

Subject: [Builder.com] The trouble with using ems

From: "Design and Usability Tactics at Builder.com"

<CNET_Networks_Member_Services@newsletter.online.com>

Date: Wed, 12 May 2004 03:17:47 -0700 (PDT)

To: kegill@u.washington.edu



[Visit Builder.com](#) | May 12, 2004

Design and Usability Tactics



Michael Meadhra

The trouble with using ems and percents for font sizing

There are several ways to specify font sizes on a Web page. You can use keywords such as small or large; you can use fixed measurements such as pixels or points; or you can use the relative measurements of em or percent. However, there are certain advantages and disadvantages for each font size specification method. Keywords are simple and easy to use, but they lack flexibility and precision. Fixed measurements can be precise, but they can create accessibility problems for visitors who want to adjust text sizes to suit their own needs and preferences. Relative measurements address the accessibility problem, but they have their own problems.

The way it's supposed to work

Relative measurements work by specifying the font size as a relative enlargement or reduction of some standard reference size. Percent sizes express the font size value as a percent of the height of the character box (the top of the capital letters to the bottom of the descenders) for the reference font.

Ems are based on the typesetting convention that defines 1em as the height of the capital letters in the reference font. For consistency with other measurements, CSS redefines an em as the height of the character box. Sizes in ems are decimal fractions or multiples of the size of the reference font. Therefore, 2em and 200 percent are the same and so are .75em and 75 percent.

In theory, using relative measurements to specify font size should have distinct advantages for both the Web builder and visitor. By specifying font size as a relative change from a standard reference size, you focus on the size relationships rather than a specific numerical size for each text style.

If you change the size of the reference font, all the fonts that are sized with a relative measurement change proportionately to maintain the same size relationship with the reference font. This means that the Web builder can tweak all the font sizes on a page by making a single change to the font size in the body tag. Similarly, it allows visitors to adjust their browser's font size preference settings to resize the text in their browser windows without losing any meaning that relative font sizes might convey.

Trouble in paradise

Unfortunately, what sounds great in theory doesn't always work so well in practice.

advertisement

TechRepublic's
*Developer's
Guide to XML*

Click Here

- ▶ Introduction to SOAP
- ▶ XML Security
- ▶ Generating XML via JAVA
- ▶ Converting XML Documents
- ▶ Remedial XML for Programmers
- ▶ X-factor: Supporting XML Technologies
- ▶ VoiceXML
- ▶ Sybase XML Basics

 **TechRepublic**
Real World. Real Time. Real IT.

One major problem with using relative measurements for sizing fonts is the lack of a consistent standard for the base font size on which the relative measurements are based. The traditional default font size for Windows-based browsers is 16 pixels and assumes a screen resolution of 96 pixels per inch. On Macs, the traditional default is 12 pixels at a screen resolution of 72 pixels per inch.

Although the Windows default font size is the official standard recommendation, many diehard Mac users (including a lot of Web builders) continue to use the traditional Mac default settings. These settings tend to skew Mac users' perceptions of "normal" text sizes. This can be a problem no matter which method a Web builder uses for specifying font sizes, but it's particularly troublesome with relative sizes because any difference in the base font size is exaggerated as you get farther from the normal size.

Another major problem for relative font sizes is that all the size calculations aren't necessarily based on the browser's default text size or even the font size as specified in the body tag. Instead, the relative size calculations are based on the font size of the parent element. That complicates things considerably because elements can, and often are, nested several levels deep. For example, you might have a paragraph of text, within a table cell, within a table, within a named div, within another div. If there are font size rules using relative measurements at each level, the relative measurements can compound each other, leading to exaggerated font sizes that are dramatically larger or smaller than what you intend.

The compounding effect that results if you apply relative sizes to nested elements can make using relative font sizes confusing and difficult. To make matters worse, there are disturbing inconsistencies in the way the various browsers identify the parent element for a given element and, therefore, what font size to use as the basis for any relative size calculations. As usual, Netscape 4.x is the worst offender, but there are still enough inconsistencies among the current browsers to cause frequent problems with relative text sizing.

When good intentions fail

Relative font sizing is a great idea that fails to live up to its promises of user accessibility or design flexibility for the Web builder. In order to use relative font sizing successfully, you need to plan your page design and CSS styles very carefully to avoid the potential problems of nesting elements. You can do it but it's not easy, and it imposes significant restrictions on your design options. Otherwise, you need to rely on keywords or absolute measurements for font sizing.

Michael Meadhra has been working in the field since the earliest days of the Web. His book credits span some three dozen titles.

We want your feedback

Do you agree with Michael Meadhra that using relative measurements for font sizing sounds good in theory but not in reality? Did this tip teach you something new? If so, let us know in our [discussion forum](#), or [send us an e-mail](#).

Correction

Last week's Design and Usability Tactics e-newsletter contained a code syntax error. One of the code samples contained an unnecessary character (}) after </style>. Here is the correct code for the example:

```
<link href="youarehere.css" rel="stylesheet" type="text/css" />
<style type="text/css">
<!--
div#menu li#page2 a {
  color: #000099;
  background-color: #66CCFF;
}
-->
</style>
```

We apologize for any inconvenience this may have caused.

Read More About Development

[Learn how to position windows](#)

To enhance the user experience on your Web site, use consistent window positioning. This article from our Web Development Library shows you how.

[Discussion: Submit your favorite Dreamweaver templates](#)

Last year we posted a story about a Dreamweaver MX template designed for intranet sites. We know a lot of you have downloaded it and tried it out, so now it's your turn--share your own Dreamweaver templates with the Builder.com community. Tell us what you use the template for, and any special hints or tricks that folks need to know when they use your template.

Audiocast: How often should I audit my network?



A network audit can present many opportunities for improvement in the security control environment. But before you jump in to perform a penetration ("pen") test, or any other type of audit, there are high level concerns that must be addressed. For instance, the number of audits conducted can play a large role in determining the effectiveness of your security infrastructure. To help you decide the best audit frequency for you company, [this audio Webcast](#) will focus on how often an enterprise should conduct a security audit to achieve optimal network security.

New at Builder.com

[Download: SecurePlay: Anti-Cheating Network Gaming Software](#)

IT GlobalSecure presents its SecurePlay anti-cheating library for Java and Flash network games. SecurePlay simplifies development with an API for standard game transactions that wrap a suite of fair-play cryptographic protocols. Download SecurePlay today! (Free registration to ITPapers.com is required.)

[Script tracing in ASP.NET](#)

Debugging an ASP.NET script doesn't have to entail embedding lots of print statements. Find out how to use the handy trace functions to easily debug your code.

[Download: Ubiquity RP 2.0](#)

Develop interactive, descriptive, browser-based prototypes faster and easier than creating static mockups with your current tools like Dreamweaver and FrontPage. Download Ubiquity RP 2.0 today! (Free registration to ITPapers.com is required.)

Get More From CNET Networks

Launch RSS to receive daily content updates

RSS, commonly known as Really Simple Syndication, is an easy way to streamline your day. Builder.com and TechRepublic now offer daily RSS feeds that deliver a concise list of new articles and downloads posted on our sites. The RSS list includes headlines, summaries, and links to the complete story. [Check out our RSS page](#) for details on how to set up RSS on your desktop and for information on feed readers, which you can use to manage feeds from Builder.com, TechRepublic, and other sites.



SIGN UP FOR FREE E-NEWSLETTERS

Builder.com's Web Development Zone

From XML to Active Server Pages, design aesthetics to database connectivity, this weekly e-newsletter provides the latest dev information to keep your skills sharp. Click the "Sign Me Up" button to automatically subscribe to this e-newsletter.

[SIGN ME UP](#)

[Unsubscribe from this mail](#) | [Update/Unsubscribe all Builder.com e-mail](#) | [Questions & comments](#) | [Visit Builder.com](#)

[Copyright](#) © 2004 CNET Networks, Inc. All rights reserved. Builder is a registered service mark of CNET Networks, Inc.
CNET Networks, Inc.
235 Second Street
San Francisco, CA 94105
U.S.A.