

PHP Arrays

Dr. Charles Severance

www.wa4e.com

<http://www.wa4e.com/code/arrays>

<http://www.wa4e.com/code/arrays.zip>





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

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" => "WA4E" );
echo $stuff[ "course" ], "\n";
?>
```

WA4E

Dumping an Array

The function `print_r()` shows PHP data - it is good for debugging.

```
<?php
    $stuff = array( "name" => "Chuck",
                    "course" => "WA4E");
    echo( "<pre>\n");
    print_r($something);
    echo( "\n</pre>\n");
?>
```

```
Array(
    [name] => Chuck
    [course] => WA4E
)
```

var_dump vs. print_r

```
<?php
$stuff = array("name" => "Chuck",
               "course" => "SI664");
var_dump($stuff);
?>
```

```
array(2) {
  ["name"]=>
  string(5) "Chuck"
  ["course"]=>
  string(5) "SI664"
}
```

var_dump() and false

```
<?php
    $thing = FALSE;
    echo( "One\n");
    print_r($thing);
    echo( "Two\n");
    var_dump($thing);
?>
```

One
Two
bool(false)

Building Up an Array

You can allocate a new item in the array and append 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.

```
$za = array();  
$za[ "name" ] = "Chuck";  
$za[ "course" ] = "WA4E";  
print_r($za);
```

```
Array(  
    [name] => Chuck  
    [course] => WA4E  
)
```

Looping Through an Array

```
<?php
$stuff = array( "name" => "Chuck",
                "course" => "SI664");
foreach($stuff as $k => $v ) {
    echo "Key=", $k, " Val=", $v, "\n";
}
?>
```

Key=name Val=Chuck
Key=course Val=SI664

Looping Through an Array

```
<?php
$stuff = array( "Chuck", "SI664");
foreach($stuff as $k => $v ) {
    echo "Key=", $k, " Val=", $v, "\n";
}
?>
```

Key=0 Val=Chuck
Key=1 Val=SI664

Counted Loop Through an Array

```
<?php
$stuff = array( "Chuck", "SI664");
for($i=0; $i < count($stuff); $i++) {
    echo "I=", $i, " Val=", $stuff[$i], "\n";
}
?>
```

I=0 Val=Chuck
I=1 Val=SI664

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["pens"]["marker"];
```

Markers



Array Functions

PHP: Array Functions – Manual

php http://php.net/manual/en/ref.array.php

RSS Q php array functions

PHP downloads | documentation | faq | getting help | mailing lists | licenses | wiki | reporting bugs | php.net sites | links | conferences | my php.net

search for In the function list

«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

`array_key_exists($key, $ar)` - Returns TRUE if key is set in the array

`isset($ar['key'])` - Returns TRUE if key is set in the array

`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

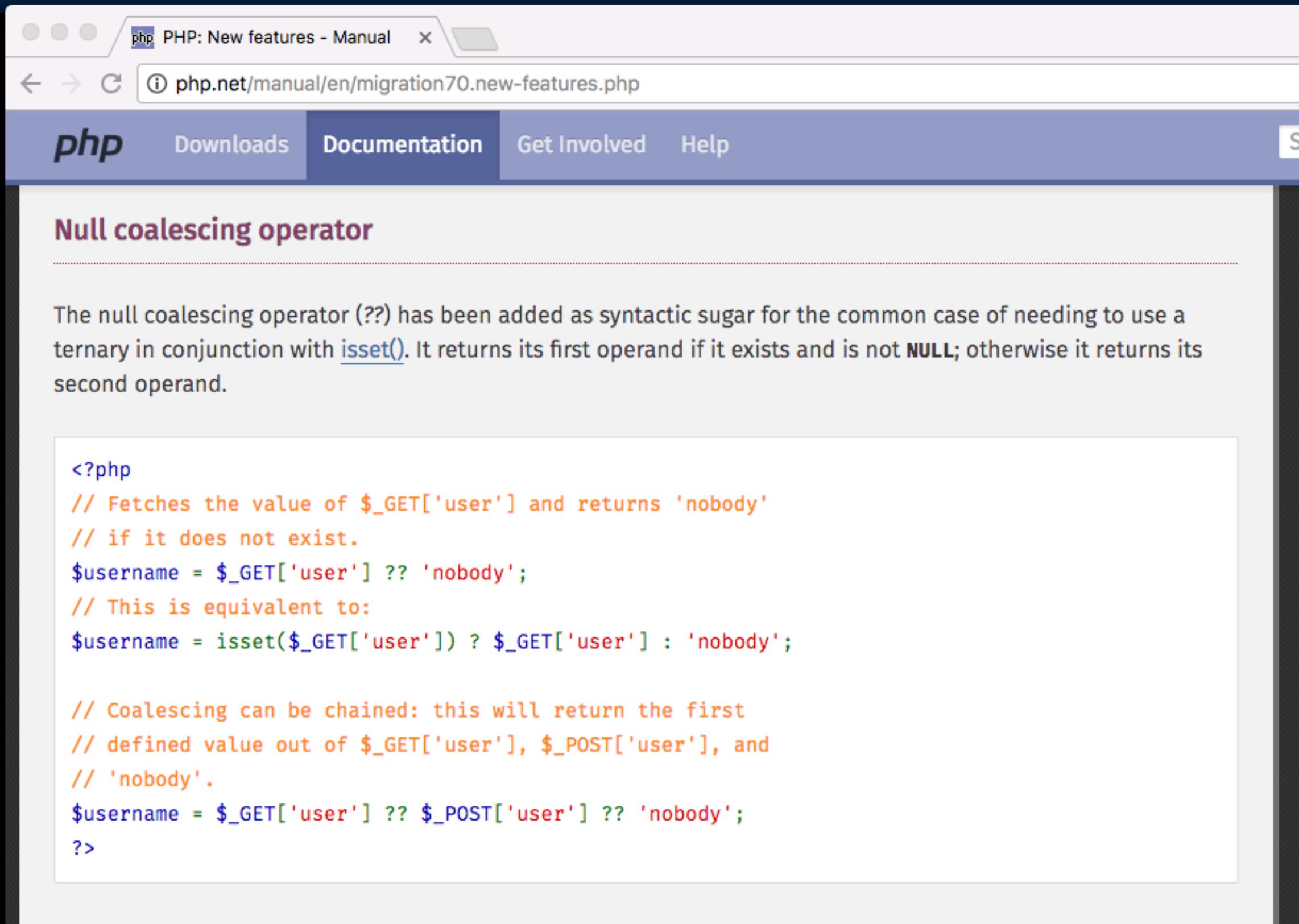
`shuffle($ar)` - Shuffles the array into random order

```
$za = array();  
$za["name"] = "Chuck";  
$za["course"] = "WA4E";
```

Course exists
name is set
addr is not set

```
if (array_key_exists('course', $za) ) {  
    echo("Course exists\n");  
} else {  
    echo("Course does not exist\n");  
}
```

```
echo isset($za['name']) ? "name is set\n" : "name is not set\n";  
echo isset($za['addr']) ? "addr is set\n" : "addr is not set\n";
```



The screenshot shows a web browser window displaying the PHP documentation. The title bar says "php PHP: New features - Manual". The address bar shows the URL "php.net/manual/en/migration70.new-features.php". The navigation bar includes links for "php", "Downloads", "Documentation" (which is the active tab), "Get Involved", and "Help". The main content area has a purple header "Null coalescing operator". Below it, a text block explains the operator's purpose: "The null coalescing operator (??) has been added as syntactic sugar for the common case of needing to use a ternary in conjunction with [isset\(\)](#). It returns its first operand if it exists and is not **NULL**; otherwise it returns its second operand." A code block below illustrates the usage of the operator:

```
<?php
// Fetches the value of $_GET['user'] and returns 'nobody'
// if it does not exist.
$username = $_GET['user'] ?? 'nobody';
// This is equivalent to:
$username = isset($_GET['user']) ? $_GET['user'] : 'nobody';

// Coalescing can be chained: this will return the first
// defined value out of $_GET['user'], $_POST['user'], and
// 'nobody'.
$username = $_GET['user'] ?? $_POST['user'] ?? 'nobody';
?>
```

```
<?php
$za = array();
$za[ "name" ] = "Chuck";
$za[ "course" ] = "WA4E";
```

```
// PHP >= 7.0.0 only
$name = $za[ 'name' ] ?? 'not found';
$addr = $za[ 'addr' ] ?? 'not found';

echo( "Name=$name\n" );
echo( "Addr=$addr\n" );
```

```
// PHP < 7.0.0 equivalent
$name = isset($za[ 'name' ]) ? $za[ 'name' ] : 'not found';
```

Null
Coalesce

Name=Chuck
Addr=not found

```
$za = array();
$za[ "name" ] = "Chuck";
$za[ "course" ] = "WA4E";
print "Count: " . count($za) . "\n";
if ( is_array($za) ) {
    echo '$za Is an array' . "\n";
} else {
    echo '$za Is not an array' . "\n";
}
```

Count: 2
\$za Is an array

```
$za = array();  
$za[ "name" ] = "Chuck";  
$za[ "course" ] = "WA4E";  
$za[ "topic" ] = "PHP";  
print_r($za);  
sort($za);  
print_r($za);  
  
Array(  
      [name] => Chuck  
      [course] => WA4E  
      [topic] => PHP  
    )  
Array(  
      [0] => Chuck  
      [1] => PHP  
      [2] => WA4E  
    )
```

```
$za = array();  
$za[ "name" ] = "Chuck";  
$za[ "course" ] = "WA4E";  
$za[ "topic" ] = "PHP";  
print_r($za);  
ksort($za);  
print_r($za);  
asort($za);  
print_r($za);  
  
Array(  
    [name] => Chuck  
    [course] => WA4E  
    [topic] => PHP  
)  
  
Array(  
    [course] => WA4E  
    [name] => Chuck  
    [topic] => PHP  
)  
  
Array(  
    [name] => Chuck  
    [topic] => PHP  
    [course] => WA4E  
)
```

Exploding Arrays

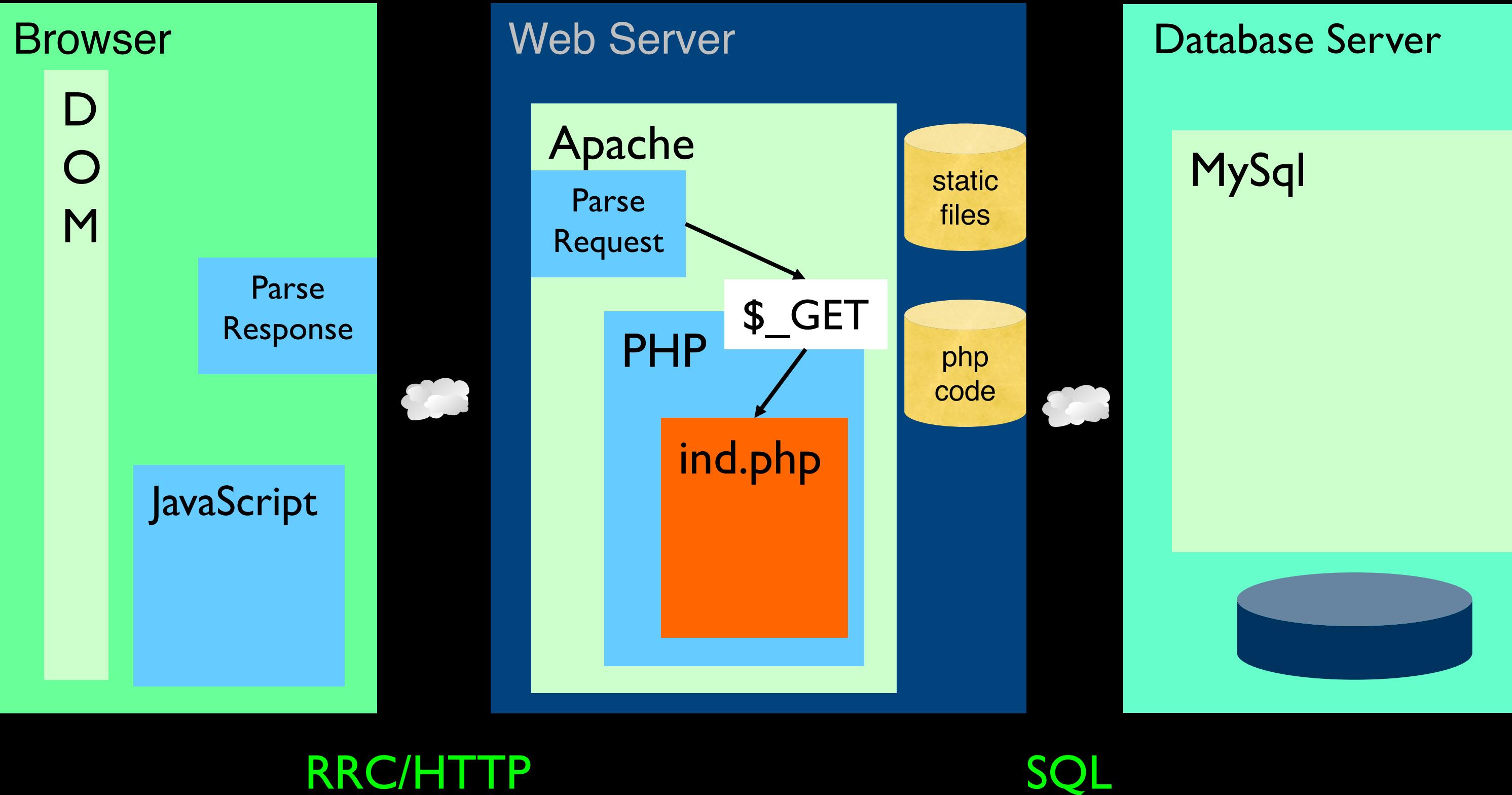
```
$inp = "This is a sentence with seven words";
$temp = explode(' ', $inp);
print_r($temp);
```

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

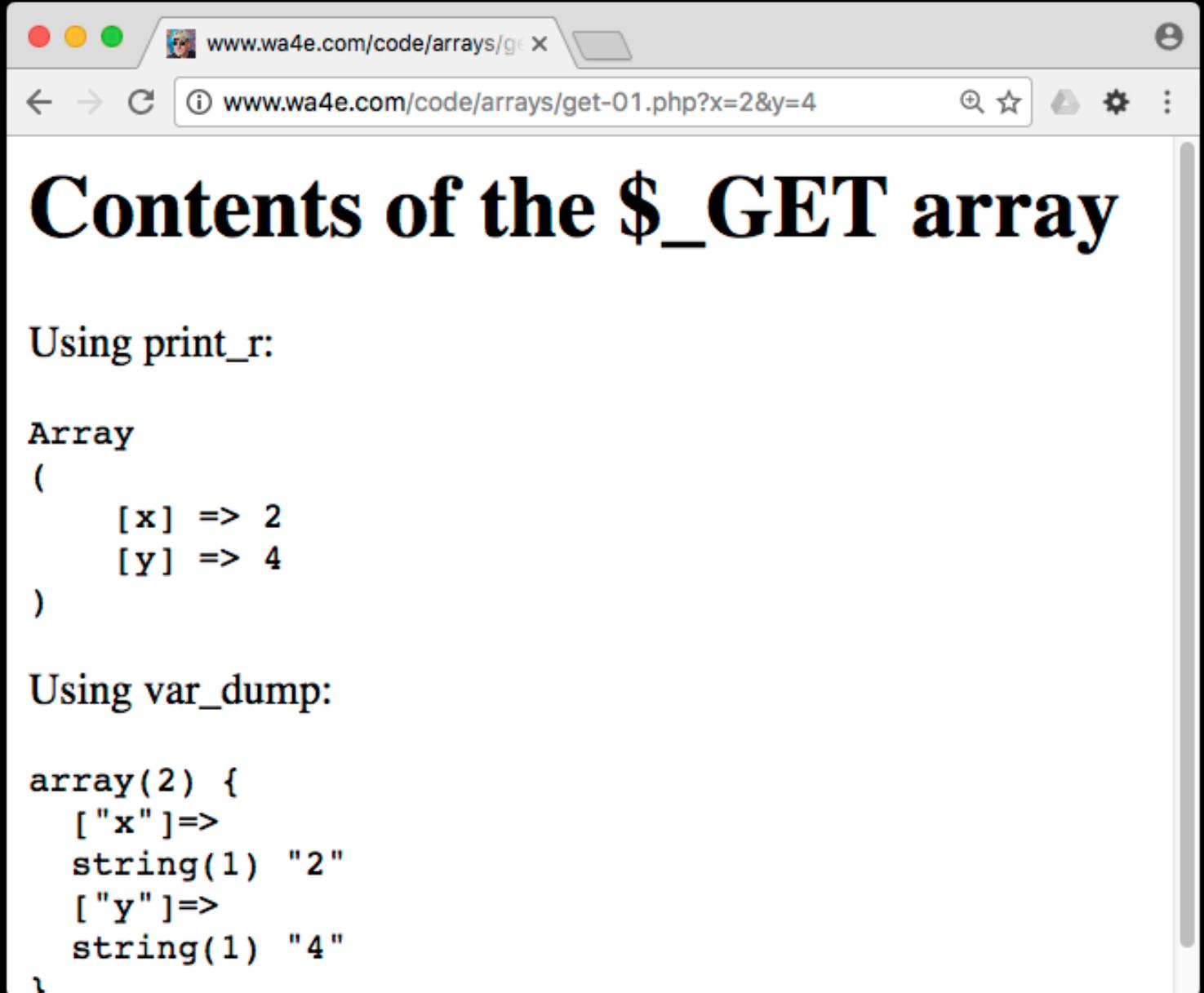


HTTP & PHP Arrays

Time



```
<h1>Contents of the $_GET array</h1>
<p>Using print_r:</p>
<pre>
<?php
    print_r($_GET);
?>
</pre>
<p>Using var_dump:</p>
<pre>
<?php
    var_dump($_GET);
?>
</pre>
```



<http://www.wa4e.com/code/arrays/get-01.php?x=2&y=4>



Summary

This is a sprint through some of the unique language features of PHP Arrays.

Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) as part of www.wa4e.com and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information

Insert new Contributors and Translators here including names and dates

Continue new Contributors and Translators here