

SYNTAX

- PHP script can be placed anywhere in the document.
- A PHP script starts with **<?php** and ends with **?>**:
- **<?php**
// PHP code goes here
?>
- The default file extension for PHP files is ".php".
- A PHP file normally contains HTML tags, and some PHP scripting code.

In PHP there are two basic ways to get output: echo and print.

- ```
<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!
";
echo "I'm about to learn PHP!
";
echo "This ", "string ", "was ", "made ", "with
multiple parameters."
?>
```

- <!DOCTYPE html>  
<html>  
<body>

Preview from Notesale.co.uk  
Page 15 of 58

```
<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

print "<h2>" . $txt1 . "</h2>";
print "Study PHP at " . $txt2 . "
";
print $x + $y;
?>
```

```
</body>
</html>
```

Preview from Notesale.co.uk  
Page 18 of 58

- ***PHP is a Loosely Typed Language***

- In PHP, a variable does not need to be declared before adding a value to it.
- PHP automatically converts the variable to the correct data type, depending on its value.
- In a strongly typed programming language, you have to declare (define) the type and name of the variable before using it.
- In PHP, the variable is declared automatically when you use it.

- A variable declared within a function has a LOCAL SCOPE and can only be accessed within that function:

- **Example**

- ```
<?php
function myTest() {
    $x = 5; // local scope
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();
```

```
// using x outside the function will generate an error
echo "<p>Variable x outside function is: $x</p>";
?>
```

PHP The global Keyword

- The global keyword is used to access a global variable from within a function.
- To do this, use the global keyword before the variables (inside the function):

- **Example**

- ```
<?php
$x = 5;
$y = 10;
```

```
function myTest() {
 global $x, $y;
 $y = $x + $y;
}
```

```
myTest();
echo $y; // outputs 15
?>
```

# PHP Functions - Returning values

- <?php

```
function sum($x, $y) {
```

```
 $z = $x + $y;
```

```
 return $z;
```

```
}
```

```
echo "5 + 10 = " . sum(5, 10) . "
";
```

```
echo "7 + 13 = " . sum(7, 13) . "
";
```

```
echo "2 + 4 = " . sum(2, 4);
```

```
?>
```

# Accessing Cookies with PHP

- <html>
- <head>
- <title>Accessing Cookies with PHP</title>
- </head>
- <body>
- <?php echo \$\_COOKIE["name"]. "<br />";
- /\* is equivalent to \*/
- echo \$HTTP\_COOKIE\_VARS["name"]. "<br />";
- echo \$\_COOKIE["age"] . "<br />";
- /\* is equivalent to \*/
- echo \$HTTP\_COOKIE\_VARS["name"] . "<br />";
- ?>
- </body>
- </html>

- ```
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30),
"/"); // 86400 = 1 day
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```