PHP Basics

PHP

- PHP is a scripting language commonly used on web servers.
  - Stands for “PHP: Hypertext Preprocessor”
  - Open source
  - Embedded code
  - Comparable with ASP
  - Multiple operating systems/web servers

PHP Resource Link

www.php.net

What can it do?

- Dynamic generation of web-page content
- Database interaction
- Processing of user supplied data
- Email
- File handling
- Text processing
- Network interaction
- And more…

Fundamentals

- PHP is embedded within xhtml pages within the tags: <?php … ?>
- The short version of these tags can also be used: <? … ?>
- Each line of PHP is terminated, like MySQL, with a semi-colon.

Hello World!

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

Preparing to code with PHP

Literals..

- All strings must be enclosed in single of double quotes: ‘Hello’ or “Hello”.
- Numbers are not in enclosed in quotes: 1 or 45 or 34.564
- Booleans (true/flase) can be written directly as true or false.

Comments

```
// This is a comment
# This is also a comment
/* This is a comment
that is spread over
multiple lines */
```

- Do not nest multi-line comments
- // recommended over #

Comments

```
<?php
// this is a comment
echo ‘Hello World!’;
/* another
   multi-line comment */
?>
```

Displaying Data

- There are two language constructs available to display data: print() and echo().
- They can be used with or without brackets.
- Note that the data ‘displayed’ by PHP is actually parsed by your browser as HTML. View source to see actual output.

Displaying data

```
<?php
echo ‘Hello World!<br />’;
echo(‘Hello World!<br />’);
print ‘Hello World!<br />’;
print(‘Hello World!<br />’);
?>
```

Escaping Characters

- Some characters are considered ‘special’
- Escape these with a backslash \
- Special characters will be flagged when they arise, for example a double or single quote belong in this group…

Escaping Characters

```
<?php
// Claire O’Reilly said “Hello”.
echo ‘Claire O\’Reilly ’;
echo “said \”Hello\”.”;
?>
```

Variables: What are they?

- When we work in PHP, we often need a labelled place to store a value (be it a string, number, whatever) so we can use it in multiple places in our script.
- These labelled ‘places’ are called

  VARIABLES

Variables: Naming

- **`$`** followed by variable name
- Case sensitive
  - **`$variable`** differs from **`$Variable`**
  - Stick to lower-case to be sure!
- Name must started with a letter or an underscore
  - Followed by any number of letters, numbers and underscores

Variables: example

```
<?php
$name = ‘Phil’;
$age  = 23;
echo $name;
echo ’ is ‘;
echo $age;
// Phil is 23
?>
```

Constants

- Constants (unchangeable variables) can also be defined.
- Each constant is given a name (note no preceding dollar is applied here).
- By convention, constant names are usually in UPPERCASE.

Constants

```
<?php
define(‘NAME’,‘Phil’);
define(‘AGE’,23);
echo NAME;
echo ’ is ‘;
echo AGE;
// Phil is 23
?>
```

“ or ‘ ?

- There is a difference between strings written in single and double quotes.
- In a double-quoted string any variable names are expanded to their values.
- In a single-quoted string, no variable expansion takes place.

“ or ‘ ?

```
<?php
$name = ‘Phil’;
$age  = 23;
echo “$name is $age”;
// Phil is 23
?>
```

“ or ‘ ?

```
<?php
$name = ‘Phil’;
$age  = 23;
echo ‘$name is $age’;
// $name is $age
?>
```