INTRODUCTION TO PHP

SAMITHA SOMATHILAKA

Department of Computing & Mathematics, WIT

CLIENT/SERVER ON THE WWW

- Standard web sites operate on a request/response basis
- A user requests a resource E.g., HTML document
- Server responds by delivering the document to the client
- The client processes the document and displays it to user

SERVER-SIDE PROGRAMMING

- Provides web site developers to utilise resources on the web server
- Non-public resources do not require direct access from the clients
- Allows web sites to be client agnostic (unless JavaScript is used also)
- Most server side programming script is embedded within markup (although does not have to be, sometimes better not to)

PHP

INTRODUCTION TO PHP

"PHP is a server-side scripting language designed specifically for the Web. Within an HTML page, you can embed PHP code that will be executed each time the page is visited. Your PHP code is interpreted at the Web server and generates HTML or other output that the visitor will see" ("PHP and MySQL Web Development", Luke Welling and Laura Thomson, SAMS)

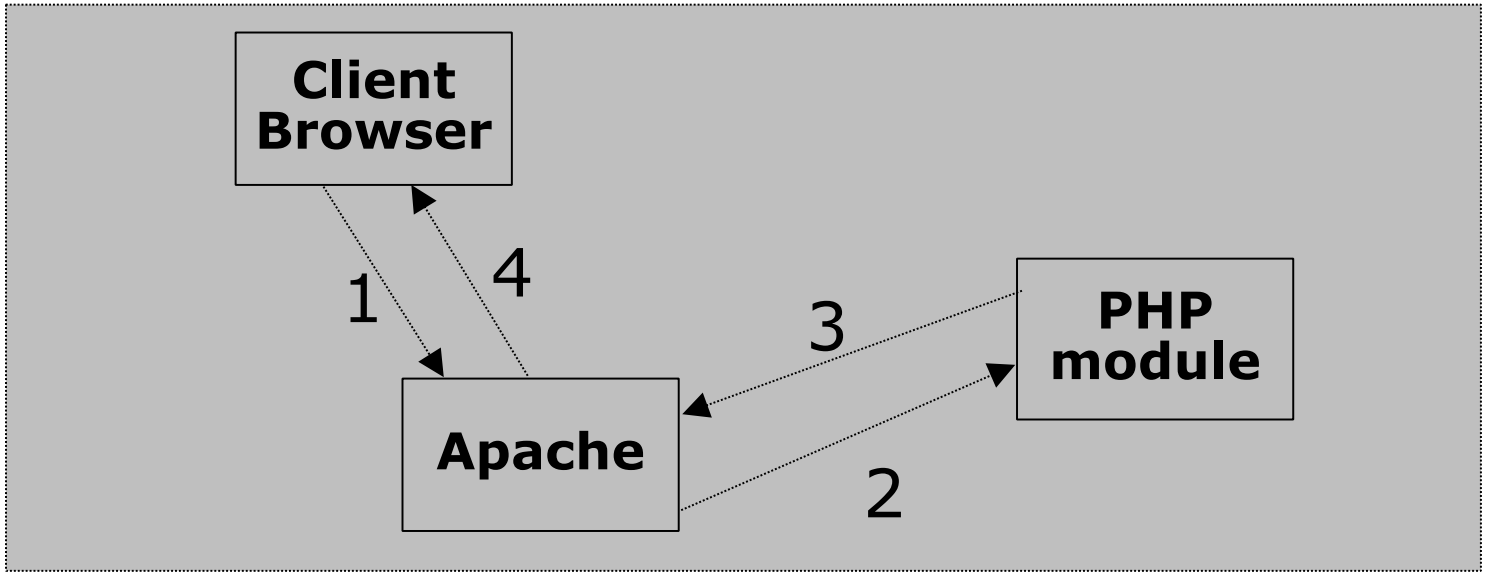
ALTERNATIVES TO PHP

- Practical extraction and Report Language (Perl)
- Active Server Pages (ASP)
- Java server pages (JSP)
- Ruby

ADVANTAGES OF PHP

- Open-source
- Easy to use (C-like and Perl-like syntax)
- Stable and fast
- Multiplatform
- Many databases support
- Many common built-in libraries
- Pre-installed in Linux distributions

HOW PHP WORKS



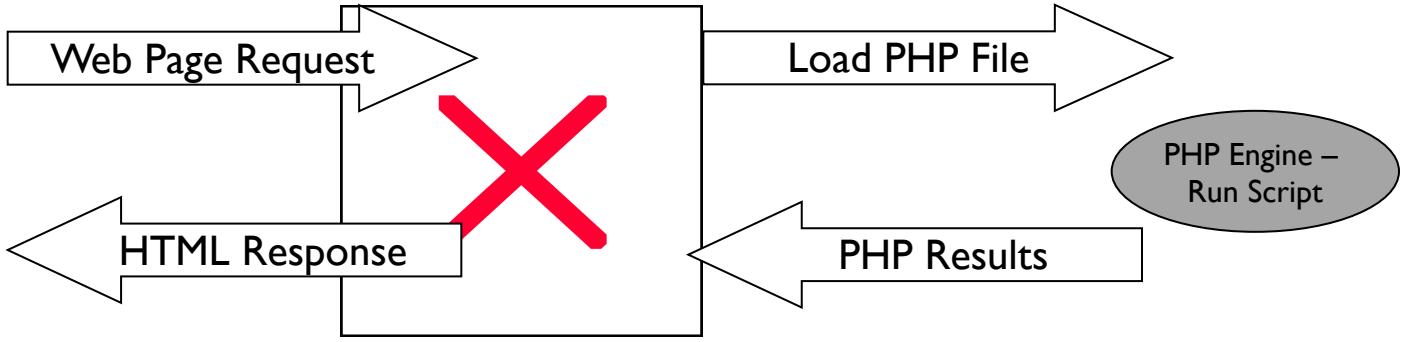
- Client from browser send HTTP request (with POST/GET variables)
- Apache recognizes that a PHP script is requested and sends the request to PHP module
- PHP interpreter executes PHP script, collects script output and sends it back
- Apache replies to client using the PHP script output as HTML output

PHP - WHAT IS IT / DOES IT DO?

- PHP: **P**HP **H**ypertext **P**re-processor
- Programming language that is interpreted and executed on the server
- Execution is done before delivering content to the client
- Contains a **vast** library of functionality that programmers can harness
- Executes entirely on the server, requiring no specific features from the client

PHP - WHAT IS IT / DOES IT DO?

- Static resources such as regular HTML are simply output to the client from the server
- Dynamic resources such as PHP scripts are processed on the server prior to being output to the client
- PHP has the capability of connecting to many database systems making the entire process transparent to the client



SIDE-BY-SIDE

PHP File:

```
<html>
<head>
<title> PHP Introduction </title> </head>
<body>
This is HTML! <br />
<?php
echo 'This is PHP! <br />’;
?>
</body> </html>
```

Output: resulting HTML

```
<html>
<head>
<title> PHP Introduction </title> </head>
<body>


This is HTML! <br />
This is PHP! <br />
</body>
</html>
```

PHP LANGUAGE BASICS

- Look at the building blocks of the PHP language
  - Syntax and structure
  - Variables, constants and operators
  - Data types and conversions
  - Decision making IF and switch
  - Interacting with the client application (HTML forms)

PHP - SYNTAX AND STRUCTURE

- PHP is similar to C
- All scripts start with **<?php** and with with **?>**
- Line separator: **;** (semi-colon)
- Code block: **{ //code here } (brace brackets)**
- White space is generally ignored (not in strings)
- Comments are created using:
  - // single line quote
  - /* Multiple line block quote */

PHP - VARIABLES

- Prefixed with a $
- Assign values with **=** operator
- Example: **$author = “Trevor Adams”;**
- No need to define type
- Variable names are case sensitive
  - **$author** and **$Author** are different

PHP - EXAMPLE SCRIPT

```
<?php
    $author = “Trevor Adams”;
    $msg = “Hello world!”;
    echo $author . “ says ” . $msg;
?>
```

PHP - CONSTANTS

- Constants are special variables that cannot be changed
- Use them for named items that will not change
- Created using a define function
  - **define('milestokm', 1.6);**
  - Used without $
  - $km = 5 * milestokm;

PHP - OPERATORS

- Standard mathematical operators
  - +, -, *, / and % (modulus)
- String concatenation with a period (.)
  - $car = “SEAT” . “ Altea”;
  - echo $car; would output “SEATAltea”
- Basic Boolean comparison with “==”
  - Using only = will overwrite a variable value
  - Less than < and greater than >
  - <= and >= as above but include equality

PHP - DATA TYPES

- PHP is **not** strictly typed
  - Different to JAVA where all variables are declared
- A data type is either text or numeric
  - PHP decides what type a variable is
  - PHP can use variables in an appropriate way automatically
- E.g.
  - **$vat_rate = 0.175;** /* VAT Rate is numeric */

PHP - EMBEDDED LANGUAGE

- PHP can be placed directly inside HTML E.g.
- <html>
  - <head><title>Basic PHP page</title></head>
  - <body>
  - <h1><?php echo “Hello World!; ?></h1>
  - </body>
- </html>

DECISION MAKING - BASICS

- Decision making involves evaluating Boolean expressions (true / false)
- If($catishungry) { /* feed cat */ }
- “true” and “false” are reserved words
- Initialise as $valid = false;
- Compare with ==
- AND and OR for combinations
  - E.g. if($catishungry AND $havefood) {/* feed cat*/}

PHP - IF STATEMENT

- Used to perform a conditional branch
- If (Boolean expression) {
  - // one or more commands if true
- } else {
  - // one or more commands if false
- }

PHP - SWITCH STATEMENTS

- Useful when a Boolean expression may have many options E.g.
- switch($choice) {
  - case 0: { /* do things if choice equal 0 */ }
  - Case 1: {/* do things if choice equal 1 */ }
  - Case 2: {/* do things if choice equal 2 */ }
  - Default: {/* do if choice is none of the above */}
- }

PHP - ONE SMALL STEP FOR MAN…

- … One giant leap for level 1 students
- First few steps are crucial - topics covered:
  - Basic structure and syntax
  - Variables, constants and operators
  - Data types and conversions
  - Decision making
- Any questions so far?

PHP - DEALING WITH THE CLIENT

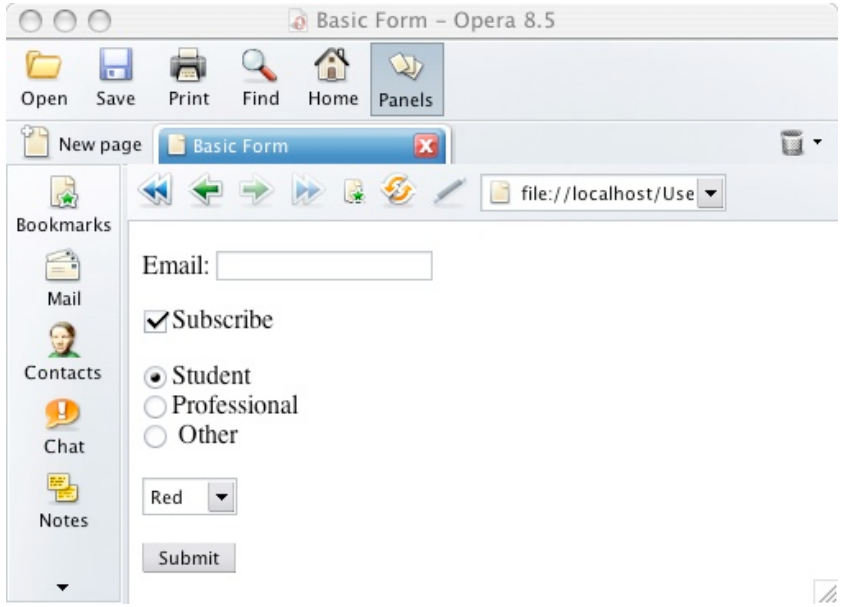
- All very nice but …
- … How is it useful in your web site?
- PHP allows you to use HTML forms
- Forms require technology at the server to process them
- PHP is a feasible and good choice for the processing of HTML forms

PHP - DEALING WITH THE CLIENT

- Quick re-cap on forms
- Implemented with a <form> element in HTML
- Contains other input, text area, list controls and options
- Has some method of submitting

PHP - DEALING WITH THE CLIENT

- Text fields
- Checkbox
- Radio button
- List boxes
- Hidden form fields
- Password box
- Submit and reset buttons



PHP - DEALING WITH THE CLIENT

- <form method="post" action="file.php" name="frmid" >
  - Method specifies how the data will be sent
  - Action specifies the file to go to. E.g. file.php
  - id gives the form a unique name
- Post method sends all contents of a form with basically hidden headers (not easily visible to users)
- Get method sends all form input in the URL requested using name=value pairs separated by ampersands (&)
  - E.g. process.php?name=trevor&number=345
  - Is visible in the URL shown in the browser

PHP - DEALING WITH THE CLIENT

- All form values are placed into an array
- Assume a form contains one textbox called “txtName” and the form is submitted using the post method, invoking process.php
- process.php could access the form data using:
  - $_POST[‘txtName’]
- If the form used the get method, the form data would be available as:
  - $_GET[‘txtName’]

PHP - DEALING WITH THE CLIENT

- For example, an HTML form:
  - <form id="showmsg" action="show.php" method="post">
    - <input type="text" id="txtMsg" value="Hello World" />
    - <input type="submit" id="submit" value="Submit">
  - </form>

PHP - DEALING WITH THE CLIENT

- A file called show.php would receive the submitted data
- It could output the message, for example:
- <html>
  - <head><title>Show Message</title></head>
  - <body>
    - <h1><?php echo $_POST[“txtMsg”]; ?></h1>
  - </body>
- </html>

Hello World

PHP - DEALING WITH THE CLIENT

- Summary
  - Form elements contain input elements
  - Each input element has an id
  - If a form is posted, the file stated as the action can use:
    - $_POST[“inputid”]
  - If a form uses the get method:
    - $_GET[“inputid”]
- Ensure you set all id attributes for form elements and their contents

PHP INTRODUCTION - SUMMARY

- Topics covered
  - Server side architecture brief overview
  - Basic PHP language topics
    - Syntax
    - Variables, Constants and Operators
    - Decision making, IF and Switch statements
    - Dealing with the client