

By Mellonfire

Takeaway

This tutorial will show you how to add such a text size switcher to your Web pages using PHP and CSS, thereby immediately making your Web site more accessible and scoring you useful brownie points from everyone over the age of 50.

Switcher

When designing a Web site, it's important to remember that all of its visitors will not necessarily be young, in fine fettle, and fully conversant with the intricacies of using a Web browser. The smart designer knows this and always builds special [accessibility features](#) into a Web site so that even older or handicapped users can use the site comfortably, without excessive strain or stress.

One of the most useful accessibility features any Web site can possess is a text size switcher. Simply put, this is a tool that allows users to alter the size of the text on a Web page, usually to make it larger and therefore easier to read. Some browsers come with this feature built-in, but novice Web users are not likely to know about it; for this reason, site designers often place easily-accessible control buttons for this purpose on each Web page.

This tutorial will show you how to add such a text size switcher to your Web pages using [PHP](#) and [CSS](#), thereby immediately making your Web site more accessible and scoring you useful brownie points from everyone over the age of 50. Keep reading, and find out how!

Note: *This tutorial assumes a working Apache/PHP installation.*

How it works

Before writing any code, it's instructive to spend a few minutes understanding how the size switcher is supposed to work. Each page of the Web site will sport a set of control buttons, allowing the user to select one of three sizes for the text on the page: small, medium and large. Each of these sizes corresponds to a CSS stylesheet, which holds the rules needed to render the page in the selected size.

When the user makes a selection, PHP internally stores the selected size in a session variable and then reloads the page. The reloaded page reads the selected size from the session variable and dynamically loads the corresponding stylesheet to re-render the page in a smaller or larger size.

Procedure

Step 1: Create a Web page

Begin by creating a HTML document, complete with placeholder content. Here's an example (Listing A):

Listing A:

```
<html>
<head>
</head>
<body>

<!-- font size buttons -->
Text size: <a href="resize.php?s=small">small</a> | <a
href="resize.php?s=medium">medium</a> | <a href="resize.php?s=large">large</a>

<p />

<!-- dummy page content -->
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute
irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.

</body>
</html>
```

Pay special attention to the text hyperlinks at the top of this page. Each hyperlink points to the script *resize.php*, and passes it the selected text size through the URL GET method.

Save this document to your Web server directory with a *.php* extension—for example, *index.php*.

Step 2: Create stylesheets

Next, create stylesheets for each of the available text sizes for the page: *small.css*, *medium.css* and *large.css*. Here's what *small.css* looks like:

```
body {
    font: 10px
}
```

In a similar manner, create *medium.css* and *large.css*, with font sizes of *17px* and *25px* respectively. Save these stylesheets to the same directory as the Web page created in the previous step.

Step 3: Create the text resizing mechanism

As previously described, the Web page "knows" which stylesheet to load by looking in a pre-defined session variable. This session variable is controlled by the script *resize.php*, (**Listing B**) which is activated whenever a user clicks one of the text-sizing buttons at the top of each page. Here's what *resize.php* looks like:

Listing B

```
<?php
// start session
// import selected size into session
session_start();
$_SESSION['textsize'] = $_GET['s'];
header("Location: " . $_SERVER['HTTP_REFERER']);
?>
```

This is fairly simple. When a user selects a new text size, *resize.php* receives that size value through the *GET* method and stores it in the session variable `$_SESSION['textsize']`. It then redirects the browser back to the page from whence it came.

Of course, there's still one missing component: the intelligence that lets the page detect which text size is currently selected and load the appropriate stylesheet. To add this, pop open your Web page *index.php*, and add the following lines at the beginning of the page (**Listing C**):

Listing C

```
<?php
// start session
// import variables
session_start();

// set default text size for this page
if (!isset($_SESSION['textsize'])) {
    $_SESSION['textsize'] = 'medium';
}
?>
```

You should also add a stylesheet link between the `<head>...</head>` elements, as follows:

```
<link rel="stylesheet" href="<?php echo $_SESSION['textsize']; ?>.css"
type="text/css">
```

Here's (**Listing D**) what the completed *index.php* will look like:

Listing D

```
<?php
// start session
// import variables
session_start();

// set default text size for this page
if (!isset($_SESSION['textsize'])) {
    $_SESSION['textsize'] = 'medium';
}
?>

<html>
<head>
<link rel="stylesheet" href="<?php echo $_SESSION['textsize']; ?>.css"
type="text/css">
</head>
<body>

<!-- font size buttons -->
Text size: <a href="resize.php?s=small">small</a> | <a
href="resize.php?s=medium">medium</a> | <a href="resize.php?s=large">large</a>

<p />

<!-- dummy page content -->
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute
irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.

</body>
</html>
```

It should be easy to understand how this works. When the Web page is loaded, it restores the current session, checks the `$_SESSION['textsize']` variable to see which text size is currently selected, and then dynamically loads the corresponding stylesheet through the `<link... />` element. This causes the page to automatically re-render in the correct size.

Using PHP and CSS in this combination is slightly different from the traditional approach, which uses JavaScript to dynamically alter CSS styles. The advantage of using PHP instead of JavaScript is that you're not dependent on the client supporting JavaScript, nor do you need to worry about creating browser-specific workarounds. Perhaps you will find a use for this tip the next time you sit down to design a Web site. Happy coding!

Additional resources

- TechRepublic's [Downloads RSS Feed](#) **XML**
- Sign up for our [Downloads Weekly Update](#) newsletter
- Sign up for TechRepublic's [Web Development Zone](#) newsletter
- Check out all of TechRepublic's [free newsletters](#)
- [Create advanced PHP 5 forms with these powerful techniques](#)
- [Use PHP 5 for console scripting](#)
- [Follow best practice by validating data with PHP's character type functions](#)

Version history

Version: 1.0

Published: December 20, 2005

Tell us what you think

TechRepublic downloads are designed to help you get your job done as painlessly and effectively as possible. Because we're continually looking for ways to improve the usefulness of these tools, we need your feedback. Please take a minute to [drop us a line](#) and tell us how well this download worked for you and offer your suggestions for improvement.

Thanks!

—The TechRepublic Downloads Team