Updated: January 21, 2017
Author: Charlie Wallin

Arrays and Cards

Objectives

- Learn the basic array functions
- Examine variable contents with **`var_dump`**
- Create a multi-dimensional array

Reading

Read chapter 3: Arrays

Assignment - Cards

Create a folder named **`my-code`** inside your **`web182`** folder

Create a folder named **`ch03`** inside your **`my-code`** folder

Create a folder named **`cards`** inside your **`ch03`** folder.

The folder path should look like

```
c:\xampp\htdocs\web182\my-code\ch03\cards
```

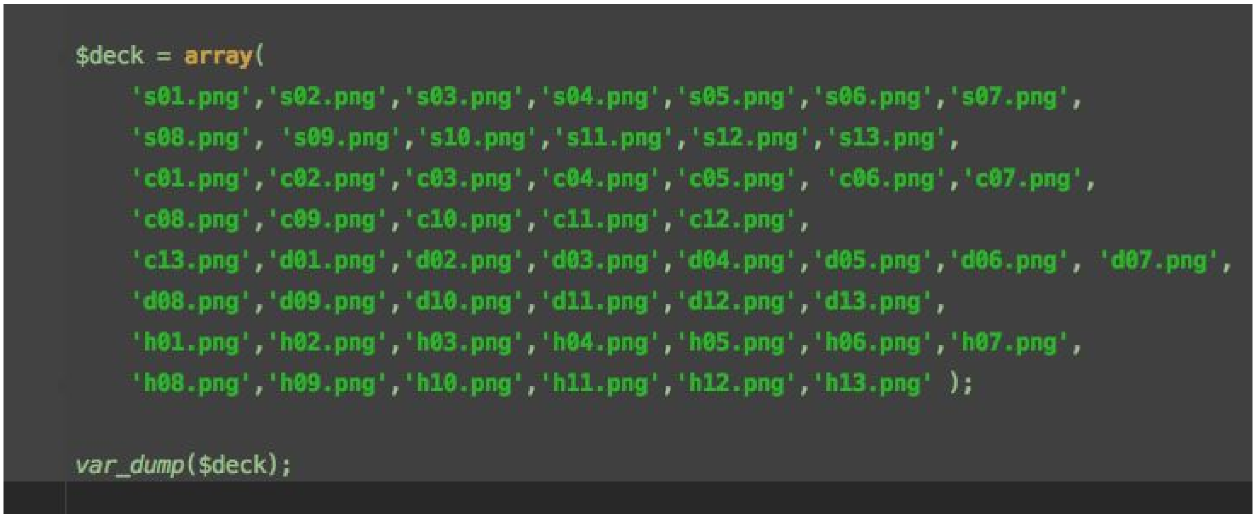
Download and unzip the **`cards.zip`** file. It contains a folder called **`card-images`** containing 52 playing cards images and a file named **`cards-array.php.`** Save both the folder and the file in your cards folder. The file paths should look like.

```
C:\xampp\htdocs\web182\my-code\ch03\cards\card-images
```

```
C:\xampp\htdocs\web182\my-code\ch03\cards\cards-array.php
```

Open up the **`card-images`** folder. Note the names for the images: aces are 01, jacks are 11, queens 12 and kings 13. Also notice that each suit is represented by the first letter of the file name. For instance d02.png is the two of diamonds.You will work with this deck to learn some basic array functions.

Open the cards-array.php file. It is just an array that contains the names of the card images so we can reference them. For this exercise you can start by copying and pasting the array within your index.php file as in the screendump below



```
$deck = array(
    's01.png','s02.png','s03.png','s04.png','s05.png','s06.png','s07.png',
    's08.png', 's09.png','s10.png','s11.png','s12.png','s13.png',
    'c01.png','c02.png','c03.png','c04.png','c05.png', 'c06.png','c07.png',
    'c08.png','c09.png','c10.png','c11.png','c12.png',
    'c13.png','d01.png','d02.png','d03.png','d04.png','d05.png','d06.png', 'd07.png',
    'd08.png','d09.png','d10.png','d11.png','d12.png','d13.png',
    'h01.png','h02.png','h03.png','h04.png','h05.png','h06.png','h07.png',
    'h08.png','h09.png','h10.png','h11.png','h12.png','h13.png' );

var_dump($deck);
```

var_dump

Notice the last line is called `var_dump`. This function dumps the contents to the screen so you can see the contents. To get started, set up the code as shown above use the **`var_dump`** function to see how it works.

You will notice that the result runs together and it is kind of difficulty to read. Fortunately there are a couple of Chrome plugins to make this easier. Download and install the Chrome plugins var_masterpiece and json_formatter.

Once the plugins are installed, run the program again and notice that your code will be displayed in a readable format. From what I have read, you should be able to get this result with just the var_masterpiece plugin but that did not work for me so I added json_formatter. Play around with it to see what works best. The same plugins are available for FireFox.

Remove the var_dump line of code once you are satisfied the program is working correctly.

Images

For the following set of exercises you are required to show the card images to prove that your code is running the array functions. Previously you unzipped the **cards-images** folder. Make sure it is in the following path

`C:\xampp\htdocs\web182\my-code\ch03\cards\card-images`

The `foreach` function

(10 points)

In this section you will display the entire deck (card images) using a foreach statement combined with HTML. Remember that you are displaying HTML so you will need to use the echo statement and the **`<img src=...>`** tag.

After the program runs as expected, add an **<hr>** tag. We are just going to use one file for this part of the assignment so we will create visual break between each part.

Here is the code for the foreach to get your started

```
foreach($deck as $card) {
   echo "<img src=\"cards-images\\$card\" alt=\"$card\">";
}
```

Notice that the code contains the backslash escape character. This is to “tell” PHP that you want to treat the quotes within quotes separately.

Jack of Hearts

(10 points)

Write code to display only the jack of hearts.

Create a break with an **`<hr>`** tag.

Display only the spades suit

(10 points)

This one gets a bit trickier. You will need to determine which file name starts with the letter “s”. You can use the PHP function substr() to do this. Here is an example that omits the starting place (x) and how many characters to count (y)

```
        if (substr($card, x, y) == "s")
```

substr() from W3Schools

Create a break with an **`<hr>`** tag

Shuffle

(10 points)

Shuffle the deck and display it. The cards will reshuffle each time the page is refreshed. Your array will now be completely different. Run the **`var_dump()`** function to check it out.

Create a break with an **`<hr>`** tag

Deal 5

(10 points)

Randomly deal five cards from the deck using a for loop. The deal will happen each time the page is refreshed. Create a break with an **`<hr>`** tag

Odd Cards Only

(15 points)

Use a foreach loop and display only the cards where the **file name** is an odd number. Check out the modulus operator for this one.

The modulus operator is only covered briefly in CIS115 but it is a really useful function. It returns the remainder value from integer division. For instance 3/2 = 1 with a remainder of 1, so the modulus would equal 1 but 2 / 2 = 1 with a remainder of zero, so the modulus would be zero. Anything divided by 2 will give you either a one or a zero. If it is a 1 then the number is odd, if it is zero then the number is even.

You have already used the **`substr()`** function to extract the first letter of the filename, now you can use it to extract the number from the file name. For instance in the filename s01.png you would want to

Extract the number 1 (you don't need the preceding zero) so you could use the **`substr()`** function as follows. This code will return the even cards.

```
foreach($deck as $card) {
   $number = substr($card, 2, 1);
   if($number % 2 ==0)
       echo "$card is even";
       // the output should be the card image. I've
       // left in the echo for proof of concept
       // Modify the code.
}
```

Create a break with an **`<hr>`** tag

Multi-dimensional Arrays

So for we have worked with some array functions with a singular list of key-value pairs. Use the Bob's Auto parts section of the book to create an associative array from the following table

Instrument	Code	Price
Fiddle	FID	$1250
Concertina	CON	$2500
Bodhran	BOD	$500

Display the contents of the array

(10 points)

Use a `for` loop to display the contents of the array (p. 89)

Create a break with an **`<hr>`** tag

Display the Concertina and it's price.

(10 points)

Your output should display **A Concertina costs $2500**

Submitting Your Work

Zip up the contents of your folder and save it as **ch03-yourLastName.zip**. Submit your zipped file in Moodle and upload your files to your host.