



Arrays and Interaction (1)

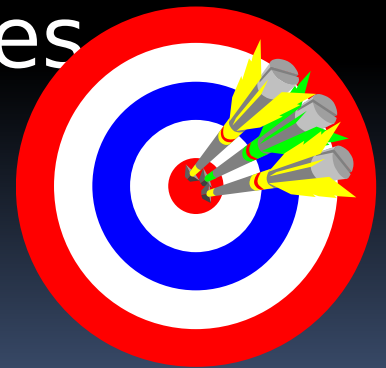
Week 6a



Website Development

In this lecture...

- We will introduce :-
 - ▣ PHP Numeric and Associative Arrays
 - ▣ Iterating (looping) over arrays
 - ▣ Interaction between pages



Arrays

- Arrays store multiple values
- Arrays names are ordinary variables
- Access maybe numeric:-

```
$Student_array[0] = "John Smith";  
$Student_array[1] = "Mary Jones";
```

or associative

```
$Student_array['top'] = "Jane Smith";
```

- Arrays can be regarded as name/value pairs

Creating and accessing arrays

- Remember that in PHP variables are created by assigning values to them
 - e.g. `$Lecturer = "JE";`
- Arrays are created in the same way
 - e.g. `$student[] = "John";` // **automatically numbered from 0**
 - `$student[3] = "Mary";`
 - `$friends['John'] = 'Mary';`
- Access is similar:
 - `$gf = $friends['John'];` // **\$gf now equals 'Mary'**
 - `$third = $student[3];` // **\$third now equals ?**

Assembling Data into Arrays

- There is a shorter way to create an array if you already know the data
- The `array()` function converts a list of arguments into an array e.g.

```
$my_pets = array("bev", "kev", "buster");
```

- Remember PHP arrays start at Zero!

Question: what is the value of

```
$my_pets[1];
```

Viewing Arrays

- Arrays may be printed using the `print_r` function
eg. `print_r($my_array);`
- **Debugging tip:**
 - ▣ To show the values of an array in your web page use the 'preformatted' tag:-

```
echo "<pre>";  
print_r($my_pets);  
echo "</pre>";
```

Looping over arrays

- Several methods
 1. For numeric arrays use 'for' loop
 2. For numeric and Associative arrays use 'foreach'
 - No need to know array length
 3. Also 'pointer' functions can be used
 - See www.php.net

foreach example -- numeric

```
$users =  
    array ("Bert", "Sharon", "Harry");  
  
foreach ( $users as $val ) {  
    print "$val<br />";  
}
```


foreach example --

Associative

```
$character = array (  
    "name" => "bob",  
    "occupation" => "superhero",  
    "age" => 30);
```

```
foreach( $character as $key => $val )  
{  
    print "$key = $val<br />";  
}
```

Useful array functions


- `count($array_1)`
 - //returns number of elements
 - e.g. `count($users)` returns 3
 - remember last element is `count($users) - 1 !`
- `sort($array)` // sorts numeric arrays
- `ksort($ass_array);` sorts by keys
- `array_key_exists()` // Checks if the given key or index exists in the array
- `array_intersect` -- Computes the intersection of arrays
- www.php.net lists 78 other functions for arrays.



PHP INTERACTION BETWEEN PAGES



Steps in client server interaction

- We will cover:
 - How to use forms to submit data
 - How to send form data to the server
 - How to use PHP to access information from a form
- 

Modifying the form

- Forms are used (primarily) used to transmit user data
- Two attributes are added to the `<form>` tag
 1. `action="scriptname.php"`
 2. `method="get"` **or** `method="post"`
- 'get' adds data to the URL
- 'post' adds data to the http request
 - Not recorded in browser history

GET Vs POST


- GET uses 'CGI' interface
 - Appends variables and values to URL
 - Start after '?'
 - Separated by '='
 - Space and other chars 'url encoded' as hex
- GET is publicly visible
- POST is hidden
- Length limit on URLs and hence GET
 - Chrome: 2MB limit for URLs, IE8 and 9 :2084 char limit

PHP Built-in Arrays

- `$_POST`:
 - Contains all the variables contained in a form where the form uses method="post".
- `$_GET`
 - Contains all the variables passed as part of the URL.
 - includes variables passed in a form using method="get".
- `$_REQUEST`
 - Contains all the variables together in `$_POST`, `$_GET`, and `$_SESSION`.
- `$_SERVER`
 - Contains information about your server.
- `$_COOKIE`
 - Contains all the cookie variables.
- `$_SESSION`
 - Contains all the session variables.



Breakout 1: Looking at Form Data

1. Start Apache
 - ✂ Test php is working
 2. Write testform.html
 - ✂ Test form constructs
 3. Write reflect.php
 - ✂ Examine form data
- 

Breakout 2: Test php is working

- ✂ Start Apache [from start, computing]

- ✂ Locate or write test.php

```
<?php
    phpinfo(); // simplest php program
?>
```

- ✂ Load into browser using apache
http://localhost/test.php

- ✂ Should show PHP configuration data

Breakout 3: Write a simple form

```
<html>
  <body>
    <form action="reflect.php" method="get">
      <p>Name:
      <input type="text" name="user" />
      </p>
      Address: <textarea name="address" rows="5"
      cols="40"> </textarea>
      </p>
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>
```

Breakout 4: Write PHP reflect.php

- This basic script shows all the data you are passing either via GET (as \$_GET) or POST, through \$_POST.

```
<?php
    echo "<pre>";
    print_r($_REQUEST);    //then $_POST
    echo "<pre>";    // or $_GET
?>
```

- Test the form (and hence the script)
 - Improve the form with html/css

Accessing individual elements

- Simple html elements may be accessed directly from \$_GET
 - e.g. try adding this to the example

```
$address = $_GET['address'];  
echo "The address is: $address <br/>";
```

Better...

```
$address =  
    htmlspecialchars($_GET['address']);
```

Other form elements ...

- There are many form elements:


- eg pull down list

```
<form action="reflect.php">
<select name="adults">
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="3">3</option>
</select>
<input type="submit" value="Submit">
</form>
```

- For different elements and examples see:
 - https://www.w3schools.com/html/html_forms.asp



Conclusion

- Introduced PHP numeric and associative arrays
 - Introduced array iteration
 - Introduced communication between html forms and PHP
 - Use GET or POST
 - Showed some basic data access from PHP
- 



Sources

- Glass et al. 2004 "Beginning PHP, Apache, MySQL Web Development" Wiley ISBN 0-7645-5744-0
 - **Ullman Larry 2005 PHP and MySQL for Dynamic Web Sites : Visual QuickPro Guide, Second Edition 9780321336576 (0321336577), Peachpit Press - Chapter 2**
 - Suehring, Steve and Valade, Janet 2013 PHP, MySQL, JavaScript & HTML5 All-in-One For Dummies . ISBN-13: 9781118228746
- 