

PHP Arrays

Unit 4

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...

Integer Indices

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

There

Key / Value

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

PHPIntro

Dumping an Array

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

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

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

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";  
$za["course"] = "PHPIntro";  
print_r($za);
```

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


Looping Through an Array

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

Key=name Val=Chuck

Key=course Val=PHPIntro

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 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
- `isset($ar['key'])` - Returns TRUE if key is set in the array
- `sort($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";  
$za["course"] = "PHPIntro";  
print "Count: " . count($za) . "\n";  
if ( is_array($za) ) {  
    echo '$za Is an array' . "\n";  
} else {  
    echo '$za Is not an array' . "\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";
```

Count: 2

\$za Is an array

name is set

addr is not set

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

```
Array(  
    [name] => Chuck  
    [course] => PHPIntro  
    [topic] => PHP  
)  
Array(  
    [0] => Chuck  
    [1] => PHP  
    [2] => PHPIntro  
)
```

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

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

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

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


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
)
```

HTTP Parameters and Arrays

PHP Global Variables

- Part of the goal of PHP is to make interacting with HTTP and HTML as easy as possible
- PHP processes the incoming HTTP Request based on the protocol specifications and drops the data into various **super global** variables (usually arrays)

Internet Engineering Task Force (IETF)
Request for Comments: 7230
Obsoletes: 2145, 2616
Updates: 2817, 2818
Category: Standards Track
ISSN: 2070-1721

R. Fielding, Ed.
Adobe
J. Reschke, Ed.
greenbytes
June 2014

Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing

Abstract

The Hypertext Transfer Protocol (HTTP) is a stateless application-level protocol for distributed, collaborative, hypertext information systems. This document provides an overview of HTTP architecture and its associated terminology, defines the "http" and "https" Uniform Resource Identifier (URI) schemes, defines the HTTP/1.1 message syntax and parsing requirements, and describes related security concerns for implementations.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in [Section 2 of RFC 5741](#).

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7230>.

5.4. Host

The "Host" header field in a request provides the host and port information from the target URI, enabling the origin server to distinguish among resources while servicing requests for multiple host names on a single IP address.

Host = uri-host [":" port] ; [Section 2.7.1](#)

A client MUST send a Host header field in all HTTP/1.1 request messages. If the target URI includes an authority component, then a client MUST send a field-value for Host that is identical to that authority component, excluding any userinfo subcomponent and its "@" delimiter ([Section 2.7.1](#)). If the authority component is missing or undefined for the target URI, then a client MUST send a Host header field with an empty field-value.

Since the Host field-value is critical information for handling a request, a user agent SHOULD generate Host as the first header field following the request-line.

For example, a GET request to the origin server for <<http://www.example.org/pub/WWW/>> would begin with:

```
GET /pub/WWW/ HTTP/1.1
Host: www.example.org
```

A client MUST send a Host header field in an HTTP/1.1 request even if the request-target is in the absolute-form, since this allows the Host information to be forwarded through ancient HTTP/1.0 proxies that might not have implemented Host.

<https://tools.ietf.org/html/rfc7230> (1 of several – 89 pages)

```
<h1>Contents of the $_GET array</h1>
```

```
<p>Using print_r:</p>
```

```
<pre>
```

```
<?php
```

```
    print_r($_GET);
```

```
?>
```

```
</pre>
```

```
<p>Using var_dump:</p>
```

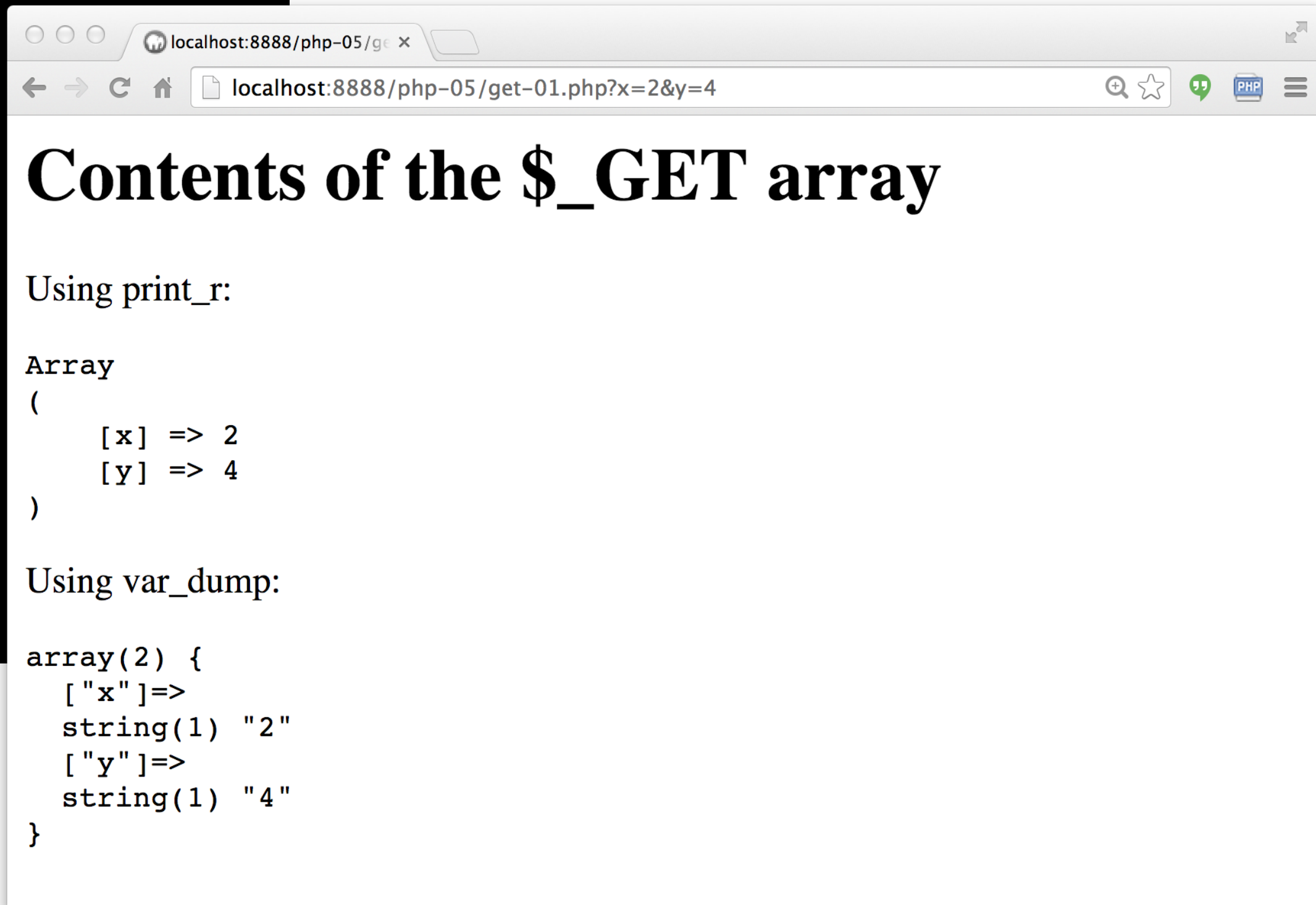
```
<pre>
```

```
<?php
```

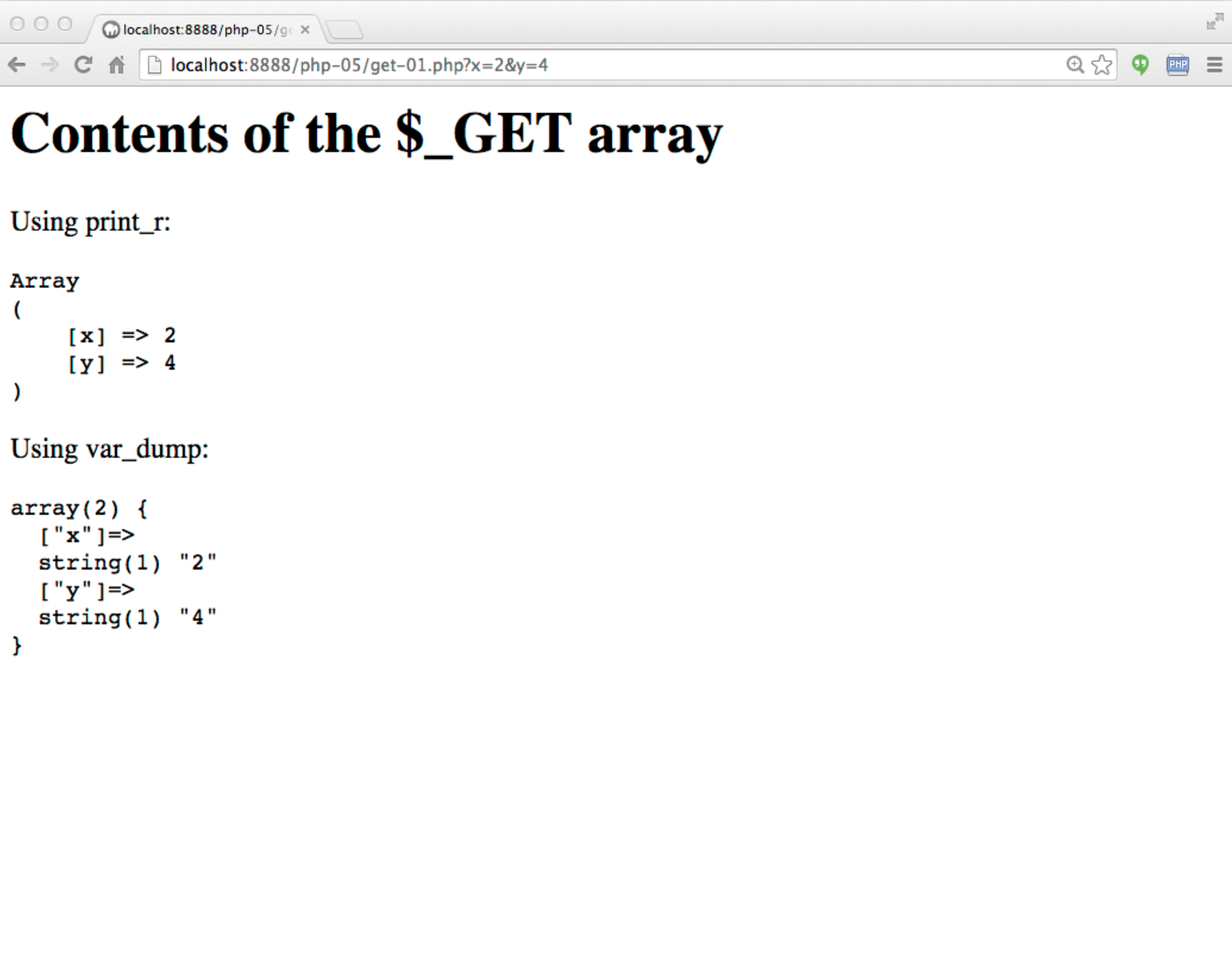
```
    var_dump($_GET);
```

```
?>
```

```
</pre>
```



<http://www.php-intro.com/code/php-05/get-01.php>



Contents of the \$_GET array

Using print_r:

Array

```
(  
    [x] => 2  
    [y] => 4  
)
```

Using var_dump:

The screenshot shows a web browser window with the address bar displaying `localhost:8888/php-05/get-01.php?x=2&y=4`. The page content displays the output of `print_r($_GET)` as an array with two elements: `[x] => 2` and `[y] => 4`. Below the page content, the Chrome DevTools Network tab is open, showing a single request to `get-01.php?x=2&y=4`. The request details are expanded, showing the following information:

- Remote Address:** `:::1:8888`
- Request URL:** `http://localhost:8888/php-05/get-01.php?x=2&y=4`
- Request Method:** `GET`
- Status Code:** `200 OK`
- Request Headers:**
 - `Accept:` `text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8`
 - `Accept-Encoding:` `gzip,deflate,sdch`
 - `Accept-Language:` `en-US,en;q=0.8,es-419;q=0.6,es;q=0.4`
 - `Cache-Control:` `no-cache`
 - `Connection:` `keep-alive`
 - `Cookie:` `SQLiteManager_currentLangue=2`
 - `Host:` `localhost:8888`
 - `Pragma:` `no-cache`
 - `User-Agent:` `Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.94 Safari/537.36`
- Query String Parameters:**
 - `x:` `2`
 - `y:` `4`
- Response Headers:** (collapsed)

The bottom status bar of DevTools indicates: `1 requests | 545 B transferred | 5 ms (load: 47 m...)`. The bottom navigation bar includes links for `Console`, `Search`, `Emulation`, and `Rendering`.

Contents of the \$_GET array

Using print_r:

Array

```
(  
    [x] => 2  
    [y] => 4  
)
```

Using var_dump:

Elements Network Sources Timeline Profiles Resources Audits Console

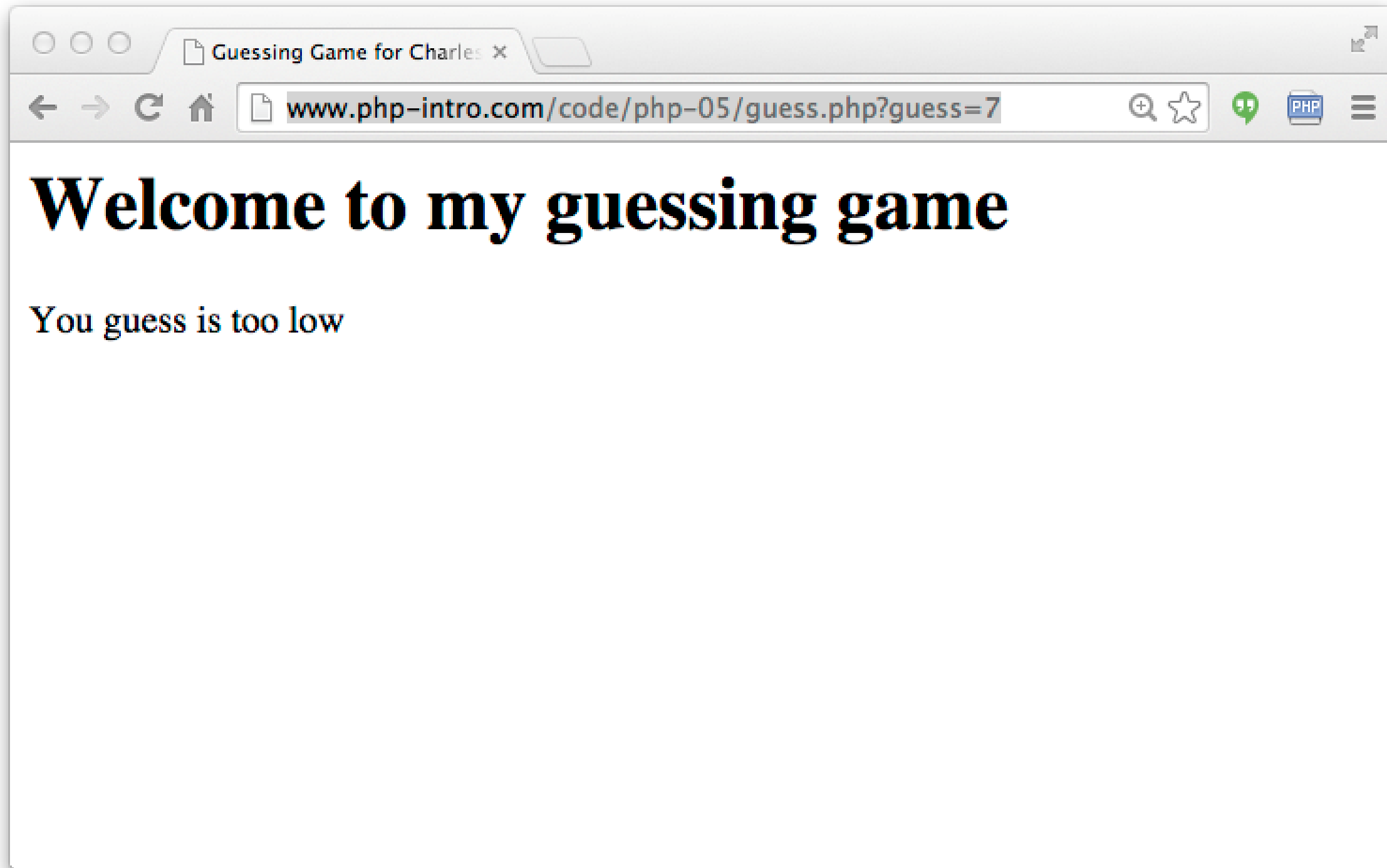
get-01.php?x=2&y=4
/php-05

Headers Preview Response Cookies Timing

```
1 <h1>Contents of the $_GET array</h1>  
2 <p>Using print_r:</p>  
3 <pre>  
4 Array  
5 (  
6     [x] => 2  
7     [y] => 4  
8 )  
9 </pre>  
10 <p>Using var_dump:</p>  
11 <pre>  
12 array(2) {  
13     ["x"]=>  
14     string(1) "2"  
15     ["y"]=>  
16     string(1) "4"  
17 }  
18 </pre>  
19  
20
```

1 requests | 545 B transferred | 5 ms (load: 47 m...)

Console Search Emulation Rendering



<http://www.php-intro.com/code/php-05/guess.php?guess=7>

```
<html>
<head>
<title>Guessing Game for Charles Severance</title>
</head>
<body>
<h1>Welcome to my guessing game</h1>
<p>
<?php
    if ( ! isset($_GET['guess']) ) {
        echo("Missing guess parameter");
    } else if ( $_GET['guess'] == 0 ) {
        echo("Your guess is not valid");
    } else if ( $_GET['guess'] < 42 ) {
        echo("You guess is too low");
    } else if ( $_GET['guess'] > 42 ) {
        echo("You guess is too high");
    } else {
        echo("Congratulations - You are right");
    }
?>
</p>
</body>
</html>
```



<http://www.php-intro.com/code/php-05/guess.php?guess=200>

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
- PHP arrays maintain order and there are many options for sorting
- We can use `explode()` to split a string into an array of strings
- HTTP Information is pre-processed and placed in super global arrays

Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) as part of www.php-intro.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