



Font optical sizing

I came to optical sizing via variable fonts. The idea is that having an optical sizing axis in your variable font you can control how the font looks at different sizes. For small text sizes, usually have thicker strokes with larger serifs. Larger text often has more contrast between thicker and thinner strokes.

Note:

Optical sizing will only work with variable fonts that have an `opsz` axis. If the font doesn't have an optical sizing axis, there will be no effect. They are also limited to operating systems that support variable fonts.

We're using [Amstelvar](#) for the code and the attached demo. We first register the font with the weight and stretch using variable font ranges rather than single values

```
@font-face {  
  src: url("AmstelvarAlpha-VF.ttf");  
  font-family: "Amstelvar";  
  font-weight: 100 900;  
  font-stretch: 35% 100%;  
  unicode-range:  
    U+0020-007E, U+4E00, U+4E2A,  
    U+4E45, U+4EE5, U+524D, U+5728,  
    U+5927, U+5F81, U+5F88, U+6211,  
    U+661F, U+662F, U+6D77, U+7684,  
    U+7CFB, U+8FB0, U+8FDC, U+9014,  
    U+9065;  
}
```

The possible values for `font-optical-sizing` are:

`auto` : The default value allows the browser to provide the optimal optical sizing for the font at the given size

none : Disables optical font sizing

The following examples use the Amstelvar font defined earlier and the default value for font-optical-sizing to provide a better reading experience for user.

The only time we use an explicit value for font-optical-sizing is in the .no-optical-sizing class when we explicitly disable it.

```
p {
  font-size: 36px;
  font-family: Amstelvar;
}

.large {
  font-size: 48px;
  font-family: Amstelvar;
}

.light {
  font-weight: 200;
  font-family: Amstelvar;
}

.bold {
  font-weight: 700;
  font-family: Amstelvar;
}

.xlarge {
  font-size: 72px;
  font-family: Amstelvar;
}

.no-optical-sizing { font-optical-sizing: none; }
```

If you want to control the optical size independent from the font size (what the browser does by default), you can set values for the opsz axis directly. In a font that has an optical size range from 16 to 64, you can set it to the maximum value like so: font-variation-settings: 'opsz' 64.

But unless you're deliberately breaking typography and design rules, what the browser does should be your best option.

This is the full example using Amstelvar:

Links and resources

- [CSS Fonts Module Level 4](#)
 - [Optical sizing control: the font-optical-sizing property](#)
- [font-optical-sizing](#) — MDN
- [font-optical-sizing](#) — CSS Tricks
- [What is optical sizing and how can it help your brand?](#) — Monotype
- [Optical size, the hidden superpower of variable fonts](#) — Pixel Ambacht