PHP Arrays and Superglobals

Chapter 12

Randy Connolly and Ricardo Hoar

Chapter 12

1. Arrays
2. $_GET and $_POST Superglobal Arrays
3. $_SERVER Array
4. $_FILES Array
5. Reading/ Writing Files
6. Summary

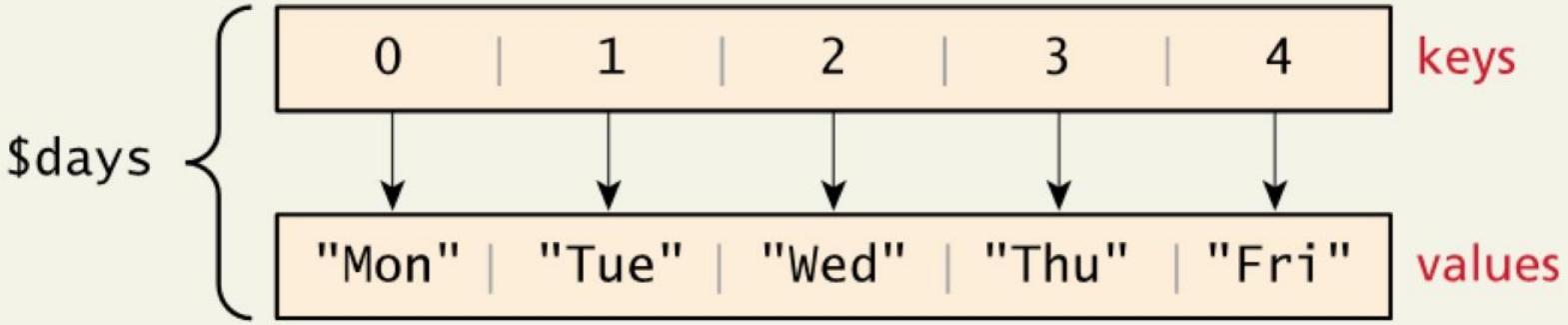
Chapter 12

1. Arrays
2. $_GET and $_POST Superglobal Arrays
3. $_SERVER Array
4. $_FILES Array
5. Reading/ Writing Files
6. Summary

Arrays

Defining and Accessing an Array

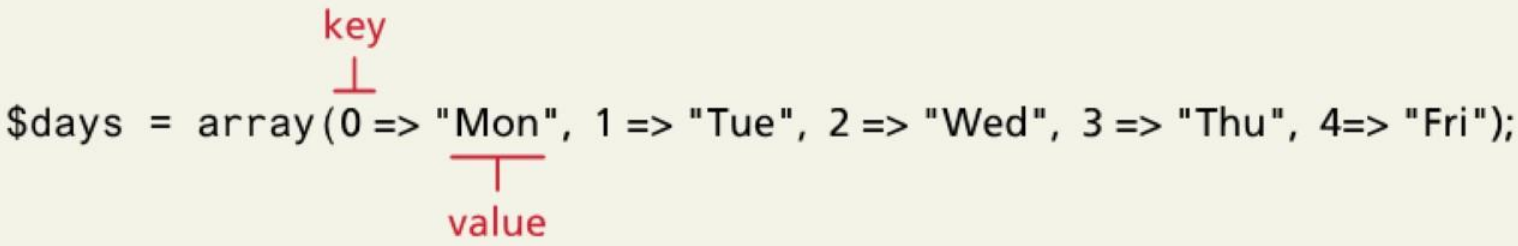
```
$days = array("Mon","Tue","Wed","Thu","Fri");

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



Arrays

Defining and Accessing an Array

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



Arrays

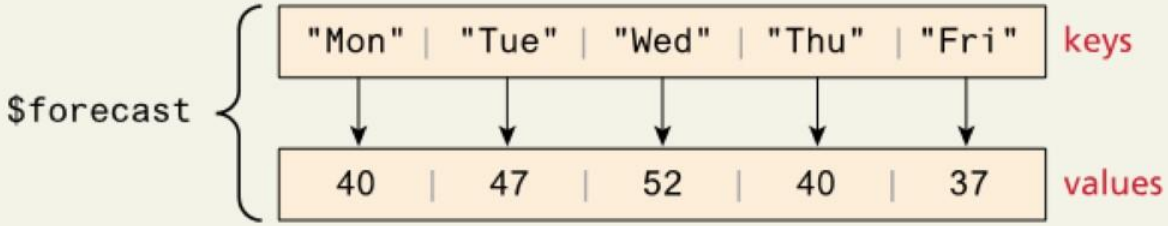
Defining and Accessing an Array

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

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

key

value

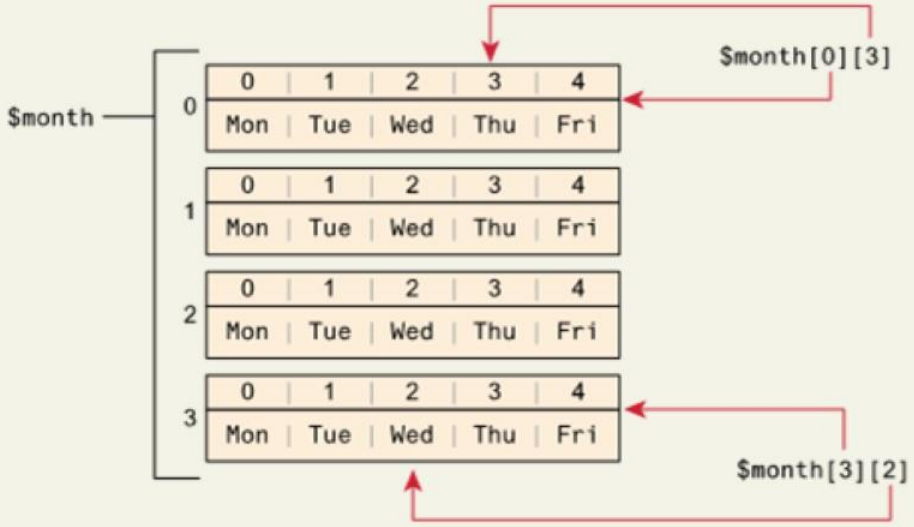



```
echo $forecast["Tue"];  // outputs 47
echo $forecast["Thu"];  // outputs 40
```

Arrays

Multidimensional Arrays

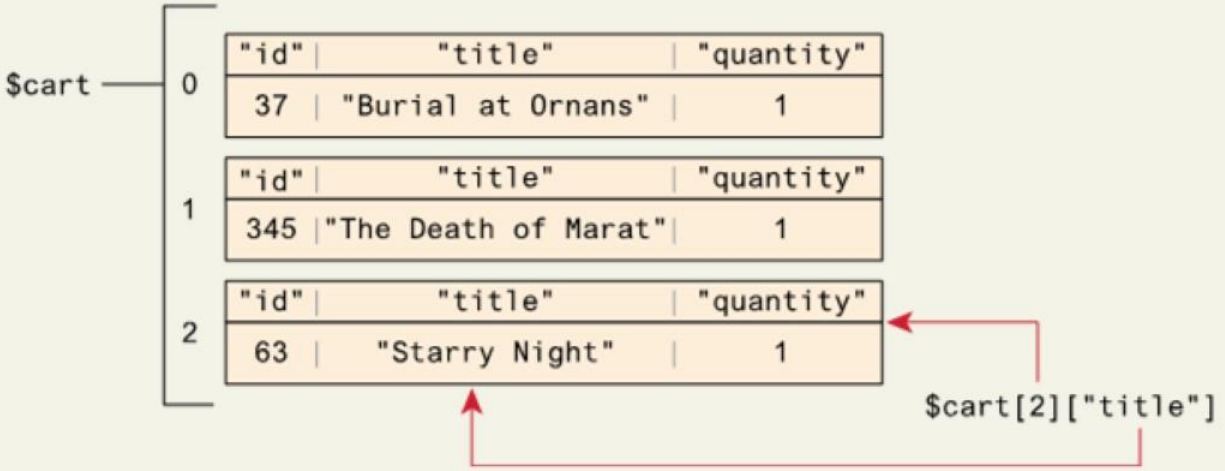
```
$month = array
(
array("Mon","Tue","Wed","Thu","Fri"),
array("Mon","Tue","Wed","Thu","Fri"),
array("Mon","Tue","Wed","Thu","Fri"),
array("Mon","Tue","Wed","Thu","Fri")
);
```



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);
```



Arrays

Iterating through an Array - while

```
// while loop

$i=0;

while ($i < count($days)) {

        echo $days[$i] . "<br>";

        $i++;

}
```

Arrays

Iterating through an Array - do while

```
// do while loop

$i=0;

do {

        echo $days[$i] . "<br>";

        $i++;

} while ($i < count($days));
```

Arrays

Iterating through an Array - for

```
// for loop

for ($i=0; $i<count($days); $i++) {

        echo $days[$i] . "<br>";

}
```

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;
}
```

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()

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)

Arrays

More Array Operations

- array_keys($someArray)
- array_values($someArray)
- array_rand($someArray, $num=1)
- array_reverse($someArray)
- array_walk($someArray, $callback, $optionalParam)
- in_array($needle, $haystack, $optionalStrict)
- shuffle($someArray)

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

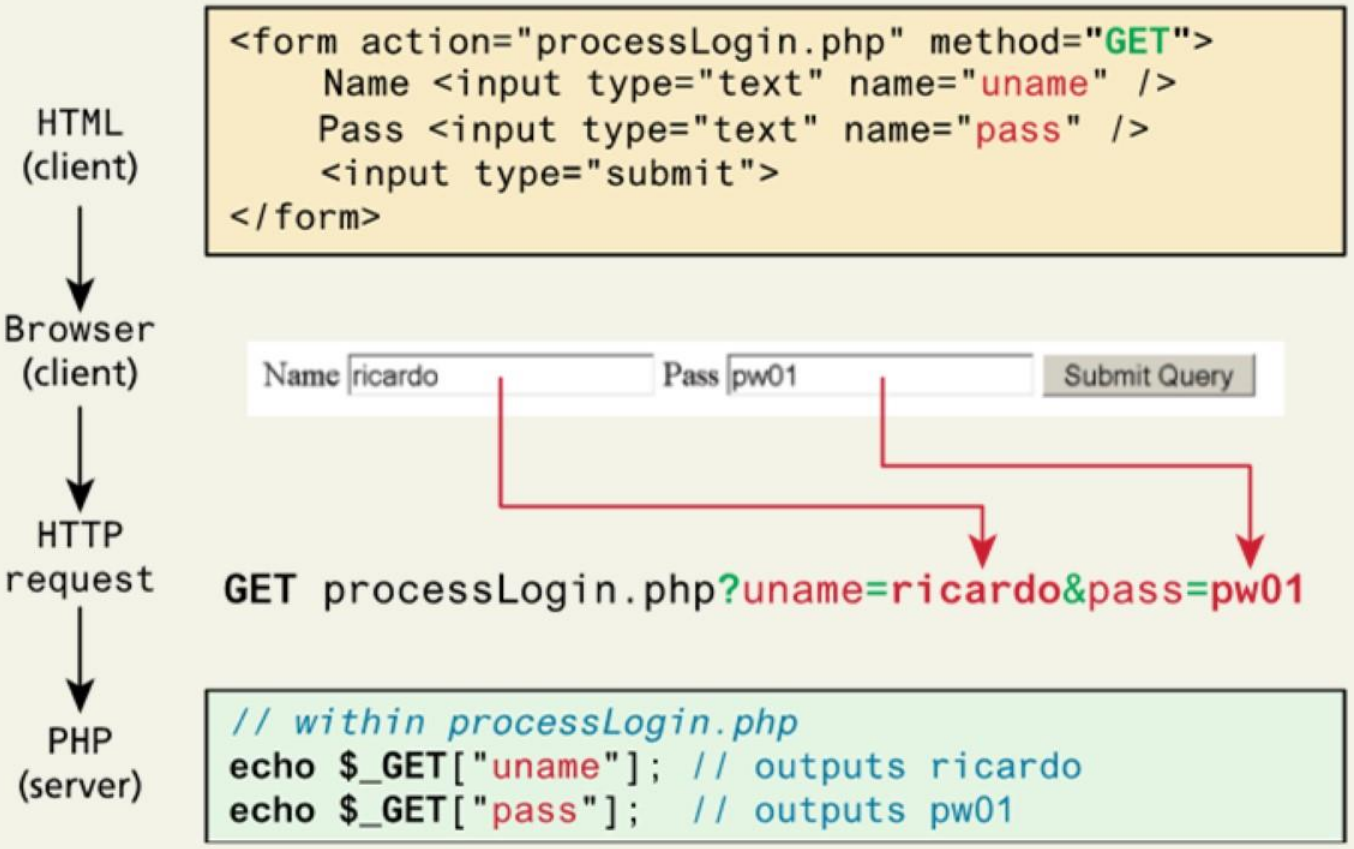
- $GLOBALS Array for storing data that needs superglobal scope
- $_COOKIES Array of cookie data passed to page via HTTP request
- $_ENV Array of server environment data
- $_FILES Array of file items uploaded to the server
- $_GET Array of query string data passed to the server via the URL
- $_POST Array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

Chapter 12

1. Arrays
2. $_GET and $_POST Superglobal Arrays
3. $_SERVER Array
4. $_FILES Array
5. Reading/ Writing Files
6. Summary

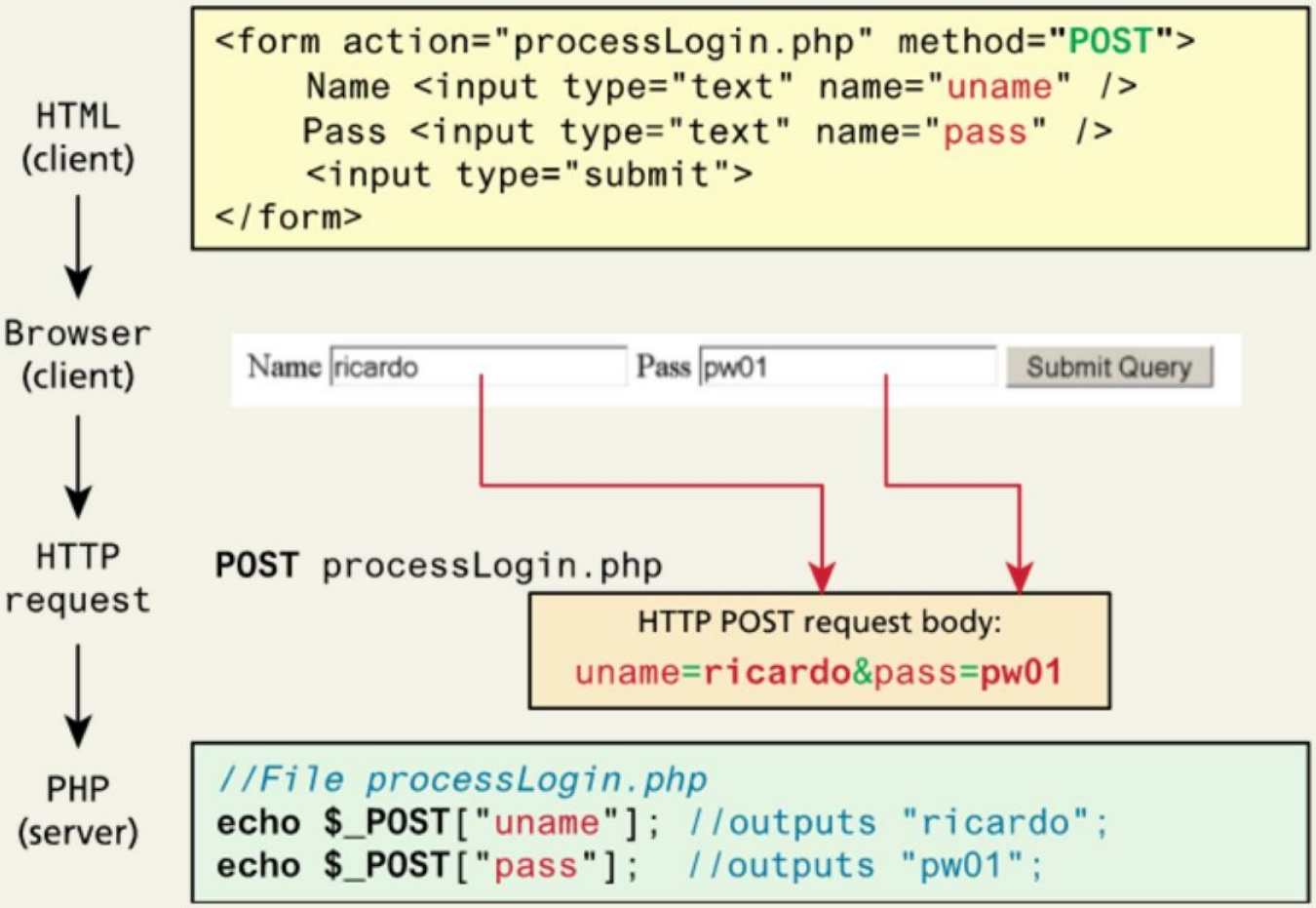
$_GET and $_POST Superglobal Arrays

Relating sent query string elements in PHP



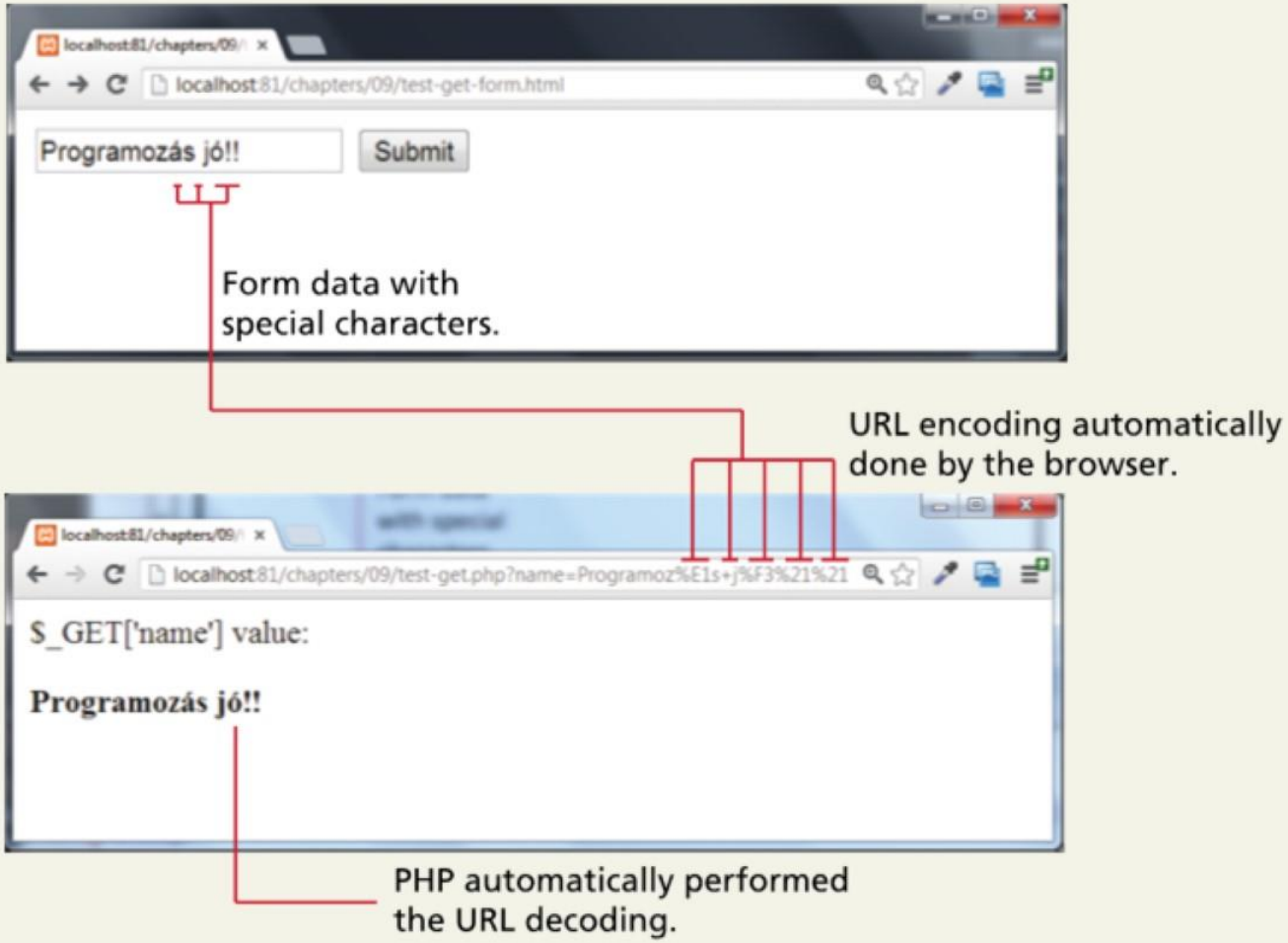
$_GET and $_POST Superglobal Arrays

Relating sent query string elements in PHP (POST)



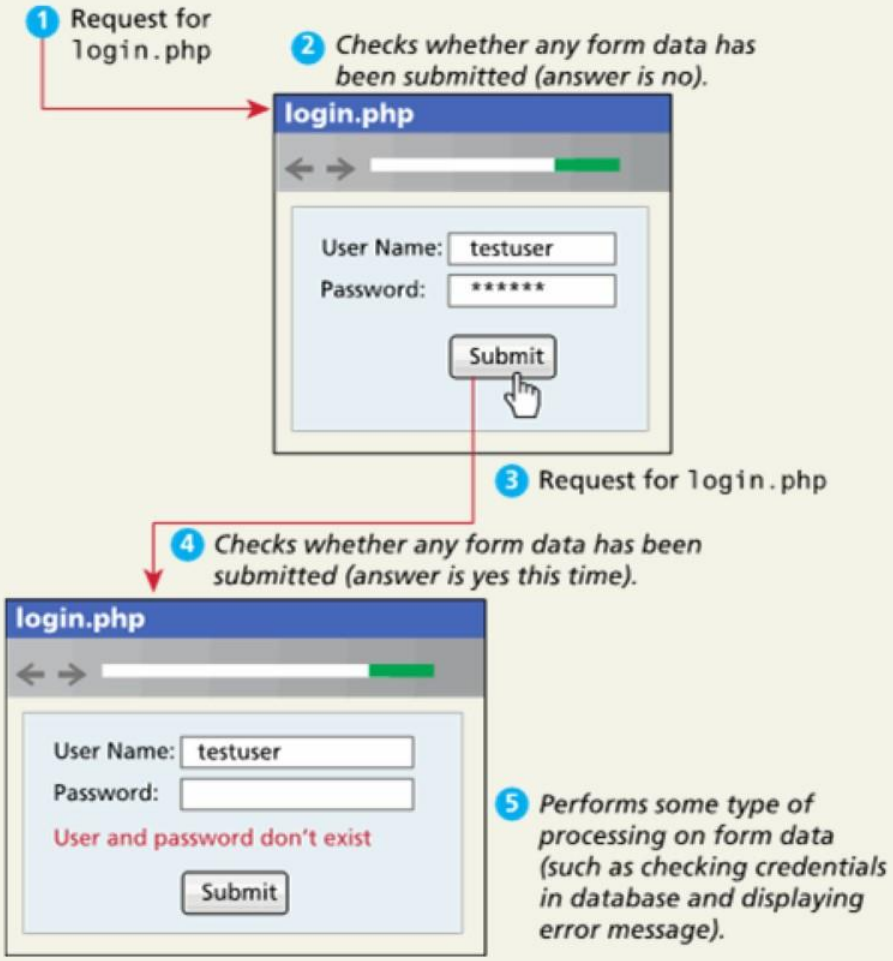
$_GET and $_POST Superglobal Arrays

Note URL encoding and Decoding



$_GET and $_POST Superglobal Arrays

Form display and processing on same page



$_GET and $_POST Superglobal Arrays

Determining If Any Data Sent

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

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

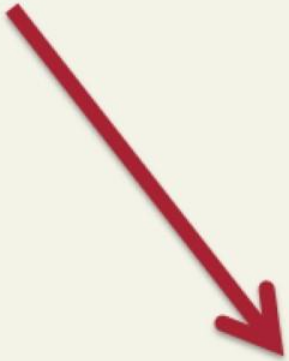
Becomes

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

$_GET and $_POST Superglobal Arrays

Accessing Form Array Data

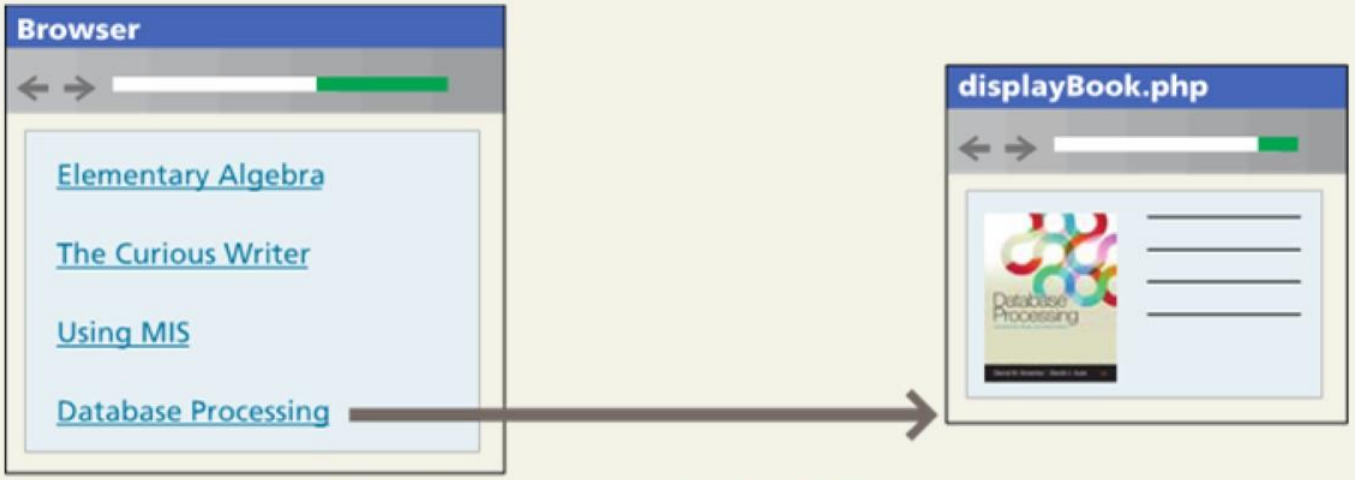
```
Monday <input type="checkbox" name="day[]" value="Monday">

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



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

$_GET and $_POST Superglobal Arrays

Using Query Strings in Hyperlinks




```
<a href="displayBook.php?isbn=0132145375">Database Processing</a>
```

Query string

$_GET and $_POST Superglobal Arrays

Sanitizing Query Strings

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:

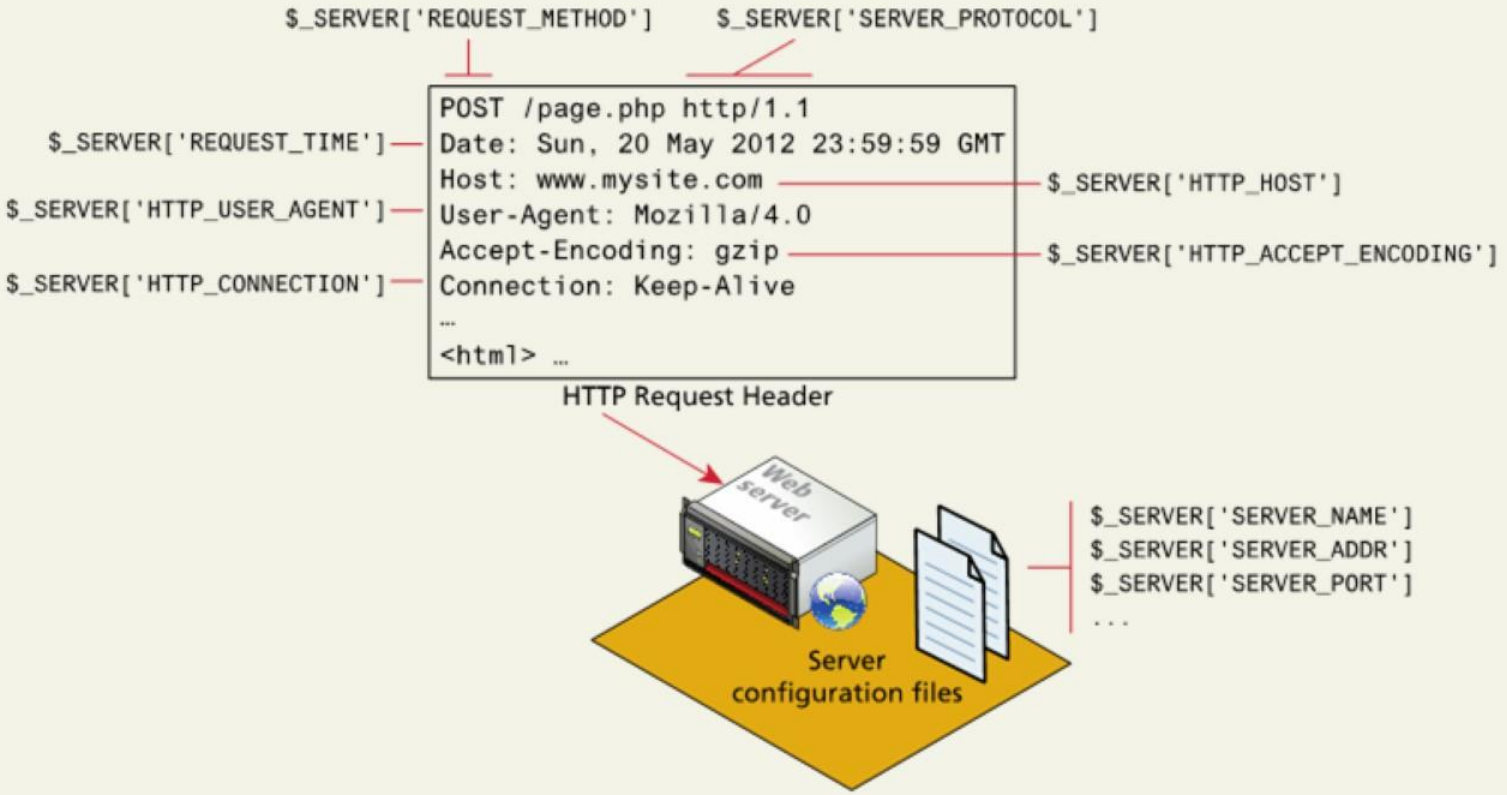
- If query string parameter doesn't exist.
- If query string parameter doesn't contain a value.
- If query string parameter value isn't the correct type or is out of acceptable range.
- If value is required for a database lookup, but provided value doesn't exist in the database table.

Chapter 12

1. Arrays
2. $_GET and $_POST Superglobal Arrays
3. $_SERVER Array
4. $_FILES Array
5. Reading/ Writing Files
6. Summary

$_SERVER Array

Server Information Keys



$_SERVER Array

Request Header Information Keys

```
<?php

echo $_SERVER['HTTP_USER_AGENT'];

//advanced browser detection

$browser = get_browser($_SERVER['HTTP_USER_AGENT'], true);

print_r($browser);

?>
```

$_SERVER Array

Request Header Information Keys

```
$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
```

Chapter 12

1. Arrays
2. $_GET and $_POST Superglobal Arrays
3. $_SERVER Array
4. $_FILES Array
5. Reading/ Writing Files
6. Summary

$_FILES Array

HTML Required for File Uploads

- 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'>
</form>
```

$_FILES Array

Handling the File Upload in PHP

HTML (client)

```
<form enctype='multipart/form-data' method='post' action='upFile.php'>
   <input type='file' name='file1'>
   <input type='submit' value="Submit Query">
</form>
```

Browser (client)

HTTP request

POST upFile.php

HTTP POST multipart/form-data

```
file1%PNG ™‡»*Î ! %çÔkáFÊ÷Ï§29%*üÄ¡Šrá vÛ„ýiN üc/©(—Ä Á Z}/vë\Á(m—%i± %6_Ë/Hí,+*" …ÀÉA`) ,þ/
_êv€oá˜™ ß%uÔaG¢ëN"ÔÓ?ÔÝ·i'm %€2h‹ ¶•˜|QÁóiçD"W)MíÛù_9ß9nÝ02'Å² km'Ñyþh $ž"›Ý˜ d®8í¬\
¡ûh»}>´‡¥[ ðB*ðèè[ ‡¥ué|ÝxR%XL)à£›[ÕZ!pâ X¢ kÔ >Ô °cÜý}=‡ ÁÔt®Q¯íÚ±¥<3%üÔ3 ×l'ó;¶l¬W÷4èê
®á,Òò9'táÊC €îÛ11Êëœ´B>{Úèà§® ®Ôê[¶îÞ%soiz³VydÙÊµ'ÅaÁ_ Ã ¯Ç jñzôce€%Ÿ~B ´=àã'~ÁÎÌ'¡íùú 5%$1 "
Z³Ñ ðë Ä¶ £…hÊGÄ ¢ª+¥i>ÈƒŽÎ€‹ûT …À| –ð€[®Å±ç€_´nú»ßó®ûOmŸa Ï ¿õ{?üö÷ï™ Û_ïé?\ËÃ /
æsÙ"8%%tëy˜¨>qäíÛõÎÔøþ Ï=ó7ßù‹ ‡š8Ä, ßøô'?÷ÁíÎlX±æìl¾x8ô ?¿ ñÚLòoþî Ú7~ï¦;¶i>d§éâ >
®¶Á:K{ªG®ð±xÀ Ûsx *wnÛ¼o×¶ñ±ýX]1Ii\ ^'fÅêuéL6>yÉÅ¼ÇÐUÛ`ý#÷×ëÕøäà³ ÊW¯ö<wÉSë'ÇGf&ÇâÎQ&›?eå
…
```

PHP (server)

```
echo $_FILES["file1"]["name"]       // "Sample1.png"
echo $_FILES["file1"]["type"]       // "image/png"
echo $_FILES["file1"]["tmp_file"]   // "/tmp/phpJ08pVh"
echo $_FILES["file1"]["error"]      //   0
echo $_FILES["file1"]["size"]       // 1219038
```

$_FILES Array

Checking for Errors

```
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 ";

        }

}
```

$_FILES Array

File Size Restrictions

You can limit in multiple ways

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

$_FILES Array

Limiting the Type of File Upload

```
$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";
        }
}
```

$_FILES Array

Moving the File

```
$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.";

}
```

Chapter 12

1. Arrays
2. $_GET and $_POST Superglobal Arrays
3. $_SERVER Array
4. $_FILES Array
5. Reading/ Writing Files
6. Summary

Reading/Writing Files

Two ways

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.

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()

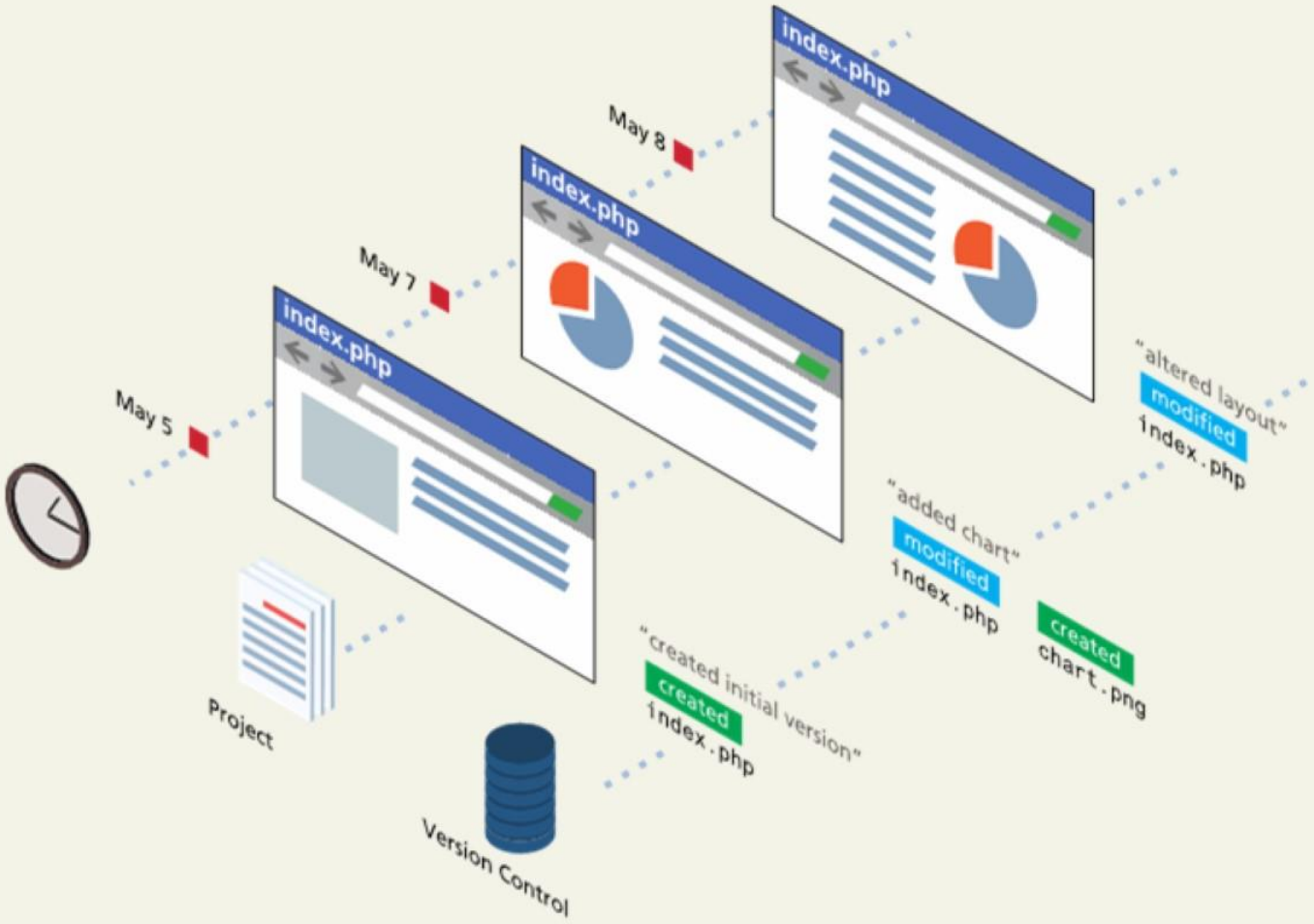
Reading/Writing Files

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.

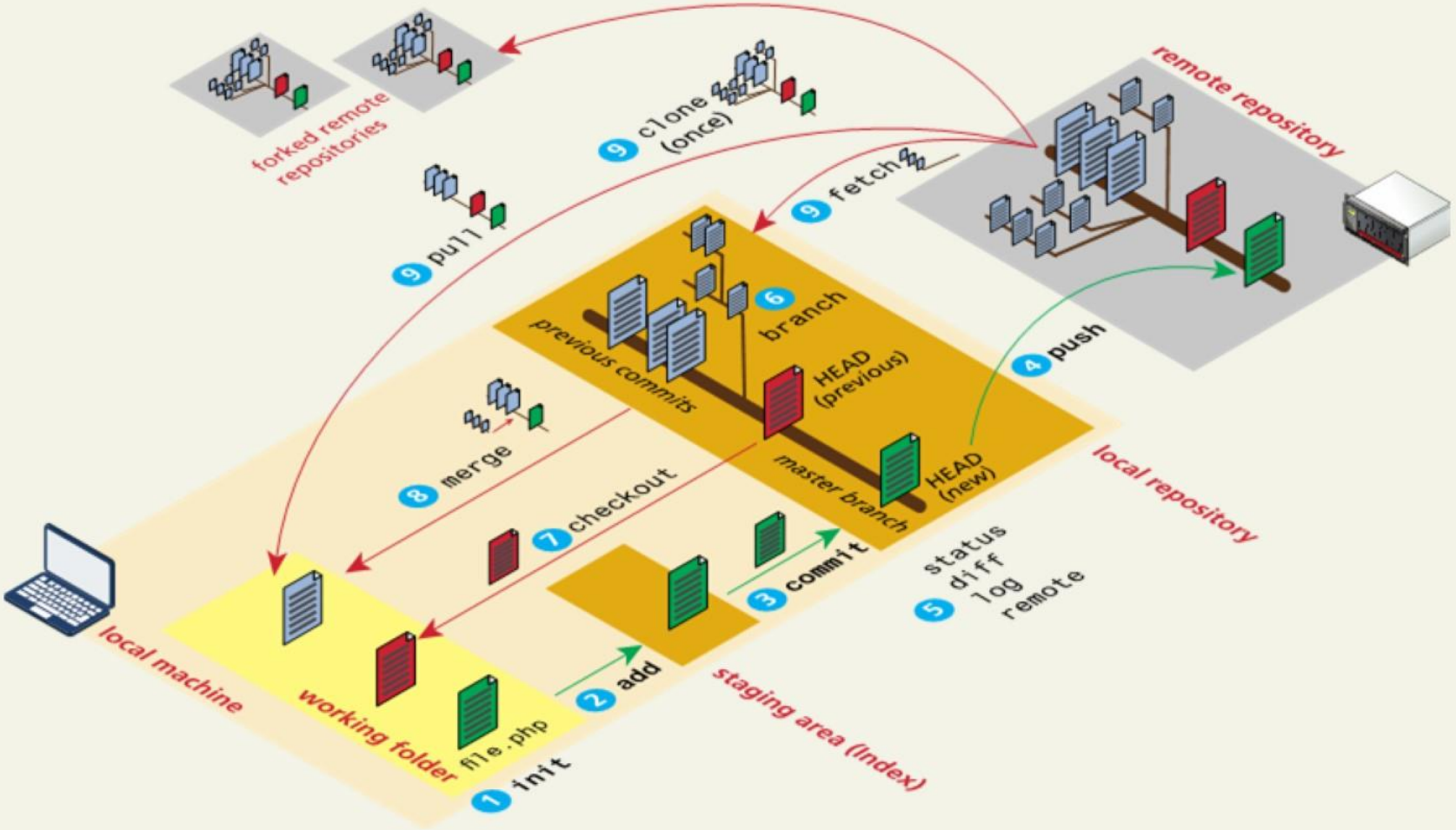
Version Control

Keep track of changes



Version Control

Widely used in industry



Chapter 12

1. Arrays
2. $_GET and $_POST Superglobal Arrays
3. $_SERVER Array
4. $_FILES Array
5. Reading/ Writing Files
6. Summary

Summary

Key Terms

All-in-memory access

array keys

array values

associative arrays

branch

forking

Git

GitHub

local repository

merge

NULL

null coalescing operator

one-way hash

ordered map

remote repository

sanitizing user inputs

stream access

stream resource

superglobal variables

user-agent

version control

Summary

Questions?