CS 325 - Introduction to Web Programming

JanPlan 2018

Lab 6: PHP (Start in class and finish by *tomorrow* midnight for full credit)

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.

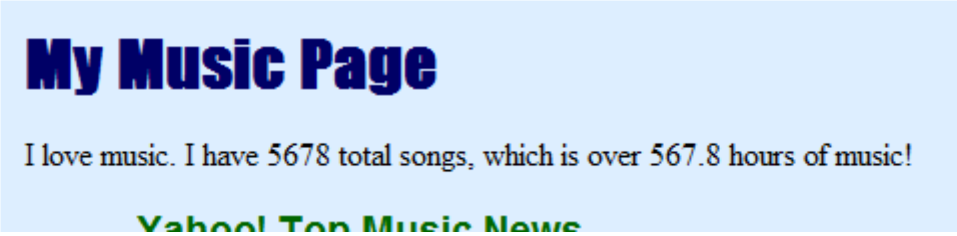
In this lab, you will **turn this page into a dynamic PHP page** that removes redundancy and dynamically lists the MP3s on the computer.

Download music.php to get started.

Exercise 1: Variable for Song Count

Brian has 1234 songs (123 hours of music). Turn this into a PHP variable you can change.

- 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.
- Upload it to the web server and point your browser to music.php to see the change.



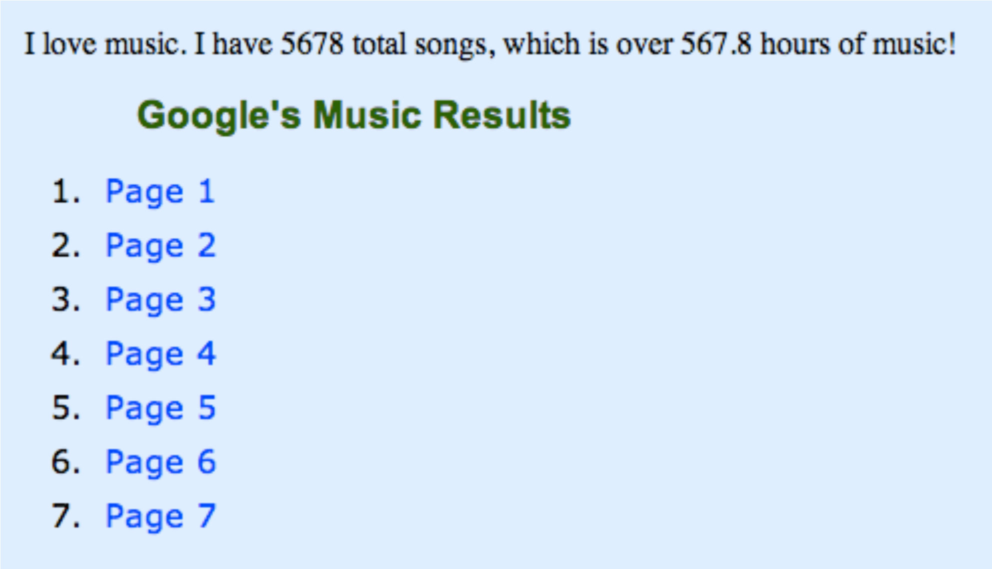
My Music Page

I love music. I have 5678 total songs, which is over 567.8 hours of music!

Yahoo! Top Music News

Exercise 2: Loop for Links

- The five Google music result 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.



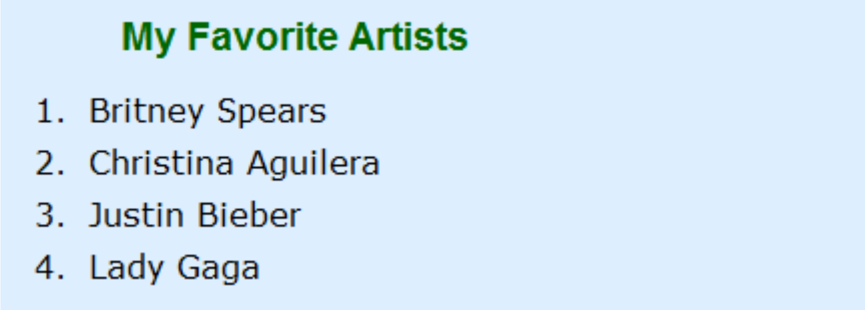
I love music. I have 5678 total songs, which is over 567.8 hours of music!

Google's Music Results

1. Page 1
2. Page 2
3. Page 3
4. Page 4
5. Page 5
6. Page 6
7. Page 7

Exercise 3: Array of Favorite Artists

- 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 (e.g. Lady Gaga).



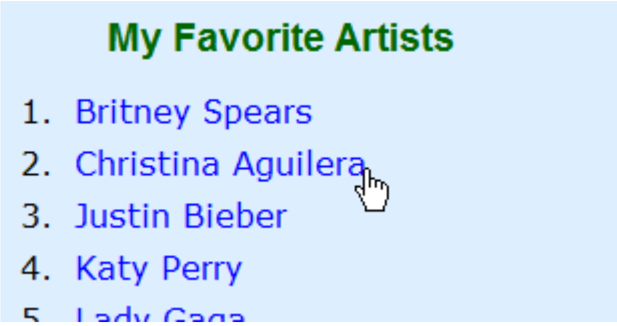
My Favorite Artists

1. Britney Spears
2. Christina Aguilera
3. Justin Bieber
4. Lady Gaga

Exercise 4: File of Favorite Artists

- Instead of writing your own array of favorite artists, read them from a file, favorite.txt.
- Download this file to your machine and upload it to web server with your PHP page. Read the artists from it, one per line.
- Make each artist's name a link.
    - The links have names similar to the artist's name but in lowercase with dashes

        - Hint: Check out PHP's String functions http://php.net/manual/en/ref.strings.php.
    - Example: Justin Bieber links to https://www.vevo.com/artist/**justin-bieber**



My Favorite Artists

1. Britney Spears
2. Christina Aguilera
3. Justin Bieber
4. Katy Perry
5. Lady Gaga

Exercise 5: List MP3 Files

(See example screenshot below)

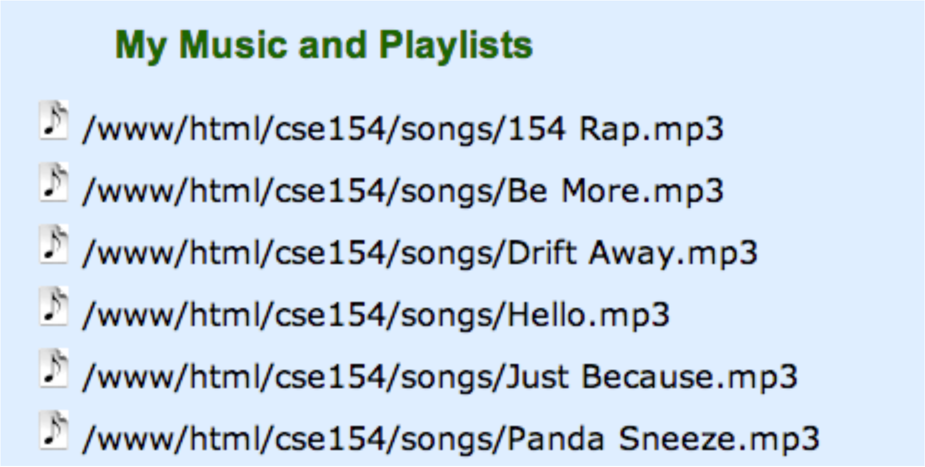
The page hard-codes the list of MP3 file names. Change it to read file names from the server.

- Create a folder called songs on the server (where you have stored your PHP file)
- Copy and paste any existing MP3 files you have in it.
- Comment out the existing list of MP3/playlist file names on the PHP page.
- Write PHP code that looks for the names of all .mp3 files in the folder you just created using the glob function. *(Pass a folder/file path to glob, NOT a URL. For debugging, you may want to print_r the array.)*
- Print each file name as an li list item with the class of mp3item.

Note: For the purpose of this exercise, you can assign any existing MP3 files you have. I am not expecting you to download the exact MP3 files from the internet to link to playlist names on your page.

Exercise 5, output

Your MP3 list should look like this when you are done:



My Music and Playlists

- ♪ /www/html/cse154/songs/154 Rap.mp3
- ♪ /www/html/cse154/songs/Be More.mp3
- ♪ /www/html/cse154/songs/Drift Away.mp3
- ♪ /www/html/cse154/songs/Hello.mp3
- ♪ /www/html/cse154/songs/Just Because.mp3
- ♪ /www/html/cse154/songs/Panda Sneeze.mp3

The “path” above will be replaced by the path where you stored your MP3 files.

Exercise 6: MP3 Formatting

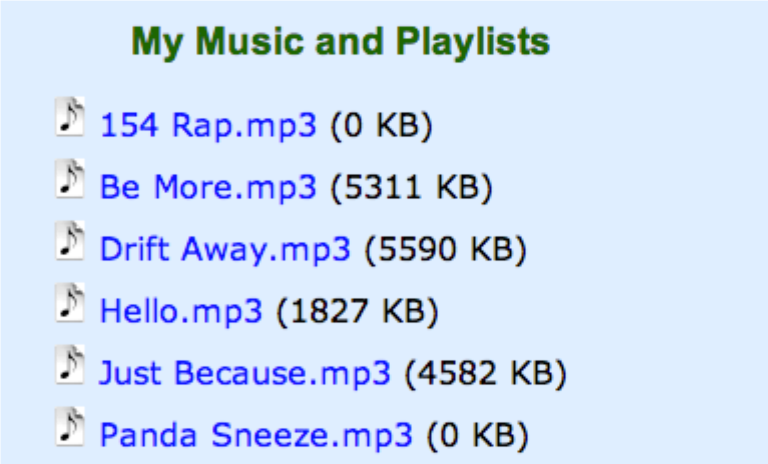
(See example screenshot below)

Now let's format the MP3 links to look better.

- Make each item display just the song file name, without the .../songs/ folder.
    - Use the basename function to extract the file name from a larger path string. e.g. basename("a/b/c.txt") returns "c.txt".
    - basename function: http://www.php.net/basename
- Make each item be a link to download that song. (Add an a tag whose href target is the same as the song file's full URL.)
- Display the size of each song file in kilobytes. You can get the size of a file in bytes using the filesize (filename) function. 1 kb is 1024 bytes.
- Use round() to round decimal numbers to whole numbers.

Exercise 6, output

Your MP3 list should look like this when you are done:



My Music and Playlists

- 154 Rap.mp3 (0 KB)
- Be More.mp3 (5311 KB)
- Drift Away.mp3 (5590 KB)
- Hello.mp3 (1827 KB)
- Just Because.mp3 (4582 KB)
- Panda Sneeze.mp3 (0 KB)

Exercise 7: (h4x0rz only): Ordering/Sorting

- **audio tag:** HTML5 includes a new audio tag for playing sounds.
- Modify the page to insert an audio tag for each song so that it can be played in the page. (You can insert any song)
- **Shuffle:** Display the songs in a random order each time the page loads.
- **Reverse order:** Display the songs in reverse ABC order. You may want to look at PHP's array functions (http://php.net/manual/en/ref.array.php) related to sorting.

Bonus:

Sort by Size : Display the songs sorted from largest to smallest size rather than in their default sorted order. The largest songs (such as drift-away.mp3) would go first and smallest songs (such as cse-154-rap.mp3) last.

Submission Process:

Upload your files to the appropriate dropbox on the course Moodle page.