Level Up Your PHP Arrays

For this lab, you will create a PHP script that parses values from a comma separated value (CSV) file and then display them as an HTML table. The functions you will use in this lab are referenced on php.net.

1. Download **the 80s_vg_sales.csv** file from the course website. Upload the file to **csphplinux.northeaststate.edu** in your **public_html** folder**.** This file is a list of video games sold in the 1980s. The file contains name, platform, year, genre, publisher, and global sales totals per million. Each line of text represents a different video game, except the top line of the file which contains the headers for each column. Each column is delimited in each line by a comma.
2. Create a file **lab03.php** using Lab 1 as a template.
3. Using the **file()** function in PHP, read in the text file 80s_vg_sales.csv and convert it into an array of lines. Use an appropriate identifier for your file array. The code below assume the CSV file is in the same directory as your PHP script.

```php
//Open a file and store in array
$lines = file( filename: '80s_vg_sales.csv');
```

4. The **file()** function returns the file as an array of string, an element for each line. At this step you can verify the file function is working by passing the variable you created for the array into the function **print_r()**. Remove the **print_r()** function once you have tested your code.
5. Create a header in the body of your HTML code, **<h2>File Array Output</h2>**
6. Now create a **foreach loop** that walks through each line of the array. Your code should
    a. echo paragraph tags around each line of the file
    b. add a line break after the closing paragraph tag (\n)
    c. use the **trim()** function in PHP on each element in the array

```php
// Loop through the array
foreach ($lines as $line) {
    echo "<p>" . trim($line) . "</p>\n";
}
```

7. Here is a sample of the output and source after running lab03.php

File Array Output

Name,Platform,Year,Genre,Publisher,Global_Sales

Asteroids,2600,1980,Shooter,Atari,4.31

Missile Command,2600,1980,Shooter,Atari,2.76

Kaboom!,2600,1980,Misc,Activision,1.15

Defender,2600,1980,Misc,Atari,1.05

```
<html>
<head>
    <title>Lab 01</title>
</head>
<body>
<h2>File Array Output</h2>
<p>Name,Platform,Year,Genre,Publisher,Global_Sales</p>
<p>Asteroids,2600,1980,Shooter,Atari,4.31</p>
<p>Missile Command,2600,1980,Shooter,Atari,2.76</p>
<p>Kaboom!,2600,1980,Misc,Activision,1.15</p>
<p>Defender,2600,1980,Misc,Atari,1.05</p>
```

8. Notice the elements in each line of the CSV file are separated by commas; the commas are used as delimiters to separate each column of data in a row. For the next part of the lab, you will create a 2-dimensional array, an array of arrays, where each sub array is an array of individual video game elements.
9. To turn each line of data into an array, you will use the **explode()** function in PHP. The explode function takes two parameters, the delimiter to split each element on, and the string to split up The function then returns an array with each column as an element in an array.
10. First create an empty array. Your code might look something like this.

```
//can haz array
$video_games = array();
```

11. Now create a foreach loop that walks through each line of the file array again. Use the explode function to create a game element array and add it to the game array. The easiest way to add an element to an array is to assign the next value in array with an empty index field, e.g., $someArray[] = “Dolores Claiborne”; This example would add Dolores Claiborne to the end of the array $someArray.

```php
foreach ($lines as $line) {
    $video_games[] = explode(",", trim($line));
}
```

12. Once again, to test to see if our code worked, you can pass the video games array to print_r() (you will see the output below in when you view the source). Once you have verified your code works, remove the print_r() function or comment it out.

```
Array
(
    [0] => Array
        (
            [0] => Name
            [1] => Platform
            [2] => Year
            [3] => Genre
            [4] => Publisher
            [5] => Global_Sales

        )

    [1] => Array
        (
            [0] => Asteroids
            [1] => 2600
            [2] => 1980
            [3] => Shooter
            [4] => Atari
            [5] => 4.31

        )

    [2] => Array
        (
            [0] => Missile Command
            [1] => 2600
            [2] => 1980
            [3] => Shooter
            [4] => Atari
            [5] => 2.76

        )
```

13. In HTML, create a header **<h2> 80s Games Sales </h2>**
14. Create a table (<table> </table>) and use a PHP foreach loop to create each row and element.

```
<h1>80's Game Sales</h1>
<table border="1">
<?
foreach ($video_games as $game){
    echo "<tr>";
    echo "<td>" . $game[0] . "</td>";
    echo "<td>" . $game[1] . "</td>";
    echo "<td>" . $game[2] . "</td>";
    echo "<td>" . $game[3] . "</td>";
    echo "<td>" . $game[4] . "</td>";
    echo "<td>" . $game[5] . "</td>";
    echo "</tr>\n";
}?>
</table>
```

15. Your output and source should look like this.

80's Game Sales

Name	Platform	Year	Genre	Publisher	Global_Sales
Asteroids	2600	1980	Shooter	Atari	4.31
Missile Command	2600	1980	Shooter	Atari	2.76
Kaboom!	2600	1980	Misc	Activision	1.15
Defender	2600	1980	Misc	Atari	1.05
Boxing	2600	1980	Fighting	Activision	0.77
Ice Hockey	2600	1980	Sports	Activision	0.49
Freeway	2600	1980	Action	Activision	0.34
Bridge	2600	1980	Misc	Activision	0.27

```
<h1>80's Game Sales</h1>
<table border="1">
<tr><td>Name</td><td>Platform</td><td>Year</td><td>Genre</td><td>Publisher</td><td>Global_Sales</td></tr>
<tr><td>Asteroids</td><td>2600</td><td>1980</td><td>Shooter</td><td>Atari</td><td>4.31</td></tr>
<tr><td>Missile Command</td><td>2600</td><td>1980</td><td>Shooter</td><td>Atari</td><td>2.76</td></tr>
<tr><td>Kaboom!</td><td>2600</td><td>1980</td><td>Misc</td><td>Activision</td><td>1.15</td></tr>
<tr><td>Defender</td><td>2600</td><td>1980</td><td>Misc</td><td>Atari</td><td>1.05</td></tr>
<tr><td>Boxing</td><td>2600</td><td>1980</td><td>Fighting</td><td>Activision</td><td>0.77</td></tr>
<tr><td>Ice Hockey</td><td>2600</td><td>1980</td><td>Sports</td><td>Activision</td><td>0.49</td></tr>
<tr><td>Freeway</td><td>2600</td><td>1980</td><td>Action</td><td>Activision</td><td>0.34</td></tr>
<tr><td>Bridge</td><td>2600</td><td>1980</td><td>Misc</td><td>Activision</td><td>0.27</td></tr>
<tr><td>Checkers</td><td>2600</td><td>1980</td><td>Misc</td><td>Atari</td><td>0.24</td></tr>
<tr><td>Pitfall!</td><td>2600</td><td>1981</td><td>Platform</td><td>Activision</td><td>4.5</td></tr>
<tr><td>Frogger</td><td>2600</td><td>1981</td><td>Action</td><td>Parker Bros.</td><td>2.2</td></tr>
<tr><td>Demon Attack</td><td>2600</td><td>1981</td><td>Shooter</td><td>Imagic</td><td>2.13</td></tr>
```

16. Add a **for** or **foreach** loop that calculates a total for the Global_Sales column and prints the result. Hint: You will have to skip the first row of your file array because the first row is a column header row.

Global sales total estimate: $376580000

17. Be prepared to answer questions about this code and data, such as, why is the global sales total estimate not an accurate representation of the actual sales data? Or, if this dataset were in an SQL database, what other operations could we perform on the data?

Deliverables

1. Demonstrate the code and execution to the instructor during this lab, during office hours, or during the next lab period.
2. Upload the instructor approved .php file (source code only) to the Lab 3 D2L drop box.

Resources

- Full video game data set: https://gist.github.com/zhonglism/f146a9423e2c975de8d03c26451f841e#file-vgsales-csv