

1

Embedded PHP

Printing HTML tags in PHP = bad style

2

```
<?php
print "<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML
1.1//EN\"\\n";
print "
\"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd\">\\n";
print "<html xmlns=\"http://www.w3.org/1999/xhtml\">\\n";
print " <head>\\n";
print " <title>Geneva's web page</title>\\n";
...
for ($i = 1; $i <= 10; $i++) {
print "<p> I can count to $i! </p>\\n";
}
?>
```

HTML

- ❑ best PHP style is to **minimize print/echo** statements in embedded PHP code
- ❑ but **without print**, how do we insert dynamic content into the page?

PHP expression blocks

3

```
<?= expression ?>
```

PHP

```
<h2> The answer is <?= 6 * 7 ?> </h2>
```

PHP

The answer is 42

output

- PHP expression block: a small piece of PHP that evaluates and embeds an expression's value into HTML
 - ▣ `<?= expression ?>` is equivalent to:

```
<?php print expression; ?>
```

PHP

Expression block example

4

```
<body>
<?php
for ($i = 99; $i >= 1; $i--) {
?>
<p> <?= $i ?> bottles of beer on the wall, <br />
<?= $i ?> bottles of beer. <br />
Take one down, pass it around, <br />
<?= $i - 1 ?> bottles of beer on the wall. </p>
<?php
}
?>
</body>
</html>
```

PHP

Complex expression blocks

5

```
...  
<body>  
<?php  
for ($i = 1; $i <= 3; $i++) {  
    ?>  
    <h<?= $i ?>>This is a level <?= $i ?>  
heading.</h<?= $i ?>>  
    <?php  
}  
?>  
</body>
```

PHP

This is a level 1 heading.

This is a level 2 heading.

This is a level 3 heading.

output

PHP form

Submitting data to a web server

7

- Though browsers mostly retrieve data, sometimes you want to **submit data to a server**
 - ▣ Hotmail: Send a message
 - ▣ Flickr: Upload a photo
 - ▣ Google Calendar: Create an appointment
- the data is sent via **HTTP requests** to the server
 1. with HTML forms
 2. with **Ajax** (will see later)
- The data is placed into the HTTP request as **parameters**

Data submission via HTML form

8

```
<form action="destination URL" method="get|post">  
  form controls  
</form>
```

HTML

- **action attribute** specifies the destination URL (php file) to whom data are being submitted.
- **method attribute** specifies how the data is going to be carried out to the server (i.e., to the destination php file).
- You can either use GET or POST method. However, **GET** is the **default** transformation method.

Data submission via GET

9

- when sending data, the GET method adds the data to the URL as name value pairs.

```
# index.php file
<form action="http://localhost/submission.php">
  <div>
    Let's submit first name and last name: <br>
    <input name="fname" /> <br> <input name="lname" />
    <br> <input type="submit" value="Submit"/>
  </div>
</form>
```

- On submission the generated url:
http://localhost/submission.php?fname='inputData1'&lname='inputData2'
- Here, the **fname='inputData1'&lname='inputData2'** part is called **query string**.

Read GET data in PHP

10

submission.php file

```
if ($_SERVER["REQUEST_METHOD"] == "GET") {  
    if (isset($_GET["fname"])) {  
        echo $_GET["fname"] #inputData1  
    }  
}
```

-
- Here, `$_GET` is a super global associative array (where key instead of index is used)
 - GET is fast data transferring method. However, it is less secure compared to POST because data sent is part of the URL (visible).
 - Should not use for transferring sensitive data.

URL-encoding

11

- certain characters are **not allowed** in URL query parameters:
 - ▣ examples: " ", "/", "=", "&"
- Therefore, passing these as a parameter, they are **URL-encoded** first.
 - ▣ “Xenia's cool!?” → “Xenia%27s+cool%3F%21”
- you don't usually need to worry about this:
 - ▣ the browser **automatically encodes** parameters before sending them
 - ▣ the **PHP \$_GET** array automatically decodes them
 - ▣ ... but occasionally the encoded version does may pop up.

Data submission via POST

12

- **POST** : **submits data** to a web server and retrieves the server's response
 - ▣ if the request has parameters, they are embedded in the request's **HTTP packet**, not the URL
- Reading POST data from PHP:

```
($_SERVER["REQUEST_METHOD"] == "POST") {  
    echo $_POST['key'];  
}
```

*

Other "Superglobal" arrays

13

Array	Description
<u>\$ REQUEST</u>	parameters passed to any type of request
<u>\$ GET</u> , <u>\$ POST</u>	parameters passed to GET and POST requests
<u>\$ SERVER</u> , <u>\$ ENV</u>	information about the web server
<u>\$ FILES</u>	files uploaded with the web request
<u>\$ SESSION</u> , <u>\$ COOKIE</u>	"cookies" used to identify the user (seen later)

- PHP superglobal arrays contain information about the **current request**, server, etc.
- These are **special kinds of arrays** called **associative arrays**.

Task

14

```
<?php
    $base = $_REQUEST["base"];
    $exp = $_GET["exponent"];
    $result = pow($base, $exp);
?>
<?= $base ?> ^ <?= $exp ?> = <?= $result ?>
```

PHP

- ❑ Open xampp → start apache server → open the 'C:/xampp/htdocs' folder → create a new folder with name 'testApp'
- ❑ inside the 'testApp' folder create two files
 1. exponential.php: where you put the above php code
 2. index.php: where write a compatible html form by your own so that both can work together.
 3. To run the code, open browser and go to:
<http://localhost/testApp/index.php>

Problem

15

```
<form action="http://localhost/test1.php" method="get">
<label><input type="radio" name="cc" /> Visa</label>
<label><input type="radio" name="cc" /> MasterCard</label>
<br />
Favorite Star Trek captain:
<select name="startrek">
    <option>James T. Kirk</option>
    <option>Jean-Luc Picard</option>
</select> <br />
</form>
```

HTML

- the form may look correct, but when you submit it...
- [cc] => on,
- [startrek] => Jean-Luc Picard
- How can we resolve this conflict?

The value attribute

16

```
<label><input type="radio" name="cc" value="visa" />
Visa</label>
<label><input type="radio" name="cc" value="mastercard" />
MasterCard</label> <br />
Favorite Star Trek captain:
<select name="startrek">
    <option value="kirk">James T. Kirk</option>
    <option value="picard">Jean-Luc Picard</option>
<input type="submit" value="submit" />
</select> <br />
```

HTML

- value attribute sets what will be submitted if a control is selected
- [cc] => visa,
- [startrek] => picard

17

Working with file submission

Uploading files

18

```
<form action="http://webster.cs.washington.edu/params.php"
method="post" enctype="multipart/form-data">
    Upload an image as your avatar:
    <input type="file" name="avatar" />
    <input type="submit" />
</form>
```

HTML

- add a file upload to your form as an input tag with **type of file**
- **must** also **set** the **enctype** attribute of the form

Processing an uploaded file in PHP

19

- uploaded files are placed into global array `$_FILES`, not `$_REQUEST`
- each element of `$_FILES` is itself an associative array, containing:
 - `name`: the local filename that the user uploaded
 - `type`: the MIME type of data that was uploaded, such as `image/jpeg`
 - `size` : file's size in bytes
 - `tmp_name` : a filename where PHP has temporarily saved the uploaded file
 - to permanently store the file, move it from this location into some other file

Uploading files

20

```
<input type="file" name="avatar" />
```

HTML

- example: if you upload tobyy.jpg as a parameter named avatar,
 - `$_FILES["avatar"]["name"]` will be "tobby.jpg"
 - `$_FILES["avatar"]["type"]` will be "image/jpeg"
 - `$_FILES["avatar"]["tmp_name"]` will be something like `"/var/tmp/phpZtR4TI"`

```
Array
(
    [file1] => Array
        (
            [name] => MyFile.txt (comes from the browser,
so treat as tainted)
            [type] => text/plain (not sure where it gets
this from - assume the browser, so treat as tainted)
            [tmp_name] => /tmp/php/php1h4j1o (could be
anywhere on your system, depending on your config
settings, but the user has no control, so this isn't
tainted)
            [error] => UPLOAD_ERR_OK (= 0)
            [size] => 123 (the size in bytes)
        )
    [file2] => Array
        (
            [name] => MyFile.jpg
            [type] => image/jpeg
            [tmp_name] => /tmp/php/php6hst32
            [error] => UPLOAD_ERR_OK
            [size] => 98174
        )
)
```

Processing uploaded file example

22

```
$username = $_REQUEST["username"];  
if (is_uploaded_file($_FILES["avatar"]["tmp_name"])) {  
    move_uploaded_file($_FILES["avatar"]["tmp_name"],  
        "$username/avatar.jpg");  
    print "Saved uploaded file as  
$username/avatar.jpg\n";  
} else {  
    print "Error: required file not uploaded";  
}
```

PHP

- ❑ functions for dealing with uploaded files:
 - ❑ `is_uploaded_file(filename)`: returns TRUE if the given filename was uploaded by the user
 - ❑ `move_uploaded_file(from, to)`: moves from a temporary file location to a more permanent file

Thank You