



# PHP – part 1

By Ned Lowe

ned.lowe@ic.ac.uk

<http://www.iesoc.com/>

# Creating a site on the 'Net

- Suggestion: [www.angelfire.com](http://www.angelfire.com)
- Suggestion: [www.spaceports.com](http://www.spaceports.com)
  - Provides PHP and MySQL support for free
  - More complicated and requires FTP
- Good for practicing and showing friends
- Don't use it for setting up your e-business!

# PHP: Hypertext PreProcessor

- Complete new way of thinking
- Server Side Programming Language
  - User makes a request
  - PHP generates HTML based on request
  - Server sends HTML results to user
  - User doesn't have access to your PHP code
- Allows your pages to be fully dynamic
  - Web Applications vs Static Pages
- Commonly coupled with databases
  - MySQL and PHP sit together perfectly
- Alternative technologies include ASP / JSP / perl

# Creating PHP pages

- We write a normal HTML page
- When we want PHP to generate the output rather than use static HTML we add

`<?php`

*commands*

`?>`

- PHP will run the commands and insert the results into the HTML file sent to user

# Sending HTML to user

- We can insert HTML into the file sent to user using the ‘echo’ command:

```
echo “<b>This text is sent to user</b>”;
```

```
<html>
```

```
<?php
```

```
  echo “<head>\n”;
```

```
  echo “<title>This is the title</title>\n”;
```

```
  echo “</head>\n”;
```

```
?>
```

```
  <body>
```

```
  </body>
```

```
</html>
```

- We need to use ‘\n’ to start a new-line in the HTML

# Variables

- Variables allow us to store values in memory
- PHP is not strongly typed
- Variables do not need to be declared
- We use the '\$' to indicate a variable
  - `$myvar = 10;`
  - `$myvar = $myvar * 10;`

```
<?php
    $var1 = 100;
    $var2 = $var 1 + 10;
    echo $var2;
?>
```

# Strings

- A string is a series of characters put together
- A literal string is wrapped in quotation marks
  - It's value never changes - “example”
- A variable string is just written as a variable
  - \$varstring holds the value “example”
- To ‘concatenate’ two strings in PHP we use the period ‘.’ symbol
- echo “Literal string”.\$varstring.“Literal”;

# Data sent to PHP

- There are two ways that data can be sent to a PHP script:
- In the URL:
  - `page.php?var1=Hello&var2=10`
  - Will automatically create `$var1` and `$var2`
- Posting data from an input form
  - `<form action='page.php' method='post' name='myform'>`  
  `<input type='text' name='text1'>`  
  `</form>`
  - Will automatically create `$text1` in 'page.php' when submitted

```
<form action='signprocess.php' method='post'>
  <table bgcolor='#6699ff' cellpadding='1' width='400' align='center'>
    <tr align='center'>
      <td class='style1'>Full Name:</td>
      <td class='style1'><input type='text' name='fullname'></td>
    </tr><tr align='center'>
      <td class='style2'>E-Mail:</td>
      <td class='style2'><input type='text' name='email'></td>
    </tr><tr align='center'>
      <td class='style1'>Department:</td>
      <td class='style1'><input type='text' name='dept'></td>
    </tr><tr align='center'>
      <td class='style2'>Year:</td>
      <td class='style2'><input type='text' name='year'></td>
    </tr><tr align='center'>
      <td colspan='2' class='style1'><input type='submit'></td></tr>
  </table>
</form>
```

```
<html>
<head><title>Entered Data</title></head>
<body>
<?php
echo "The data you entered was:\n";
echo "<table width='500' align='center'>\n"
echo "<tr><td>Name:</td>\n"
echo "<td>".$fullname."</td></tr>";
echo "<tr><td>E-Mail:</td>\n"
echo "<td>".$email."</td></tr>";
echo "<tr><td>Department:</td>\n"
echo "<td>".$dept."</td></tr>";
echo "<tr><td>Year:</td>\n"
echo "<td>".$year."</td></tr>";
echo "</table>";
?>
</body>
</html>
```

# Conditional Execution

- Test conditions and depending on result send different data in HTML

```
if (condition) {
```

```
statements
```

```
}
```

```
else {
```

```
statements
```

```
}
```

- Don't require an else statement if unnecessary.
- ' > ', ' < ', ' <= ' (less than or equal), ' >= ', ' == ', ' != ' (not equal)

```
if ($var1 == "password") {
```

```
$validpass = true;
```

```
}
```

```
<?php
if (!$fullname) {
echo "Please enter your name.\n"; }
if (!$email) {
echo "Please enter your E-Mail.\n"; }
if (!$dept) {
echo "Please enter your department.\n"; }
if (!$year) {
echo "Please enter your year.\n"; }
?>
```

- Can make compound statements using '&&' or '||' (double bar)
- if ((!\$var1) || (!\$var2)) {  
    *statements*  
}

# While Loops

- Allow an action to be repeated ad infinitum
- `while (condition) {  
    statements  
}`
- `while ($length < 10) {  
    echo $length.<br>;  
    $length = $length + 1;  
}`
- Ensure condition can change, otherwise infinite loop

```
<html>
  <head><title>PHP generates a lot of this page</title>
</head>
<body>
  <?php
    if ($num) {
      echo "<table>\n";
      while ($num > 0 ) {
        echo "<tr>\n<td>\n".$num."</td>\n</tr>\n";
        $num--;
      }
      echo "</table>\n";
    }
    else {
      echo "You must enter a number!"; }
  ?>
</body>
</html>
```

# Notes from example

- `$varname--` `===` `$varname = $varname - 1;`
- `$varname++` `===` `$varname = $varname + 1;`
- `if ($var)`
  - checks to see if the variable named 'var' exists at all, irrespective of value
  - if user does not enter value into textbox named 'num' then variable is not created – `if ($var)` returns **false**
- May be entered as text, but becomes number automatically