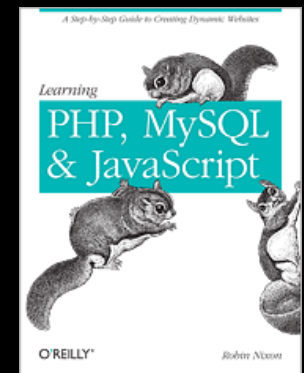


PHP Arrays

Chapter 6

Dr. Charles Severance

To be used in association with the book:
PHP, MySQL, and JavaScript by Robin Nixon



open.michigan

Unless otherwise noted, the content of this course material is licensed under a Creative Commons Attribution 3.0 License.

<http://creativecommons.org/licenses/by/3.0/>.

Copyright 2011, Charles Severance



PHP Arrays Rock!

- Better than Python Dictionaries
- Better than Java Hash Maps
- PHP Arrays have all the benefits of Python Dictionaries but they can also maintain the order of the items in the array

http://en.wikipedia.org/wiki/Associative_array

Associative Arrays

- Like Python Dictionaries - but more powerful
- Can be key => value or simply indexed by numbers
- Ignore two-dimensional arrays for now...

http://en.wikipedia.org/wiki/Associative_array

Integer Indices

```
<?php
    $stuff = array("Hi", "There");
    echo $stuff[1], "\n";
?>
```

There

Key / Value

```
<?php
    $stuff = array("name" => "Chuck",
                  "course" => "SI572");
    echo $stuff["course"], "\n";
?>
```

SI572

Dumping an Array

- The function `print_r()` dumps out PHP data - it is used mostly for debugging

```
<?php
    $stuff = array("name" => "Chuck",
                  "course" => "SI572");
    print_r($stuff);
?>
```

```
Array
(
    [name] => Chuck
    [course] => SI572
)
```

Building up an Array

- You can allocate a new item in the array and add a value at the same time using empty square braces [] on the right hand side of an assignment statement

```
$va = array();  
$va[] = "Hello";  
$va[] = "World";  
print_r($va);
```

```
Array  
(  
    [0] => Hello  
    [1] => World  
)
```

Building up an Array

- You can also add new items in an array using a key as well

```
$za = array();  
$za["name"] = "Chuck";  
$za["course"] = "SI572";  
print_r($za);
```

```
Array  
(  
    [name] => Chuck  
    [course] => SI572  
)
```

Looping Through an Array

```
<?php
    $stuff = array("name" => "Chuck",
                  "course" => "SI572");
    foreach($stuff as $k => $v ) {
        echo "Key=", $k, " Val=", $v, "\n";
    }
?>
```

```
Key=name Val=Chuck
Key=course Val=SI572
```

Arrays of Arrays

The elements of an array can be many things other than a string or integer. You can even have objects or other arrays.

```
$products = array(  
    'paper' => array(  
        'copier' => "Copier & Multipurpose",  
        'inkjet' => "Inkjet Printer",  
        'laser' => "Laser Printer",  
        'photo' => "Photographic Paper"),  
    'pens' => array(  
        'ball' => "Ball Point",  
        'hilite' => "Highlighters",  
        'marker' => "Markers"),  
    'misc' => array(  
        'tape' => "Sticky Tape",  
        'glue' => "Adhesives",  
        'clips' => "Paperclips")  
);
```

```
echo $products["paper"]["copier"];
```

```
Copier & Multipurpose
```

Array Functions

[PHP Manual](#)[Function Reference](#)[Variable and Type
Related Extensions](#)[Arrays](#)

- [Introduction](#)
- [Installing/Configuring](#)
- [Predefined Constants](#)
- [Sorting Arrays](#)
- **[Array Functions](#)**

[«Sorting Arrays](#)[array_change_key_case»](#)

view this page in

Brazilian Portuguese

[\[edit\]](#) Last updated: Fri, 16 Sep 2011

Array Functions

See Also

See also [is_array\(\)](#), [explode\(\)](#), [implode\(\)](#), [split\(\)](#), [preg_split\(\)](#), and [unset\(\)](#).

Table of Contents

- [array_change_key_case](#) — Changes all keys in an array
- [array_chunk](#) — Split an array into chunks
- [array_combine](#) — Creates an array by using one array for keys and another for its values
- [array_count_values](#) — Counts all the values of an array
- [array_diff_assoc](#) — Computes the difference of arrays with additional index check
- [array_diff_key](#) — Computes the difference of arrays using keys for comparison
- [array_diff_uassoc](#) — Computes the difference of arrays with additional index check which is performed by a user supplied callback function
- [array_diff_ukey](#) — Computes the difference of arrays using a callback function on the

Array Functions

- `count($ar)` - How many elements in an array
- `is_array($ar)` - Returns TRUE if a variable is an array
- `sort($ar)` - Sorts the array values (loses key)
- `ksort($ar)` - Sorts the array by key
- `asort($ar)` - Sorts array by value, keeping key association
- `shuffle($ar)` - Shuffles the array into random order

```
$za = array();
$za["name"] = "Chuck";
$za["course"] = "SI572";
print "Count: " . count($za) . "\n";
if ( is_array($za) ) {
    echo '$za Is an array' . "\n";
} else {
    echo '$za Is not an array' . "\n";
}
$zb = "123";
echo is_array($zb) ? '$zb Is an array' : '$zb Is not an array';
echo "\n";
```

```
Count: 2
$za Is an array
$zb Is not an array
```

```
$za = array();  
$za["name"] = "Chuck";  
$za["course"] = "SI572";  
$za["topic"] = "PHP";  
print_r($za);  
sort($za);  
print_r($za);
```

Array

```
(  
    [name] => Chuck  
    [course] => SI572  
    [topic] => PHP  
)
```

Array

```
(  
    [0] => Chuck  
    [1] => PHP  
    [2] => SI572  
)
```

```
$za = array();
$za["name"] = "Chuck";
$za["course"] = "SI572";
$za["topic"] = "PHP";
print_r($za);
ksort($za);
print_r($za);
asort($za);
print_r($za);
```

```
Array
(
    [name] => Chuck
    [course] => SI572
    [topic] => PHP
)
Array
(
    [course] => SI572
    [name] => Chuck
    [topic] => PHP
)
Array
(
    [name] => Chuck
    [topic] => PHP
    [course] => SI572
)
```

Arrays and Strings

```
$inp = "This is a sentence with seven words";  
$temp = explode(' ', $inp);  
print_r($temp);
```

Array

```
(  
    [0] => This  
    [1] => is  
    [2] => a  
    [3] => sentence  
    [4] => with  
    [5] => seven  
    [6] => words  
)
```

Summary

- PHP arrays are a very powerful associative array as they can be indexed by integers like a list, or use keys to look values up like a hash map or dictionary
- There are many options for sorting
- We can use `explode()` to split a string into an array of strings