It seemed like only yesterday the king of the wide web world was Pixel. You just needed to write a precise number in a CSS file and the object of the exact size, as you wanted it to, appeared in a browser. But then dozens of mobile gadgets uncrowned the king Pixel and new times came. In this post, we will explore the properties of icon font and share some advice on how to use it effectively.

Nowadays it’s a great challenge for web designers and developers to discover ways to create flexible and adaptive objects. So that every user of any device with any size and platform could get a perfect picture on a screen. We are all taking part in building a brand new world. And icons have become a very important part of it. The less space we have available, the smaller signs we should draw to indicate user’s interface elements. But at the same time these signs are needed to be responsive so we can’t use raster images. It looks like Icon Font is today’s hero. Like the other new, icon fonts have disadvantages too. Let’s start with the pros and cons.

Pros

Cons

- Infinitely scalable, and it works well both on standard and retina displays;
- Can be styled with a CSS including text-shadow, gradient, color, :hover and :focus pseudo-classes and other properties. It also helps when you need to change the whole set of icons: just change a CSS rule — no necessity to redraw every icon;
- Animatable;
- Cross browser compatibility;
- Fewer pictures — fewer requests to a server side.

- No ability to create multi-coloured icons: just single color or gradient;
- Not pixel-perfect sharp;
- It's long and hard work to create such a font.

So icon font is a good solution for web apps with a great number of icons of the same style.

Embedding Icon Font

1. Pick out an icon font. First of all you have to pick out an icon font that you'd like to use. But sometimes the chosen font doesn't contain the symbols that you need. Or you don't need all icon set symbols and want to reduce the size of the font package. And you may also want to combine your own font with some others. Fortunately, some people have already solved these problems, so you can go and use Fontello and IcoMoonservices. You just need to upload your own font and/or select icons from a prepared icon sets.

2. Map icons. After you determine which icons will form your font you need to map them to letters or to Unicode symbols. The second way is the preferred one, because it means that your icons won't be spoken by a screen reader. Both IcoMoon and Fontello map icons to Private User Area of Unicode can freely change the value if you'd like to.

3. Download icon font. Downloaded .zip archive will contain all the web font formats (.woff, .ttf, .svg, .eot) and also a sample of html/css.

4. Include font in your project. Surely using the @font-face rule.

CSS

```
@font-face{
```

5. Create a Markup. There are usually two ways to use icons in web: standalone and with word/phrase.

   - **Standalone icon.** These icons are surely functional but have no visual description. So markup should provide a textual explanation which is very important for SEO and for making this functionality audibly obvious with the use of screen readers. Data-icon is the HTML5 data-* attribute which allows you to store some CSS and JS accessible custom values and we use it to store the exact character which we want to use. And the absolute position (with such big negative coordinates) for the text block helps to make this text invisible on the screen though it is still presented in the HTML source code.

     HTML

     ```
                                     <span aria-hidden=Ã¢â,¬Â?trueÃ
     icon=Ã¢â,¬Â?&#9881?                                    ></span>
     ```

     CSS

   - **Icon with text.** These types of icons shouldn't be said aloud by screen readers, because we already have a text for that. So now we use an aria-hidden="true" attribute for the icon HTML container element. We used: before pseudo-element to generate symbol in browser while the HTML remains simple. Speak is the property that specifies whether text will be rendered aurally. It's a new feature so it cannot be trusted as cross-browser standard but we can still use it for future support.

     HTML

```
icon=Ã¢â,¬Â?&#9881?                                          ></span>
```

CSS