

## Web fonts

← Previous

↑ Overview: Styling text

Next →

In the first article of the module, we explored the basic CSS features available for styling fonts and text. In this article we will go further, exploring web fonts in detail — these allow you to download custom fonts along with your web page, to allow for more varied, custom text styling.

### Prerequisites:

Basic computer literacy, HTML basics (study [Introduction to HTML](#)), CSS basics (study [Introduction to CSS](#)), [CSS text and font fundamentals](#).

### Objective:

To learn how to apply web fonts to a web page, using either a third party service, or by writing your own code.

---

## Font families recap

As we looked at in [Fundamental text and font styling](#), the fonts applied to your HTML can be controlled using the `font-family` property. This takes one or more font family names, and the browser travels down the list until it finds a font it has available on the system it is running on:

```
1  p {  
2    font-family: Helvetica, "Trebuchet MS", Verdana, sans-serif;  
3  }
```

This system works well, but traditionally web developers' font choices were limited. There are only a handful of fonts that you can guarantee to be available across all common systems — the so-called [Web-safe fonts](#). You can use the font stack to specify preferable fonts, followed by web-safe alternatives, followed by the default system font, but this adds overhead in terms of testing to make sure that your designs look ok with each font, etc.

---

## Web fonts

But there is an alternative, which works very well, right back to IE version 6. Web fonts are a CSS feature that allows you to specify font files to be downloaded along with your website as it

is accessed, meaning that any browser that supports web fonts can have exactly the fonts you specify available to it. Amazing! The syntax required looks something like this:

First of all, you have a `@font-face` block at the start of the CSS, which specifies the font file(s) to download:

```
1  @font-face {  
2      font-family: "myFont";  
3      src: url("myFont.woff");  
4  }
```

Below this you can then use the font family name specified inside `@font-face` to apply your custom font to anything you like, as normal:


```
1  html {  
2      font-family: "myFont", "Bitstream Vera Serif", serif;  
3  }
```

The syntax does get a bit more complex than this; we'll go into more detail below.

There are two important things to bear in mind about web fonts:

1. Browsers support different font formats, so you'll need multiple font formats for decent cross-browser support. For example, most modern browsers support WOFF/WOFF2 (Web Open Font Format versions 1 and 2), the most efficient format around, but older versions of IE only support EOT (Embedded Open Type) fonts, and you might need to include an SVG version of the font to support older versions of iPhone and Android browsers. We'll show you below how to generate the required code.




2. Fonts generally aren't free to use. You have to pay for them, and/or follow other license conditions such as crediting the font creator in the code (or on your site). You shouldn't steal fonts and use them without giving proper credit.

 **Note:** Web fonts as a technology have been supported in Internet Explorer since version 4!

---

## Active learning: A web font example

With this in mind, let's build up a basic web font example from first principles. It is difficult to demonstrate this using an embedded live example, so instead, we would like you to follow the steps detailed in the below sections, to get an idea of the process.

You should use the  [web-font-start.html](#) and  [web-font-start.css](#) files as a starting point to add your code to (see the  [live example](#)). Make a copy of these files in a new directory on your computer now. In the `web-font-start.css` file, you'll find some minimal CSS to deal with the basic layout and typesetting of the example.

### Finding fonts

For this example, we'll use two web fonts, one for the headings, and one for the body text. To start with, we need to find the font files that contain the fonts. Fonts are created by font foundries and are stored in different file formats. There are generally three types of sites where you can obtain fonts:

- A free font distributor: This is a site that makes free fonts available for download (there may still be some license conditions, such as crediting the font creator). Examples include [Font Squirrel](#), [dafont](#), and [Everything Fonts](#).
- A paid font distributor: This is a site that makes fonts available for a charge, such as [fonts.com](#) or [myfonts.com](#). You can also buy fonts directly from font foundries, for example [Linotype](#), [Monotype](#), or [Exljbris](#).
- An online font service: This is a site that stores and serves the fonts for you, making the whole process easier. See the [Using an online font service](#) section for more details.

Let's find some fonts! Go to [Font Squirrel](#) and choose two fonts — a nice interesting font for the headings (maybe a nice display or slab serif font), and slightly less flashy and more readable font for the paragraphs. When you find each font, press on the download button, and save the file inside the same directory as the HTML and CSS files you saved earlier. It doesn't matter whether they are TTF (True Type Fonts) or OTF (Open Type Fonts).

In each case, unzip the font package (Web fonts are usually distributed in ZIP files containing the font file(s) and licensing information). You may find multiple font files in the package — some fonts are distributed as a family with different variants available, for example thin, medium, bold, italic, thin italic, etc. For this example, we just want you to concern yourself with a single font file for each choice.

**Note:** Under the "Find fonts" section in the right-hand column, you can click on the different tags and classifications to filter the displayed choices.

## Generating the required code

Now you'll need to generate the required code (and font formats). For each font, follow these steps:

1. Make sure you have satisfied any licensing requirement, if you are going to use this in a commercial and/or Web project.
2. Go to the [Fontsquirrel Webfont Generator](#).
3. Upload your two font files using the *Upload Fonts* button.
4. Check the checkbox labeled "Yes, the fonts I'm uploading are legally eligible for web embedding."
5. Click *Download your kit*.

After the generator has finished processing, you should get a ZIP file to download — save it in the same directory as your HTML and CSS.

## Implementing the code in your demo

At this point, unzip the webfont kit you just generated. Inside the unzipped directory you'll see three useful items:

- Multiple versions of each font: (for example `.ttf`, `.woff`, `.woff2`, etc.; the exact fonts provided will be updated over time as browser support requirements change). As mentioned above, multiple fonts are needed for cross browser support — this is Fontsquirrel's way of making sure you've got everything you need.
- A demo HTML file for each font — load these in your browser to see what the font will look like in different usage contexts.
- A `stylesheet.css` file, which contains the generated `@font-face` code you'll need.

To implement these fonts in your demo, follow these steps:

1. Rename the unzipped directory to something easy and simple, like `fonts`.

2. Open up the `stylesheet.css` file and copy both the `@font-face` blocks contained inside into your `web-font-start.css` file — you need to put them at the very top, before any of your CSS, as the fonts need to be imported before you can use them on your site.
3. Each of the `url()` functions points to a font file that we want to import into our CSS — we need to make sure the paths to the files are correct, so add `fonts/` to the start of each path (adjust as necessary).
4. Now you can use these fonts in your font stacks, just like any web safe or default system font. For example:

```
1 | font-family: 'zantrokeregular', serif;
```

You should end up with a demo page with some nice fonts implemented on them. Because different fonts are created at different sizes, you may have to adjust the size, spacing, etc., to sort out the look and feel.


# Hipster ipsum is the best

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Cray food truck brunch, XOXO +1 keffiyeh pickled chambray waistcoat ennui. Organic small batch paleo 8-bit, Intelligentsia umami wayfarers pickled, asymmetrical kombucha letterpress kitsch leggings cold-pressed squid chartreuse put a bird on it. Listicle pickled man bun cornhole heirloom art party.

## It is the quaintest

Bespoke green juice aesthetic leggings DIY williamsburg selvage. Bespoke health goth tote bag, fingerstache venmo ennui thundercats butcher trust fund cardigan hella. Wolf vinyl you probably haven't heard of them taxidermy, ugh quinoa neutra meditation asymmetrical mixtape church-key kitsch man bun occupy. Knausgaard butcher raw denim ramps, offal seitan williamsburg venmo gastropub mlkshk cardigan chillwave chartreuse single-origin coffee twee. Ethical asymmetrical banjo typewriter fap. Polaroid waistcoat tousled selfies four dollar toast locavore thundercats. Truffaut post-ironic skateboard trust fund.

 **Note:** If you have any problems getting this to work, feel free to compare your version to our finished files — see [web-font-finished.html](#) and [web-font-finished.css](#) ([run the finished example live](#)).

---

Using an online font service 



Online font services generally store and serve fonts for you, so you don't have to worry about writing the `@font-face` code, and generally just need to insert a simple line or two of code into your site to make everything work. Examples include [Adobe Fonts](#) and [Cloud.typography](#). Most of these services are subscription-based, with the notable exception of [Google Fonts](#), a useful free service, especially for rapid testing work and writing demos.

Most of these services are easy to use, so we won't cover them in great detail. Let's have a quick look at Google fonts, so you can get the idea. Again, use copies of `web-font-start.html` and `web-font-start.css` as your starting point.

1. Go to [Google Fonts](#).
2. Use the filters on the left-hand side to display the kinds of fonts you want to choose and choose a couple of fonts you like.
3. To select a font family, press the  $\oplus$  button alongside it.
4. When you've chosen the font families, press the *[Number] Families Selected* bar at the bottom of the page.
5. In the resulting screen, you first need to copy the line of HTML code shown and paste it into the head of your HTML file. Put it above the existing `<link>` element, so that the font is imported before you try to use it in your CSS.
6. You then need to copy the CSS declarations listed into your CSS as appropriate, to apply the custom fonts to your HTML.

**Note:** You can find a completed version at [google-font.html](#) and [google-font.css](#), if you need to check your work against ours ([see it live](#)).

---

## @font-face in more detail [↗](#)

Let's explore that `@font-face` syntax generated for you by fontsquirrel. This is what one of the blocks looks like:

```
1  @font-face {  
2    font-family: 'ciclefina';  
3    src: url('fonts/cicle_fina-webfont.eot');  
4    src: url('fonts/cicle_fina-webfont.eot?#iefix') format('embedded-opentype'),  
5          url('fonts/cicle_fina-webfont.woff2') format('woff2'),  
6          url('fonts/cicle_fina-webfont.woff') format('woff'),  
7          url('fonts/cicle_fina-webfont.ttf') format('truetype'),  
8          url('fonts/cicle_fina-webfont.svg#ciclefina') format('svg');  
9    font-weight: normal;  
10   font-style: normal;  
11 }
```

This is referred to as "bulletproof @font-face syntax", after a post by Paul Irish from early on when `@font-face` started to get popular ([↗ Bulletproof @font-face Syntax](#)). Let's go through it to see what it does:

- `font-family`: This line specifies the name you want to refer to the font as. You can put this as anything you like, as long as you use it consistently throughout your CSS.
- `src`: These lines specify the paths to the font files to be imported into your CSS (the `url` part), and the format of each font file (the `format` part). The latter part in each case is optional, but it is useful to declare it because it allows browsers to find a font they can use faster. Multiple declarations can be listed, separated by commas — the browser will search through them and use the first one it can find that it understands — it is therefore

best to put newer, better formats like WOFF2 earlier on, and older, not so good formats like TTF later on. The one exception to this is the EOT fonts — they are placed first to fix a couple of bugs in older versions of IE whereby it will try to use the first thing it finds, even if it can't actually use the font.

- `font-weight`/`font-style`: These lines specify what weight the font has, and whether it is italic or not. If you are importing multiple weights of the same font, you can specify what their weight/style is and then use different values of `font-weight`/`font-style` to choose between them, rather than having to call all the different members of the font family different names. ↗ [@font-face tip: define font-weight and font-style to keep your CSS simple](#) by Roger Johansson shows what to do in more detail.

**Note:** You can also specify particular `font-variant` and `font-stretch` values for your web fonts. In newer browsers, you can also specify a `unicode-range` value, which is a specific range of characters you want to use out of the web font — in supporting browsers, only the specified characters will be downloaded, saving unnecessary downloading. ↗ [Creating Custom Font Stacks with Unicode-Range](#) by Drew McLellan provides some useful ideas on how to make use of this.

---

## Summary [↗](#)

Now that you have worked through our articles on text styling fundamentals, it is time to test your comprehension with our assessment for the module, Typesetting a community school homepage.

---

## In this module

- Fundamental text and font styling
  - Styling lists
  - Styling links
  - Web fonts
  - Typesetting a community school homepage
-