

Language cross-reference for PHP, Javascript and Perl.

	PHP 5	Javascript	Perl
scalar variables	<code>\$v = 1;</code>	<code>var v = 1;</code>	<code>my \$v = 1;</code>
arrays	<code>\$a = array('a', 'b', 'c'); \$a[2] = 'B';</code>	<code>a = []; a[9] = 2; la = new Array("a","b","c");</code>	<code>@a = ("a", "b"); \$a[1] = "c";</code>
multidimensional arrays	<code>\$aa = array(array(0,1), array(2,3)); \$aa[0][1]=1;</code>	<code>var aa = new Array(new Array(0,1), new Array(2,3)); aa[0][1]=1;</code>	<code>@aa = ([0,1], [2,3]); \$aa[0][1]=1;</code>
array length	<code>count(\$a)</code>	<code>a.length</code>	<code>scalar(@a)</code>
array push and pop	<code>\$a[] = \$x; \$x = array_pop(\$a); array_unshift(\$a,\$x); \$x = array_shift(\$a);</code>	<code>a.push(x); x = a.pop(); a.unshift(x); x = a.shift();</code>	<code>push @a, \$x; \$x = pop @a; unshift @a, \$x; \$x = shift @a;</code>
associative arrays	<code>\$h = array('a' => 'b'); if(!array_key_exists('c', \$h)) { \$h['c'] = "C"; }</code>	<code>var h = {"a":"A", "b":"B"}; if(!h['c']) { h['c'] = "C"; }</code>	<code>%h = ("a"=>"A", "b"=>"B"); if(!exists \$h{"c"}) { \$h{"c"} = "C"; }</code>
iterate array	<code>foreach(\$a as \$value) { \$y = \$value; } foreach(\$h as \$key=>\$value) { echo \$key . '=>' . \$value; }</code>	<code>for(var i in a) { y = a[i]; } for(var i in h) { print i+"=>"h[i]; }</code>	<code>foreach \$x (@a) { \$y = \$x; } foreach \$key (keys %h) { print \$key."=>".\$h{\$key}; }</code>
iterate characters	<code>for(\$i=0; \$i<strlen(\$str); ++\$i) { \$c = str{\$i}; }</code>	<code>for(var i=0; i<s.length; ++i) { var c = s.charAt(i); }</code>	<code>@a = unpack("C*", \$s); @a = split(//,\$s); while(\$s~/(.)/g) { my \$c = \$1; }</code>
seconds since epoch	<code>time();</code>	<code>new Date().GetTime() / 1000;</code>	<code>time;</code>
comparison to null	<code>isset(\$a); isset(NULL) == false;</code>	<code>if(v == undefined)... if(!x){} // matches NULL or</code>	<code>if(defined \$v) ... undef \$v;</code>

	PHP 5	Javascript	Perl
test for non-zero, true and false	<pre>// 0, 0.0, "0", NULL, unset, false, FALSE, !true, !TRUE and "" are all treated as FALSE. // Use ! to invert the boolean value. isset(var) // tests to see if var has ever been assigned.</pre>	<pre>undefined or 0 if(x==null){} // matches NULL only if(x == undefined){} // matches undefined and NULL // 'true' and 'false' are valid keywords.</pre>	<pre># 0, 0.0, "", "0", undef are all treated as FALSE. Everything else TRUE. # There are no native 'true' and 'false' keywords.</pre>
string concatenation	<pre>\$a . \$b</pre>	<pre>a + b</pre>	<pre>\$a . \$b</pre>
string replace	<pre>\$r = str_replace("this", "withthat", \$s);</pre>	<pre>s=""+s; // Stringify numbers s.replace(/this/g, "withthat");</pre>	<pre>\$s =~ s/this/that/g;</pre>
functions	<pre>function f(\$x, &\$y, \$z=1) { return \$x + (\$y++) + \$z; }</pre>	<pre>function f(x) { return x; }</pre>	<pre>sub f { my \$x = shift; return \$x; }</pre>
objects	<pre>class C extends B { protected \$a; protected \$b; function __construct(\$a,\$b) { // Note \$ signs \$this->a = \$a; \$this->b = \$b; } function __destruct() {} public function GetA(\$x) { // Note only one \$ sign return \$this->a; } }</pre>	<pre>function C(a, b) { this.a = a; this.b = b; } C.prototype.getA = function() { return this.a; }</pre>	<pre>package C; sub new { my \$class = shift; my \$self = { a=>shift, b=>shift }; bless \$self, \$class; return \$self; } sub DESTROY {} sub getA { my \$self = shift; return \$self->{a}; }</pre>
instances	<pre>\$c = new C(1,2);</pre>	<pre>var c = new C(1,2);</pre>	<pre>my \$c = new C(1,2);</pre>
split string	<pre>\$a = explode(' ', \$str);</pre>	<pre>var a = str.split(' ');</pre>	<pre>my @a = split /,/, \$str;</pre>
print to stdout	<pre>echo "x=\$x\n";</pre>	<pre>// It varies. Can fall back on: alert("x="+x);</pre>	<pre>print "x=\$x";</pre>
string length	<pre>strlen(\$x)</pre>	<pre>s.length</pre>	<pre>length \$s;</pre>

	PHP 5	Javascript	Perl
<<=	<code>\$a <<= 4;</code>	<code>a <<= 4;</code>	<code>\$x <<= 4;</code>
for	<code>for(i=0;i<10;++i)...</code>	<code>for(var i=0; i<10; ++i) ...</code>	<code>for(my \$i=0; \$i<10; ++\$i)...</code>
ternary if	<code>\$a ? \$b : \$c</code>	<code>a ? b : c</code>	<code>\$a ? \$b : \$c</code>
enums	<code>// Some people use arrays, e.g.: \$commands['CMD_ENTER']</code>	<code>var Command={ Enter:0, Exit:1 }; var cmd = Command.Enter;</code>	<code>my (\$ENTER, \$EXIT) = (0..999);</code>
constants	<code>define(PI, 3.1415965358979)</code>	<code>const PI = 3.141596;</code>	<code>use constant PI => 3.141; use constant { Enter => 0, Exit => 1 };</code>
case sensitivity	<code>// variable names are case sensitive</code>	<code>// variable names are case sensitive</code>	<code>// variable names are case sensitive</code>
references	<code>\$ra = &\$a;</code>	<code>// Non-simple data types are always passed by reference. // To copy an array use a2=a1.splice(); // To copy an object, write a clone() method.</code>	<code>\$ra = \@a; \$ra = ["a","b"];</code>
(int)x, and string to int	<code>\$x = (int)12.9; // Rounds down \$x = (int)"123"; \$x = "123" * 1;</code>	<code>Math.floor(x); parseInt(x); x = "123" * 1;</code>	<code>int(\$x) // Rounds down my \$x = "123" * 1;</code>
to lower case	<code>strtolower(\$str)</code>	<code>str.toLowerCase()</code>	<code>lc \$str</code>
dump variable	<code>dump_var(x)</code>	N/A	N/A
output literally	<code>>? This is literal <?php</code>	N/A	<code>print <<END; this gets printed literally END # Note: no ; on previous line</code>
isdigit	<code>ctype_digit(\$str)</code>	<code>((c >= "0") && (c <= "9"))</code>	<code>\$c =~ /[\\d]/;</code>
int to char	<code>chr("A") // 65 ord(65) // A</code>	<code>"A".charCodeAt(0) // 65 String.fromCharCode(65) // "A"</code>	<code>chr("A") // 65 ord(65) // A</code>
random (0 < x < 1)	<code>\$x = rand() / getrandmax(); // rand() returns an integer.</code>	<code>x = Math.random();</code>	<code>\$x = rand();</code>
substring	<code>\$s=substr(\$s, \$start, \$len); \$s=substr(\$s, \$start);</code>	<code>s.substring(start,stop); s.substring(start);</code>	<code>substr \$s, \$start; substr \$s, \$start, \$len;</code>

	PHP 5	Javascript	Perl
index of char	<code>strpos(\$s, \$c, \$start);</code> <code>strpos(\$s, \$c);</code>	<code>s.indexOf(c, start);</code> <code>s.indexOf(c);</code>	<code>index \$s,\$c,0;</code> <code>index \$s,\$c;</code>
file i/o	<code>if file_exists("file");</code> <code>\$handle=fopen("file","r");</code> <code>if(\$handle)</code> <code>{</code> <code> while(!feof(\$handle))</code> <code> {</code> <code> \$line = fgets(\$handle,4096);</code> <code> }</code> <code> fclose(\$handle);</code> <code>}</code>	N/A	<code>open(FH, \$filename) or</code> <code>die("Can't open file");</code> <code>while(my \$line = <FH>)</code> <code>{</code> <code> chomp(\$line);</code> <code>}</code> <code>close FH;</code>
current directory	<code>\$dir = getcwd()</code>	N/A	<code>use Cwd;</code> <code>my \$dir = getcwd;</code>