Commonly Available Fonts

Because a font will display only if it is available on a user's hard drive, it makes sense to design with the most commonly available fonts, particularly for sites with wide-reaching audiences. So, which fonts can you rely on?

In general web design practice, designers tend to specify fonts from Microsoft's Core Web Fonts collection. This is a set of TrueType fonts (for both Windows and Mac) that have been specially designed to be easy to read on screens at small sizes. Microsoft released the fonts in 1996 and initially made them available for download. Today, they are installed automatically with Internet Explorer and other Microsoft software, so you can count on the majority of users having them available. Table 18-1 lists the fonts in the Core Web Fonts collection.

Table 18-1. Core Web Fonts from Microsoft

Serif	Georgia Times New Roman
Sans Serif	Arial Arial Black Trebuchet MS Verdana
Monospace	Courier New Andale Mono
Miscellaneous	Comic Sans MS Impact Webdings

Microsoft publishes an interesting online resource that lists which fonts are installed with its various popular applications and each version of the Windows operating system. There are also lists of the fonts that come installed with Macintosh OS X, Unix systems, and Adobe Type Manager. You'll find the font lists at *www.microsoft.com/typography/fonts/default.aspx*.

If you know your audience might have more specialized fonts installed, by all means, make a statement and go off the beaten path. You can always provide a more commonly available font as a backup in the list of font names.

Font Size

CSS provides the `font-size` property for specifying the size of text. There are many value options for specifying font size, each with its own pros and cons. This section discusses the various keyword and unit options and their impact on usability.

font-size

Values: xx-small | x-small | small | medium | large | x-large | xx-large | smaller | larger | <length> | <percentage> | inherit

Initial value: medium

Applies to: All elements

Inherited: Yes

These examples demonstrate the font-size property used with several different value types.

```
p.copyright {font-size: x-small;}
strong {font-size: larger;}
h2 {font-size: 1.2em;}
p#intro {font-size: 120%;}
```

Absolute Versus Relative Sizes

Before diving into the details of specifying type size, it is worth pausing to clarify the difference between absolute and relative sizes. *Absolute sizes* have predefined meanings or an understood real-world equivalent. In CSS, absolute values may be expressed as keywords, such as small or x-large (discussed next) or by using absolute length values, such as cm (centimeter), in (inch), or pt (point, 1/72 of an inch).

Relative sizes, on the other hand, are based on the size of something else, like the parent element or the em measurement of the text (see the sidebar "A Word About Ems"). Relative values, such as em and percentages, are generally preferred for web text for reasons that are covered in the upcoming sections.

Absolute Size Keywords

Absolute sizes are descriptive terms that reference a table of sizes kept by the browser. There are seven absolute size keywords in CSS: xx-small, x-small, small, medium, large, x-large, and xx-large. The keywords do not correspond to a particular measurement, but rather are scaled consistently in relation to one another. The default size is medium in current standards-conformant browsers.

Figure 18-3 shows how the following examples of text sized with absolute keywords look in Firefox 1.0.

```
<span style="font-size: xx-small">xx-small</span>
<span style="font-size: x-small">x-small</span>
<span style="font-size: small">small</span>
<span style="font-size: medium">medium</span>
<span style="font-size: large">large</span>
<span style="font-size: x-large">x-large</span>
<span style="font-size: xx-large">xx-large </span>
```

A Word About Ems

In traditional typography, the em has been a measurement of width approximately equal to the width of the capital letter M for the given typeface. Using that measurement, you arrive at the width of an em-space or an em-dash.

As typography has adapted to digital media, the em has become a measure of width and height, or often height alone. For purposes of CSS, the em is calculated as the distance between the baselines when the font is set without any additional *interlinear space*, also called *leading* (extra space added between lines of text for legibility).

This distance forms the basis of an implied *em-square* measurement based on the design of the typeface (also called the *em-box*). It is possible that ascenders and descenders of a particular typeface may exceed the boundaries of the em-square, or that no characters of another face fill it completely. The font's em-box measurement can be used as a relative unit of measurement.

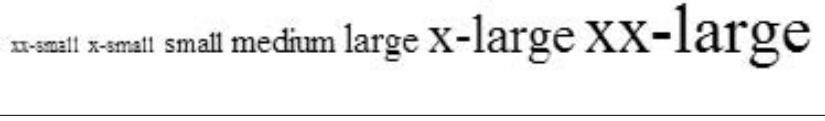

Figure 18-3. Text sized with absolute keywords

This figure and other figures in this book use inline styles as a means to save space on the page. In the real world, inline styles should be avoided in favor of external or embedded style sheets.

The CSS 2.1 specification leaves the scaling factor (how much each consecutive keyword is enlarged or reduced) up to the user agent. Chances are, it will be somewhere around 1.2 (the most recent recommended scaling factor) or as large as 1.5 (the CSS 1 recommended scaling factor), varying between different browsers.

At a scaling factor of 1.2, if `medium` (default) text is 16 pixels, then `large` text would be 19 pixels (after some rounding). The upshot of it all is absolute size keywords vary in size from browser to browser, so they are not the best option if you are looking for consistency.

Internet Explorer 5 and 5.5 for Windows use `small` as the default, which can seriously throw off an attempt to use absolute size keywords throughout a document.

Relative Size Keywords

There are two relative keywords: `larger` and `smaller`. They are used to shift the size of the text relative to the parent object according to the seven-step absolute-size scale (using the same scaling factor). For example, if the text of a paragraph is set to `large`, applying the keyword `smaller` to a child `em` element would cause the emphasized text to display at `medium` size. Figure 18-4's examples use relative size keywords.

```
There are two relative keywords: <span style="font-size: larger">larger</
span> and <span style="font-size: smaller">smaller</span>. They are used...
```

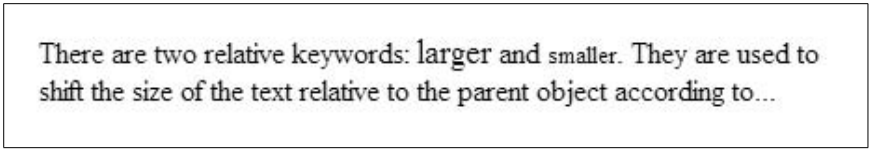

Figure 18-4. Relative size keywords

Percentage Measurements

One fairly reliable way to specify text size is in percent values. Percent values are calculated relative to the inherited size of the parent text. That "inherited" part is important, because it means that if you nest similar elements with percentage values, the affect is cumulative. It doesn't take many levels of nesting before the text is unreadable.

In Figure 18-5, the `ul` element is set to a relative size of 80%. If the body of the document is 16 pixels, that means the `ul` text will be 13 pixels (after rounding). The nested `ul` within that list takes the same size setting (80%), but this time it is applied to its inherited size (13 pixels), resulting in 10 pixel text, and so on for each nested level.

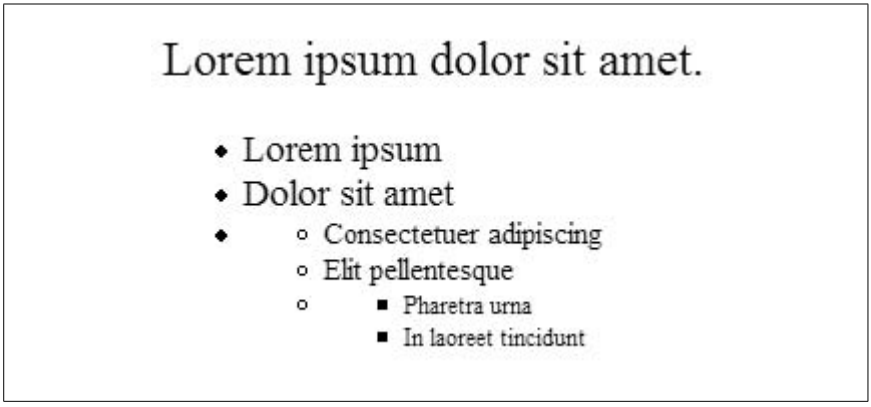

Figure 18-5. Nested elements with percentage size values

Style sheet

```
body {font-size: 24px;}
ul {font-size: 80%;}
```