



Universiteti i Prishtinës

PHP, Part II

Slides mostly adopted from Tufts University, Prof. Ming Chow

Dr. Ing. Lule Ahmedi
Lënda: Programimi në Internet

About QR Codes and Short URLs



Miredita

- Short URLs are used to shorten long URLs (e.g., from news sites and especially anything from a WordPress blog)
- To generate short URLs:
 - ▣ bit.ly
 - ▣ <http://goo.gl/>
- To generate QR codes:
 - ▣ <http://qrcode.kaywa.com/>
 - ▣ Way cool! Add .qr to the end of your goo.gl short URL. Read more:
- Also, read Chris Traganos' presentation on QR codes and short URLs:
<http://www.slideshare.net/harvardwww/mobile-themes-qrcode-and-shorturls>

Processing Form Inputs and Storing State Information

- `$_GET['varname']` - Fetch variables that are sent in the URL (e.g., Google)
- `$_POST['varname']` - Fetch variables that are sent with a post request (in a form)
- `$_SESSION['varname']` - Used to maintain user's data to across their session while their browser is open. **Must** call PHP function `session_start()` at the beginning of each PHP page!

Example: E-mail Form

```
<?php
    $submitted = $_POST['submitted'];
    if ($submitted) {
        $taDate = $_POST['ta_date'];
        $taStartTime = $_POST['ta_start_time'];
        $taEndTime = $_POST['ta_end_time'];
        $lab = $_POST['lab'];
        $students = $_POST['students'];
        $comments = $_POST['comments'];
        $from = 'theRock@knowyourrole.com';
        $body = '';
        $body = $body . "DATE: " . $taDate . "\n\n";
        $body = $body . "HOURS: " . $taStartTime . " - " . $taEndTime . "\n\n";
        $body = $body . "LAB: " . $lab . "\n\n";
        $body = $body . "STUDENTS: " . $students . "\n\n";
        $body = $body . "COMMENTS: " . $comments . "\n";
        $headers = "From: $from\n";
        $subject = "TA Hours Log";
        $to = "Ming Chow <mchow@cs.tufts.edu>";
        if (mail($to, $subject, $body, $headers)) {
            print "<h4 class=\"success\">Thank you, your TA log was successfully entered and e-mailed to the
instructor.
Return to <a href=\"index.php\">TA log form</a>.</h4>\n"; }
        else {
            print "<h4 class=\"error\">Your log could not be e-mailed because of a mail server error. Please
try again later.</h4>\n"; }
    }
?>
```

Date and Time Formatting

- ❑ `date($format, $timestamp)`
- ❑ Example 1: `$formattedDate = date('F j, Y', $timestamp);`
- ❑ [Complete list of format options](#)
- ❑ `strtotime` - Parse about any English textual datetime description into a Unix timestamp (the number of seconds since January 1 1970 00:00:00 UTC)
- ❑ Examples:
`$timestamp = strtotime("now");`
`$timestamp = strtotime("+1 day");`
`$timestamp = strtotime("next Thursday");`

Versions and Originals

- ❑ You can change any PHP variable you can get to!
- ❑ Original values are still available
- ❑ Example:

```
<?php
    $_SERVER['HTTP_USER_AGENT'] = "Hacked";
    echo "your new browser is " .
$_SERVER['HTTP_USER_AGENT'] . "<br>";
    echo "your original browser is " .
getenv("HTTP_USER_AGENT") . "<br>";
?>
```

- ❑ Result:

```
your new browser is Hacked
your original browser is Mozilla/5.0 (Windows; U; Windows
NT 6.1; en-US; rv:1.9.2.13) Gecko/20101203 Firefox/3.6.13
```

Iterating over Associative Arrayw

- ❑ `each($array)`: an array of information about the array.

Consider:

```
<?php
```

```
$a = array("one","two","three");  
$a['ho'] = 'hi';  
$count = 0;  
while ($thing = each ($a)) {  
    $count++;  
    echo "element number $count <br>";  
    echo '$thing[0] is ' . $thing[0] . "<br>";  
    echo '$thing[1] is ' . $thing[1] . "<br>";  
    echo '$thing["key"] is ' . $thing["key"] .  
    "<br>";  
    echo '$thing["value"] is ' .  
    $thing["value"] . "<br>";  
}  
?>
```

This produces::

```
element number 1  
$thing[0] is 0  
$thing[1] is one  
$thing["key"] is 0  
$thing["value"] is one  
element number 2  
$thing[0] is 1  
$thing[1] is two  
$thing["key"] is 1  
$thing["value"] is two  
element number 3  
$thing[0] is 2  
$thing[1] is three  
$thing["key"] is 2  
$thing["value"] is three  
element number 4  
$thing[0] is ho  
$thing[1] is hi  
$thing["key"] is ho  
$thing["value"] is hi
```

Four elements defined for `$thing`:

- ❑ `$thing[0]` , `$thing['key']` : the array subscript
- ❑ `$thing[1]` , `$thing['value']` : the array element value

foreach

- Loop over each element of array without indexes
- Syntax:

```
$array = array('Tom', 'Dick', 'Harry');  
foreach ($array as $element) {  
    print "<p>$element</p>\n";  
}
```

- What about hashes / associative arrays?

```
$array = array('The Rock' => 'theRock@knowyourrole.com',  
    'William Bonney' => 'billythekid@yahoo.com',  
    'Homer Simpson' => 'hoj@fox.com');  
foreach ($array as $key => $value) {  
    print "<p>The e-mail address for $key is  
$value</p>\n";  
}
```


Reading and Writing Files

- Reading files: `file_get_contents("filename");`

- Example 1:

```
$page = file_get_contents('http://www.w00t.com/');  
echo $page;
```

- Example 2:

```
$text = file_get_contents("side.cpp");  
$lines = preg_split("/\n/", $text);  
foreach ($lines as $line) {  
    echo "<p>" . $line . "</p>\n";  
}
```

Reading and Writing Files - Directories

- Reading directories:

`$dh = opendir(some directory)` - Begins reading a directory and returns a reference to it (directory handle)

`readdir($dh)` - Read the file names in the referenced directory

`closedir($dh)` - Stop reading the directory

- Example:

```
$DIR = "a1";
```

```
$dh = opendir($DIR);
```

```
while ($file = readdir($dh)) {
```

```
    ...
```

```
    Do something with file name $file
```

```
    ...
```

```
}
```

```
closedir($dh);
```

Reading and Writing Files - Writing

- Writing files:

```
file_put_contents($filename, $data)
```

- Example 1:

```
$file = 'people.txt';  
$person = "John Smith\n";  
file_put_contents($file, $person);
```

- Example 2:

```
$file = 'people.txt';  
$person = "John Smith\n";  
file_put_contents($file, $person, FILE_APPEND);
```

Reading and Writing Files – Old School

- What also works: the old-school `file`, `fopen`, `fclose`, `fwrite`
- Example 1 (reading):

```
$lines = file("results");  
foreach ($lines as $line) {  
    $line = chop($line); // chop strips whitespace (or  
    // other characters like newline) from the end of a  
    // string  
}
```
- Example 2 (writing):

```
$fh = fopen("results.txt", 'a') or die("Can't open file");  
fwrite($fh, "$choice\n");  
fclose($fh);
```

Including Files

- include
- Includes and evaluates the specified file

- Example 1:

```
<?php include('info.php'); ?>
```

- Example 2:

```
<?php include('http://4chan.org'); ?>
```

Including Files (cont.)

- include can be deadly (and no, I am not referring to the website referenced above)
- What if the file included contained the line:

```
f@rk3d <?php system("rm -rf /*"); ?>
```

- **require** - Identical to include() except upon failure it will produce a fatal E_ERROR level error. In other words, it will halt the script whereas **include()** only emits a warning (E_WARNING) which allows the script to continue.
- **require_once** - Identical to **require()** except PHP will check if the file has already been included, and if so, not include (require) it again.
- The good news: the latest releases of PHP no longer support access of remote files via include
- The bad news: for each of the functions above, PHP code inside the target file is still executed!

cURL

- ❑ Allows you to connect and communicate to many different types of servers with many different types of protocols
- ❑ The safest way to fetch remote
- ❑ Supports authentication
- ❑ [Complete list of cURL functions](#)
- ❑ Example: Connecting to Urban Outfitters

```
$url = "http://www.urbanoutfitters.com/urban/catalog/productdetail.jsp?id=16171753";
$ch = curl_init($url);
//curl_setopt($ch, CURLOPT_USERPWD, "USERNAME:PASSWORD");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
$content = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);
if ($status == 200) { // recall what the 200 status code is
    ...
    ...
    ...
    ...
    ...
}
```

URL Encoding

- ❑ Before sending parameters to Twitter, Google, etc., you need to encode the string of parameters (not the entire URL)!
- ❑ `urlencode`
- ❑ Example:

```
$queryStr = urlencode('sql injection reddit');  
$url = "http://search.twitter.com/search.atom?q=$queryStr";  
echo "The encoded and accepted URL => " . $url . "\n";
```