INTRODUCTION TO PHP

SAMITHA SOMATHILAKA

Department of Computing & Mathematics, WIT

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

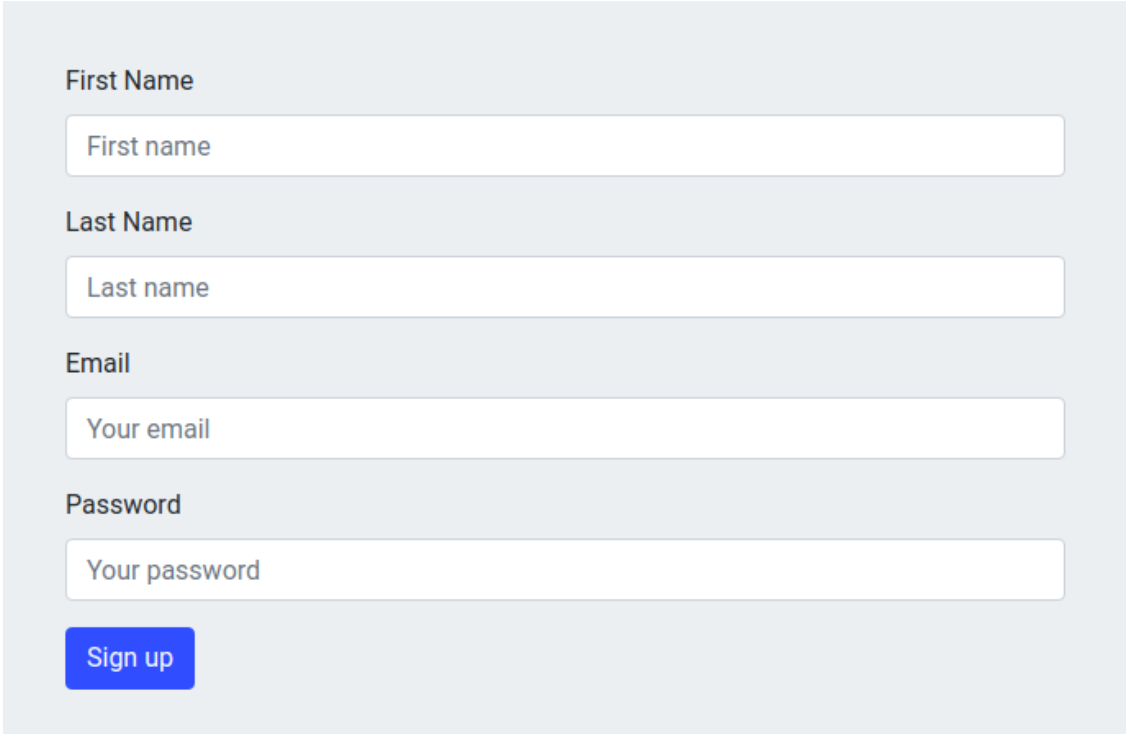
- All very nice but …
- … How is it useful in your web site?
- PHP allows you to use HTML forms
- Forms require technology at the server to process them
- PHP is a feasible and good choice for the processing of HTML forms

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>
```

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