

Introduction to PHP (Hypertext PreProcessor, HyperText Processor)

Part 3

PHP Arrays and Superglobals

References

- Chapter 8 Introduction to Server-Side Development with PHP of the Text Book Entitled Fundamentals of Web Development, by Randy Connolly and Ricardo Hoar
- PHP 5.6.14 Released Oct. 1, 2015
- PHP 5.6.1 Released Oct. 2, 2014, <http://php.net/>
- PPH Documentation, <http://php.net/urlhowto.php>
- PHP Language Reference, <http://php.net/manual/en/langref.php>

Predefined Superglobals, <http://php.net/manual/en/language.variables.superglobals.php>

Superglobals // Built-in variables that are always available in all scopes

- `$GLOBALS` // All variables available in global scope (array containing info about the request and the server)
- `$_SERVER` // Server and execution environment info
- `$_GET` // HTTP GET variables (array of query string data passed to the server via the URL)
- `$_POST` // HTTP POST variables (array of query string data passed to the server via the HTTP header)
- `$_FILES` // HTTP file upload variables (array of file items uploaded to the server)
- `$_REQUEST` // HTTP request variables (array containing the contents of `$_GET`, `$_POST`, and `$_COOKIE`)
- `$_SESSION` // Session variables (array contains session data)
- `$_ENV` // Environment variables (array) of a server
- `$_COOKIEs` // Array of Cookie data passed to page via HTTP request

HTTP Cookies, <http://php.net/manual/en/features.cookies.php>

- PHP Supports for HTTP Cookies
- Set the session cookie parameters, <http://php.net/manual/en/function.session-set-cookie-params.php>

`$_SERVER`, <http://php.net/manual/en/reserved.variables.server.php>

- An array containing information such as Headers, Paths, Script Location
- There is no guarantee that every web server will provide any of these info
- See CGI/1.1 Specification, <http://www.faqs.org/rfcs/rfc3875.html>

Example 1. A PHP Page that uses print Server retained info in the \$_SERVER

Also use date(), <http://php.net/manual/en/function.date.php>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
<meta content="en-us" http-equiv="Content-Language" />
<title>This HTML page prints out your b</title>
</head>
<body>
<p>This HTML page prints out your browser info and the IP addresses of the HTTP
Request. </p>
Your Computer Name: <?php echo $_SERVER['HTTP_HOST'] ?><br />
Your Browser is : <?php echo $_SERVER['HTTP_USER_AGENT'] ?><br />
Your IP address : <?php echo $_SERVER['REMOTE_ADDR'] ?> <br/>
Your IP address : <?php echo $_SERVER['REMOTE_PORT'] ?> <br/>
Server Name: : <?php echo $_SERVER['SERVER_NAME'] ?> <br/>
Server Protocol : <?php echo $_SERVER['SERVER_PROTOCOL'] ?><br />
Server Software : <?php echo $_SERVER['SERVER_SOFTWARE'] ?><br />
Request Method : <?php echo $_SERVER['REQUEST_METHOD'] ?><br />
Request Time : <?php echo $_SERVER['REQUEST_TIME'] ?><br />
Request Date: <?php echo date("D:d:M:Y:H:i:s") ?>;</body>
</html>
```

Example 2: A PHP page that shows the contents kept in \$_COOKIE from the server that process your form data.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- readYourCookies.php -->
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
<meta content="en-us" http-equiv="Content-Language" />
<title>Displaying the Cookie Contents</title>
</head>

<body>

<p>This PHP page prints out your Cookie Contents. </p>
<?php
    setcookie("color", "red");
    echo $_COOKIE["color"];
```

```
?>
</body>
</html>
```

Example 3. A PHP Page that shows Server and browser info in the \$_SERVER.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- PHPbrowser-server-InfoAll.php -->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
<meta content="en-us" http-equiv="Content-Language" />
<title>This HTML page prints out your b</title>
</head>
<body>
<p>This HTML page prints out your browser info and the IP addresses of the HTTP
Request. </p>
<p>
<?php
echo "Your Computer Name: " . $_SERVER['HTTP_HOST'];
echo "<br/>HTTP User Agent: " . $_SERVER['HTTP_USER_AGENT'];
echo "<br/>Your Browser is: " . $_SERVER['HTTP_USER_AGENT'];
echo "<br/>Your IP address: " . $_SERVER['REMOTE_ADDR'];
echo "<br/>Your Port address: " . $_SERVER['REMOTE_PORT'];
echo "<br/>Request Method: " . $_SERVER['REQUEST_METHOD'];
echo "<br/>Request Time: " . $_SERVER['REQUEST_TIME'];
echo "<br/>Server Name: " . $_SERVER['SERVER_NAME'];
echo "<br/>HTTP Host: " . $_SERVER['HTTP_HOST'];
echo "<br/>Server IP address " . $_SERVER['SERVER_ADDR'];
echo "<br/>Server Port " . $_SERVER['SERVER_PORT'];
echo "<br/>Server Protocol: " . $_SERVER['SERVER_PROTOCOL'];
echo "<br/>HTTP Connection: " . $_SERVER['HTTP_CONNECTION'];
echo "<br/>HTTP Accept Encoding" . $_SERVER['HTTP_ACCEPT_ENCODING'];
echo "<br/>Server Software: " . $_SERVER['SERVER_SOFTWARE'];
?>
<?php echo "<br/>Request Date: " . date("D:d:M:Y:H:i:s"); ?>
</p>
</body>
</html>
```

Example 4. Handling the File Upload

- Using a HTML form for upload:
 - o POST method
- PHP File Upload

- o Configure php.ini
 - file_uploads = on
 - upload_file_maxsize
 - post_max_size
 - memory_limit
 - max_execution_time
 - max_input_time
- o POST method, <http://php.net/manual/en/features.file-upload.post-method.php>
- o \$_FILES array, keys
 - name – full file name
 - type – MIME (Multipurpose Internet Mail Extension) type of the file
 - application, audio, image, message, multipart, text, video
 - tmp_name – full path to the location on your server
 - error
 - size
- o Figure 9.12 Example (page 387), keys - <http://php.net/manual/en/reserved.variables.files.php>
 - echo \$_FILES["file1"]["name"]; // "sample1.png"
 - echo \$_FILES["file1"]["type"]; // "image/png"
 - echo \$_FILES["file1"]["tmp_name"]; // tmp/phpJ08pVh
 - echo \$_FILES["file1"]["error"]; // 0
 - echo \$_FILES["file1"]["size"]; // 1219038
- References
 - o Handling File Uploads, <http://php.net/manual/en/features.file-upload.php>
 - o http://www.w3schools.com/php/php_file_upload.asp
 - o The Content-Type Header Field, http://www.w3.org/Protocols/rfc1341/4_Content-Type.html
 - o Media Types, 2015-10-12, <http://www.iana.org/assignments/media-types/media-types.xhtml>

```

<html>
<title> File Upload </title>
<head> </head>
<body>
<form encrypt='multipart/form-data' method='post'>
  <input type='file' name='thisFile' />
  <input type='submit' />
</form>
</body>
</html>

```

PHP Reading/Writing Files Support

- o Stream Access
 - o Read just a small portion of the file at a time
 - o Most memory-efficient approach, but I/O intensive

- o In-Memory File Access
 - o Read entire file into memory (PHP variable)
 - o Easier for file processing

Listing 9.19

```
<?php
$f = fopen("sample.txt", "r");
$ln = 0;
while ($line = fgets($f)) {
    $ln++;
    printf("%2d: ", $ln);
    echo $line . "<br>";
}
fclose($f);
?>
```

sample.txt

```
1: 01070,Picasso,The Actor,1904
2: 01080,Picasso,Family of Saltimbanques,1905
3: 02070,Matisse,The Red Madras Headdress,1907
4: 05010,David,The Oath of the Horatii,1784
```

Listing 9.20

PHP functions used

- o array file(string \$filename), read entire file into an array,
<http://php.net/manual/en/function.file.php>
- o array explode(string \$delimiter, string \$string), split a string by string,
<http://php.net/manual/en/function.explode.php>

```
<?php
```

```
$filename="sample.txt";
```

```
// read the file into memory; if there is an error then stop processing
$paintings = file($filename) or die('ERROR: Cannot find file');
```

```
// our data is comma-delimited
$delimiter = ',';
```

```
// loop through each line of the file
foreach ($paintings as $painting) {
    // returns an array of strings where each element in the array
    // corresponds to each substring between the delimiters
```

```

    $paintingFields = explode($delimiter, $painting);
    $id= $paintingFields[0];
    $artist = $paintingFields[1];
    $title = $paintingFields[2];
    $year = $paintingFields[3];

    // do something with this data
    //Not in book listing - but here for demo.
    echo "Painting (#$id) $title by $artist in $year<br>";
}

?>

```

PHP Functions Reference

- **echo()**
- **unset()** // Destroy variables
- **func_get_args()** // get function arguments
- **func_num_args()** // get function argument count

String Manipulation Functions

- **strcmp()** // string comparison, case sensitive
- **strcasecmp()** // string comparison, non case sensitive
- **substr()** // the sub string
- **strlen()** // The number of chars in the string
- **strpos()** // The character position
- **chop()** // Remove all white spaces from its ends
- **trim()** // remove all white spaces from both ends
- **strtolower()** // covert to lower case characters
- **strtoupper()** // convert to upper case characters

Arithmetic Functions

- **floor(), ceil(), round(), srand(), rand(), abs(), min(), max()**

Output Functions

- **print()**
- **printf()** // formatted output

PHP Functions

- **func_get_args()** // <http://us3.php.net/manual/en/function.func-get-args.php>
- **func_num_args()**

<!DOCTYPE html>

<!--func_arguments.php - A trivial example to illustrate a php document -->

<html lang = "en">

```
<head>
  <title> funct_arguments.php </title>
<meta charset = "utf-8" />
</head>
<body>
<?php
  function func_dynamic() {
    echo "ITC 250/CPET 499 Web Systems: ".func_num_args(). "Number of arguments.<br>";
    $args = func_get_args();
    for($i = 0; $i < count($args); $i++){
      echo "Passed arguments: {args[$i]} <br>";
    }
  }
  func_dynamic(5, 4, 3, 2, 1 );
?>
</body>
</html>
```