

Dear Typekit...

AUTHOR: Steven Robertson
CONTACT: steven@strobe.cc
PUBLISHED: 2009-10-28
TAGS: Typography, Web, Article

ABSTRACT

If you're using a supported browser, the fonts you see on this site are provided by [Typekit](#). It's generally an awesome service, but it doesn't yet work for most Linux users.

UPDATE: Typekit has done one better, not only using Gecko browser detection but also [using WOFF for Firefox 3.6 and up](#). Took a few months, but they got it done the right way.

Dear Typekit,

Your service is currently broken. The good news: it's a one-line fix.

The JavaScript generated for each client includes the following regex against `navigator.userAgent`, intended to check whether or not the browser is compatible with `@font-face` and Typekit:

```
1 function(D){  
2     var C=D.match(/Firefox\/(\d+\.\d+)/);  
3     if(C){  
4         var B=C[1];  
5         return parseFloat(B)>=3.5  
6     }  
7 }
```

The problem is that many Linux distributors [can't legally call their browser Mozilla Firefox](#). The Debian project is a notable example; they've rebranded their Firefox build "Iceweasel", and chosen similar names for other Mozilla software. To avoid this dispute, other distributions have taken to using the code-name for a particular Firefox build as the name—I'm currently posting this from a browser called "Shiretoko". This doesn't even cover contexts in which the Gecko engine is embedded by other applications which fully support TypeKit, such as Mozilla SeaMonkey.

It would be unreasonable to check for every variant of the browser name in the JavaScript handed out by your application. Fortunately, Mozilla has made a provision allowing the right decision to be made, regardless of browser name. Here's the value of `navigator.userAgent` from the official build of Firefox:

```
Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1.3) Gecko/20091021 Firefox/3.5.3
```

Here's what it looks like in my browser:

```
Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1.3) Gecko/20091021 Shiretoko/3.5.3
```

Note the common string component `rv:1.9.1.3`; this identifies the Gecko release version. Since browsers based on the Gecko rendering engine get most of their characteristics from that engine, you can in almost all cases simply check the Gecko version instead of the Firefox version¹.

Doing so is really this simple:

```
1 function(D){
2     var C=D.match(/rv:(\d+\.\d+).*Gecko\/\//);
3     if(C){
4         var B=C[1];
5         return parseFloat(B)>=1.9
6     }
7 }
```

Since this change is a simple one, and since Gecko's user-agent strings have been standardized for a while, I hope that you can make it soon, and enable support for the many users using browsers which would otherwise work flawlessly with Typekit.

Thank you for providing an awesome and much-needed service.

Steven

PS: For users of unbranded Firefox versions who just want things to work now: go to `about:config` and change the string `general.useragent.extra.firefox` to `Firefox/3.5.3` (or your current version). But remember to update it along with your browser.

¹Ideally, you shouldn't use the user-agent string at all to do browser detection; [Mozilla says why, and what to do instead](#). But often user-agent checking is the pragmatic solution.