

PHP Arrays and Superglobals

Chapter 12

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development

© 2017 Pearson
<http://www.funwebdev.com>

Chapter 12

1

Arrays

2

\$_GET and
\$_POST
Superglobal Arrays

3

\$_SERVER Array

4

\$_FILES Array

5

Reading/
Writing Files

6

Summary

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Chapter 12

1 Arrays

2 \$_GET and
\$_POST
Superglobal Arrays

3 \$_SERVER Array

4 \$_FILES Array

5 Reading/
Writing Files

6 Summary

Randy Connolly and Ricardo Hoar

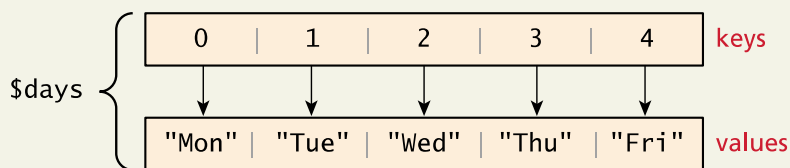
Fundamentals of Web Development - 2nd Ed.

Arrays

Defining and Accessing an Array

```
$days = array("Mon","Tue","Wed","Thu","Fri");
```

```
$days = ["Mon","Tue","Wed","Thu","Fri"]; // alternate syntax
```



Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Arrays

Defining and Accessing an Array

All arrays in PHP are generally referred to as **associative arrays**

```
$days = array(0 => "Mon", 1 => "Tue", 2 => "Wed", 3 => "Thu", 4 => "Fri");
```

Diagram illustrating the structure of the array:
 - **key**: 0, 1, 2, 3, 4 (pointing to the first part of each pair)
 - **value**: "Mon", "Tue", "Wed", "Thu", "Fri" (pointing to the second part of each pair)

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

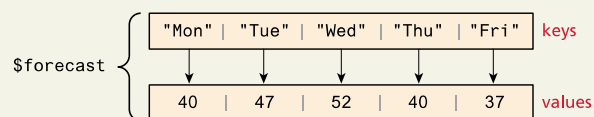
Arrays

Defining and Accessing an Array

You can use integer and string keys, not necessarily in order.

```
$forecast = array("Mon" => 40, "Tue" => 47, "Wed" => 52, "Thu" => 40, "Fri" => 37);
```

Diagram illustrating the structure of the array:
 - **key**: "Mon", "Tue", "Wed", "Thu", "Fri" (pointing to the first part of each pair)
 - **value**: 40, 47, 52, 40, 37 (pointing to the second part of each pair)



```
echo $forecast["Tue"]; // outputs 47
echo $forecast["Thu"]; // outputs 40
```

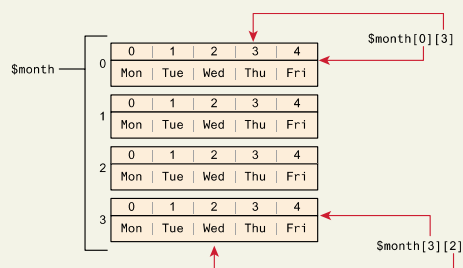
Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Arrays

Multidimensional Arrays

```
$month = array
(
    array("Mon","Tue","Wed","Thu","Fri"),
    array("Mon","Tue","Wed","Thu","Fri"),
    array("Mon","Tue","Wed","Thu","Fri"),
    array("Mon","Tue","Wed","Thu","Fri")
);
```



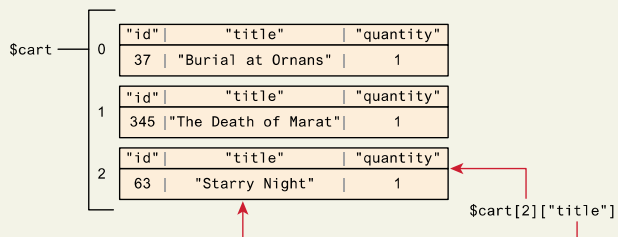
Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Arrays

Multidimensional Arrays

```
$cart = array();
$cart[] = array("id" => 37, "title" => "Burial at Ornans", "quantity" => 1);
$cart[] = array("id" => 345, "title" => "The Death of Marat", "quantity" => 1);
$cart[] = array("id" => 63, "title" => "Starry Night", "quantity" => 1);
```



Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Arrays

Iterating through an Array - while

```
// while loop
$i=0;
while ($i < count($days)) {
    echo $days[$i] . "<br>";
    $i++;
}
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Arrays

Iterating through an Array - do while

```
// do while loop
$i=0;
do {
    echo $days[$i] . "<br>";
    $i++;
} while ($i < count($days));
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Arrays

Iterating through an Array - for

```
// for loop  
for ($i=0; $i<count($days); $i++) {  
    echo $days[$i] . "<br>";  
}
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Arrays

Iterating through an Array - foreach

```
// foreach: iterating through the values  
foreach ($forecast as $value) {  
    echo $value . "<br>";  
}  
  
// foreach: iterating through the values AND the keys  
foreach ($forecast as $key => $value) {  
    echo "day[" . $key . "]=" . $value;  
}
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Arrays

Adding and Deleting Elements

An element can be added to an array simply by using a key/index that hasn't been used, as shown below:

```
$days[5]= "Sat";
```

As an alternative to specifying the index, a new element can be added to the end of any array using empty square brackets after the array name, as follows:

```
$days[]= "Sun";
```

Delete with unset()

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Arrays

Array Sorting

```
sort($days);
```

As the values are all strings, the resulting array would be:

```
Array ([0] => Fri [1] => Mon [2] => Sat [3] => Sun [4] => Thu  
[5] => Tue [6] => Wed)
```

```
asort($days);
```

The resulting array in this case keeps associations so is:

```
Array ([4] => Fri [0] => Mon [5] => Sat [6] => Sun [3] => Thu  
[1] => Tue [2] => Wed)
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Arrays

More Array Operations – pp. 546-547

- `array_keys($someArray)` – returns an indexed array with the values being the key of `$someArray`.
- `array_values($someArray)` - returns an indexed array with the values being the values of `$someArray`.
- `array_rand($someArray, $num=1)` – returns an many random keys as are requested.
- `array_reverse($someArray)` - returns `$someArray` array in reverse order.
- `array_walk($someArray, $callback, $optionalParam)` – call a method for each value in `$someArray`.
- `in_array($needle, $haystack, $optionalStrict)` – search `$haystack` array for a value (`%needle`), returns true if found
- `shuffle($someArray)` – removes any existing keys and `$someArray` will be an indexed array.

Arrays

Superglobal Arrays

PHP uses special predefined associative arrays called **superglobal variables** that allow the programmer to easily access **HTTP headers**, **query string parameters**, and other **commonly needed information**.

Arrays

Superglobal Arrays – Table 12.1, page 547

- `$GLOBALS` Array for storing data that needs superglobal scope
- `$_COOKIE` Array of cookie data passed to page via HTTP request
- `$_ENV` Array of server environment data
- `$_FILES` Array of file items uploaded to the server
- `$_GET` Array of query string data passed to the server via the URL
- `$_POST` Array of query string data passed to the server via the HTTP header
- `$_REQUEST` Array containing the contents of `$_GET`, `$_POST`, and `$_COOKIE`
- `$_SESSION` Array that contains session data
- `$_SERVER` Array containing information about the request and the server

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Chapter 12

1 Arrays

2 `$_GET` and
`$_POST`
Superglobal Arrays

3 `$_SERVER` Array

4 `$_FILES` Array

5 Reading/
Writing Files

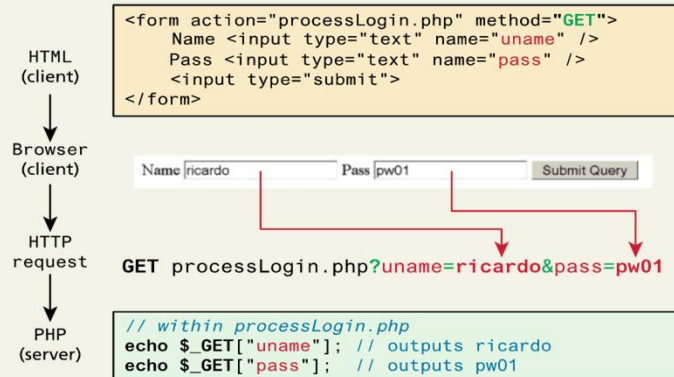
6 Summary

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$_GET and \$_POST Superglobal Arrays

Relating sent query string elements in PHP, Figure 12.5, page 548

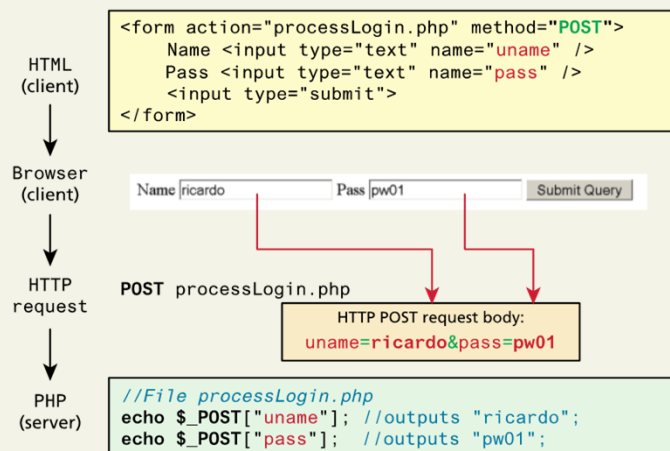


Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$_GET and \$_POST Superglobal Arrays

Relating sent query string elements in PHP (POST), Figure 12.6, page 549



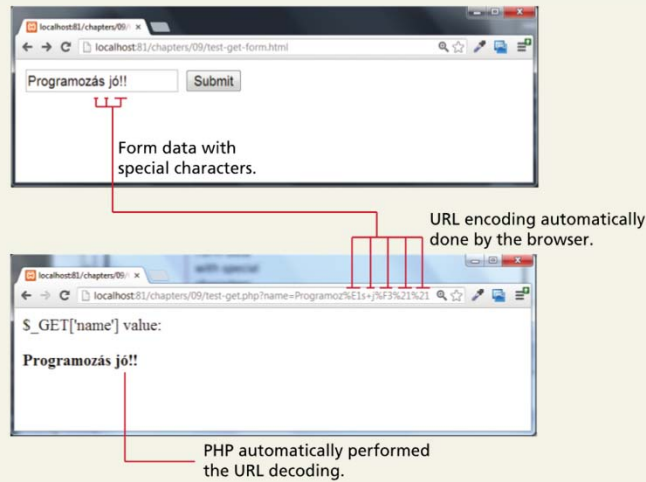
Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$_GET and \$_POST Superglobal Arrays

Figure 12.2

Note URL encoding and Decoding: urlencode(), urldecode()

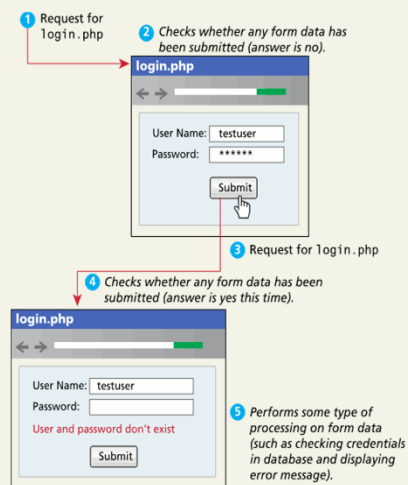


Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$_GET and \$_POST Superglobal Arrays

Figure 12.8 Form display and processing on same page, page 550



Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$_GET and \$_POST Superglobal Arrays

Determining If Any Data Sent
Listing 12.6, page 551 (incomplete code)

use the **isset()** function in PHP to see if there is any value set for a particular expected key

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if ( isset($_POST["uname"]) && isset($_POST["pass"]) ) {
        // handle the posted data.
```

\$_GET and \$_POST Superglobal Arrays

null coalescing operator, page 552

```
$username = isset($_GET['uname']) ? $_GET['uname'] : 'nobody';
```

Becomes


```
$username = $_GET['uname'] ?? 'nobody';
```

\$_GET and \$_POST Superglobal Arrays

Accessing Form Array Data, pp. 552-553
Listing 12.8

Monday <input type="checkbox" name="day[]" value="Monday">

Tuesday <input type="checkbox" name="day[]" value="Tuesday">



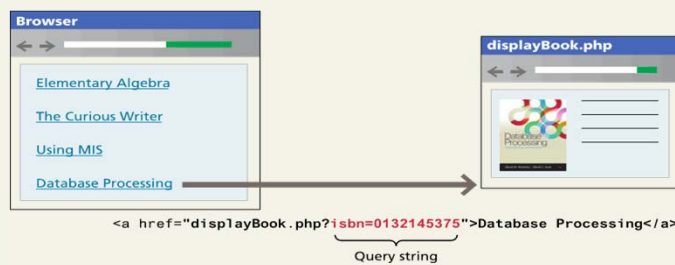
```
<?php
    echo "You submitted " . count($_GET['day']) .
    " values";
    foreach ($_GET['day'] as $d) {
        echo $d . " <br>";
    }
?>
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$_GET and \$_POST Superglobal Arrays

Using Query Strings in Hyperlinks, Figure 12.10, page 554



Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$_GET and \$_POST Superglobal Arrays

Sanitizing Query Strings, page 554-555

That is, just because you are expecting a proper query string, it doesn't mean that you are going to get one. your program must be able to handle:

- If query string parameter **doesn't exist**.
- If query string parameter **doesn't contain a value**.
- If query string parameter **value** isn't the correct type or is out of acceptable range.
- If value is required for a **database lookup**, but provided value doesn't exist in the database table.

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Chapter 12

1 Arrays

2 \$_GET and
\$_POST
Superglobal Arrays

3 \$_SERVER Array

4 \$_FILES Array

5 Reading/
Writing Files

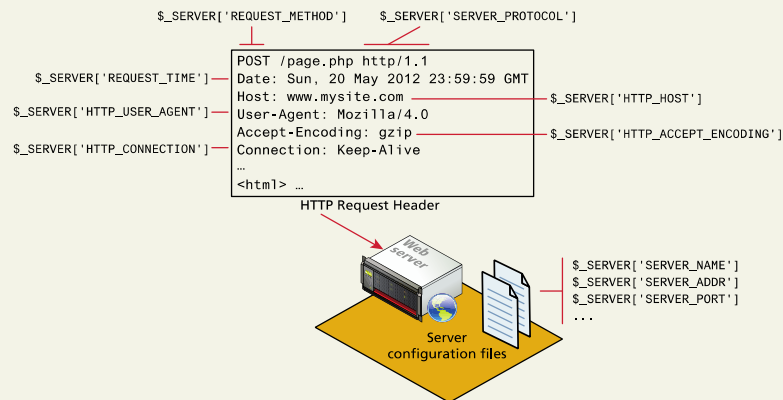
6 Summary

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$_SERVER Array

Server Information Keys, Figure 12.11, page 559



Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$_SERVER Array

Request Header Information Keys, Listing 12.10, page 560

```
<?php
echo $_SERVER['HTTP_USER_AGENT'];

//advanced browser detection
$browser = get_browser($_SERVER['HTTP_USER_AGENT'], true);
print_r($browser);
?>
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

`$_SERVER` Array

Request Header Information Keys, Listing 12.11, page 561

```
$previousPage = $_SERVER['HTTP_REFERER'];  
  
// Check to see if referer was our search page  
if (strpos($previousPage,"search.php") != 0) {  
    echo "<a href='search.php'>Back to search</a>";  
}  
  
// Rest of HTML output
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Chapter 12

1 Arrays

2 `$_GET` and
`$_POST`
Superglobal Arrays

3 `$_SERVER` Array

4 `$_FILES` Array

5 Reading/
Writing Files

6 Summary

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$ _FILES Array

HTML Required for File Uploads, page 562, Listing 12.12

- First, you must ensure that the HTML form uses the HTTP POST method
- Second, you must add the enctype="multipart/form-data" attribute to the HTML form that is performing the upload
- Finally you must include an input type of file in your form.

```
<form enctype='multipart/form-data' method='post'>
```

```
    <input type='file' name='file1' id='file1'>
```

```
    <input type='submit' value='Submit Query'>
```

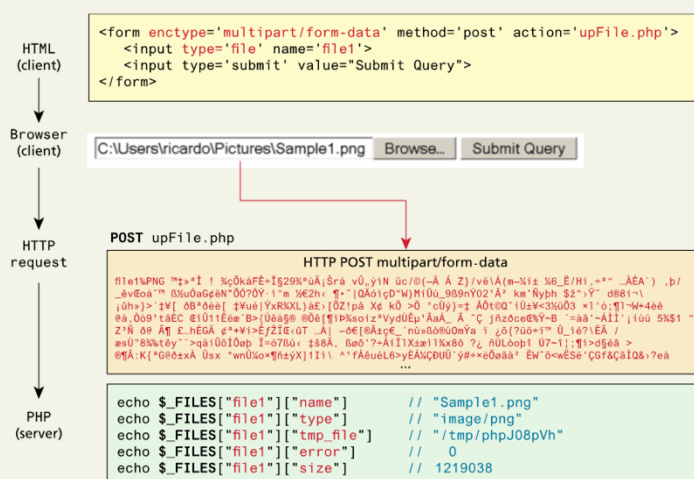
```
</form>
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$ _FILES Array

Handling the File Upload in PHP, Figure 12.12, page 563



Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

`$_FILES` Array

Checking for Errors, Listing 12.13, page 565

Table 12.2 Error Codes in PHP for File Upload, page 564

```
foreach ($_FILES as $fileKey => $fileArray) {  
    if ($fileArray["error"] != UPLOAD_ERR_OK) { // error  
        echo "Error: " . $fileKey . " has error" .  
        $fileArray["error"] . "<br>";  
    }  
    else { // no error  
        echo $fileKey . "Uploaded successfully ";  
    }  
}
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

`$_FILES` Array

File Size Restrictions, page 565

You can limit in multiple ways

- HTML form attributes in inputs (browser)
- JavaScript (browser)
- Php validation (server)

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$ _FILES Array

Limiting the Type of File Upload, Listing 12.17, page 567

```
$validExt = array("jpg", "png");
$validMime = array("image/jpeg", "image/png");
foreach($_FILES as $fileKey => $fileArray ){
    $extension = end(explode(".", $fileArray["name"]));
    if (in_array($fileArray["type"],$validMime) && in_array($extension, $validExt)) {
        echo "All is well. Extension and mime types valid";
    }
    else {
        echo $fileKey." has an invalid mime type or extension";
    }
}
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

\$ _FILES Array

Moving the File, Listing 12.1, page 568

```
$fileToMove = $_FILES['file1']['tmp_name'];
$destination = "./upload/" . $_FILES["file1"]["name"];
if (move_uploaded_file($fileToMove,$destination)) {
    echo "The file was uploaded and moved successfully!";
}
else {
    echo "There was a problem moving the file.";
}
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Chapter 12

1

Arrays

2\$_GET and
\$_POST
Superglobal Arrays**3**

\$_SERVER Array

4

\$_FILES Array

5Reading/
Writing Files**6**

Summary

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Reading/Writing Files

Two ways, page 568

There are two basic techniques for read/writing files in PHP:

- **Stream access**. Our code will read just a small portion of the file at a time.
- **All-In-Memory access**. In this technique, we can read the entire file into memory (i.e., into a PHP variable). While not appropriate for large files, it does make processing of the file extremely easy.

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Reading/Writing Files

Stream Access

To those of you familiar with functions like `fopen()` , `fclose()` , and `fgets()` from the C programming language, this first technique will be familiar

- Open `fopen()`
- Read data `fgets()`
- Close the file `fclose()`

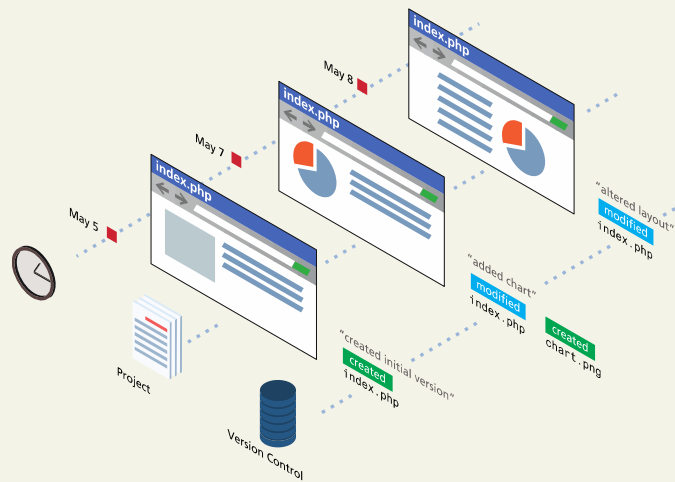
Reading/Writing Files

In-Memory File Access

- `file()` Reads the entire file and returns an array, with each array element corresponding to one line in the file.
- `file_get_contents()` Reads the entire file and returns a string variable.
- `file_put_contents()` Writes the contents of a string variable out to a file.

Version Control

Keep track of changes

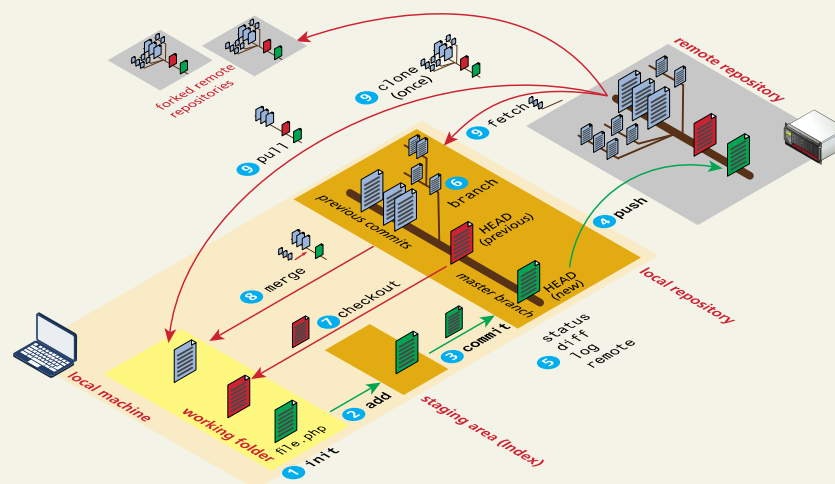


Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Version Control

Widely used in industry



Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2nd Ed.

Chapter 12

1

Arrays

2

 \$_GET and
\$_POST
Superglobal Arrays

3

\$_SERVER Array

4

\$_FILES Array

5

 Reading/
Writing Files

6

Summary

Randy Connolly and Ricardo Hoar

 Fundamentals of Web Development - 2nd Ed.

Summary

Key Terms

All-in-memory access	GitHub	remote repository
array keys	local repository	sanitizing user inputs
array values	merge	stream access
associative arrays	NULL	stream resource
branch	null coalescing operator	superglobal variables
forking	one-way hash	user-agent
Git	ordered map	version control

Randy Connolly and Ricardo Hoar

 Fundamentals of Web Development - 2nd Ed.

Summary

Questions?