

IT420: Database Management and Organization

PHP Arrays, Files, Functions

PHP Summary

- PHP tags `<?php ?>`
 - Mixed with HTML tags
 - File extension `.php`
- Statements
 - Separated by semicolon
 - `if..else..`, `while`, `do`, `for`, `switch`
- Variables
 - `$varname`
 - Type determined by content; variables not declared; case sensitive
- Strings
 - Single quotes – literal string
 - Double quotes – interpolated string (variables are replaced with their value)
- Accessing form variables
 - `$_POST[age]`, `$_GET[age]` (if method is GET), `$_REQUEST[age]`

PHP Summary

- PHP objects
 - Java-like inheritance
 - `public`, `private`, or `protected` attributes and methods
 - `__construct()`, `__destruct()`,
 - `__set()`, `__get()`
- PHP functions
 - function `myFunction($param1, $param2){...}`
- Files
 - `resource fopen(string $fileName, string $mode)`
 - `int fwrite(resource $handle, string $someText)`
 - `int fclose(resource $handle)`
 - `string fgets(resource $handle)`
 - `boolean feof(resource $handle)`

```
class_page.inc.php
<?php
class Page{
    //attributes
    public $content;
    public $title = "IT420 page";

    private $header = "<?xml version = '1.0' ?>
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    'http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd'>";

    //constructor
    public function __construct(){
    }

    //set public attributes
    public function __set($name, $value){
        $this->$name = $value;
    }

    //display page
    public function display(){
        echo $this->header;
        echo "<head><title> $this->title </title></head>";
        echo "<body>";
        echo $this->content;
        echo "</body></html>";
    } //end class definition
?>
```

Outline

- Arrays
- More About Files
- More About Functions

Arrays

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"
<html><head><title>IT420 Arrays</title></head>
<body>
<?php
require_once('array_functions.inc.php');

//create array
$numbers = array('zero', 'one', 'two');
$numbers[3] = 'three';

//display array
display_array('The content of $numbers array is:', $numbers);

//create associative array
$numbers2 = array('zero' => 0, 'one' => 1, 'two' => 2);
$numbers2['three'] = 3;

//display associative array
display_assoc_array('The content of $numbers2 array is:', $numbers2);

//sort array
sort($numbers);
display_array('The content of $numbers array in sorted order is:', $numbers);

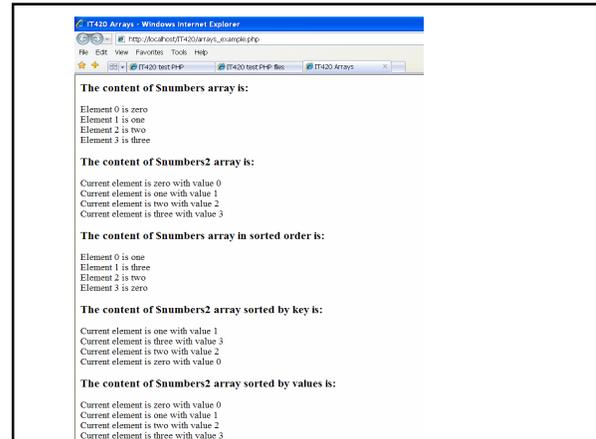
$numbers3 = $numbers2;
ksort($numbers2);
display_assoc_array('The content of $numbers2 array sorted by key is:', $numbers2);

asort($numbers3);
display_assoc_array('The content of $numbers2 array sorted by values is:', $numbers3);
?>
</body></html>
```

array_functions.inc.php

```
<?php
function display_array($caption, $array){
    echo "<h3> $caption </h3> <p>";
    for($i=0; $i< count($array); $i++){
        echo "Element $i is $array[$i] <br />";
    }
    echo "</p>";
}

function display_assoc_array($caption, $array){
    echo "<h3> $caption </h3> <p>";
    foreach($array as $key => $value){
        echo "Current element is $key with value $value <br />";
    }
    echo "</p>";
}
?>
```



Class Exercise

- Create an array containing the following values: (1900, 2000, 2004, 2005).
- Use the array in a “foreach” loop to test if the value is a leap year.
- If it is a leap year print “XXXX is a leap year”
- Else print “XXXX is not a leap year”
- A year is a leap year if it is divisible by 4. If the year is also divisible by 100 then it is NOT a leap year unless it also divisible by 400. Thus, 1900 is not a leap year (divisible by 100 but not 400) while 2000 is a leap year (divisible by 400).

Outline

- Arrays
- **More About Files**
- More About Functions

fopen() modes

r	Read only. Starts at beginning of file
r+	Read/Write. Starts at beginning of file
w	Write only. Opens and clears contents of file; or creates new file if it doesn't exist
w+	Read/Write. Opens and clears contents of file; or creates new file if it doesn't exist
a	Write only. Opens and writes to end of file or creates new file if it doesn't exist
a+	Read/Write. Preserves file content by writing to end of file
x	Write only. Creates new file. Returns FALSE and error if file already exists
x+	Read/Write. Creates new file. Returns FALSE and error if file already exists

More Files: Read Line from File

- *string* = `fgets(filePointer, [maxLength])`
- *string* = `fgetss(filePointer, [maxLength] [, allowableTags])`
- *array* = `fgetcsv(filePointer, [maxLength] [, string delimiter])`

Read Whole File

- *int* `readfile(fileName)`
- *int* `fpasssthru(filePointer)`
- *array* `file(fileName)`
- *string* `file_get_contents(fileName)`

Other Reads

- *char* `fgetc(filePointer)`
- *string* `fread(filePointer, nbBytes)`

Useful File Functions

- *bool* `file_exists(fileName)`
- *int* `filesize(fileName)`
- *bool* `unlink(fileName)`

File Locking

```
function save_to_file($text, $fileName = "myFile.txt"){  
    $fp = @fopen($fileName, 'a');  
  
    if (!$fp){  
        echo "<p>ERROR: Could not open file $fileName. </p>";  
        return FALSE;  
    }  
    else{  
        flock($fp, LOCK_EX);  
  
        fwrite($fp, $text);  
  
        flock($fp, LOCK_UN);  
  
        fclose($fp);  
        return TRUE;  
    }  
}
```

File Permissions

- Ownership: Input/Output files usually **NOT** owned by "Web Server".
- Operating system may enforce read, write, and/or modify restrictions on I/O files
 - For file output/append, may need to create file prior to first use
 - File permissions need set for access by the "web server" account (Right-click on file, pick Properties, then set permissions like example on right)

Class Exercise

- Create PHP script to:
 - Open/create a file, without overwriting it
 - Write the numbers 1 to 20 to file, separated by space
 - Close the file

Outline

- Arrays
- More About Files
- **More About Functions**

Function Parameters

```
<?php
function start_table($border, $cellspacing=2,
    $cellpadding=2){
    echo "<table border = $border
        cellspacing = $cellspacing
        cellpadding = $cellpadding>";
    }
?>
```

- start_table(2,3,4)
- start_table(1)
- start_table(2,3)

Functions with Variable Number of Parameters

```
<?php
function print_function_params(){
    echo 'Number of parameters:';
    echo func_num_args();

    echo '<br />';
    $args = func_get_args();
    foreach($args as $arg){
        echo $arg . '<br />';
    }
}

print_function_params('one', 'two');
print_function_params();
print_function_params(1,2,3);
?>
```

Variables Scope

- Variables declared in functions are visible from declaration line to end of **function – local variables**
- Variables declared outside functions are visible from declaration line to end of **file, but not inside functions – global variables**
- **Superglobal variables** (\$_POST, \$_SERVER, ...) are visible **everywhere**
- Keyword *global* makes local variables global – not recommended

Variables Scope Example

```
<?php
function fn(){
    $var = 'content';
}
fn();
echo $var;
?>
```

Variables Scope Example 2

```
<?php
$var = 'content 1 <br/>';
echo $var;
function fn(){
    echo $var;
    $var = 'content 2 <br/>';
    echo $var
}
fn();
echo $var;
?>
```

Writing Reusable Code

- Typical page header:

```
<?xml version = '1.0' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML1.1//EN"
'http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd'>
<html>
<head>
<title> My Page Name </title>
</head>
<body>
```

- Type same code for every page?
- What if want to change title?

Include files

- header.php

```
<?xml version = '1.0' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML1.1//EN"
'http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd'>
<html>
<head>
<title> <?php echo $pageTitle; ?></title>
</head>
<body>
```

Include example (cont)

- index.php

```
<?php
$pageTitle = "IT420 Page";
include ("header.php");
echo "Some content for my page";
?>
```

Generated HTML Page

```
<?xml version = '1.0' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML1.1//EN"
'http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd'>
<html>
<head>
<title> IT420 Page </title>
</head>
<body>
Some content for my page
```

Summary

- Arrays
 - Numerically indexed
 - Associative
 - Sort
- Files
 - Functions to work with files
 - File locking
- Functions
 - Optional/variable number of parameters
 - Variable scope