

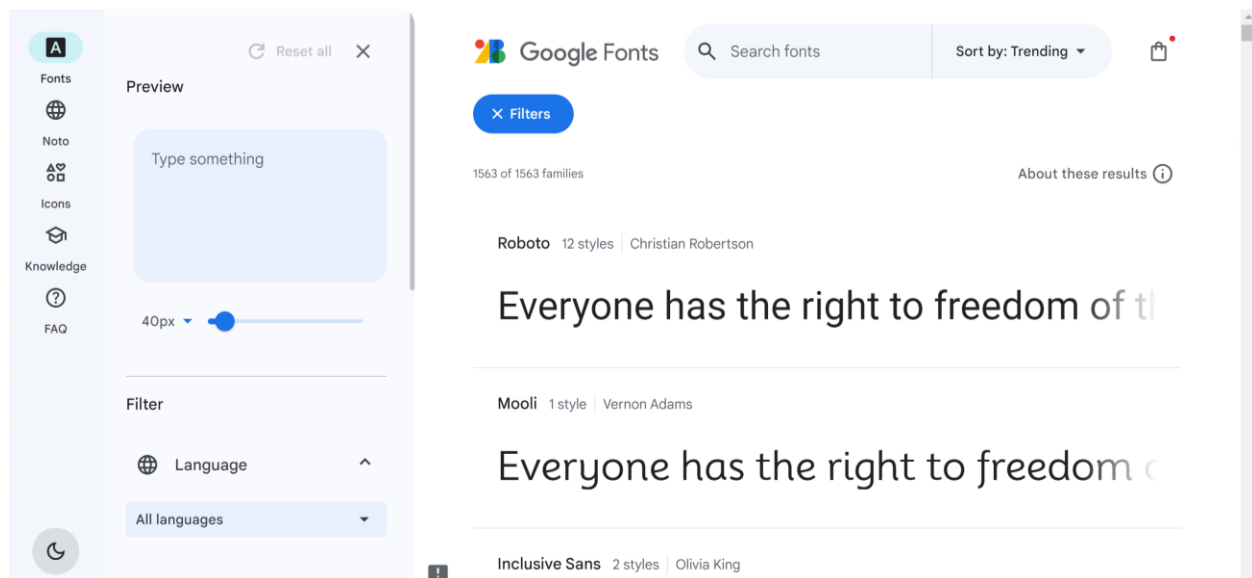
Webfonts

Prior to this, you have had a very brief intro to adding fonts to your CSS, but what if you want non-standard fonts (i.e. not fonts from standard font stacks)? This week, we'll look at fonts from other font sources.

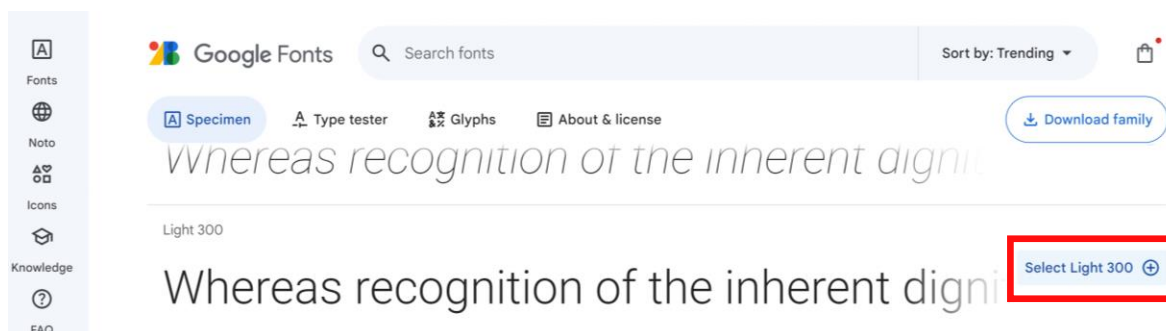
Google Fonts (<https://fonts.google.com/>)

This is a huge source of fonts for developers and it's easy to use. You simply go the site, choose the font you want then use the simple CSS `@import` statement to add the font to your CSS.

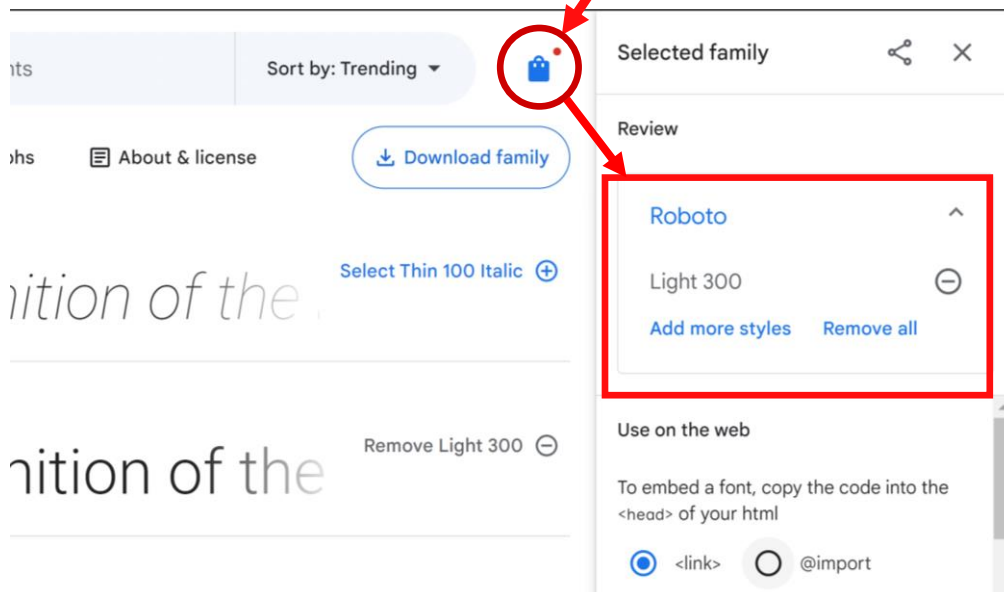
1. On Google Fonts, find a font you like. (You can filter by font type using the filters pictured below.)



2. Select the font you want to use by clicking on the font you like, then scroll down to find all the available styles. Click the + to add only the styles you'll use. Don't select all styles if you're not using them because the more styles selected, the more negative the impact on page load times.



3. Your selected fonts will show up in a panel accessible from the icon in the top-right corner of the page which looks like some tiles and a plus sign.



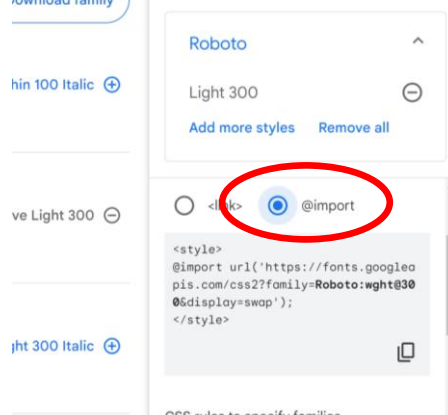
4. You can continue to add more styles from the same font or from different fonts.

Caution!

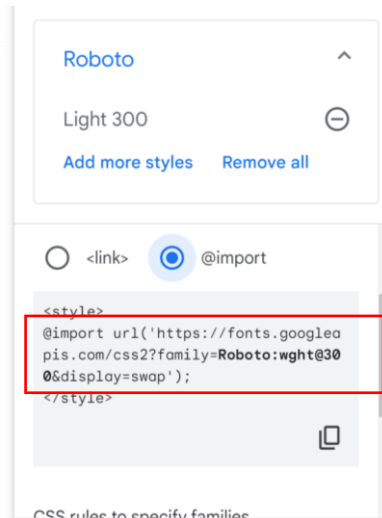
Be aware that the more you choose the slower your page will load. In older versions of Google Fonts, Google displayed a message indicating how fast your page will load with your selected font styles, but this has been removed.

Try to keep the number of styles selected down to no more than three if you can help it to have fast loading times.

5. To add the font to your webpage, click on the **@import** radio button in your style selection pane to find the CSS @import statement.



6. Copy the @import statement. You only need the line which starts with "@import".



Tip!

Select **all of the fonts/font styles** you want to use **BEFORE** copying the @import statement. Google will amalgamate all styles in one @import statement. This is more efficient and faster than running multiple @import statements for multiple font families.

7. Paste the @import statement at the **top** of your CSS file. Although you can add the <link> into your HTML page as an alternative, putting the import statement in your CSS file keeps everything neat and keeps the styling stuff separate from your HTML.
8. Below the "<link>" and "@import" embed methods, there are instructions on how to use the font in your CSS.

CSS rules to specify families

```
font-family: 'Roboto', sans-serif;
```

Tip!

You can download the fonts from Google Fonts. This would allow you to install them on your computer to be used in your design software. You can use the "Download family" link on the font page **OR** you can select all the styles you want and click on the "Download all" button in the sidebar to download all selected fonts in one go. Install the fonts using your normal font-installation procedure for your computer.

Font Squirrel (<http://www.fontsquirrel.com>)

This is an online source of fonts for desktop, web, and ebook formats. You will need to be careful when selecting fonts from this website, however. Make sure the font is for web use and that you have license to use the fonts (not all fonts are free OFL fonts).

What is the difference between a desktop font and webfont?

The main difference is in how they render in a browser. Webfonts are usually optimized to be used in browsers and in your CSS via the @font-face declaration.

Embedding Font Files (@font-face)

You can also embed your own font files into your CSS by declaring a new custom font-family and pointing the new font-family to a set of files. For the most part, I would recommend you use webfonts (such as Google Fonts) as much as possible. Sometimes it is necessary to use a font file provided by the client. This—if there is no online webfont version available—is when you would need to embed the font. (Note that technically you can also download fonts from Google Fonts and embed for a slightly faster font loading time.)

For the highest compatibility, you will need the following font file formats:

- EOT
- WOFF (and WOFF2)
- TTF
- SVG

EOT (Embedded OpenType)

This is a file format developed by Microsoft to be embedded in web pages. This is for use with IE4+, IE8 and under **will not recognize any format other than EOT.**

WOFF (Web Open Font Format)

This format was developed by Mozilla and some other organizations for web use. This was a reaction to OTF and TTF and WOFF actually is a compression of the structures used by OTF and TTF. The newest browsers use WOFF2 (.woff2).

This format loads pretty quickly. This format is supported by Firefox 3.6+. (See the format compatibility table below for full compatibility.)

TTF (TrueType Font)

TTF is recognized by Safari, Android, and iOS.

SVG (Scalable Vector Graphics)

This is a font created as vectors. This is the **only format accepted by Safari 4.1 and under for iOS.**
Font Format-Browser Compatibility Table

	<i>WOFF</i>	<i>OTF</i>	<i>TTF</i>	<i>SVG</i>	<i>EOT</i>
IE	9	9	9	none	4
Firefox	3.6	3.5	3.5	none	none
Opera	11.1	10	10	10	none
Opera Mobile	11.0	9.7	9.7	9.7	none
Safari	5.1	3.1	3.1	3.1	none
Chrome	6	4	4	0.3	none
Safari on iOS	5	none	4.2	3.1	none

Note: For OTF/TTF, Chrome 3 *could* technically support it but you needed to do a command-line switch to enable support.

The above table was taken from <http://www.stunningcss3.com/resources/fontface-file-types-browser-support.html>. This is a pretty complete table, but **nowadays, you probably only need WOFF, WOFF2 and perhaps TTF.**

To set your own font-face declaration, use the CSS below. Put the snippet at or the top of your CSS file. (It is preferable to put your @ statements such as @import and @font-face at the top of your CSS file.)

```
@font-face {
  font-family: 'CustomFont';
  src: url('webfont.eot'); /* IE9 Compat Modes */
  src: url('webfont.eot?#iefix') format('embedded-opentype'), /* IE6-IE8 */
       url('webfont.woff2') format('woff2'), /* Super Modern Browsers */
       url('webfont.woff') format('woff'), /* Pretty Modern Browsers */
       url('webfont.ttf') format('truetype'), /* Older Safari, Android, iOS */
       url('webfont.svg#svgFontName') format('svg'); /* Legacy iOS */
}
```

Note that in the above CSS, there is a "?#iefix" for IE6 to IE8 but this is not required for IE9. The actual words (iefix) doesn't matter. You could even put "?#foovalue". The word is arbitrary but "?#iefix" is used out of convention. The reason for putting the word for IE6 to IE8 is because of two possible bugs which may or may not occur:

- IE8 and under sometimes struggles to parse src properly and will cause an error when there are multiple file formats. The "?" tricks the browser into thinking that the rest of the src text is a query string and will only load the EOT. Without it you may get a 404 error.
- Sometimes IE will fail because of too many characters in the @font-face declaration.

The embedding code above is for the highest compatibility. In newer documentation you may only see WOFF and WOFF2 font files embedded because these are the recommended font formats for modern browsers. If so, you would only need (if you don't need TTF you can omit that line and change the comma for the second source to a semi-colon):

```
@font-face {
  font-family: 'CustomFont';
  src: url('webfont.woff2') format('woff2'), /* Super Modern Browsers */
       url('webfont.woff') format('woff'), /* Pretty Modern Browsers */
       url('webfont.ttf') format('truetype'); /* Older Safari, Android, iOS */
}
```

}

Note that this may not be realistic depending on how broad your compatibility needs to be.

What if you are missing a file format?

You can use Font Squirrel's Webfont Generator Tool (<http://www.fontsquirrel.com/tools/webfont-generator>) to generate the formats. This tool is free to use but **be sure that you have license to use the font in your website before creating font formats for the web** (see below).

OFL (Open Font License)

You need to be careful when you use non-standard fonts in your website. If you want to use a font you need to be sure that it is free for you to use. Look for OFL when you are searching for fonts if you want free fonts.

Non-OFL fonts usually need to be purchased to give you the license to use them. Look out for which fonts are free to use and which need to be purchased. Also check if fonts are free to use for **commercial** use if they are free.

Caution!

Whoever is using the font needs to purchase the font (if it's not free).

Make sure you read the fine print. Be sure that you know how many computer *machines* the license is good for. (Your license may only be applicable for ONE computer.) The font license is only needed for the computer which is using the font. This means that if your font file is stored on the same computer as the CSS file which is using the font, the font license should be for that one computer. If your font is stored on one computer while the CSS using the font is on another computer you will need a license for TWO computers.

Typography in design

Fonts can really set the mood or tone of a website and using them thoughtfully can set your website apart. One trend is to use very large image backgrounds with large text over top.

Some general rules of thumb are:

- Fancy fonts should be used sparingly. Too many fancy fonts make the webpage look unprofessional and too busy on the eyes. They are also harder to read.
- Don't use too many different font types. **If design is not your strong suit, try not to use more than two different font types.** Too many can make your design look childish or amateurish.
- If you are using decorative fonts, does it match the branding of your site? Decorative fonts can be nice if they go with the branding and reinforce your branding. But again, use sparingly. Just because it goes with your brand, doesn't mean you should use it as your paragraph font.

- For a more modern look, consider clean, sans-serif fonts. Serif fonts, although they can also look modern, tend to feel more traditional.

Google Fonts has helpful [articles about choosing fonts](#). A useful one is about [pairing typefaces](#).

Impact of Web Fonts

Web fonts *do* make an impact on page load times so you need to be careful about how many you use and how large the fonts are. The reason for this is that fonts still need to be downloaded before they can be displayed so if the font file size is large, this adversely affects page load times.

Another big issue about font loading is that, by default, many browsers will not show the fallback font while loading the web font (when using `@font-face`). Depending on the browser (and how old it is), some will show the fallback font right away while some will show none at all during loading. This will cause you to have blank areas of the page until the browser has finished loading the font. Some browsers will take a while to load before showing the font while others keep trying to load and essentially freeze the page enough that you'll have to refresh the page. For usability, this is bad.

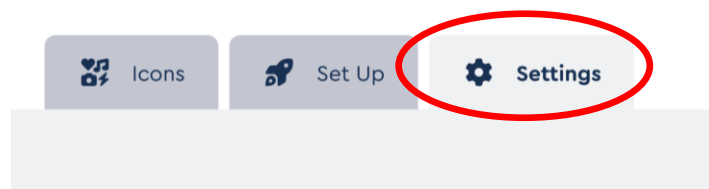
Due to the above issues with font loading, be sure to look at the font sizes when selecting fonts. If possible choose fonts with smaller sizes (e.g. 70KB) as opposed to large fonts (e.g. 500KB).

Icon Fonts

There are CSS fonts which are just icons (not letters). One such font is Font Awesome (<http://fontawesome.com/>).

Use the font in your site

1. As of version 5 and up, you will need to sign up to use the kit (the latest is version 6). Go to the Font Awesome site and go to the **Start** link in the menu.
2. Complete the account creation to access your kit. Under the Free account, you are allowed to create just one kit, but that's good enough for our purposes.
3. Once you've created your account, you may see a kit available. If you don't create a new kit, create one now.
4. To use the kit, copy the `<script>` line and insert into your `<head>`.
5. By default, your kit will be using the Webfont version, but this does not allow you to fully leverage new features. To take advantage of new features, we need to make sure we're using the SVG+JS version.
 - a. Click on the **Settings** link to view the settings for your kit.



- b. If you don't see the above link, click on the **Kits** link in the menu. You should see your kit which you've created.

- c. Click on the Settings tab to view the **Settings** page.
6. On the settings page, under *Technology* select "SVG".
7. Scroll down and click on the **Save Changes** button.
8. Go to the *Icons* page (<http://fontawesome.com/icons/>) to get a list of all the icon names.
9. To display an icon, you will need the CSS class. Take the icon name and prefix the name with "fa-". This will be the name of your CSS class. For example, there is a database icon with the name "database". The class name will be "fa-database".
10. To display the icon, use the following HTML (you will also need the "fa-solid" class all the time for **solid** icons):

```
<span class="fa-solid fa-database"></span>
```

Note

In the documentation, the `<i>` tag is used rather than ``, but `` is technically more semantically correct. Many icon fonts now use `<i>` for icons since `<i>` for italicizing is now deprecated.

You can also insert Font Awesome icons in your CSS, but you'll need to add the following attribute to your `<script>` line:

data-search-pseudo-elements

This is used by Font Awesome's script to tell it to search for CSS pseudo-elements and insert icons where specified.

Now, in your CSS you can use (assuming your target class is named myicon):

```
.myicon::before {  
  font-family:'Font Awesome 6 Free'; /* alt-> font: var(--fa-font-solid); */  
  font-weight:900; /* this may be required if the icon doesn't show */  
  content:'\f1c0'; /* the Unicode is found on the icon details page */  
  display:none; /* under the SVG method, this line is required */  
}
```

We can further modify the above. As of version 6, Font Awesome now recommends using the CSS custom property names (think of them like CSS variables).

For the free version, there are only the brands (CSS var `--fa-font-brands`) and solid (CSS var `--fa-font-solid`) styles available.

[View the documentation page about using pseudo-elements](#)

Accessibility Issues

To avoid accessibility issues, add text in addition to the icon rather than displaying only an icon.

If you must have no text (e.g. an icon link), add an aria-label. For example:

```
<span class="fa-brands fa-instagram" aria-label="Instagram"></span>
```


Alternatively, add a `` with text inside and hide it for regular users by moving it off-screen though using `aria-label` is less awkward.

So now you have:

```
<span class="fa-brands fa-instagram" ></span>  
<span class="hidden">Instagram</span>
```

where the "hidden" class has the following CSS:

```
.hidden {  
  position: absolute;  
  left: -10000px;  
}
```

Some design considerations

When choosing fonts for your design, make sure you take into account the readability of the font. For example, if you are using a fancy font, don't make the font size too small because this becomes hard to read. Also consider letters which may look similar to each other. This causes accessibility issues. For example, take a look at the word "ill" capitalized.

In Times New Roman: "Ill"

In Trebuchet MS: "Ill"

In Arial: "Ill"

You can see where Arial may cause a problem because a capital "I" and lowercase "l" look very similar. Common problem scenarios are (as above) "I" and lowercase "l", lowercase "rn" (rn) and lowercase "M" (m), and also the number "1" (1) with lowercase "l" (l).