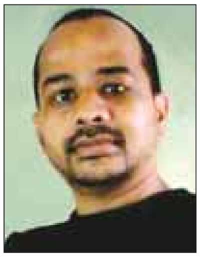
Cufón: Custom Font Embedding Technique for Web

Cufón, pronounced as Koofo, is one among the popular font replacement techniques of its kind, which has attained much attention and popularity among web developers. Thanks to Simo Kinnuenen, a Finland-based web guru who developed this technique, which has evolved and enabled us to use non-web, based fonts on web pages with much ease.



MOHAN DAS
Technical Director
NIC HQ
mohandas@nic.in

Many of you would have been curious to know how some websites and blog pages display awesome choice of fonts with smoother curves on your browsers. You would have also noticed that such web pages stand out visually attractive and mind grabbing. Well, kudos to the several researchers and web developers for their efforts in making it possible for us to embed numerous customized fonts so as to make the page look unique and special. Now, with some extra effort along the process of website development, you too can make this happen on your website. Needless to say, all these come with some concerns of fonts licensing and certain technology limitations for accessibility and user experience. Like many other designers who admire and get inspired of the beauty and function of fonts, which significantly enhance the web surfing experience, I happen to be one who is curiously observant about how emerging technologies draw and extend boundaries for expanding possibilities.

CUFÓN

Cufon is a JavaScript based font replacement technique, a popular of its kind and an effective alternative to the earlier techniques such as sIFR (Scalable Inman Flash Replacement) and FLIR (Face Lift Image Replacement). Cufon emerged trumps among the others for many, being faster, by using normal CSS (Cascading Style Sheet) to change the attributes of Cufón font selectors. Well, such other techniques, even with their certain merits, still remain painfully tricky to set up and use. Prior to these developments, web sites were constrained by pages displaying only the web-safe fonts, which are still a handful. Though designers and developers used images and splash as alternatives (many still do the same), these are either non-accessible, non-scalable or require plug-ins to display such variety of fonts on web pages.

WHY CUFÓN?

Cufón has a better edge over the others due to its specific merits:

1. It uses features that are natively supported. Hence the client machine requires no plug-ins.
2. Ithas better Browser Compatibility. Cufon works on every major browser available on net now.
3. It is easy to use. Cufon requires no or almost zero configuration for standard use cases.
4. Speed of rendering is fast, even for sufficiently large amounts of text.

HOW DOES CUFÓN WORK?

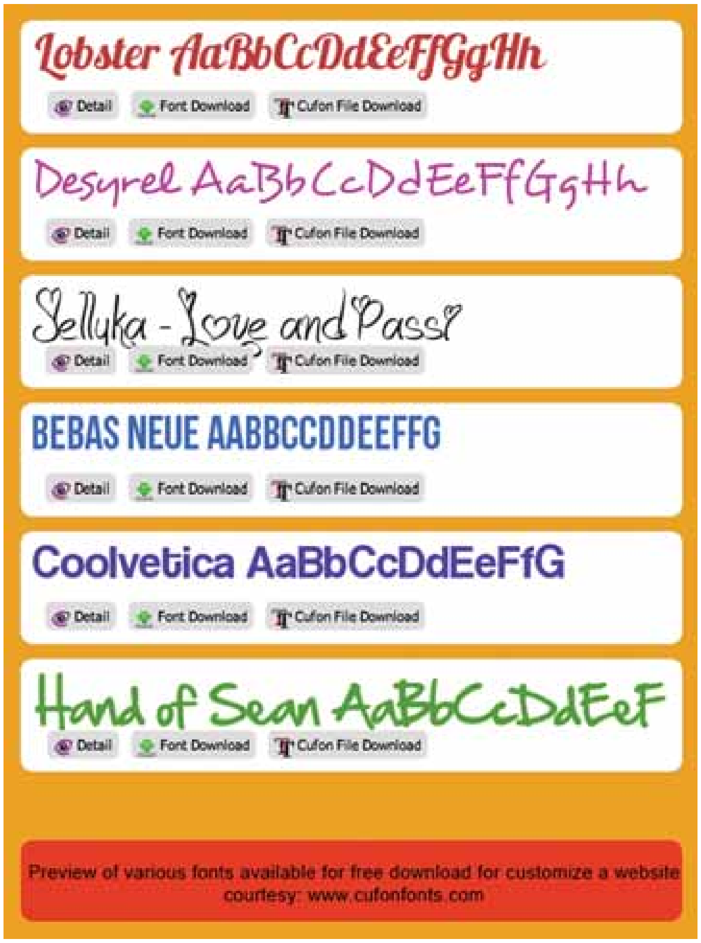
Cufón setup and use consists of two individual parts - a **Font Generator**, which converts fonts to a proprietary format and a **Rendering Engine** written in JavaScript.

THE FONT GENERATOR

This is a bit more than a web-based interface to FontForge (a typeface editor program). Initially, the generator builds a custom FontForge script based on user input and then runs it, which saves the font as an SVG

(Scalable Vector Graphic)font. The SVG font is then parsed and its path converted to VML (Vector Markup Language) paths. The resulting document is then converted to JSON (JavaScript Object Notation) with a blend of functional Javascript. This has many advantages:

- To include a font, one just needs to load it with the standard <script> tag as any other JavaScript file and it would be registered automatically.
- No requirement to manually parse the file on client-side repeatedly.
- External JavaScript files block execution until they are loaded, which helps us to achieve a flicker-free, clean replacement.
- It compresses fonts extremely well. A compressed font usually weighs 80 to 60% less than the original. The process is then repeated for the rest of the fonts provided to the Generator and the resulting JavaScript file is sent back to the client with a distinctive filename.



Preview of various fonts available for free download for customize a website courtesy: www.cufonfonts.com

THE RENDERER

In comparison, this part is slightly tougher. The Renderer consists of further 3 parts: **A Core** and **two Rendering Engines**. A Core provides the API (Application Program Interface) and common functional aspects. One Renderer out of the two, renders VML shapes, while the other Rendering Engine uses the widely supported HTML5<canvas> element. Since the path data is already VML, a bit of work is needed in the VML engine. The canvas engine thus must convert all paths to the corresponding sets of drawing commands provided by the canvas API. This turns out to be somewhat tricky initially, but a solution emerged after the second Engine complete rewrites in the form of code generation and caching, resulting in a very fast renderer. Inline SVG is surprisingly slow in a few browsers that really supported it properly.

CUFÓN AND ACCESSIBILITY ISSUES

Along with many advantages of using Cufón over other alternatives, there are also issues regarding web accessibility which are yet to be resolved. But the initial inability of text selection in Cufón has been worked out to a considerable extent. In IE(6-8), Safari 4, Chrome and FireFox3 you are still able to select the original HTML text. (FireFox3 won't show the text selection visually but when you select it and hit ctrl+c you will be able to paste it.) In screen readers, each word in text rendered by Cufón being a separate span element, it makes several screen readers such as VoiceOver, FireVox, ORCA, Window-Eyes and Opera Voice read: The. Text. As. If. Each. Word. Was. Separated. By. A. Full. Stop. In some cases the screen reader will stop after reading the first word of a Cufón-replaced sentence. Setting the separate option to 'none' in the Cufón API is found to be a way to avoid this problem. By doing this, Cufón does not wrap each word in a separate span element.

CONCLUSION

Despite having some yet to be resolved issues of font security, for a huge lot, Cufón still remains a favourite technique for font embedding. For some who feels '@font-face' or a 'TypeKit' style of service better for usage of large number of fonts, they are still constrained with certain font rendering issues on some OS's or browsers. Fonts that look brilliant on Chrome in OSX (mac), look terrible in Windows XP Firefox, for example....

With the evolving research activities and innovative web font managing techniques, we can anticipate newer developments and surprises from Cufón and other font handling techniques supported by robust technology. In general, embedding websites with fonts enabling accessibility, which enhances user experience for all, will always be an integral part of an ideal process of web development, thus supporting the well-accepted thought of Inclusive Web design.