

# PHP FOR C PROGRAMMERS

## QUICK REFERENCE

By SaEd Mansour

<b>Code Wrappers</b>		<p>The rule: PHP code goes inside the wrappers.</p> <pre>&lt;?php /*PHP code*/ ?&gt; &lt;html&gt; &lt;head&gt;&lt;/head&gt; &lt;body&gt; &lt;/body&gt; &lt;?php //PHP CODE ?&gt; &lt;/html&gt; &lt;?php /*PHP code*/ ?&gt;</pre> <p>Don't forget: -Files that contain PHP code have the file extension "php". (filename.php)</p>
<b>Variables</b>	<pre>int x = 5; double x = 5.5; char *x = "hello world";</pre>	<pre>\$x = 5; \$x = 5.5 \$x = "hello world"</pre> <p>Variables don't have types.</p> <p>Variables aren't declared.</p> <p>Use \$ before a variable.</p>
<b>Operators</b>		<p>Arithmetic ,assignment and auto increment/decrement are all the same in PHP.</p>

<p style="text-align: center;"><b>Output</b></p>	<pre>printf( "hello world");  char* x = " world"; printf("hello%s", world);  int x = 5; printf("hello %d", x);</pre>	<pre>echo "hello world";  \$x = " world"; echo "hello\$x";  \$x = 5; printf("hello %d", \$x); //printf also in php  \$x = "world"; echo 'hello \$x'; (output: hello \$x)  Single quotes don't allow variable substitution.</pre>
<p style="text-align: center;"><b>Arrays</b></p>	<pre>char arr[] = {1,2,3,4};  arr[0]; arr[1];</pre>	<pre>\$arr = array(1,2,3,4);  \$arr[0]; \$arr[1];  Multi types:  \$arr = array (1, 2.4444, "string");</pre>

<p><b>Flow Control (if, for, while)</b></p>	<pre> for (int i = 0; i &lt; 10; i++) {     printf( "what\n"); }  If (x &gt; 5) {     x = 7; } <b>else if</b> ( x == 5) {     x = 20; } else {     x = 1; }  while ( x &lt; 5) {     x++; }  int array[10] = {0,1,2,3,4,5,6,7,8,9};  for(int i = 0; i &lt; 10; i++) {     printf( "%d\n", array[i]); } </pre>	<pre> for (\$i = 0; \$i &lt; 10; \$i++) {     echo "what&lt;br /&gt;"; }  if (\$x &gt; 5) {     \$x = 7; } <b>elseif</b> ( \$x == 5) {     \$x = 20; } else {     \$x = 1; }  while ( \$x &lt; 5) {     \$x++; }  \$array = array(0,1,2,3,4,5,7,8,9);  <b>foreach</b>(\$array as \$val) {     echo "{\$val}&lt;br /&gt;"; }  Foreach runs over all array elements, each iteration \$val gets the value of the next array element.  \${val} is used to make it know that the variable isn't "\$val&lt;br" because the variable \$val&lt;br doesn't exist...  continue/ break are also in php. </pre>
<p><b>String Concatenation</b></p>	<pre> char str[10] ="what"; strcat(str, " the!"); printf("%s", str); (output: what the!) </pre>	<pre> echo 'what' . ' the!'; (output: what the!)  Operator dot attaches the two strings together (concatenation). </pre>

<p><b>Functions</b></p>	<pre>double mul(double a, double b) {     return a * b; }</pre>	<pre>function mul(\$a, \$b) {     return \$a * \$b; }</pre>
<p><b>Switch</b></p>	<pre>switch(x) {     case 1:         x = 1;         break;     case 2:         x = 2;         break;     default:         x = 0;         break; }</pre>	<pre>switch(\$x) {     case 1:         \$x = 1;         break;     case 2:         \$x = 2;         break;     default:         \$x = 0;         break; }</pre> <p><b>switch accepts strings:</b></p> <pre>\$scream = "fire!"; switch(\$scream) {     case "boom!":         echo "hide!";     case "fire!":         echo "run away!"; }</pre>
<p><b>Comments</b></p>	<pre>// this is a single line comment  /*     this is a comment */</pre>	<pre>// this is a single line comment  /*     this is a comment */  # this is a single line comment</pre>
<p><b>Pointers</b></p>		<p>There are no pointers, use references.</p> <pre>function swap( &amp;\$a, &amp;\$b) {     \$temp = \$a;     \$a = \$b;     \$b = \$temp; }</pre>

		Function call: swap(\$a, \$b);
--	--	-----------------------------------

if there was no & in swap's  
declaration the values wouldn't be  
swapped.

**Useful functions in php:**

**require "fileName";**

//includes the text of fileName into the file where the function require was called