PHP Homework

1. The Basics

   a. On your turing account, create a public_html (use lowercase and use an underscore to separate words!). Be sure to correctly set your file permission (should be 705 – or rwx for the first, --- for the middle, and r-x for the third)

   b. Create a basic html file, called temp.php, and enter the following information. Verify that it opens in a browser (URL is *turing.cs.olemiss.edu/~yourWebID*):

```
<html lang = “en”>
        <head>
                <title>basic</title>
        </head>

        <body>
        My PHP Information Page
        <?php
                //My first real php file
                echo “<hr />”;
                phpinfo();
        ?>
        </body>
</html>
```

NOTE: <hr /> is for horizontal ruler & <br /> is for line break. Yes, there is a space between the r & /

   c. Create a second php file, ***yourLastNameFirstInitialPHPBasics.php***; add the following variables:

```
        <?php
                $addOn = "again";
                $addOn2 = 'and again';
                $addOn3 = ' and again';
                echo "First Attempt:  Hello World $addOn  $addOn2 <br />";
                echo "Second Attempt: Hello World $addOn $addOn2{$addOn3}";
                echo "Third Attempt: "."{$addOn}"." {$addOn2}"."{$addOn3} <br /><hr />";
        ?>
```

   Upload your file to your public_html folder. Open a browser and navigate to your file:
   https://turing.cs.olemiss.edu/~yourWebid (replace with your webid – click on your file)

   If your browser is blank, it means you have a syntax error. To identify your error, open PuTTY (or Terminal) and browse to the folder with your file (use ***cd folderName → cd public_html***). Execute your php file on command line by typing ***php filename.php → php phpBasics.php***

   d. Open phpBasics.php on turing and add the following code which uses constants. Verify your code works before continuing to part e:

```
<?php
        define("DaysInYear", 365);
```

```
            echo DaysInYear."<br /><hr />";

            define("DaysInYear", 235);
            echo DaysInYear."<br /><hr />";
    ?>
```

e. Continuing in PHPBasics.php, typecast two variables, test1 and test2:

```
<?php
    $test1 = 3;
    $test2 = 3;
    settype($test1, "string");
     (string) $test2;
    “Data Type test1: “.gettype($test1).”<br />”;
    “Data Type test2: “.gettype($test2).”<br />”;
 ?>
```

f. In PHPBasics.php, create 5 variables, *first, second, third, fourth and fifth* with the following values (if you copy & paste code below there will be errors):

first: *The quick red fox*
second: *jumped over the lazy brown dog.*
third: assign **first** to **third**
fourth: concatenate **first** variable, a **space**, and **second** variable, assigning to **fourth**
The quick red fox jumped over the lazy brown dog
fifth: six **spaces** + *jumped over the lazy brown dog.*

Now, use the different string functions to modify these five variables:

```
echo “Lower: “.strtolower($first).”<br />“;
echo “Upper: “.strtoupper($second) .”<br />“;
echo “First Word: “.ucfirst($second) .”<br />“;
echo “All words:  “.ucwords($second) .”<br />“;
echo “Trim: “.trim($fifth) .”<br />“;
echo “Replace: “. trim(str_replace(“dog”, “cat”, $fifth)) .”<br />“;
echo “Length:  “.strlen($first) .”<br />“;
echo “Substring:  “.substr($second, 16, 4) .”<br />“;
echo “Find: “.strstr($second, “lazy”) .”<br />“;
echo “Repeat: {$first}”.str_repeat($second, 2) .”<br />“;
echo “The position of brown begins at index: “.strpos($fourth, “brown”) .”<br />“;
echo “String Fragment: “. strchr($fourth, “z”) .”<br />“;
echo “String comparison: “.strcmp($first, $third) .”<br />“;
echo “String comparison ignoring case: “.strcasecmp($first, strtoupper($third)) .”<br />“;
```

g. Php does type juggling – it will try to convert to a data type, picking out what it needs and throwing away the rest. Add to phpBasics.php the following:

```
echo “<hr />Type Juggling<br />“;
$count = “2 cats”;
```

```
echo “$count Data Type: “. gettype($count). “<br />“;

$count += 3;
echo “$count Data Type: “.gettype($count). “<br />“;
$cats = “I have “ . $count . “ cats.”;
echo “$cats Data Type: “.gettype($cats). “<br />“;
echo “<hr />Type Casting<br />“;
$count3 = count;
settype($count3, “integer”);
echo “Count 3: “.gettype($count3). “<br />“;

 $count4 = (string) $count3;
echo “Count 3 Again: “.gettype($count3).”<br />“;
echo “Count 4: “.gettype($count4).”<br />“;
```

2. Arrays: Standard and Associative
   NOTE: You may READ instructions a. – i below without actually adding to the file PHPArrays.php. This part of the homework only requires you to complete j. – m.

   Standard

   a. Download arrays.php from Blackboard and copy into your public_html file on Blackboard.
   b. Add an element to $numbers3 as follows:

      ```
      $numbers3[7] = “mountain lion”;
      ```

   c. Now, display the new array using the print-readable function: print_r($numbers3);
   d. Last, display each element using the foreach loop

      ```
      foreach($numbers3 as $oneNumber)  {
              echo $oneNumber.”<br />“;
      }
      ```

   Associative

   e. Create a standard array as follows:

      ```
      $sportsLeagues = [“ncaa”, “nfl”, “nba”, “mlb”, “fifa”];
      ```

   f. Now, create associative arrays for each element in the *sportsLeagues*. For example, for ncaa and nfl, you would code the following:

      ```
      $ncaa = [“sport” => “basketball”, “championship” => “March Madness”];
      $nfl = [“sport” => “football”, “championship” => “Super Bowl”];
      ```

      The remaining sports & championship game are as follows:
      NBA basketball – Championship Series
      MLB baseball – World Series
      FIFA soccer – World Cup

   g. Output some of the values:

      ```
      echo $fifa[“sport”].”<br />“;
      echo $fifa[“championship”].”<br />“;echo
      ${$sportsLeagues[2]}[“sport”].”<br />“;
      ```

   h. Now, write a foreach loop that steps through ***$sportsLeagues*** using these values to reference their corresponding associative array. Ouput the *sport* and *championship game/series*.

```
foreach($sportsLeagues as $oneSport) {
        echo ${$oneSport}['sport'].”<br />“;
        echo ${$oneSport}['championship'].”<br />“;
}
```

i. Php has an array_search function that will search for a value. Enter the following code:

```
If (array_search(“nba”, $sportsLeagues) )  {
        echo “NBA found! <br />“;
}
else  {
        echo “NBA not found! <br />“;
}
```

Write an if-statement to search for MLS (Major League Soccer)?

j. Now create a file ***yourLastNameFirstInitialPHPArrays.php.***

k. First, define the following standard array, which will be used to reference each SEC school’s corresponding associative arrays (if you simply copy & paste there will be errors):

```
$sec = [‘auburn’, ‘lsu’, ‘msu’, ‘texasAM’, 'alabama', 'arkansas', 'florida', 'georgia', 'kentucky',
‘oleMiss’, ‘mizzou’, ‘southCarolina’, ‘tennessee’, ‘vanderbilt’];
```

l. Now, define separate associative arrays for each school with the following key values:

school: name of school
mascot: school’s mascot
city
state
division: whether school is in SEC East or SEC West

School	Mascot	City	State	Division
Auburn	Tigers	Auburn	AL	West
LSU	Fighting Tigers	Baton Rouge	LA	West
MSU	Bulldogs	Starkville	MS	West
Texas A&M	Aggies	College Station	TX	West
Alabama	Crimson Tide	Tuscaloosa	AL	West
Arkansas	Razorbacks	Fayetteville	AR	West
Florida	Gators	Gainesville	FL	East
Georgia	Bulldogs	Athens	GA	East
Kentucky	Wildcats	Lexington	KY	East
Ole Miss	Rebels	Oxford	MS	West
Mizzou	Tigers	Columbia	MO	East
South Carolina	Gamecocks	Columbia	SC	East
Tennessee	Volunteers	Knoxville	TN	East
Vanderbilt	Commodores	Nashville	TN	East

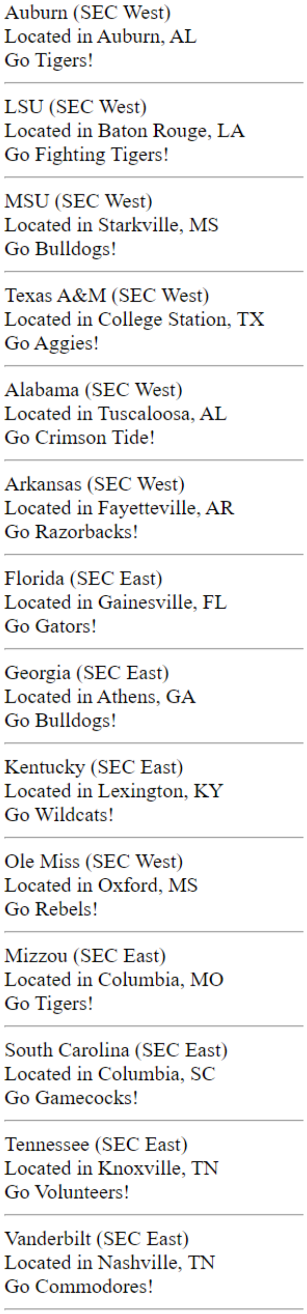
For example, the first associative array would be as follows:
$auburn = ['school' => 'Auburn', 'mascot' => 'Tigers', 'city' => 'Auburn', 'state' => 'AL', 'division' => 'West'];

m. Once you have defined each associative array, **write a foreach loop** to step through *$sec*. For each element of *$sec*, "build" a new variable referencing the corresponding associative array (the first element in *$sec* – *'auburn'* – would be used to access values in the *$auburn* associative array) – refer to e. through h. above as a guide. Also refer to the handout on arrays.

For each school, output the school, mascot, city, state, division and a <hr />. See Sample Output for how to format each element. Your output must look the same to get full credit.

n. Submit both PHPBasics.php and PHPArrays.php on Blackboard.

Sample Output:



```
Auburn (SEC West)
Located in Auburn, AL
Go Tigers!

LSU (SEC West)
Located in Baton Rouge, LA
Go Fighting Tigers!

MSU (SEC West)
Located in Starkville, MS
Go Bulldogs!

Texas A&M (SEC West)
Located in College Station, TX
Go Aggies!

Alabama (SEC West)
Located in Tuscaloosa, AL
Go Crimson Tide!

Arkansas (SEC West)
Located in Fayetteville, AR
Go Razorbacks!

Florida (SEC East)
Located in Gainesville, FL
Go Gators!

Georgia (SEC East)
Located in Athens, GA
Go Bulldogs!

Kentucky (SEC East)
Located in Lexington, KY
Go Wildcats!

Ole Miss (SEC West)
Located in Oxford, MS
Go Rebels!

Mizzou (SEC East)
Located in Columbia, MO
Go Tigers!

South Carolina (SEC East)
Located in Columbia, SC
Go Gamecocks!

Tennessee (SEC East)
Located in Knoxville, TN
Go Volunteers!

Vanderbilt (SEC East)
Located in Nashville, TN
Go Commodores!
```