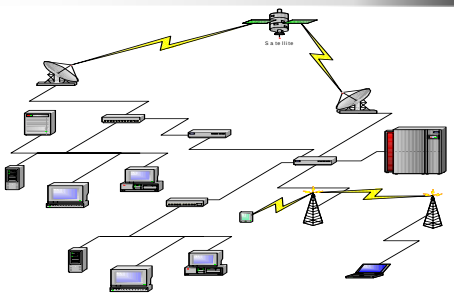
Web Development (Serverside) Using php

Some figures copied from Database Management with Web Site Development Applications by Greg Riccardi (with permission)

© Northern Illinois University
June 2003
Updated March 2009

Internet as Collection of Connected Computers, etc.



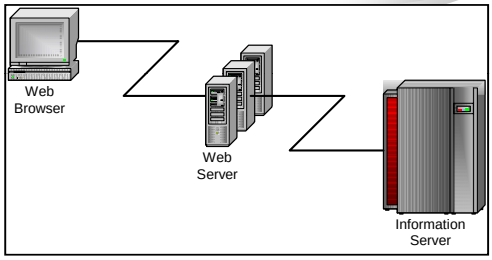
2

Components of Web-Services

- Web browser
  - formats and displays Web pages
- Web server
  - sends Web pages to browsers and lets site visitors enter and request information
- Information server
  - accepts requests from the Web server and uses its stored data to respond appropriately

3

Components of Web-Services



4

Database Server

- A database server is a kind of information server
- It stores information in databases
- Web servers, etc. connect to the database server
  - to send queries
  - to update data

5

Web Pages and HTML

- An HTML document is used to create the format and structure of a Web page
- HTTP is a communication protocol that specifies how two or more things are expected to interact on the Web

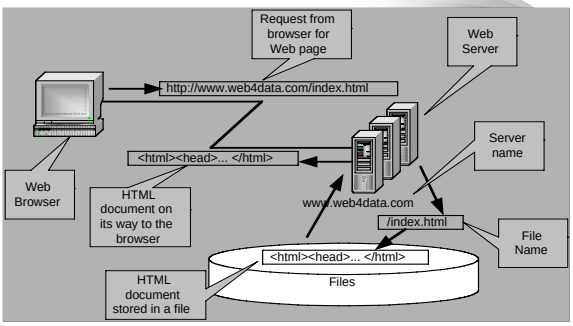
6

Web and Information Systems

- Web server
  - a computer system
  - responds
    - to requests for Web pages
    - by processing the requests and returning a new Web page to the browser

7

Architecture of Web Sites



8

Preparing to Use php

- The php language was designed to help developers create dynamic and data-driven web pages
- php interacts with one main external tool, the MySQL database management system, to access data stored in a database
- MySQL must be installed on a functional Web server to interact with php, but this is a relatively easy step in setting up the php environment

9

Preparing to Use php

- php is a server-side scripting language that you can embed into HTML documents
- You can also embed HTML in php scripts
- php scripts are parsed and interpreted on the server side of a Web application
- php is popular with web developers and web designers alike, and is powerful and easy to use

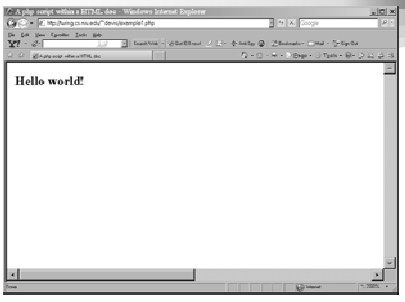
10

Exploring php for the First Time

- Internet references for php:
  - http://php.net/docs.php
  - http://www.w3schools.com/php/
- To start working with php, you can create a script that contains HTML code
- You can create the script using any text editor, such as Notepad in Windows or TextPad on the Macintosh

11

Exploring php for the First Time



12

Exploring php for the First Time

- One method:

```
<? php
print("Hello, world!");
?>
```

Start and end tags of php

- Another method:

```
<SCRIPT LANGUAGE="php"
Echo "Hello, world!";
</SCRIPT>
```

Start and end tags of php

13

Displaying php Output

- php has two functions that allow display: echo and print
  - The only difference between echo and print is that the print function returns a 1 or 0 integer (denoting success or failure, respectively), for the contents of the function being displayed
- Also, be aware that if you want to send php reserved characters (such as double quotations) to the Web browser within the echo command, you must use the backslash character

14

Understanding php Basics

- All statements in php are terminated with a semicolon (;)
- If you forget to end a statement with a semicolon and you receive an error, you should first look for a missing semicolon on the first or second line prior to the reported line
- Parsing errors are often caused by missing semicolons

15

Combining HTML and php

example1.php

```
<html>
<head>
<title> A php script within a HTML doc</title>
</head>
<body>
<b>
<?php
echo "Hello world!";
?>
</b>
</body>
</html>
```

php script

16

Combining HTML and php

```
<html>
<head>
<title> A php script within a HTML doc</title>
</head>
<body>
<b>
Hello world!
</b>
</body>
</html>
```

Source that would be seen in the browser—notice no php.

17

Defining php Variables

- Variables in php are preceded with a dollar sign ($) and contain either letters or numbers
- php is called a loosely typed programming language, meaning that you don't have to predefine your variables; you can define and use them as needed

18

Defining php Variables

- You do have to follow certain rules for naming a variable:
  - Precede the variable name with a dollar sign ($)
  - Assign the variable a meaningful name that you can remember in the future
  - Name the variable with uppercase or lowercase letters, numbers, or the underscore (_) character

19

Defining php Variables

- You do have to follow certain rules for naming a variable:
  - Do not allow the first character after the ($) to be a number
  - Variable names are case sensitive
  - Assign the variable an initial value with a single equals (=) sign

20

Using Variable Scope

- If a variable is defined at the start of a php file, it stays in memory until the end of that file
- This is known as the variable's *scope*
- If a variable is assigned a value of 5 in one php file, and that file calls another php file that has a variable of the same name, then the first variable is terminated and its value is lost

21

Using Variable Scope

- One major distinction that relates to a variable's scope involves the processing of web-based forms
- Any variables that are defined within a php/HTML form and sent to the server with the form's Post method are automatically sent with the called Post action and named in php by the same name used in the HTML form

22

Variable Data Types

Type	*Example*	*Description*
Boolean	TRUE	
Integer	5	
Float or double	3.14	
String	"hello"	
Object		An instance of a class
Array		Ordered set of keys and values
Resource		Reference to a third-party resource (a database for example)
NULL		An unintialized variable

23

Variable Data Types

- Test the type of a variable by using the built-in php function gettype().
- There are also many functions you can use with numbers. Two nice ones are round() and number_format().
  - round() – rounds a decimal to either the nearest integer or to a specified number of digits. Round($n,2) will give 2 digits to the right of the decimal point.
  - number_format() - makes a number appear in the more commonly written format (adding commas where appropriate) and you can specify digits to the right of the decimal point.

24

gettype()

```
<html>
<head>
<title> A php script using getttype</title>
</head>
<body>
<b>
<?php
    $test;  //declare with no type
    echo gettype($test);  //should be null
    echo "<br>";   //inserting an html formatting tag
    $test = 5;
    echo gettype($test);   //now should be an integer
    echo "<br>";   //inserting an html formatting tag
    $test = "five";
```

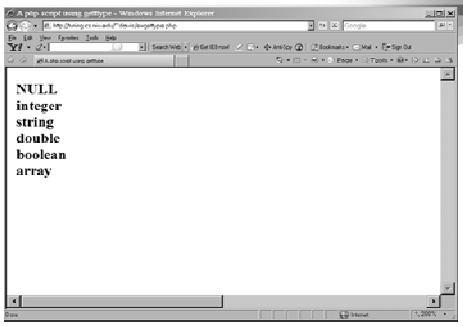
25

gettype()

```
    echo gettype($test);   //now should be a string
    echo "<br>";   //inserting an html formatting tag
    $test = 5.123;
    echo gettype($test);   //now should be a float/double
    echo "<br>";   //inserting an html formatting tag
    $test = true;
    echo gettype($test);   //now should be boolean
    echo "<br>";   //inserting an html formatting tag
    $test = array('apple','orange','pear');
    echo gettype($test);   //now should be an array
    echo "<br>";
?>
</b>
</body>
</html>
```

26

gettype()



27

Super Global Variables

- Super Global Variables – pre defined in php, these are always present and their values available to all your scripts.
  - $_GET contains any variables provided to a script through the GET method
  - $_POST contains any variables provided to a script through the POST method
  - $_COOKIE contains any variables provided to a script through a cookie
  - $_FILES contains any variables provided to a script through file uploads

28

Super Global Variables

- Continued: Super Global Variables – pre defined in php, these are always present and their values available to all your scripts.
  - $_SERVER contains information such as headers, file paths, and script locations
  - $_ENV contains any variables provided to a script as part of the server environment
  - $_REQUEST contains any variables provided to a script via any user input mechanism
  - $_SESSION contains any variables that are currently registered to a session

29

php Operators

PHP operator	Description
Variable operators	
=	Value assignment operator: $user = "Peter";
+	Addition operator: $value = $one + $two;
-	Subtraction operator: $value = $one - $two;
*	Multiplication operator: $value = $one * $two;
/	Division operator: $value = $one / $two;
%	Modulus (remainder) operator: $value = $one % $two;
Comparison operators	
= =	Equality: $this == $that
===	Identical values (identical value and data type): $this === $that
!= (or) <>	Not equal: $this != $that
<	Less than: $this < $that
>	Greater than: $this > $that
<=	Less than or equal to: $this <= $that
>=	Greater than or equal to: $this >= $that
Logical operators	
!	NOT: ! $this => Returns TRUE if $this is False
&&	AND: $this && $that => Returns TRUE if both $this and $that are true
||	OR: $this || $that => Returns TRUE if either or both $this and $that are true

30

Using Comments in Code

- Like most computer languages, php allows you to add explanations to the code in the form of comments
- These comments are ignored by the php parserComments should be added whenever necessary to explain code that is hard to follow
- To insert a comment in a single line of php code, you preface the comment with either a pound symbol (#) or two forward slashes (//)

31

Constants in php

- use php's builtin define() function
- define("YOUR_CONSTANT_NAME", value)
- You can set your constant to a number, a string, or a boolean.
- By convention, use all caps for name of a constant.
- You don't use a $ when accessing a constant.

32

Managing php Program Flow

- You use the following four constructs to manage the flow of your php programs:
  - If-then-else
  - Switch-case
  - For-next
  - Do-while

33

if-then-else

```
if (expression)
{
    //code to execute
}
```

```
if (expression)
{
    //code to execute
}
else
{
    //code to execute
}
```

34

if-then-else

```
if (expression)
{
    //code to execute
}
else if (expression)  //else if == elseif
{
    //code to execute
}
else
{
    /code to execute
}
```

35

Switch-Case

```
switch (expression) {
   case result1:
        //code to execute
        break;
   case result2:
        //code to execute
        break;
   default:
        //execute if no break has been encountered
        break;
   }
```

36

For Loop

```
for (init expression; test expression;
   modification expression)
{
   //code to execute
}
```

Note: zero, an undefined variable or an empty string will all evaluate to false, all others will evaluate to true.

37

Do While

```
while (expression)
{
   //code to execute
}
```

```
do {
   //code to execute
} while (expression)
```

38

Arrays

- There are two ways to define an array in php.
  - $colors = array("red","green","blue");
  - $colors[0] = "red";
    $colors[1] = "green";
    $colors[2] = "blue";
- These are both numerically indexed arrays.

39

Arrays

- You can also have associative arrays which have named keys.
  - $character = array(
    "name" => "Monk",
    "occupation" => "detective"
    );
    - name and occupation are the keys,
    - Monk and detective are the associated values.
- You access an element of an associative array by using the key name rather than a number.

40

Arrays

- There are approximately 60 array functions built into php. You can find them all at www.php.net/array
  - count() and *sizeof()* return the number of elements in the array.
  - *foreach()* steps through an array
  - *each()* and list() usually appear together in the context of stepping through an array and returning keys and values

41

Arrays

- *reset()* rewinds the pointer to the beginning of the array
- *array_push()* adds elements at the end of an existing array
- *array_pop()* removes and returns the last element in an existing array
- *array_merge()* combines two or more existing arrays
- *shuffle()* randomizes the elements of a given

42

Including Files

- When developing more than a single home page for the Internet, you probably want the pages to have a common look and feel
- To make this possible, php has provided a method called include files
- These files let you incorporate common artwork, contact information, and menu and link options into your Web pages with a minimum of code

43

Including Files

- **Basic include() example**
- vars.php

```
<?php
$color = 'green';
$fruit = 'apple';
?>
```

test.php

```
<?php
echo "A $color $fruit";      // Output - A
include 'vars.php';
echo "A $color $fruit";      // Output - A green apple
?>
```

44