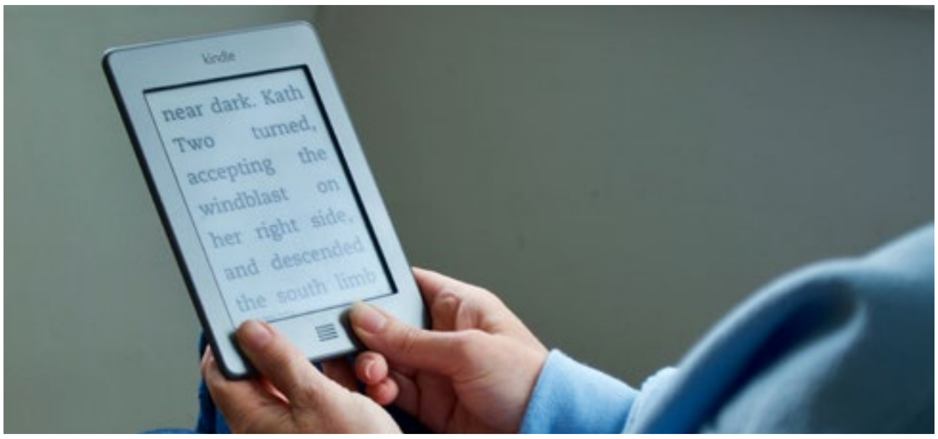
Designing paragraphs: text size

Text size is the second leg of the readability stool. Start with a size suitable for comfortable reading, which could vary with your choice of typeface and the distance of the screen from your reader's eyes.

Use the default font size for paragraphs

Your starting point for paragraph text should be whatever size has been set as the default in the browser. Device manufacturers will have determined an appropriate initial size, usually 16 px. Be aware that your reader may have adjusted the default size to better suit their needs. You should respect your reader's wishes – if they've gone to the effort of adjusting the font size (especially the browser default) it will be for a good reason.



Some people set their default text very large indeed.

Many designers, especially those new to the web, are surprised at how large the default text size is on screen compared with printed material. But this comparison only holds up if you compare the two media side by side. We tend to read screens from further away than the distance at which we hold books, newspapers or magazines, so the perceived screen and print text sizes are actually similar.

Your inner designer may err towards smaller text, but from your reader's perspective it is safer to make the text a little too big than too small. It is true that your reader can adjust the size of text within their browsers, but that's no excuse for setting type too small in the first place. Why make them go through the effort of correcting the type size? You're the typographer – it's your job to set the type correctly, and the default is the size at which browsers were intended to display text.

Take small print into account

You should consider whether the text you are designing is likely to include long passages set smaller than your regular body text. Examples of this include small print, annotations and comments.

You may not yet have decided whether your small print should actually be set in a smaller size, but now's the time to start thinking about the possibility. Your small print will need to be differentiated from your regular text, but if dropping down a text size means your small print becomes illegible, you'll need to bump up its size a bit and increase the body text to accommodate it.

Reversed-out text

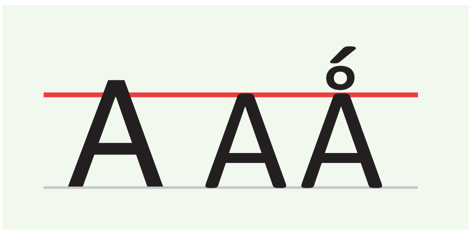
Be aware that light text set on a dark background can often do with being set a little bigger, particularly when viewed on lower-resolution screens. Experiment to see what works for you.

Adjust the font size if the typeface requires it

The browser default is always the best place to *start* for text size. However, typefaces can vary significantly – even among ones suitable for continuous reading – and your choice of typeface may require you to adjust your paragraph text size up or down from the default.

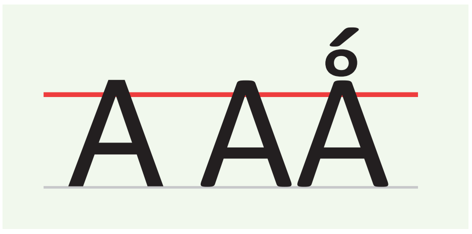
Some fonts are simply smaller owing to the way they were built. Often these are fonts which were designed from the start with heavily accented characters or the need to accommodate non-Latin scripts such as Arabic.

A good example of this is Calibri, which contains the character Ǻ, a rarely used double-accented Danish glyph. It is an extremely tall character and, to fit the diacritics into the font's vertical metrics, Calibri is built with smaller letters compared to similar typefaces.



Helvetica's (*left*) is drawn bigger than Calibri (*right*). Here the glyphs are set at the same font size, despite appearances to the contrary.

Calibri's capital letters are about 10% shorter than the version of Helvetica included in macOS, so to achieve parity with Helvetica at a typical default of 16 px, you should set Calibri at 18 px.

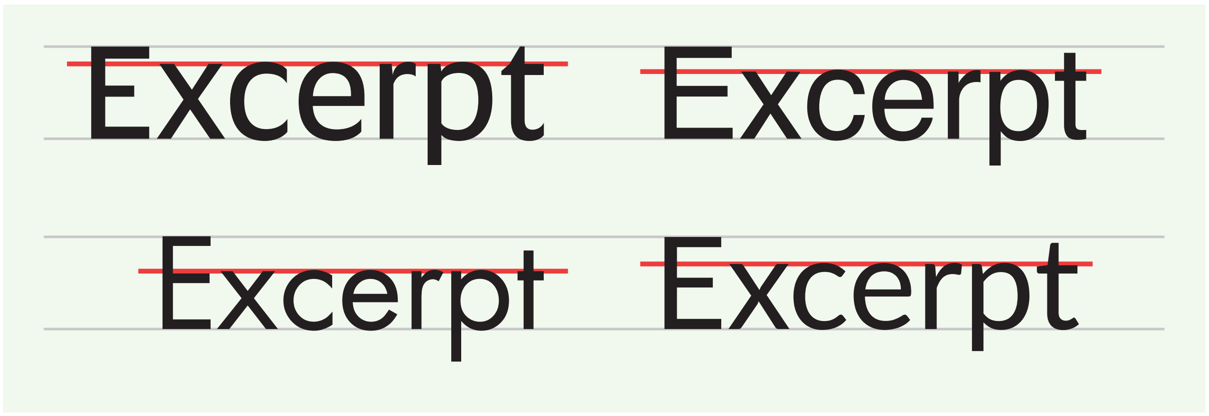


Helvetica set at 16 px compared with Calibri at 18 px (*magnified*).

Fonts smaller (or bigger) by design

A given typeface may appear smaller or bigger compared with another, despite the two fonts being constructed with similar vertical metrics. It is the design of the typefaces that makes them look different sizes, rather than the way they have been developed. The main difference in design is usually the x-height. X-height refers to the size of the lowercase letters relative to the size of capital letters. A shorter x-height leads to a typeface which looks smaller.

Helvetica has a more generous x-height than many other fonts, but as it's commonly used as a default typeface we can employ it as our benchmark. Compared with Helvetica, Altis has a tall x-height, so to match the apparent type size of Helvetica you would need to set Altis at a smaller size. Conversely, Lato has a slightly shorter x-height than Helvetica, so you might consider setting Lato slightly bigger than the default. Futura has an even smaller x-height, so you would need to set Futura significantly bigger to match the perceived type size of Helvetica.



Clockwise (from top-left): Altis, Helvetica, Lato and Futura showing a variety of x-heights.

When deciding precisely how much to adjust your typefaces there are two approaches. First, you could do so by eye and experiment. Set a paragraph in Helvetica alongside the same paragraph set in your choice of typeface. Using your judgement, tweak the type size of your font one pixel at a time until the lowercase letters look about the same size as Helvetica. It's not an exact science but any nudge in the right direction will make a difference to your reader.

Helvetica 16px

One more attribute the modern typographer must have: the capacity for taking great pains with seemingly unimportant detail. To them, one typographical point must be as important as one inch, and they must harden their heart against the accusation of being too fussy.

Futura 20px

One more attribute the modern typographer must have: the capacity for taking great pains with seemingly unimportant detail. To them, one typographical point must be as important as one inch, and they must harden their heart against the accusation of being too fussy.

Futura must be set at 20 px to match the perceived size of Helvetica at 16 px.

Calculating type size adjustments

An experienced typographer will be able to tweak type sizes by eye. It is also possible to use the internal font metrics to calculate a size adjustment to make one font appear the same size as another. The theory is to adjust the font size so that the x-heights are the same. The relative height of lowercase letters to their uppercase counterparts is commonly referred to as the *aspect value*; it is this value we will use in our calculations. Precisely defined, the aspect value is equal to the x-height of a font divided by the font size.

A typical aspect value is about 0.5, meaning the x-height is half that of the type size. The beauty of type is that there's no such thing as an average typeface, and it's the differences we are making adjustments for. If you have access to font editing software, you can determine the aspect value from the font metrics used in constructing the fonts. A quicker and easier way is to use an online aspect value calculator[1]. Using such tools you'll find that the fonts we've been looking at have the following aspect values:

Typeface	**Aspect Value**
Helvetica	0.521
Lato	0.507
Futura	0.417
Altis	0.542

Given this information, we can calculate a text size for our alternative fonts that will be make them *look* about the same size as Helvetica. Here's how. Assuming that Helvetica is set at 16 px, then the equivalent size for Lato is:

```
16 × 0.521 ÷ 0.507 = 16.5px
```

Using the same formula with Futura:

```
16 × 0.521 ÷ 0.417 = 20.1px
```

And for Altis:

```
16 × 0.521 ÷ 0.542 = 15.2px
```

1 Online aspect value calculators by Bruno Fassino (http://wbtyp.net/59) and Richard Rutter (http://wbtyp.net/58).

Altis 15.2px
One more attribute the modern typographer must have: the capacity for taking great pains with seemingly unimportant detail. To them, one typographical point must be as important as one inch, and they must harden their heart against the accusation of being too fussy.

Helvetica 16px
One more attribute the modern typographer must have: the capacity for taking great pains with seemingly unimportant detail. To them, one typographical point must be as important as one inch, and they must harden their heart against the accusation of being too fussy.

Lato 16.5px
One more attribute the modern typographer must have: the capacity for taking great pains with seemingly unimportant detail. To them, one typographical point must be as important as one inch, and they must harden their heart against the accusation of being too fussy.

Futura 20px
One more attribute the modern typographer must have: the capacity for taking great pains with seemingly unimportant detail. To them, one typographical point must be as important as one inch, and they must harden their heart against the accusation of being too fussy.

Four typefaces with actual size adjusted to match the perceived size of Helvetica.

The calculations we just went through resulted in font sizes at fractions of a pixel. In reality, it's better not to set your fonts in anything other than whole pixels. Nudging one whole pixel at a time will prevent unevenness in rendering on lower-resolution screens, and also make layout and type-setting calculations easier further down the line.

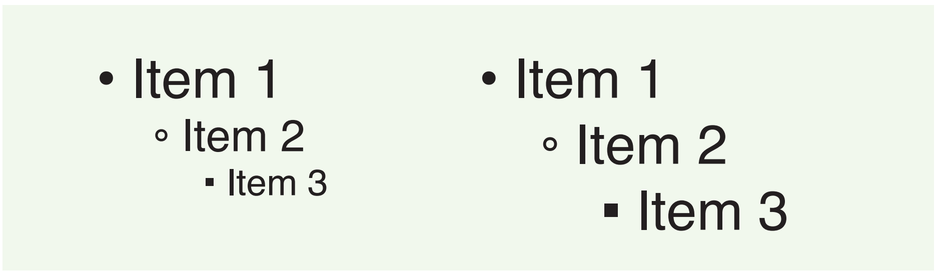
Use rems to size your text

Up to this point we've talked about font sizes in terms of pixels. This was just for the convenience of explanation. When it comes to CSS you should use rem units to size your text. In the chapter 'The amazing em (and friends)' we learned that rems are relative units which enable you to specify sizes in relation to the root `<html>` element's font size. If the default text size of the browser is 16 px then text sized at 2 rem would always be displayed at 32 px, regardless of its context.

In contrast, text sized in ems is always relative to the parent element and is thereby prone to inheritance complications. A common example of this is the shrinking list. Consider the following nested list:

```
<ul>
   <li>Item 1
      <ul>
         <li>Item 2
            <ul>
               <li>Item 3</li>
            </ul>
         </li>
      </ul>
   </li>
</ul>
```

If you size the text using something like `font-size:0.9em`, the list items will get progressively smaller. If you size the text using rems instead of ems, `font-size:0.9rem`, the list items will all stay the same size.



Left: List items sized in ems. ***Right***: List items sized in rems.

So why not just use pixels? In CSS, pixels are absolute units anchored to a physical measurement. A pixel – short for *picture element* – used to refer to a physical pixel on a device, but with the advent of high-resolution screens the definition has changed. As explained in 'The amazing em (and friends)', CSS3 defines a pixel to be 1/96th of 1 inch (0.26 mm). The idea is that a CSS pixel is equivalent to the size of a pixel if the screen had a resolution of 96 dpi.

In theory, this means that because pixels refer to a physical length, text sized in pixels should stay the same regardless of any settings changed by the reader. In reality, users of most devices can change the text size even when it is sized in pixels; this wasn't always the case, however.

Microsoft's Internet Explorer 6 (and earlier) wouldn't resize text set in pixels, causing a major accessibility problem for any reader needing to adjust the text size to better suit their eyesight. The same happened more recently with an update to Chrome 52. By the letter of the CSS specification, these implementations could not really be said to be wrong – and that logic still stands. For the sake of future-proofing, pixels are best avoided for sizing text. Once bitten, twice shy.

Having said that, when web designers think of text size we do tend to think in pixels. To think in rems we need to do a simple calculation: for the size in rems, divide the required size in pixels by 16 (the typical size of the root element). For example, to change your body text to 18 px by setting it in rems, you would calculate the following:

```
18 ÷ 16 = 1.125
```

with the resulting CSS being:

```
body { font-size: 1.125rem; }
```

Rems provide the same non-inheriting convenience as pixels, but because they are a relative unit not tied to any physical dimensions, text sized in rems can be resized at will.

Next we will attach the third leg of the readability stool, and consider how line spacing adds stability to the situation.