopfli	83KB
OpenSans-Italic-VariableFont	ASCII	WOFF2	70KB

However unless you're absolutely sure that you will use English only with no phrases from other Latin languages like Spanish or French then you should use the ASCII subset; otherwise I would recommend the Latin Subset.

The sizes will also vary with the font or fonts you use. There are fonts that deliver a single variable font to cover all possible styles and variations. These fonts will be larger overall, but the larger size is offset by it being a single file to download.