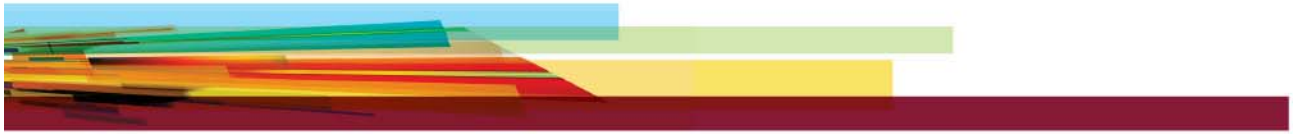
CEWP 459

PHP Programming with MySQL – Level I

Web Techniques



HTTP Specific Topics

GLOBAL Server Arrays

$HTTP_COOKIE_VARS

- Contains any cookie values passed as part of the request, where the keys of the array are the names of the cookies

$HTTP_GET_VARS

- Contains any parameters that are part of a GET request, where the keys of the array are the names of the form parameters

$HTTP_POST_VARS

- Contains any parameters that are part of a POST request, where the keys of the array are the names of the form parameters

$HTTP_POST_FILES

- Contains information about any uploaded files

$HTTP_SERVER_VARS

- Contains useful information about the web server, as described in the next section

$HTTP_ENV_VARS

- Contains the values of any environment variables, where the keys of the array are the names of the environment variables

- $_COOKIE
- $_GET
- $_POST
- $_FILES
- $_SERVER
- $_EN

$_REQUEST

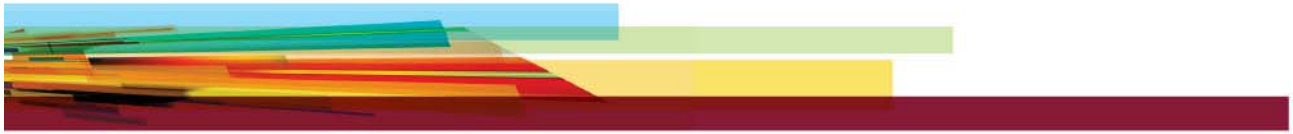
Contains

- $_GET
- $_POST
- $_COOKIE

$_SERVER

- SERVER_NAME
- SERVER_PORT
- REQUEST_METHOD : get / post
- PATH_INFO
- QUERY_STRING
- REMOTE_HOST
- REMOTE_ADDR
- HTTP_REFERER

```
if ($_SERVER['REQUEST_METHOD'] ==
'GET') {
// handle a GET request
} else {
die("You may only GET this page.");
}
```



Forms

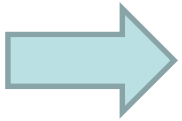
Forms

- A form is a HTML structure which allows the user to submit data.
- Example:

```
<!DOCTYPE HTML>
<html>
<body>

<form action="welcome.php"
method="post">
Name: <input type="text"
name="name"><br>
E-mail: <input type="text"
name="email"><br>
<input type="submit">
</form>

</body>
</html>
```



Name:
E-mail:
Submit

Common input types

Keyword	State	Data type	Control type
hidden	Hidden	An arbitrary string	n/a
text	Text	Text with no line breaks	Text field
password	Password	Text with no line breaks (sensitive information)	Text field that obscures data entry
checkbox	Checkbox	A set of zero or more values from a predefined list	A checkbox
radio	Radio Button	An enumerated value	A radio button
file	File Upload	Zero or more files each with a MIME type and optionally a file name	A label and a button
submit	Submit Button	An enumerated value, with the extra semantic that it must be the last value selected and initiates form submission	A button
image	Image Button	A coordinate, relative to a particular image's size, with the extra semantic that it must be the last value selected and initiates form submission	Either a clickable image, or a button
reset	Reset Button	n/a	A button
button	Button	n/a	A button

GET

- URL
- http://example.com/page.php?name=Bob

- PHP in page.php
  - $_GET["name"]

Get

- <html><body>
  <form action="welcome_get.php" method="get">
  Name: <input type="text" name="name">
  E-mail: <input type="text" name="email">
  <input type="submit">
  </form>
  </body></html>
- welcome_get.php
- <html><body>
- Welcome <?php echo $_GET["name"]; ?>
- Your email address is: <?php echo $_GET["email"]; ?>
- </body></html>

POST

- Same as get but the information is not passed in the query string.
- Passed in the body.
- More secure

Self Superglobal

- $_SERVER["PHP_SELF"]

Forms

- Standard form, submits another form
- action="form1_get.php"

- Form that submits to itself.
- action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"

Exercises (FORMS)

- [3] Create a GET form – submitting to another page.
  - On the form, use two text boxes to ask for NAME and WEEKDAY.
  - On the next page, simply display “Hello xxxx the day is xxxx”.

- [4] Create a POST form – submitting to SAME page.
  - Ask for two numbers, then when the user clicks SUBMIT, add them and display the results eg: “The result of 1 + 2 is 3”.

Isset

- To validate if your return value is present, use this function.

ISSET vs. EMPTY vs. ISNULL

- *isset — Determine if a variable is set and is not NULL*
- *empty — Determine whether a variable is empty*
- *is_null — Finds whether a variable is NULL*

Group Mini-Project – part 1

- Create a PHP program that goes through each of the following cases in the following table, and then displays the result in a properly formatted table.

Value of variable ($var)	isset($var)	empty($var)	is_null($var)
“” (an empty string)			
” ” (space)			
FALSE			
TRUE			
array() (an empty array)			
NULL			
“0” (0 as a string)			
0 (0 as an integer)			
0.0 (0 as a float)			
var $var; (a variable declared, but without a value)			
NULL byte (“\ 0”)			

Value of variable ($var)	isset($var)	empty($var)	is_null($var)
"String Value"			
123			

Forms

- Standard form, submits another form
- action="form1_get.php"

- Form that submits to itself.
- action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"

htmlspecialcharacters

```
<?php
$new = htmlspecialchars("<a href='test'>T
est</a>", ENT_QUOTES);
echo $new; // &lt;a href=&#039;test&#039;
&gt;Test&lt;/a&gt;
?>
```

```
<!- Using the GET method -->
<html>
<head><title>Temperature Conversion</title></head>
<body>
<?php
$fahr = $_GET['fahrenheit'];
if (is_null($fahr)) {
?>
<form action="<?php echo $_SERVER['PHP_SELF'] ?>"
method="GET">
Fahrenheit temperature:
<input type="text" name="fahrenheit" /> <br />
<input type="submit" name="Convert to Celsius!" />
</form>
<?php
} else {
$celsius = ($fahr - 32) * 5/9;
printf("%.2fF is %.2fC", $fahr, $celsius);
}
?>
</body>
</html>
```

```
<html> <!- Using REQUEST METHOD -->
<head><title>Temperature Conversion</title></head>
<body>
<?php
if ($_SERVER['REQUEST_METHOD'] == 'GET') {
?>
<form action="<?php echo $_SERVER['PHP_SELF'] ?>"
method="POST">
Fahrenheit temperature:
<input type="text" name="fahrenheit" /> <br />
<input type="submit" name="Convert to Celsius!" />
</form>
<?php
} elseif ($_SERVER['REQUEST_METHOD'] == 'POST') {
$fahr = $_POST['fahrenheit'];
$celsius = ($fahr - 32) * 5/9;
printf("%.2fF is %.2fC", $fahr, $celsius);
} else {
die("This script only works with GET and POST requests.");
}
?>
</body>
</html>
```

Sticky Forms

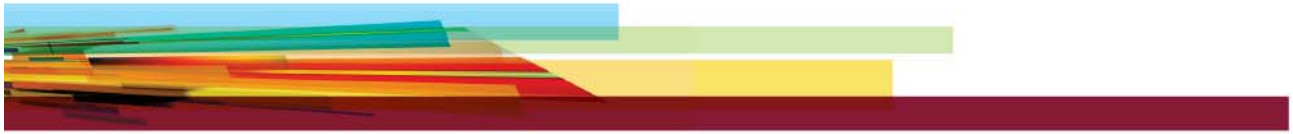
- Many web sites use a technique known as *sticky forms*, in which the results of a query are accompanied by a search form whose default values are those of the previous query.

```
<html>
<head><title>Temperature Conversion</title></head>
<body>
<?php
$fahr = $_GET['fahrenheit'];
?>
<form action="<?php echo $_SERVER['PHP_SELF'] ?>" method="GET">
Fahrenheit temperature:
<input type="text" name="fahrenheit" value="<?php echo $fahr ?>" />
<br />
<input type="submit" name="Convert to Celsius!" />
</form>
<?php
if (! is_null($fahr)) {
$celsius = ($fahr - 32) * 5/9;
printf("%.2fF is %.2fC", $fahr, $celsius);
}
?>
</body>
</html>
```

Multi Value SELECT

- Use [] array marker in the name of the select.
- Eg: `name="foobar[]"`

- It is not recommended to make multivalue form values sticky. Reconsider the form strategy if so.



File Upload via Form

Upload a File - Form

- Create form to upload a file.

- ```
  <form action="upload_file.php"
  method="post"
  enctype="multipart/form-data">
  <label for="file">Filename:</label>
  <input type="file" name="file"
  id="file"><br>
  <input type="submit" name="submit"
  value="Submit">
  </form>
  ```
  ```

Exercise
Create your HTML to upload your file.

- Follow example on previous slide.

Upload file - Script

- Note: This should upload to a temporary location, then copy the file once complete.

```
<?php
if ($_FILES["file"]["error"] > 0)
  {
  echo "Error: " . $_FILES["file"]["error"] . "<br>";
  }
else
  {
  echo "Upload: " . $_FILES["file"]["name"] . "<br>";
  echo "Type: " . $_FILES["file"]["type"] . "<br>";
  echo "Size: " . ($_FILES["file"]["size"] / 1024) . "
kB<br>";
  echo "Stored in: " . $_FILES["file"]["tmp_name"];
  }
?>
```

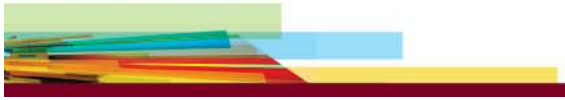
This is an Array

Exercise
Create your PHP to upload your file.

- Follow example on previous slide.

Add Upload Restrictions

```
$allowedExts = array("gif", "jpeg", "jpg");
$temp = explode(".", $_FILES["file"]["name"]);
$extension = end($temp);
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/jpg"))
&& ($_FILES["file"]["size"] < 20000)
&& in_array($extension, $allowedExts))
  {
  if ($_FILES["file"]["error"] > 0)
    { echo "Error: " . $_FILES["file"]["error"] . "<br>"; }
  else
    {
    echo "Upload: " . $_FILES["file"]["name"] . "<br>";
    echo "Type: " . $_FILES["file"]["type"] . "<br>";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . " kB<br>";
    echo "Stored in: " . $_FILES["file"]["tmp_name"];
    }
  }
else
  { echo "Invalid file"; }
```



Exercise
Retrict to TXT file only, 100bytes max.

- Follow example on previous slide.

Saving the Uploaded File

- Check if the file exists, if it does not, MOVE IT.

```
if (file_exists("upload/" .
_FILES["file"]["name"]))
{
 echo $_FILES["file"]["name"] . " already
exists.";
}
 else
{
 move_uploaded_file($_FILES["file"]["tmp_name"],
      "upload/" . $_FILES["file"]["name"]);
      echo "Stored in: " . "upload/" .
      $_FILES["file"]["name"];
}
```

Exercise
Move the file if it doesn't exist.

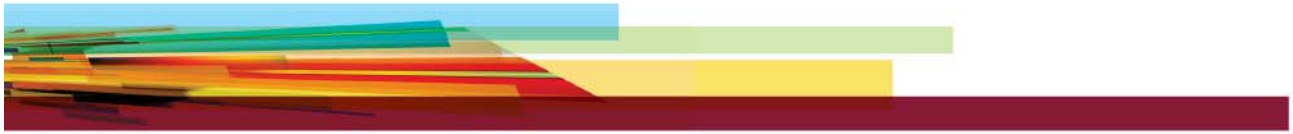
- Follow example on previous slide.

List files in directory

```
$path    = '/tmp';
$files = scandir($path);
```

- Following code will remove . and .. from the returned array from scandir:

```
$files = array_diff(scandir($path),
array('.', '..'));
```



Other HTTP functions

Redirection

- Redirect to another physical web page:

```
<?php
header('Location:
http://www.example.com/a.html');
exit();
?>
```

Other uses for header

- Status codes.

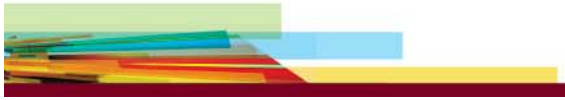
```
header("HTTP/1.0 404 Not Found");
```

- Downloading a file.

```
header('Content-
Type: application/pdf');

header('Content-
Disposition: attachment; filename="down
loaded.pdf"');

readfile('original.pdf');
```



Cookies

Write a Cookie

- setcookie(name, value, expire, path, domain);
- setcookie("name", "Brendan", time()+3600);
- Time()+3600 = 1 hour.
- $expire=time()+60*60*24*30;
- setcookie("name", "Brendan", $expire);

Retrieve the cookie

- Print a cookie
  `echo $_COOKIE["user"];`

- A way to view all cookies
  `print_r($_COOKIE);`

Delete a cookie

- Just set the time to the past. (1 hour before now).

```
setcookie("name", "", time()-3600);
```

Uses for a cookie

- Store the user name
- Show if user visited site
- Show if a user visited a page
- Store a preference (language)

Exercise (cookie)

- Make two php pages.
  - Web page #1: sets a color name (choose a primary color like red, blue or green.
  - Web page #2: Read the cookie and color the background according to the cookie value.

Hint: webpage background can be changed by adding and attribute to the <body> tag.

<body bgcolor="#E6E6FA">

Exercise

- Create a view page counter.

Session

- Used to store information about, or change settings for a user session. Session variables hold information about one single user, and are available to all pages in one application.
- MUST be set before the <html> tag.

```
<?php session_start(); ?>
```

```
<?php
session_start();
$_SESSION['views']=1;
?>

<html><body>
<? echo "Pageviews=". $_SESSION['views']; ?>
```

Notes on Sessions

- By default, the session ID is passed from page to page in the PHPSESSID cookie.
- Arrays, objects can be stored in session variables.

isset

- Simple counter within session.

```
<?php
	session_start();
	if(isset($_SESSION['views']))

	$_SESSION['views']=$_SESSION['views']+
1;
	else
		$_SESSION['views']=1;
	echo "Views=" . $_SESSION['views'];
?>
```

Destroy a Session

- Destroy session variable.

```
<?php
session_start();
if(isset($_SESSION['views']))
  unset($_SESSION['views']);
?>
```

- Destroy all the session.

```
<?php session_destroy(); ?>
```

Exercise (Session)

- [6] Write just one web page that creates a session at the beginning, then it creates a session variable called “name” with your name in it, then READ the session variable called “name” and display it to the screen.

- [6b] Write a webpage that creates a session, and a session variable called “page views” and display the value. The value should increment itself each time the page is refreshed.

SSL

- Validate if the current web browser is connected via SSL.

```
if ($_SERVER{'HTTPS'] !== 'on') {
die("Must be a secure connection.");
}
```