Lesson Proper for Week 17

Understanding Arrays

Arrays are indexed, which means that each entry is made up of a *key* and a *value*. The key is the index position, beginning with 0 and increasing incrementally by 1 with each new element in the array. The value is whatever value you associate with that position—a string, an integer, or whatever you want. Think of an array as a filing cabinet and each key/value pair as a file folder. The key is the label written on the top of the folder, and the value is what is inside.

PHP Arrays

PHP array is an ordered map (contains value on the basis of key). It is used to hold multiple values of similar type in a single variable.

Advantage of PHP Array

ü **Less Code**: We don't need to define multiple variables.

ü **Easy to traverse**: By the help of single loop, we can traverse all the elements of an array.

ü **Sorting**: We can sort the elements of array.

PHP Array Types

There are 3 types of array in PHP.

1. **Indexed Array / Numeric array** − An array with a numeric index. Values are stored and accessed in linear fashion.

2. **Associative Array** - An array with strings as index. This stores element values in association with key values rather than in a strict linear index order.

3. **Multidimensional Array** - An array containing one or more arrays and values are accessed using multiple indices

PHP Indexed Array

PHP index is represented by number which starts from 0. We can store number, string and object in the PHP array. All PHP array elements are assigned to an index number by default.

Creating Arrays

You can create an array using either the array() function or the array operator[]. The array() function is usually used when you want to create a new array and populate it with more than one element, all in one fell swoop. The array operator is often used when you want to create a new array with just one element at the outset, or when you want to add to an existing array element.

There are two ways to define indexed array / Numerical Array:

1: The following code snippet shows how to create an array

```
$season=array("summer","winter","spring","autumn");
```

2: The following snippet shows the same array being created incrementally using the array operator:

```
$season[0]="summer";
$season[1]="winter";
$season[2]="spring";
$season[3]="autumn";
```

PHP Associative Array

The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index. Associative array will have their index as string so that you can establish a strong association between key and values. Whereas numerically indexed arrays use an index position as the key—0, 1, 2, and so forth—associative arrays use actual named keys. We can associate name with each array elements in PHP using => symbol.

There are two ways to define associative array:

Example 1:

```
$salary=array("Sonoo"=>"350000","John"=>"450000","Kartik"=>"200000");
```

Example 2:

```
$salary["Sonoo"]="350000";
$salary["John"]="450000";
$salary["Kartik"]="200000";
```

PHP Multidimensional Array

PHP multidimensional array is also known as array of arrays. It allows you to store tabular data in an array. PHP multidimensional array can be represented in the form of matrix which is represented by row * column. A multi-dimensional array each element in the main array can also be an array. And each element in the sub-array can be an array, and so on. Values in the multi-dimensional array are accessed using multiple index.

Example:

```
$emp = array
 (
 array(1,"sonoo",400000),
 array(2,"john",500000),
 array(3,"rahul",300000)
 );
```

PHP Array Functions

More than 70 array-related functions are built in to PHP, which you can read about in detail at http://php.net/manual/en/ref.array.php. Some of the more common (and useful) functions are described briefly in this section:

► **count()** and **sizeof()** —Each of these functions counts the number of elements in an array; sizeof() is an alias of count(). Given the array

$colors = array("blue", "black", "red", "green");

both count($colors); and sizeof($colors); return a value of 4.

► **each()** and **list()** —These functions (well, list() is a language construct that *looks* like a function) usually appear together, in the context of stepping through an array and returning its keys and values. You saw an example of this previously, where we stepped through the $c array and printed its contents.

► **foreach()** —This control structure (which looks like a function) is used to step through an array, assigning the value of an element to a given variable, as you saw in the previous section.

► **reset()** —This function rewinds the pointer to the beginning of an array, as in this example:

reset($character);

This function proves useful when you are performing multiple manipulations on an array, such as sorting, extracting values, and so forth.

► **array_push()** —This function adds one or more elements to the end of an existing array, as in this example:

array_push($existingArray, "element 1", "element 2", "element 3");

► **array_pop()** —This function removes (and returns) the last element of an existing array, as in this example:

$last_element = array_pop($existingArray);

► **array_unshift()** —This function adds one or more elements to the beginning of an existing array, as in this example:

```
array_unshift($existingArray, "element 1", "element 2", "element 3");
```

► **array_shift()** —This function removes (and returns) the first element of an existing array, as in this example, where the value of the element in the first position of $existingArray is assigned to the variable $first_element:

```
$first_element = array_shift($existingArray);
```

► **array_merge()** —This function combines two or more existing arrays, as in this example:

```
$newArray = array_merge($array1, $array2);
```

► **array_keys()** —This function returns an array containing all the key names within a given array, as in this example:

```
$keysArray = array_keys($existingArray);
```

► **array_values()** —This function returns an array containing all the values within a given array, as in this example:

```
$valuesArray = array_values($existingArray);
```

► **shuffle()** —This function randomizes the elements of a given array. The syntax of this function is simply as follows:

```
shuffle($existingArray);
```

PHP - Strings

A collection of characters enclosed within either single or double quotation marks. A string variable may consist of a single letter, a word, a sentence, a paragraph, HTML code, or even a jumble of nonsensical letters, numbers, and symbols (which might represent a password). Strings may be the most common variable type used in PHP. PHP string is a sequence of characters i.e., used to store and manipulate text. PHP supports only 256-character set and so that it does not offer native Unicode support.

1. single quoted - This method in used when we want to the **string** to be exactly as it is written.
2. double quoted - By using Double quotes the PHP code is forced to evaluate the whole string.
3. heredoc syntax - is a way to write large block of text inside **PHP**, without the classic single quote, double quotes delimiters.

Single Quoted

We can create a string in PHP by enclosing the text in a single-quote. It is the easiest way to specify string in PHP. For specifying a literal single quote, escape it with a backslash (\) and to specify a literal backslash (\) use double backslash (\\). All the other instances with backslash such as \r or \n, will be output same as they specified instead of having any special meaning.

Double Quoted

In PHP, we can specify string through enclosing text within double quote also. But escape sequences and variables will be interpreted using double quote PHP strings.

Heredoc

Heredoc syntax (<<<) is the third way to delimit strings. In Heredoc syntax, an identifier is provided after this heredoc <<< operator, and immediately a new line is started to write any text. To close the quotation, the string follows itself and then again that same identifier is provided. That closing identifier must begin from the new line without any whitespace or tab.

◀ Preliminary Activity for Week 17 Jump to...

Analysis, Application, and Exploration for Week 17 ▶

Navigation

Home
Dashboard
Site pages
My courses
121 - CC106
121 - BPM101 / DM103
121 - OAELEC2
121 - ITE3
121 - MUL101
121 - ITSP2B
121 - WEB101 / CCS3218
Participants
Grades
General
01 Introduction to Internet and World Wide Web
02 Full Stack Development Overview
03 The Basics of HTML
04 Mark Up Text
05 HTML Link, Image and Table
06 - Preliminary Examination
07 HTML Block, Inline Element and Forms Control
08 Introduction To Cascading Style Sheets

09 The Font Style Properties
10 The List and Table style properties
11 Advanced CSS
12 - Midterm Examination
13 JavaScript Basics
14 JavaScript Objects
15 Introduction to PHP
16 PHP Control Statement and Function
17 PHP Array and String
Preliminary Activity for Week 17
Lesson Proper for Week 17
Analysis, Application, and Exploration for Week 17
Generalization for Week 17
Evaluation for Week 17
Assignment for Week 17

Courses

Fair Warning

NOTICE: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for ***free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission***.

PROSECUTION: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following: Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense. Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.

COURSE OF ACTION: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.

2nd Semester Enrollment

Activities

- Assignments
- Forums
- Quizzes
- Resources