PHP Arrays

Lecture 20

Robb T. Koether

Hampden-Sydney College

Wed, Feb 28, 2018

PHP Arrays

Initializing an Array

```
$a = array(5, 10, 15, 20);
$b[0] = 5;
$b[1] = 10;
$b[2] = 15;
$b[3] = 20;
```

- Arrays may be initialized by using the keyword **`array`**.
- Or, the array elements may be assigned individually without using the keyword.

`while` Statements

The `while` Statement

```
while (condition)
    while-block;
```

- **`while`** statements in PHP are just like **`while`** statements in C.

Displaying Arrays

Displaying an Array

```
$a = array(5, 10, 15, 20);
$i = 0;
while ($i < count($a))
{
echo $a[$i] . ' ';
$i++;
}
```

produces

Output

```
5 10 15 20
```

`for` Statements

The `for` Statement

```
for (initial; condition; update)
    for-block;
```

- **`for`** statements in PHP are just like **`for`** statements in C.

Displaying Arrays

Displaying an Array

```
$a = array(5, 10, 15, 20);
for ($i = 0, $i < count($a), $i++)
echo $a[$i] . ' ';
```

produces

Output

```
5 10 15 20
```

`foreach` Statements

The `foreach` Statement

```
foreach (array_name as var_name)
    Process var_name
```

- The **`foreach`** statement will process an array's elements.

Displaying Arrays

The `foreach` Statement

```
foreach ($a as $b)
echo $b . ' ';
```

- Or, we may use a **`foreach`** loop.

foreach Statements

The foreach Statement

```
foreach (array_name as index_name => var_name)
    Process index_name and var_name
```

- The **foreach** statement will also process an array's elements and their indexes.

Displaying Arrays

The `foreach` Statement

```
foreach ($a as $i => $b)
echo "Element $i is $b<br/>";
```

- Or, we may use a **`foreach`** loop.

Displaying Arrays

The `print_r()` Function

```
print_r($a);
```

- Or, we may use the `print_r` function.
- This is more appropriate for debugging.
- The above statement will produce the following.

Output

```
Array ( [0] => 5 [1] => 10 [2] => 15 [3] => 20 )
```

Displaying Arrays

The `var_dump()` Function

```
var_dump($a);
```

- Or, we may use the `var_dump` function.
- This is also very helpful in debugging.
- The above statement will produce the following.

Output

```
array(4) { [0]=> int(5) [1]=> int(10) [2]=> int(15)
    [3]=> int(20) }
```

Displaying Arrays

Heterogeneous Arrays

```
$a = array(5, "dog", true, 6.99);
var_dump($a);
array(4) { [0]=> int(5) [1]=> string(3) "dog"
    [2]=> bool(true) [3]=> float(6.99) }
```

- Arrays in PHP may be heterogeneous.

Associative Arrays

Associative Arrays

```
variable_name = array(index => value,...)
```

- In PHP, arrays may be associative instead of numerically indexed.
- In an associative array, two lists of values are associated as a list of paired elements.
  - The first member of each pair serves as the index.
  - The second member of each pair serves as the value.
- Each index must be unique.

Associative Arrays

Associative Arrays

```
$author = array(
    "A Tale of Two Cities" => "Charles Dickens",
    "Treasure Island" => "Robert Louis Stevenson",
    "Tom Sawyer" => "Mark Twain",
    "Oliver Twist" => "Charles Dickens",
    "Roughing It" => "Mark Twain"
);
```

Associative Arrays

Associative Arrays

```
$title = "Treasure Island";
echo "The book " . $title . " was written by "
     . $author[$title];
```

produces

```
The book Treasure Island was written by Robert
Louis Stevenson
```

Associative Arrays

Associative Arrays and `foreach`

```
foreach ($author as $a)
    echo "The author is $a<br/>";
```

produces

```
The author is Charles Dickens
The author is Robert Louis Stevenson
The author is Mark Twain
The author is Charles Dickens
The author is Mark Twain
```

Associative Arrays

Associative Arrays and `foreach`

```
foreach ($author as $ttl => $aut)
    echo "The author of $ttl is $aut<br/>";
```

produces

```
The author of A Tale of Two Cities is Charles Dickens
The author of Treasure Island is Robert Louis Stevenson
The author of Tom Sawyer is Mark Twain
The author of Oliver Twist is Charles Dickens
The author of Roughing It is Mark Twain
```

Associative Arrays

PHP Arrays and HTML Tables

```
echo "<table>";
echo "<tr><th>Title</th><th>Author</th></tr>";
foreach ($author as $ttl => $aut)
    echo "<tr><td>$ttl</td><td>$aut</td></tr>";
echo "</table>";
```

- The above code will produce a table of titles and authors.

Accessing MySQL from PHP

- PHP provides several functions for opening and querying a database through MySQL.
- The function `mysql_connect()` will connect to the MySQL server.
- The function `mysql_select_db()` will open a database.
- See `http://php.net/manual/en/book.mysql.php`.
- See `http://php.net/manual/en/book.mysqli.php` for the new, improved interface.
- See `http://php.net/manual/en/mysqli.overview.php`.

Connecting to MySQL

The `mysqli_connect()` Function

mysqli mysqli_connect(*hostname*,*username*, *password*)

where

- *hostname* is the name of the server (`localhost`).
- *username* is the MySQL username
- *password* is the MySQL password

- This function returns a “link” that represents the connection to the database.
- If the connection fails, the value of the link is **`false`**.

Connecting to MySQL

Example

```
$hostname = "localhost";
$username = "billybob";
$password = "euclid";
$mysqli = mysqli_connect($hostname, $username, $password);
```

- This example logs in to MySQL.

Testing the Connection

Testing the Connection

```
$mysqli = mysqli_connect($hostname, $username, $password);
if (!$mysqli)
{
    die("Failed to connect (" . mysqli_connect_errno() . ") "
        . mysqli_connect_error());
}
```

- We should always test whether the connection to MySQL was successful.
- The `die()` function will display the message and exit the PHP program.

Opening a Database

The `mysqli_select_db()` Function

```
bool mysqli_select_db(mysqli, database)
```

where

- *`mysqli`* is the link returned by `mysqli_connect()`.
- *`database`* is the name of the database.

- This function returns a boolean value that tells whether it was successful.
- The resource parameter is unnecessary if there is only one connection.

Opening a Database

Example

```
$database = "company";
$success = mysqli_select_db($link, $database);
if (!$success)
    die("Failed to open database " . $database);
```

- This example will open the database `company`.

Assignment

Assigment

- Read the documentation at the web page `http://php.net/manual/en/mysqli.overview.php`.