PHP

Comp-206 : Introduction to Software Systems
Lecture 24

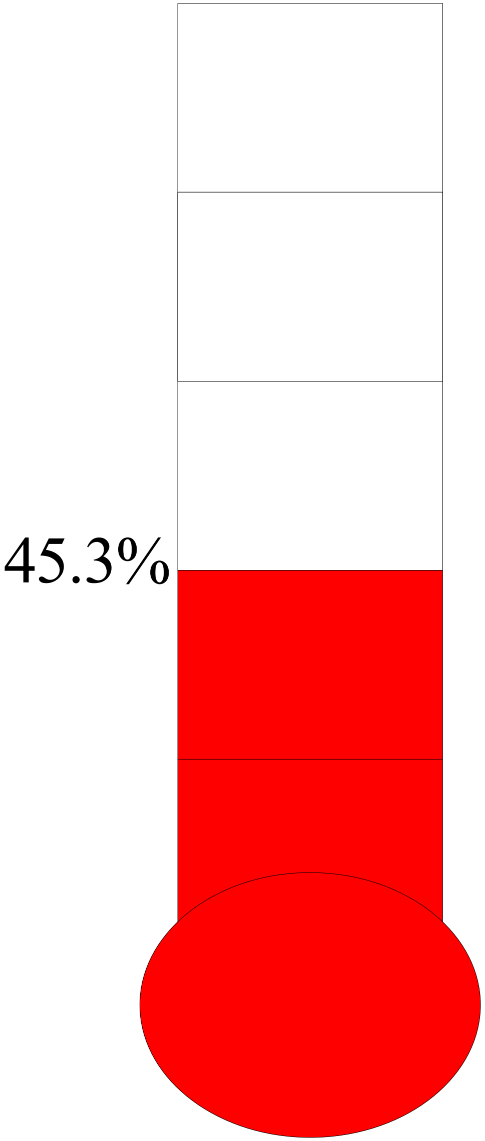
Alexandre Denault
Computer Science
McGill University
Fall 2006

Office Hours

- Office hours on Tuesday, December 5th are canceled
- However, I will be in the office in the morning (from 9h00 to 13h00).
- In addition, I should be in the office on the following days:
  - Wednesday, December 6th
  - Thursday, December 7th
  - Monday, December 11th
  - Tuesday, December 12th (afternoon)
- To make sure I'll be there, send me a mail.

Course Evaluation - Mercury

24 / 53

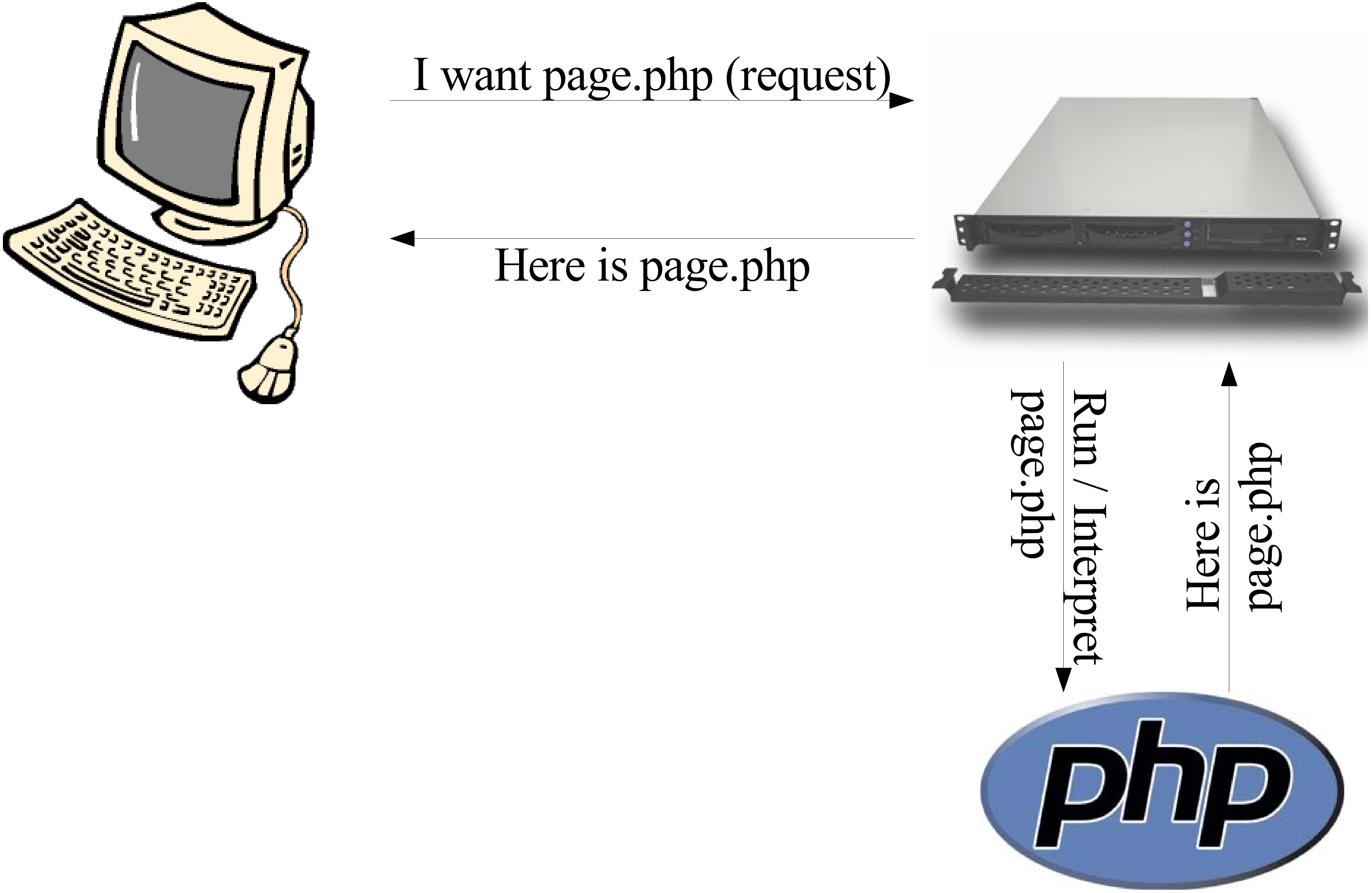


What is PHP?

PHP ...

- initially stood for Personal Home Page Tools.
- was created as libraries to write pages in C.
- is a web scripting language.
  - can also be used for shell scripts or graphic applications.
- is used to build dynamic websites.
- is an object oriented language.
- has a syntax very similar to C.
- is open source (free).

PHP Execution Model



I want page.php (request)

Here is page.php

Run / Interpret page.php

Here is page.php

Your first PHP Page

```
<html>
    <head>
       <title>PHP Test</title>
    </head>
    <body>
       <?php
       echo '<p>Hello World</p>';
       ?>
    </body>
</html>
```

The PHP tags

- A php file must have the .php extension to be properly interpreted by the web server.
- Any text found between the “<?php” and “?>” tags will be considered PHP code.
- Most instructions in PHP must be terminate by a semi-colon.

Variable

- Variables are always preceded by the $ sign.
- Variables do not need to be declared before they are used.
  - You simply need to assign them a value.
  - PHP has a very loose type system.
  - Unlike C, PHP has strings and booleans.

```
$firstVar = "Hello";
$secondVar = "World!";
$boolVar = true;
```

Variable Examples

```
<?php
 error_reporting(E_ALL);
 $firstVar = "Hello";
 $secondVar = "World!";
?>
<html>
   <head>
      <title>PHP Test</title>
   </head>
   <body>
   <?php
      echo '<p>'.$firstVar.' '.$secondVar.'</p>';
   ?>
   </body>
</html>
```

Language Constructs

- Using conditional statements and looping statements is identical to C.
- Creating functions and using them is identical to C.
    - PHP is not single pass. You do not need to use function prototypes.

Simple function

```
<?php
function makeCountString($num = 8) {

   $value = " ";

   for ($i = 0; $i < $num; $i++) {
      $value = $value . $i;
   }

   return $value;
}
?>
```

Arrays

- Arrays in PHP function like hash tables, or array maps.
  - They are data structures that hold key/value pairs.
- They can be created using the array() function.

```
<?php
$trad = array(“hello”, “to”, “the”, “world”);
echo $trad[0]; // “hello”
echo $trad[3]; // “world”

$spec = array("foo" => "bar", 12 => true);
echo $spec["foo"]; // “bar”
$spec[“foo”] = “test”;
echo $spec[12];    // true
?>
```

The $_REQUEST array

- When a form is “POST” or “GET”, the values of the form are stored in the $_REQUEST array.
- You can use these values however you want.

```
Hi <?php echo
  htmlspecialchars($_REQUEST['name']); ?>.
You are <?php echo (int)$_REQUEST['age']; ?>
  years old.
```

- Note that $_REQUEST is data from the user. As such, it should not be trusted.

Session Variables

- Each time a page loads, it's a new instance of that page (variables only exist for that page load).
- If you want a variable, for a given user, to persist from one page to the other, you need to save it into the session array.
- To access values in the session, you need to call session_start() as the first function in your script.
- You can then use $_SESSION as any regular array.
- When the page is processed, the content of $_SESSION will be saved.
- To save an object in session, you will need to *serialize* it.
- To restore an object from session, you will need to *unserialize* it.

Classes

- PHP is an object-oriented language, like Java or Python.
- As such, it allows the creation of classes.
  - Classes are grouping of variables and functions.
- Classes can be declared using the “class” keyword”.
  - Member variables must be defined using the “var” keyword.
  - Functions now need to be contained in the scope of the class.
- Like Java, variables can be declared as public, protected and private.
- You can also create static members of classes.

Class Example

```
<?php

class Book {

    var $author;
    var $title;
    var $pages;

    function  __construct($title, $author, $pages) {
        $this->title = $title;
        $this->author = $author;
        $this->pages = $pages;
    }
```

Class Example (cont.)

```
    function prettyString() {
        return $this->title . “ | “ . $this->author . “ | “ . $this->pages . “ pages.”
    }

    function set($var, $value) {
        $this->$var = $value;
    }
}

?>
```

Using a Class

- Instantiating an object is pretty simple, you simply need to use the new keyword.

```
$myBook = new Book(“Lotr”, “Tolkien”, 300);
```

- You can then use the “->” to access the variables and method in the object.

```
echo $myBook->title;
echo $myBook->set(“title”, “Lord of the Rings”);
echo $myBook->prettyString();
```

Inheritance

- PHP classes can extend other PHP classes.
- If the subclass declares a constructor, the constructor of the parent is no longer called.
  - You need to call it explicitly: parent::__construct();

```
class TextBook extends Book {

   var $subject;

   function prettyString() {
      return $this->title . “ | “ . $this->author . “ |
        “ . $this->pages . “ pages.”
   }

}
```

Include

- Although classes and functions do not need to be declared before they are used, they do need to be declared.
- For scripts split into multiple files, PHP has the include() function that evaluates the specified file.

```
 include ("Book.class.php");
```

- If the file is not found, include just throws a warning. To do a force include, use the require function() instead.
- A function or a class should only be defined once. To prevent multiple inclusion, you can use the include_once() function or the require_once() function.

Common Functions (Variables)

- empty -- Determine whether a variable is empty
- floatval -- Get float value of a variable
- intval -- Get the integer value of a variable
- is_array -- Finds whether a variable is an array
- is_bool -- Finds out whether a variable is a boolean
- is_callable -- Verify that the contents of a variable can be called as a function
- is_double -- Alias of is_float()
- is_float -- Finds whether a variable is a float
- is_int -- Find whether a variable is an integer
- is_null -- Finds whether a variable is NULL

Common Functions (Variables)

- is_numeric -- Finds whether a variable is a number or a numeric string
- is_object -- Finds whether a variable is an object
- is_string -- Finds whether a variable is a string
- isset -- Determine whether a variable is set
- serialize -- Generates a storable representation of a value
- strval -- Get string value of a variable
- unserialize -- Creates a PHP value from a stored representation
- unset -- Unset a given variable
- var_dump -- Dumps information about a variable

Common Functions (String)

- echo -- Output one or more strings
- htmlspecialchars -- Convert special characters to HTML entities
- ltrim -- Strip whitespace (or other characters) from the beginning of a string
- rtrim -- Strip whitespace (or other characters) from the end of a string
- nl2br -- Inserts HTML line breaks before all newlines in a string
- printf -- Output a formatted string
- strlen -- Get string length
- strpos -- Find position of first occurrence of a string
- strstr -- Find first occurrence of a string
- strtolower -- Make a string lowercase
- strtoupper -- Make a string uppercase
- strtr -- Translate certain characters
- substr -- Return part of a string

Common Functions (Files)

- copy -- Copies file
- disk_free_space -- Returns available space in directory
- feof -- Tests for end-of-file on a file pointer
- fclose -- Closes an open file pointer
- fgetc -- Gets character from file pointer
- fgets -- Gets line from file pointer
- fgetss -- Gets line from file pointer and strip HTML tags
- file -- Reads entire file into an array
- fopen -- Opens file or URL
- fpassthru -- Output all remaining data on a file pointer
- fread -- Binary-safe file read
- fwrite -- Binary-safe file write
- mkdir -- Makes directory
- touch -- Sets access and modification time of file
- unlink -- Deletes a file

Libraries

- PHP has libraries (functions) to manipulate many different types of files:
  - Images : png, jpg, gif
  - Documents : pdf
  - Archives : zip, rar, bzip2
  - etc
- PHP can also communicate with different types of databases:
  - MySQL
  - Microsoft SQL
  - PostgreSQL
  - Oracle
  - Sybase
  - etc

phpinfo()

- This function outputs a lot of information about PHP including :
  - The version of PHP
  - The currently compiled modules (MySQL, Images, Zip, etc)
  - The current configuration of PHP (including the err. rep. Level)
  - The current environment (variables)
  - All predefined variables (including $_REQUEST)
  - And more ...

Error Level

- The amount of debug information given is determined by the current error reporting level.
- The default error reporting level is determined by the system administrator.
  - On a production machine, the error reporting level will be very low (only critical errors will be shown).
  - On a development machine, the error reporting level will be very high (all errors and warnings will be displayed).
- You can change the error reporting level using the error_reporting() function.

```
// Turn off all error reporting
error_reporting(0);
// Report all PHP errors (bitwise 63 may be used in PHP 3)
error_reporting(E_ALL);
```

Databases

- To truly build dynamic website, you will need a form of perpetual storage.
- Although files can be used this way, they are not efficient when a large number of them are needed.
- The practical solution is to store information inside a database.
  - Databases are outside the scope of this course.
  - If you are serious about web development, you should start taking a look at MySQL.
- As previously mentioned, MySQL has all the necessary functions to access a database.

Ressources

- PHP : http://www.php.net
- PHP Classes : http://www.phpclasses.org/

Tools

- To program in PHP, you need a webserver with the PHP module.
  - Apache
  - PHP
- A database could also be useful.
  - MySQL
- You can download packages with all three:
  - VentrigoServ http://vertrigo.sourceforge.net/
  - Xampp http://www.apachefriends.org/en/xampp.html