

---

## How to Size Text in CSS

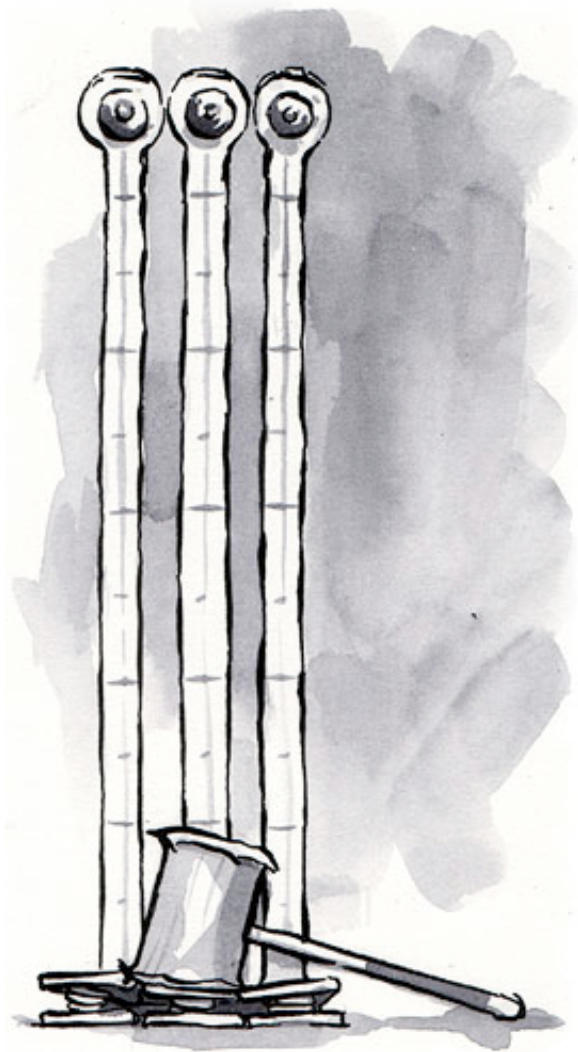
by **RICHARD RUTTER**

---

Published in: Browsers, CSS, HTML and XHTML, Layout, Typography

---

There's been a welcome resurgence of interest (<http://ilovetypography.com/2007/09/19/15-excellent-examples-of-web-typography/>) in web typography (<http://webtypography.net>) over the past year or so, with many articles ([http://www.digital-web.com/articles/css\\_typography/](http://www.digital-web.com/articles/css_typography/)) and conference (<http://webtypography.net/sxsw2007/>) talks



([http://www.markboulton.co.uk/journal/comments/type\\_in\\_berlin/](http://www.markboulton.co.uk/journal/comments/type_in_berlin/)) offering techniques (<http://www.sitepoint.com/blogs/2007/04/30/typography-baseline-rhythm-deciphered/>) and theory ([http://markboulton.co.uk/articles/detail/five simple steps to better typography/](http://markboulton.co.uk/articles/detail/five_simple_steps_to_better_typography/)). Frequently asserted is the notion that good typography requires accurate control of font size and line-height. But this is the web: it's a special medium where the reader can have as much control as the designer—the implication

being that text on the web, while bending to the designer's will, must also be reliably resizable across browsers and platforms.

In this article, we will reconcile the designer's requirement for accuracy with the user's need to resize text on demand, arriving at a best practice that satisfies designers *and* users and works across browsers and platforms.

We'll reach our destination by the traditional method of trial and error. With more than a nod to Owen Briggs's [pioneering work](http://www.thenoodleincident.com/tutorials/box_lesson/font/index.html) (http://www.thenoodleincident.com/tutorials/box\_lesson/font/index.html) of 2002, I have created a base case with six iterations and 161 screenshots. Follow along, won't you?

## The test suite

The [content used for testing purposes](http://alistapart.com/d/howtosizetextincss/test-0.html) (http://alistapart.com/d/howtosizetextincss/test-0.html) was a two-column layout with body copy on the left and a sidebar on the right. Text was set in Arial to aid consistency across operating systems and platforms.

The browsers used for testing were Safari 2, Firefox 2 and Opera 9.5a running on Mac OS X Tiger, along with Internet Explorer 6 (IE6) and Internet Explorer 7 (IE7) running on Windows XP with ClearType turned on. Clearly this is not an exhaustive list of browsers, operating systems, or rendering engines, but it covers the majority of users out there today.

Each operating system and browser was run using its default settings. Every iteration was tested to see how each browser rendered text at smaller, medium, larger, and largest sizes, along with 90%, 100%, 110%, and 120% page zoom levels, where applicable.

## Base case

First it was necessary to verify that browsers provided a consistent baseline from which to start. The [base case](http://alistapart.com/d/howtosizetextincss/ss-test-0.html) (http://alistapart.com/d/howtosizetextincss/ss-test-0.html) shows that in each browser, the default text size is consistently 16px when no styles are applied (other than the browser defaults), and the text scales fairly consistently across the board.

## Text size in pixels – iteration 1

The default text size of the base case is a good starting point, but for most people (designers, clients, and their customers) 16px is too large for body text. In our example, the body text was reduced to 14px, with the sidebar set at 12px. This first iteration does just that, setting the fonts in pixels:

```
.bodytext p {
    font-size:14px;
}
```

```
.sidenote {
    font-size:12px;
}
```

The result (<http://alistapart.com/d/howtosizetextincss/ss-test-1.html>) is that Safari and Firefox still resize the text, whereas IE6 and IE7 do not. The text can be resized in Opera and IE7 by using the page zoom tool, which magnifies the page layout, text and images within.

## Text size in ems – iteration 2

Although browser market share differs from site to site, and browser share statistics are drawn in sand, it's safe to say that IE6 is still used by [many people \(http://marketshare.hitslink.com/report.aspx?qprid=6\)](http://marketshare.hitslink.com/report.aspx?qprid=6). So setting text in pixels would leave many people no means of resizing it. There's also an argument that says IE7 users should be able to resize text without being forced to use the zoom control.

The next unit to try for text sizing is ems. The em is a true typographic unit, [recommended by the W3C \(http://www.w3.org/WAI/GL/css2em.htm\)](http://www.w3.org/WAI/GL/css2em.htm), and affords a precision [keywords \(http://www.alistapart.com/articles/sizematters/\)](http://www.alistapart.com/articles/sizematters/) lack. Working from a default of 16px, the following styles should give the desired text sizes:

```
.bodytext p {
    font-size:0.875em; /* 16x.875=14 */
}

.sidenote {
    font-size:0.75em; /* 16x0.75=12 */
}
```

The results (<http://alistapart.com/d/howtosizetextincss/ss-test-2.html>) show that, across all browsers, text at the medium browser setting is rendered identically to text set in pixels. It also demonstrates that text sized in ems can be resized across all browsers. However IE6 and IE7 unacceptably exaggerate the smallness and largeness of the resized text.

## Body sized as percentage – iteration 3

A fix to the exaggerated text resizing of IE6 and IE7 is to size the body using a percentage. So retaining the ems on our content, the following styles were tested:

```
body {
    font-size:100%;
}
```

```
.bodytext p {
    font-size:0.875em;
}

.sidenote {
    font-size:0.75em;
}
```

The results (<http://alistapart.com/d/howtosizetextincss/ss-test-3.html>) show that the difference between larger and smaller browser settings in IE6 and IE7 is now less pronounced, meaning we now have all browsers rendering text at an identical size on their medium setting, and resizing text consistently.

## Setting line height in pixels – iteration 4

Recent web typography articles such as "[Setting Type on the Web to a Baseline Grid](http://www.alistapart.com/articles/settingtypeontheweb)" (*A List Apart*, April 2007) stress that good typography requires a vertical grid, that is to say a solid vertical rhythm achieved with a consistent, measured line-height. The key implication is that line-height should be the same regardless of the size of the text (so that line-height, or the vertical grid, remains consistent, regardless of font size).

For our example, a suitable line-height is 18px, so that is added to the body as follows:

```
body {
    font-size:100%;
    line-height:18px;
}

.bodytext p {
    font-size:0.875em;
}

.sidenote {
    font-size:0.75em;
}
```

The results (<http://alistapart.com/d/howtosizetextincss/ss-test-4.html>) show that the 18px line-height is inherited by all text on the page—note how the sidebar text has the same regular rhythm as the body copy. Specifying a unit (in this case, px) when setting the line-height enables the **value** to be inherited throughout the page. If a [unitless line-height](http://meyerweb.com/eric/thoughts/2006/02/08/unitless-line-heights/) had been specified, the **multiplier** would have been inherited, resulting in line-heights being rendered proportionally to the text size, thus breaking the vertical rhythm.

Unfortunately the results show that the 18px line-height is not scaled by IE6 and IE7 when text is resized, meaning the largest setting appears to squash the text.

## Setting line height in ems – iteration 5

When pixels failed before, we turned to ems. Repeating the logic gives us the following styles:

```
body {
    font-size:100%;
    line-height:1.125em; /* 16x1.125=18 */
}

.bodytext p {
    font-size:0.875em;
}

.sidenote {
    font-size:0.75em;
}
```

The results (<http://alistapart.com/d/howtosizetextincss/ss-test-5.html>) show accurate, consistently resized text and line-height across all browsers. Perfect. Or nearly so.

## The Safari monospace problem – iteration 6

The observant among you may have noticed a wee glitch in the Safari [screenshots](http://alistapart.com/d/howtosizetextincss/ss-test-5.html) (<http://alistapart.com/d/howtosizetextincss/ss-test-5.html>): the monospaced font included in the body text is rendered inconsistently. For text set in pixels, Safari renders the monospaced font at the same size as the proportional-width text surrounding it. When text is set in ems, however, Safari renders monospace text much smaller than the surrounding text. The inconsistency appears to stem from Safari's default text sizes, which are 16px for "standard fonts" and 13px for "fixed-width fonts." Safari 3a on OS X does not appear to suffer from this problem.

You could decide that undersized monospace text in Safari is something you and your readers can live with, and as Safari 3 is included in OS X Leopard and the latest update to Tiger, it will not be long until the problem pretty much disappears. For the nervous control freak who can't wait, an alternative fix is to send text sized in pixels to Safari.

The following code appends a [downlevel-revealed conditional comment](http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/overview/ccomment_ovw.asp) ([http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/overview/ccomment\\_ovw.asp](http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/overview/ccomment_ovw.asp)) to our styles, so that pixels are sent to all browsers except IE6 and IE7 (note the [if !IE] syntax, instructing IE/Win to ignore the markup that follows).

```
<style type="text/css">
body {
    font-size:100%;
    line-height:1.125em;
```

```

}

.bodytext p {
    font-size:0.875em;
}

.sidenote {
    font-size:0.75em;
}
</style>

<!--[if !IE]>-->

<style type="text/css">
body {
    font-size:16px;
}
</style>

<!--<[endif]>-->

```

The results (<http://alistapart.com/d/howtosizetextincss/ss-test-6.html>) show consistently resized text and line-height across all browsers, including the monospaced text in Safari 2.

Conditional comments are controversial, with many [detractors](http://meiert.com/en/blog/20070201/why-conditional-comments-are-bad-repeat-bad/) and [proponents](http://www.456bereastreet.com/archive/200510/stop_using_css_hacks_now/) ([http://www.456bereastreet.com/archive/200510/stop\\_using\\_css\\_hacks\\_now/](http://www.456bereastreet.com/archive/200510/stop_using_css_hacks_now/)), but I believe the approach is appropriate in this case, as we are using a browser feature (conditional comments) to work around a browser behaviour (non-resizing of pixels). It should also be noted that, for the sake of clarity, the code listed above presents CSS rules within `style` elements; best practice would dictate the use of [linked style sheets](http://htmlhelp.com/reference/css/style-html.html#external) (<http://htmlhelp.com/reference/css/style-html.html#external>) instead.

## Conclusion

Our task was to find a way to size text that allows designers to retain accurate control of typography, without sacrificing the user's ability to adjust his or her reading environment. We tested various units across common browsers. Sizing text and line-height in ems, with a percentage specified on the body (and an optional caveat for Safari 2), was shown to provide accurate, resizable text across all browsers in common use today. This is a technique you can put in your kit bag and use as a best practice for sizing text in CSS that satisfies both designers and readers.

### ADDENDUM

Ems can be tricky to work with, especially when nesting elements deeply, as it can be hard to keep track of the maths. However, commenting your style sheets well and styling elements from the body

inwards can keep things easier to follow. This more [complex example](http://alistapart.com/d/howtosizetextincss/complexexample.html) (<http://alistapart.com/d/howtosizetextincss/complexexample.html>) and its [accompanying style sheet](http://alistapart.com/d/howtosizetextincss/complexexample.css) (<http://alistapart.com/d/howtosizetextincss/complexexample.css>) demonstrate how to size nested elements using the body as the starting point. ↵

*Illustration by [Kevin Cornell](http://alistapart.com/about/kevincornell) (<http://alistapart.com/about/kevincornell>)*

---

## Learn More

Related Topics: [Browsers](#), [CSS](#), [HTML and XHTML](#), [Layout](#), [Typography](#)

---

## About the Author



Richard Rutter is Production Director at [Clearleft](http://www.clearleft.com) (<http://www.clearleft.com>), a user experience consultancy based in Brighton, UK. He curates [The Elements of Typographic Style Applied to the Web](http://www.webtypography.net) (<http://www.webtypography.net>), a practical guide to web typography. Richard blogs on a site called [Clagnut](http://www.clagnut.com) (<http://www.clagnut.com>).

---

ISSN: 1534-0295 ♦ Copyright © (<http://alistapart.com/copyright/>) 1998-2009 A List Apart Magazine and the authors.

---