**PHP Music Lab**

Brian has written a page named music.php to display his MP3s. He has already written the basic HTML and CSS code. But the HTML is redundant, and he must manually edit it every time he adds songs to his collection. In this lab, you will **turn this page into a dynamic PHP page**that removes redundancy and dynamically lists the MP3s on the computer.

* Download music.php and open it in your text editor.
* Add a variable for the number of songs, and use it in the HTML/PHP code. Change its value to something other than 1234, such as 5678. Express the number of hours of music in terms of your variable. Assume that 10 songs can be played per hour.
* The five Yahoo music links at the top of the page are redundant, so express them using a for-loop. You should be able to change your loop bound to get more/fewer page links. Try that. Also, the links should still work when you click on them. Challenge: Add a GET query parameter, newspages, for the number of Yahoo links to display. (See the folder with a query example.) If the parameter is not passed, newspages should default to 5 pages.
* The page lists Brian's favorite artists as a bulleted list. Create an **array** containing the favorite artists' names and use the array to generate the bullets in the list. Test that it's working by adding at least one more favorite artist. Challenge: Instead of writing your own array of favorite artists, read them from a file, [favorite.txt](http://courses.cs.washington.edu/courses/cse190m/12sp/labs/3/favorite.txt). Read the artists from it, one per line. (Day 3 examples illustrate working with files.)
* For added challenge, make each artist's name a link. Example: Justin Bieber links to <http://music.yahoo.com/videos/justin-bieber/> (The im\_explode example illustrates String methods that are helpful.)
* The page hard-codes the list of MP3 file names. Change it to read MP3 file names from the server in folder /songs/. Comment out the existing list of MP3/playlist file names on the page. Write PHP code that looks for the names of all .mp3 files in the /songs/ folder using the [glob](http://www.php.net/glob) function.  (Pass a folder/file path to glob, NOT a URL. For debugging, you may want to [print\_r](http://www.php.net/print_r" \t "_blank) the array.) Print each file name as a list item with the class of mp3item.
* Now format the MP3 links to look better. Make each item display just the song file name, without the .../songs/ folder. Use the [basename](http://www.php.net/basename" \t "_blank) function to extract the file name from a larger path string. For example,  basename("a/b/c.txt") returns"c.txt". Make each item be a link to download that song. (Add a tag whose href target is the same as the song file's full URL.)
* For added challenge, display the size of each song file in kilobytes. You can get the size of a file in bytes using the [filesize(filename)](http://php.net/filesize" \t "_blank) function. 1 kb is 1024 bytes.