SMASHING MAGAZINE

MENU

# The @Font-Face Rule And Useful Web Font Tricks

By Ralf Hermann

March 2nd, 2011   Fonts, Typography   69 Comments

The possibility of embedding any font you like into websites via `@font-face` is an additional stylistic device which promises to abolish the monotony of the usual system fonts. It surely would be all too easy if there was only one Web font format out there. Instead, there's quite a variety, as you will get to know in this article.
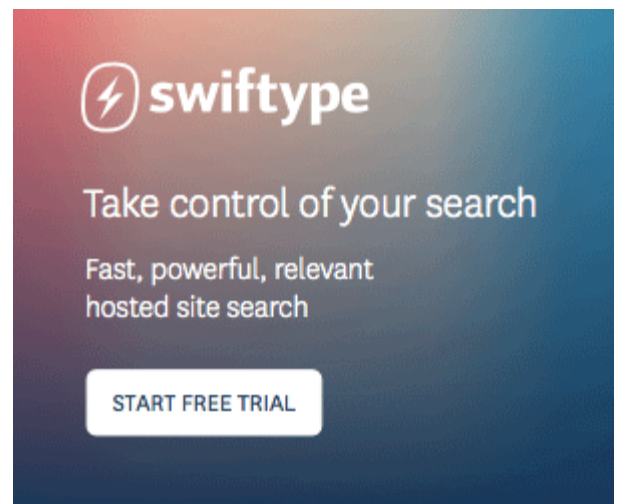
This quick introduction to `@font-face` will lead you towards a guide through the `@font-face` kit generator. If you want to make Web use of your already licensed desktop fonts, read up on how to embed them from your own server. Topped up with some helpful tips, tricks and workarounds, this article will hopefully provide some useful insights.

Web Font Formats 🔗

EOT, TTF, OTF, CFF, AFM, LWFN, FFIL, FON, PFM, PFB, WOFF, SVG, STD, PRO, XSF, and the list goes on.

To find one's way through in this veritable jungle of font formats is not exactly easy. Let's have a closer look at the pros and cons of font formats that are particularly relevant for their use on websites.

TRUETYPE 🔗

This format was developed in the late 1980s as a competitor to Adobe's Type 1 fonts used in PostScript. As a scalable outline format, it replaced the common bitmap fonts that were used for screen display at that time. Microsoft took up the TrueType format as well and it soon evolved into the standard format for system fonts due to the fact that it offered fine-tuned control for a precise display of font in particular sizes.

OPENTYPE 🔗

Microsoft and Adobe teamed up in developing this font format. Based on the TrueType format, OpenType offers additional typographical features such as ligatures, fractions or context sensitive glyphs and the like. However, browser support of these features which are common in sophisticated layout and illustration programs is still unsatisfactory. There are two different versions of OpenType fonts, depending on the outline technology used. There are:

- OpenType fonts with TrueType Outlines (OpenType TT) and
- OpenType fonts with PostScript Outlines (OpenType PS)

OpenType comes in two different versions.

OpenType PS is a so-called *CFF based file format* (CFF = compact file format). This is relevant when using OpenType PS fonts as Web fonts, because PostScript based formats are displayed without subpixel

rendering on Windows platforms which affects the rendering quality considerably. That's why TrueType based fonts are the better choice as Web fonts, even though Microsoft will solve this rendering issue in the future. The structures of TrueType and OpenType fonts are very similar and browser support is available in Safari 3.1 and higher, Firefox 3.5 and Opera 10 (and of course newer versions).

EOT 🔗

Internet Explorer has supported the proprietary Embedded OpenType (EOT) standard from the late 1990s. It's a variation of the TrueType and OpenType formats that provide the following particularities:

- EOT fonts are a compact form of OpenType optimized for quick delivery on the Web due to data compression.

- By means of URL-binding, EOT fonts can be tied to a specific domain. The fonts can then only be delivered to and used on those Web pages. This technique helps prevent fonts from being copied and used without a licence.

EOT is exclusively supported by Internet Explorer. Even though it might not succeed as a Web font format in the future, it still makes sense to use this format today in order to supply the users of various IE versions with Web fonts. Current IE versions (< 9) do not use any other format.

If you want to convert TTF fonts to natively compressed EOT files, you can use EOTFast (a free application) currently available only for Windows.

WOFF 🔗

Unlike EOT, the Web Open Font Format (WOFF) is in the process of being standardized as a recommendation by the W3C which published WOFF as a working draft back in July 2010.

WOFF came into existence as a kind of a compromise between font foundries and browser companies, so it's no wonder that WOFF has been developed by two font designers (Erik van Blokland and Tal Leming) in cooperation with Mozilla developer Jonathan Kew. Essentially, WOFF is a wrapper that contains TrueType and OpenType fonts, and it's not really a new format of its own.

WOFF uses an integrated compression algorithm named *zlib*, which offers file size reduction for TrueType fonts exceeding 40%. Furthermore, meta data can be added, e.g. a user's licence. However, this data presents only meta information and is not validated by browsers.

Thanks to WOFF, Mozilla can use its corporate typeface FF Meta.

The format has been supported by Firefox since version 3.6, and by Google Chrome since version 5.0. All other browser manufacturers are working on adding full support in future releases. Fonts can be converted into the WOFF format by the online service Font Squirrel free of charge.

SVG 🔗

SVG fonts are text files that contain the glyph outlines represented as standard SVG elements and attributes, as if they were single vector objects in the SVG image. But this is also one of the biggest disadvantages of SVG fonts. While EOT, WOFF and PostScript-flavoured OpenType have compression built into the font format — SVG fonts are always uncompressed and usually pretty large.

SVG fonts are not really an alternative to the other Web font formats, and iOS 4.2 is the first version of Mobile Safari to support native Web fonts (in TrueType format) instead of SVG. However, SVG is the only format that can be used for the iPhone and iPad prior to iOS 4.2.

Tools, such as Font Squirrel, can be used to convert fonts into this format. Another possibility to obtain SVG fonts is to rent them from one of the numerous Web font providers.

~~Typekit also serves its fonts as SVG files for the iPhone and iPad.~~ Typekit no longer servers SVG fonts to iOS devices. The service serves TrueType fonts to iOS 4.2 devices and higher.

@font-face Revolution 🔗

The CSS3 property `@font-face` presents so many new possibilities that a veritable gold-digging mentality is taking hold of Web designers. There's hope that regular system fonts will soon be abolished by Web font embedding, which enables us to choose practically any typeface and font style they want — just like in print design.

With regard to typography, the Web is *way* behind print. Take headlines: in print, condensed typefaces come in handy because they allow more words to fit on one line. System font collections, however, usually have no condensed fonts. Also, companies cannot use their proprietary fonts on their own websites. Instead, they have to replace them with standard fonts, such as Arial, which makes establishing a consistent corporate identity across all media impossible.

@FONT-FACE IN THE LATE 1990S 🔗

The ability to embed any font into a website has been around for a while. Netscape 4 and Internet Explorer 4 supported `@font-face` by the end of the 1990s. The rule allowed us to deposit a font on a

server and deliver it through a Web page.

```
@font-face {
font-family: Gentium;
src: url(fonts/gentium.eot);
}
```

This technique was ahead of its time. At that point it was used for simple grayscale anti-aliasing. This was no problem for system fonts, which were laboriously optimized for rendering on screen, but other fonts were not rendered properly; they lacked the benefits of manual on-screen optimization. Instead of improving Web typography, the non-system fonts made websites look worse.

ALIAS **five**   GREYSCALE ANTI-ALIAS **five**

It's no wonder that the `@font-face` rule was removed in the CSS 2.1 specification. Using system fonts was the general practice in Web design, especially for copy. For headlines, several workarounds have been established. Some designers replaced text with a bitmap file or a Flash movie that displayed the headline in a particular font.

Another approach that has emerged in the past few years is to replace headlines with vector graphics, with the help of JavaScript. Typeface.js and Cufón offer this functionality. Each of these techniques, though, has one problem or the other, be it incompatibility with search engines or issues with zooming.

SUCCESSFUL SECOND RUN, THANKS TO SUBPIXEL RENDERING 🔗

The introduction of Safari 3.1 by Apple marked a turning point in the use of Web fonts. The browser update brought the old `@font-face` rule back. A significant improvement came with the introduction of flat-panel LCD displays, which have high screen resolutions and anti-aliasing via subpixel rendering. Subpixel rendering takes advantage of the fact that each pixel on a color LCD is composed of individual red, green and blue subpixels. It uses these subpixels to anti-alias text, which increases the apparent resolution of an LCD display and thus improves the rendering of text — even text set at very small sizes.

On Mac OS X platforms, this function is activated by default. Windows uses its own trademark, ClearType, which is activated by default on Windows Vista and Windows 7 but turned off on Windows XP. In Microsoft Office 2007 and 2010, Internet Explorer 7+ and Windows Live Messenger, ClearType is turned

on by default, even if it's not enabled throughout the operating system.

Safari 3.1 offered support for standard font formats for the first time ([demo](#)).

Subpixel rendering is displayed in Safari on Mac OS X. What's remarkable about Safari's `@font-face` support is that, for the first time, embedding standard formats — namely, TrueType (TTF) and OpenType (TTF/OTF) — is possible without any prior conversion, as the following CSS shows:

```css
@font-face {
font-family: Gentium;
src: url(fonts/gentium.otf);
}
```

THE FONT FOUNDRIES' CONCERNS 🔗

Suppliers of commercial fonts, however, were not exactly happy about the new functionality. Anyone could now download and embed fonts without paying the licensing fee. A lively debate between browser makers and font foundries began… and continues. The foundries want commercial fonts to be protected from bootlegging and unlawful use, while the browser makers do not feel obligated to claim the foundries' copyrights; they instead argue that content providers are responsible for declaring any copyright-protected fonts on their websites, just as they are obliged to do with images, video, text and other assets.

Using the @font face Kit Generator 🔗

For a practical test you might want to download the font package at Graublau Sans Web. This font package offers a PostScript-based OpenType font that can actually be used on Safari 3.1+, Firefox 3.5+ and Opera 10+ without any changes. In order to provide browser support for Internet Explorer and improve the screen display, you can use the @font-face Kit Generator. Click the "Add Fonts" button and upload the respective font with all its weights. Then choose the "Expert" radio button and check all the necessary options, which we'll go through in more detail now:

FONT FORMATS 🔗

The main problem is that you have to activate EOT in order to obtain `@font-face` support on Internet Explorer IE4 to IE8 (in IE9 RC, WOFF is supported as well). SVG fonts will mainly be needed for Mobile Safari on the iPhone and iPad prior to iOS 4.2, although Chrome and Opera can handle this format as well. Alternatively or rather in addition to SVG, there's the compressed version SVGZ, which offers a much smaller file size. Unfortunately, it doesn't run on the iPhone so you'll need the SVG font anyway.

Creating a TrueType font format will allow for support in Safari (since 3.1), Firefox (since 3.5) and Opera (since 10). Checking the WOFF option won't increase browser support these days, but it sure will in the near future, when WOFF has become the standard Web font format.

RENDERING AND MISCELLANEOUS OPTIONS 🔗

- *Add Hinting*: This option offers improved font display in Windows. You should only uncheck it if you are absolutely positive that the respective fonts are screen-optimized already.

- *WebOnly™* adds specific modifications which allow browsers to use the modified fonts but avoid installation of these fonts on common operating systems. Checking this option helps to avoid unintended illegal copies.

- *Keep OT Features*: Desktop fonts may have a number of OpenType functions. As there is little browser support for these functions today, you might want to remove them with this option (rendering and miscellaneous options). In order to keep some of the standard, i.e. already supported OpenType functions such as ligatures, you can activate this option.

- *Remove Kerning*: Another possibility to reduce the file size is by checking this option which will erase all kerning values that are contained for specific letter combinations. In case you are going

to use the font for body text, this option is highly recommended. For headlines it should be handled with care or not be done at all as the missing kerning values might lead to the unpleasant effect of the shape of words looking like a Swiss cheese.

- *Simplify Outlines* does exactly that: it tries to simplify the outline of characters. As this option reduces the quality of screen display, it is not advisable to use it.

- *Build Cufón File* isn't directly part of the options for @font-face embedding. For further details on building a file you can also check the Cufón website.

SUBSETTING 🔗

Subsetting means that all superfluous characters are removed. Whether or not this is actually necessary depends on the respective font. Some fonts can easily contain thousands of characters of various writing systems which bloats the font file considerably. Without subsetting such fonts, they are not suitable for use as Web fonts.

The option *Basic Subsetting* is set as default and offers the usual Western European glyph allocation based on the character set MacRoman. *Custom Subsetting* allows for a custom defined scope of contained characters and glyphs. Whereas, the option *No Subsetting* deactivates subsetting completely and converts the font with all contained characters and glyphs.

CSS-FORMATS 🔗

The @font-face Kit Generator creates both the converted fonts and the matching CSS files which is helpful as the CSS code can be huge, especially when several fonts with various font formats are involved.

You can choose three different versions:

- Fontspring Syntax

- Further Hardening of the Bulletproof Syntax

- Mo' Bulletproofer

- Smiley

The Fontspring syntax is currently the most simple and compatible one. It can deliver the WOFF file to IE9 and EOT to IE prior version 9 and it also works on mobile operating systems like iOS and Android.

CSS OPTIONS 🔗

The option *Style Linking* groups styles by family. This allows for addressing the fonts later through the CSS properties `font weight` and `font-style`.

This option will only work properly if the font family doesn't contain more than the common four styles, i.e. regular, italic, bold and bold italic. Otherwise, you should leave the option unchecked so that fonts can be addressed by independent family names.

Last but not least, the option *Base64 Encode* embeds Web fonts with a base64 encoding into the CSS code instead of creating a separate font file. As a result, the fonts don't appear as font files in the browser's cache.

Code Sample 🔗

The following example illustrates what your CSS code for @font-face embedding might look like:

```
@font-face {
font-family: Graublauweb;
src: url('Graublauweb.eot'); /* IE9 Compatibility Modes */
src: url('Graublauweb.eot?') format('eot'),  /* IE6-IE8 */
url('Graublauweb.woff') format('woff'), /* Modern Browsers */
url('Graublauweb.ttf')  format('truetype'), /* Safari, Android, iOS */
url('Graublauweb.svg#svgGraublauweb') format('svg'); /* Legacy iOS */
}
```

The first statement is for IE9 in IE7 & IE8 render modes. In the second source declaration, the EOT file for Internet Explorer is declared first in this comma-separated list and the name is followed by a question mark. This fools IE into thinking the rest of the string is a query string and loads just the EOT file.

The other browsers follow the spec and select the format they need based on the `src` cascade and the format hint. The SVG specification contains an additional hash tag as a unique identification number. This is necessary as SVG files may contain several fonts. However, the Fontsquirrel @font-face generator takes care of the identification number and its embedding into the CSS code automatically.

Please notice that the syntax suggested by Ethan Dunham in his article The New Bulletproof @Font-Face

Syntax and revised by Richard Fink in an article on New @Font-Face Syntax: Simpler, Easier no longer works in Internet Explorer 9. Also, take a look at Ethan's follow-up Further Hardening of the Bulletproof Syntax.

LOADING TIME INCREASES WITH AMOUNT OF FONTS 🔗

By means of the above-mentioned options, file sizes of Web fonts can be reduced to approximately 30 to 60 kilobytes. Larger font files or too many fonts on one web page can slow down loading of the page, especially on mobile devices.

WEIRD INTERIM SOLUTION IN FIREFOX 🔗

Most browsers won't show any text before all Web fonts are imported. Firefox, however, displays the text using a system font and renders the text again, when the embedded Web fonts are completely loaded. This technique results in a "flash of unstyled text" that sometimes leads to side-effects. Web designers can control this behaviour by using Google's Webfont Loader.

The result of the above process looks like this in Safari on Mac OS X.

Embedding Web Fonts From Your Server 🔗

You don't necessarily always have to rent or licence fonts for use on Web pages. You can also pimp your already licenced desktop fonts for cross-browser Web font embedding and upload the fonts to your own Web server.

FONTS TO CHOOSE AND FONTS TO AVOID 🔗

Before you start converting your desktop fonts into Web fonts, make sure that the licence with the respective font entitles you to do so. Generally, commercial licence agreements do not permit the storing of font software on a publicly accessible Web server. This, however, is a prerequisite when using the `@font-face` rule.

Currently, commercial font providers count on Web font embedding services. Only a few suppliers deliver special Web font packages for storage on the customer's own Web server, among them FSI FontShop International and Fontspring. A growing number of webfont packages can also be licensed via MyFonts.com.

Once the Web Open Font Format (WOFF) is accepted as a standard, more font vendors might offer this service.

EMBEDDING FREE AND OPEN-SOURCE FONTS 🔗

Besides commercial fonts, there's a wide variety of freeware and open source fonts which you can embed into your Web pages. A list with suitable fonts that are free for use via @font-face embedding can be found on webfonts.info.

Another huge collection of such fonts is offered by Font Squirrel. It is a useful tool when it comes to converting desktop fonts into Web font formats as it offers a powerful tool, the @font-face Kit Generator.

Please note that the fonts you want to convert have to be legally eligible for Web font embedding!

Creating web fonts made easy by Font Squirrel's generator.

Tips, Tricks and Workarounds 🔗

The possibilities for using custom fonts on Web pages have developed more quickly over the past two years than anyone had expected. But Web designers still have to struggle with a clutter of formats in order to provide cross-browser support for a given font. This problem will subside as soon as the Web Open Font Format (WOFF) is established as the standard Web font format.
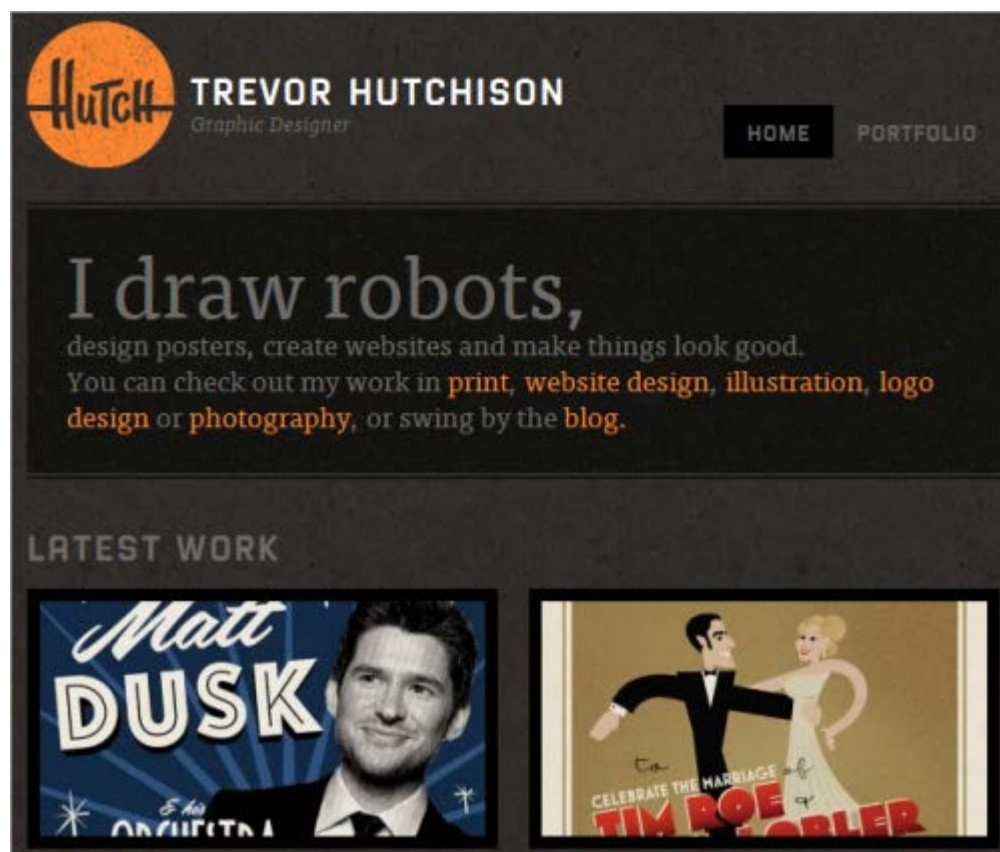
Typographic variety beyond the monotony of system fonts.

Another obstacle, mentioned earlier, is the prevalence of Windows computers on which subpixel rendering is deactivated (either by default on Windows XP or by preference on the part of users). Compared to system fonts, most Web fonts are displayed at low quality on screen displays that lack subpixel rendering. Time will solve this problem, as users replace old hardware and operating systems. In the meantime, Internet Explorer 9 works with a text engine called *DirectWrite* that delivers significantly improved rendering.

Using Web fonts in your design **requires thorough testing on as many different browsers** and

platforms as possible, with a close look at various options for rendering text. If the screen display is of poor quality and lacks subpixel rendering, then opt for graceful degradation by serving system fonts to older browsers and OS. Conditional comments are the easiest way to exclude older browsers and operating systems from style sheets with Web fonts. Of course, JavaScript is a more elegant way to detect whether a client's subpixel rendering is turned on.

TEXT LAYOUT 🔗

Even though some Web layouts are drawing nearer in sophistication to print layouts (thanks to rich typography and Web font embedding), there's still a big difference: **browsers do not automatically hyphenate text**. Especially in languages with many long words (such as German), justifying text is not possible without creating spacious gaps, and thus reducing the readability of the body text.

With JavaScript, you can provide language-based client-side hyphenation. A server-side solution is offered by phpHyphenator. Still, these are band-aid solutions for a function that should be integral to every browser — hopefully in the near future.

OPENTYPE TYPOGRAPHY FEATURES 🔗

Another text-layout shortcoming shared by common browsers has come into focus as Web fonts spread: high-end desktop publishing programs, such as InDesign and QuarkXPress 7+, do not support OpenType typographic features.

OpenType functions add several smart options that enhance a font's typographic and language-support capabilities. If an application supports these options, then characters can be replaced by additional features automatically. For Roman scripts, these mainly concern ligatures, fractions and small capitals.

FAKE SMALL CAPS
REAL SMALL CAPS 123

Computer-generated small caps (gray above) and true small caps (blue below) in Firefox 4 beta.

Some fonts can be put to use only through OpenType functions in the first place. Take joined Roman lettering or Arabic writing, in which the shape of a character depends on its position in the word and on

adjacent characters. Comprehensive OpenType fonts might, therefore, contain alternative glyphs for a given character. By means of OpenType functions, the basic version of a character would be replaced by a version with matching connections, based on context. The current beta version of Firefox 4 offers access to this OpenType feature for the first time. You can define with this browser-specific attribute:

```
h1 {
-moz-font-feature-settings: 'smcp=1';
}
```

This would display the headline in true small caps, provided that the font has a small caps case.

Microsoft has a list of common OpenType features. So far, there are no standards to address these features in your CSS style sheets, but the CSS Fonts Module Level 3 draft mentions the possibility. Other browser companies will likely follow this example sooner or later.

Browser developers still have a long way to go before the Web catches up to the text layout in print design, let alone overtakes it. It's good to know, though, that there's a growing awareness of typographic subtleties among browser manufacturers. No doubt, further progress can be expected soon!

(al) (sp) (ik)

It's finally here! Smashing Book 5: "Real-Life Responsive Web Design" is now shipping. With **front-end techniques and design patterns** from actual projects, it's a playbook to master all the tricky facets and hurdles of responsive design. Print hardcover and eBook. Free worldwide shipping.

Tweet it   Share on Facebook   ↑ Back to top

Ralf Hermann

Ralf Herrmann studied graphic design at Bauhaus University in Weimar, Germany.

He's the author of several books on typography and he founded webfonts.info, fonts.info and typografie.info (in German).

Related Articles

118

118

118

Typographic Patterns In HTML Email Newsletter Design

Responsive Typography With Sass Maps

Benton Modern, A Case Study On Art-Directed Responsive Web Typography

Advertisement

69 Comments

Vishnu H   [March 2, 2011 9:58](#)   1

[am](#)

That's a very much informative post. Thanks for this one.

It's very nice that you've started with the basics – TT, PS and SVG types, which often
confuses people. And it moves to the details.

Good article. Keep writing. :-)

2

Yvonne   [March 2, 2011 9:59](#)   2

[am](#)

Thanks for the detailed info! Any thoughts on adding "local" for src? I've always used that to
save loading time in case the users already have the font installed on their computers.
Does it really choke up the Android phones though?

0

Ralf H.   [March 3, 2011 4:17](#)   3

[am](#)

Seems to be still true. But since it is a bug, it might get fixed in the future.
You might check this page: http://code.google.com/p/android/issues/detail?id=10609

1

Yvonne   March 3, 2011 9:10 am

4

Thank you, Ralf. That's useful info.

0

Amey Gokarn   March 2, 2011 9:59 am

5

really useful article =)

0

Pothi   March 2, 2011 10:12 am

6

I used google's web fonts with varied success. Now I know why. ;-)

0

judy   March 2, 2011 10:12

am

7

and the stars align, this is just what we needed..thanks!

0

Edward M   March 2, 2011 10:34

am

8

Good article!

0

Alexander Filatov   March 2, 2011 11:17

am

9

Thank you so much for such a detailed article! Was a good and useful read!

0

Brett   March 2, 2011 12:37

pm

10

Very thorough article Ralf. This is an excellent source for how to use @font-face!

0

11

Rob   [March 2, 2011 12:37](#)

[pm](#)

I didn't see it mentioned in here, but people using this should check their work in Firefox. Firefox does not support cross-domain loading of fonts. This means that it will load a font on the www sub-domain but not on the same domain without the www.

I found a workaround for this using .htaccess posted on Stack Overflow (link below for those interested).

http://stackoverflow.com/questions/2892691/font-face-fonts-only-work-on-their-own-domain

1

12

Matthieu   [March 2, 2011 3:58](#)

[pm](#)

the other solution I found is too embed the font in the css file (select Base64 Encode in the kit generator css options). This also fixes the problem.

0

13

Rob   [March 3, 2011 10:38](#)

[am](#)

I think loading them through Google's font-loader also sidesteps this as well.

Does the embedding trick always work, or do you have to get them from a site

that will specifically embed them for you?

0

Kacper Potega March 2, 2011 1:20 pm

14

Great article! I only miss a paragraph about issues with letter-spacing and svg-fonts ...

1

Mandy Brown March 2, 2011 1:33 pm

15

For the record, Typekit no longer servers SVG fonts to iOS devices. We now serve TrueType fonts to iOS 4.2 devices and higher.

1

16 March 3, am

Vitaly Friedman (editor-in-chief of Smashing Magazine)

Thank you, Mandy. The article was updated.

1

Austin Houser
March 2, 2011 1:34
pm

17

Great article, using non-standard fonts on the web has always been a "gray area" for me.

0

James Brocklehurst
March 2, 2011 1:58
pm

18

The problem of aliased fonts is still alive and kicking as any PC running Windows XP has ClearType disabled by default, across all browsers, not just IE. And since XP still has the largest useage stats of any OS, about 40% worldwide, it can't be ignored. Many custom fonts look terrible when aliased, and some become downright illegible at smaller sizes.

I've gone through most of the decent-ish ones available on Font Squirrel, picked out the ones that look okay when aliased, and included them in a cheat sheet along with some other useful web-font info here.

1

FlorentG
March 3, 2011 2:28
am

19

The problem of aliased fonts can sometimes be circumvented by using the source OTF instead of the TTF generated by Font Squirrel.

Here's an example on Firefox 3.6 (under Vista32). Both use the same font, Calluna.

On the top is the Font Squirrel-generated TTF. Although there's some degree of antialiasing, the contrast is too high, resulting in jagged edge.

On the bottom is the original OTF, which looks way better !

-2

James Brocklehurst   March 3, 2011 6:20

20

am

@FlorentG I didn't know about the different treatment of TTF/OTF, although this doesn't help with IE6-9 that only support EOT/WOFF, or, I suspect, if ClearType is disabled, as it is in XP, in which case fonts are aliased regardless.

0

Paratron   March 2, 2011 1:58

21

pm

I would really love to start using the @font-face option, but: the output just sucks. Font's arent nicely antialiased while the same font rendered through cufón brings out

perfect rendered, but unselectable results.
I think i'll stick with cufón just a bit more…

-1

Alice   [March 2, 2011 2:27](#)

[pm](#)

22

Very nice and informative article.

-1

Marco   [March 2, 2011 2:36](#)

[pm](#)

23

Brilliant stuff Ralf,

One quick question; how necessary is it to include the statements for mobile devices, if one
tries to do a site for both platforms? (desktop and mobile devices). Clearly, the statements
would be a bit redundant, correct?

I am talking mostly about the SVG files; my iPad seems to recognize ttf quite well, but the
iPhone doesn't.

Regards,

Marco

-1

Ralf H.   [March 3, 2011 4:22](#)

24

am

This syntax offers broad support and little redundancy:
http://www.fontspring.com/blog/further-hardening-of-the-bulletproof-syntax

The iPhone will only load TrueType/OpenType fonts if it has been upgraded to iOS4.2.

-1

Lochlan   March 2, 2011 5:20

25

pm

Excellent article, thank you.

-1

patvillaruz   March 2, 2011 7:09

26

pm

useful article, im using @font-face but im looking more light font embedding.. :)

0

catalin cimpanu   March 3, 2011 1:28

27

am

Congratulations on a great article.

0

Anh Nguyet Tran   March 3, 2011 1:59

28

am

This's a helpful and interesting topic for me. Thank you !

-1

Niall   March 3, 2011 3:59

29

am

What a great article. I've only just started to do a bit of reading on the subject and had been finding it difficult getting the low down on it all. I've only used embedded fonts on one website. It's a font from the google @font-face site. For some reason however, although the fonts display correctly on my machines at home on all browsers as far as I could test, they wont display in Firefox when I check them in work. Would this be due to a firewall or something along those lines?

0

Josh Read   March 3, 2011 4:32

30

am

Amazing and well done article. Thank you for this information.

0

Justin   [March 3, 2011 9:34 am](#)   31

Head cracking... so much info...

Seriously, thanks for all the great info!

0

Misa Brezanac   [March 3, 2011 10:40 am](#)   32

Bravo Ralf! Bravo Smashing! Spot on. Exactly the kind of read I need for a job I am about to do in the next couple of hours :)

0

Santi Banks   [March 3, 2011 3:48 pm](#)   33

Very useful and in-depth article! I really hope designers and developers pick up @font-face. It makes the web a better place :)

-1

David Rodriguez   [March 3, 2011 4:22](#)

34

pm

Thanks for posting this, it was extremely helpful. My question for your is are there any
 browser versions for which the code/strategy above would not work?

 How far back will this work in Firefox for example? I have been using some other text
 replacement methods and would like to start using this one but am worried it won't work
 on the most widely used browsers.

 Are you saying this is a good/better solution than cufon, typekit and other text replacement
 methods?

-1

Ralf H.   March 4, 2011 1:21

35

am

For browser support see here:

http://www.webfonts.info/wiki/index.php?title=@font-face_browser_support

Webfonts are the "native" technology and much better than hacks like Cufon ans sIFR.

Typekit is not a different technology. It is a subscription-bases webfont provider.

0

Vanj Dolores   March 3, 2011 10:08

36

pm

keep them coming! subscribed for a year now :D

0

Ave   March 3, 2011 10:18

37

pm

Sweet! Thanks :)

0

Advitum Webdesign   March 3, 2011 10:50

38

pm

Another useful article can be found on t3n.de. But it is in german:

http://t3n.de/news/webdesign-tipps-font-face-mehr-denkt-297320/

-1

Jeff   March 4, 2011 6:08

39

pm

nice write-up. I am still held back by Windows' crappy cleartype rendering of any text-based
 embedded fonts, but hopefully we will be able to move forward with or without them.
 *and note that InDesign's OpenType support is not so great, but it is there.

0

Vincent Flanders   March 4, 2011 9:09

40

pm

See Steve Souders article "@font-face and performance" —
http://www.stevesouders.com/blog/2009/10/13/font-face-and-performance/

0

John Romant   March 5, 2011 10:23

41

am

Good stuff. Thanks for the info and for sharing the data.

0

The HeadMaster   March 7, 2011 12:20

42

am

Exelent post.

It has always been a hassle, when trying to make a great looking website.

It is a pain in the b.. when you have made a really good picture in you picture editing
 program and wanted some fancy text to go along.

This clears it up.

Thanks.

-1

Greg RIckaby [March 7, 2011 6:17](#) [am](#)

43

Great post. As always, I'll be bookmarking for future reference.

-1

Nic [March 7, 2011 2:32](#) [pm](#)

44

Good article! I've been using Google's webfont API the last few months on some small test sites, and though it's quite easy to implement, occasional loading snafu's can make a webfont-focused designs looks like a disaster. Still fun to play with the options, but not 100% reliable across browser and platforms yet.

-1

webbeetle [March 7, 2011 8:37](#) [pm](#)

45

One of the best articles on web fonts in a looooong time!!!

Even though web fonts are all the hype, not much useful in-depth information can be found about them. THANK YOU!

0

Christa [March 8, 2011 3:32 pm](#)

46

OK I hate to be a complainer, but I was so excited to read through this article, until the Corbis ad/background popped up. Talk about distracting! I understand you need to advertise, but this is just obnoxious.

2

Dog Meat [March 9, 2011 4:16 pm](#)

47

if i read one more IE digression in a CSS article, I'm gonna scream. The best usage of CSS in my opinion at the present, and I pray that Tantik or Eric Meyer are hard at work, would be the @display:IE-related:NONE Rule!

1

Vladislavs Judins [March 15, 2011 11:01 am](#)

48

Thats really good article about the font-face thing.
I've used it on my website judins.net! (no ads-links, who'll want just copy)
Thank you a lot Smashing Magazine, you've saved my life once again :)

0

J. Hendrix

March 25, 2011 4:03 am

49

Excellent article, very detailed. It answers a lot of questions that I had about embedding fonts in webdesign.

0

Sagar Ranpise

March 30, 2011 10:41 pm

50

Is there a way of adding 2 font-face rules ? I mean if we want to add 2 different fonts ?

-2

Ralf H.

April 20, 2011 10:29 pm

51

Of course you can add multiple @font-face rules. Just make sure you apply a different family (and or) style name. Otherwise the new rule will override the first one.

0

Sagar Ranpise

March 30, 2011 10:47

52

pm

Great its working even if we use 2 font-face rules for different font!!!

-1

Ruud
June 23, 2011 10:26

53

pm

Font-face is great, but there's one thing that's very anoying:
Firefox for Mac always give bolder fonts. I first also had this on Webkit browsers but I found
a workaround to use: -webkit-font-smoothing: antialiased;

Does other people also have this strange problem?

0

Rosh
August 3, 2011 12:19

54

pm

very illuminating post. Thanks..

0

tota
September 7, 2011 7:16

55

am

what about fonts support Arabic language

@font-face not work in IE8/7 but work only in IE9

plz advice

0

Rob

[September 25, 2011 4:11](#)

[pm](#)

56

Ahh great finally a good article regarding fonts

-1

Tien Do Xuan

[September 30, 2011 10:52](#)

[pm](#)

57

http://fontface.codeandmore.com/ a simple font-face generator but works beautifully.

3

Maksim

[October 10, 2011 9:21](#)

[am](#)

58

How to convert eot format back to ttf?

0

Dave

[October 31, 2011 3:48](#)

59

am

Does anyone know how to disable web fonts/font-face in firefox?

0

Andi December 14, 2011 4:10

60

pm

Great article.

0

2

↑ Back to top

With a commitment to quality content for the design community.

Founded by Vitaly Friedman and Sven Lennartz. 2006-2011.

Made in Germany.   Write for us – Contact us – Impressum.