Language basics

script start	`<?php`
script end	`?>` or nothing
command termination	`;`
inline comment	`// simple oneliner`
block comment	`/* multiline text */`
run script from file (if error throws warning)	`include('script.php');`
run script from file (errors are fatal)	`require('script.php');`

Variables

declaration without value	`$variableName;`
string declaration	`$string = "some text";`
integer declaration	`$int = 123;`
double declaration	`$doble = 1.2`
empty array declaration	`$array = array();`
array with elements declaration	`$a = array("text",1,2.3);`
boolean declaration	`$bool = true;`
null declaration	`$empty = null;`

Variable operations

addition	`$int1 + $int2 ->` 12
divison	`6 / 3 ->` 2
multiplication	`6 * 6 ->` 36
exponentiation	`2 ** 4 ->` 16
substraction	`2 - 1 ->` 1
string + int (string get converted) #1	`"13shits" + 13 ->` 26
string + int (string get converted) #1	`"shitsof13" + 13 ->` 13
string + string	`"2dogs" + "5dogs" ->` 7
strings **concat**	`$str1 . $str2 ->` hello stranger!
string and int **concat**	`"numero uno - " . 1; ->` numero uno - 1
unset - delete variable	`unset($variable);`

```
$int1 = 5;
$int2 = 7;
$str1 = "hello ";
$str2 = " stranger!";
```

Flow control

simple condition	`if (true){`
	`echo("condition pass");`
	`}`

is var1 equal to var2?	`if ($var1 == $var2){`
	`echo("it is.");`
	`} else {`
	`echo("it ins't.")`
	`}`

two vars comperation	`($var1 == $var2)`
is var equal to int?	`($var1 == 4)`
is var lower than int?	`($int < 4)`
is var bigger than int?	`($int > 4)`
is var lower than 5 and not equal to zero	`($int < 5 && $int != 0)`
is string equal to string?	`($str == "dangerous")`
is num 1 or 4?	`($num == 1 || $num == 4)`

Array operations

get array element by index	`$array[index]`
push data to **end** of array	`array_push($array, $data);` OR `$array[] = $data;`
push data to **specific** index of array	`$array['myColor'] = "#FFF";`
remove & get last element of array	`array_pop($array);`
get **count** of elements in array	`count($array);`
is specific value in array?	`in_array($needle, $array);`

Output

simple print	`echo($variable . " and some text");`
simple print #2	`echo 1;`
simple print #3 (must be without parentheses)	`echo 1,2; ->` 12
print multiple lines	`echo("line1\nline2\n");`
print variable type and data	`print_r($var);`
print variable type and data (detailed)	`var_dump($var);`

Output (cont)

conditional printing	`echo($var1 == 1 ? "its 1!" : "its not 1!");`

Loops

for - classic loop (start instruction; condition on start of each round;end instruction of each round)	`for ($i = 1; $i <= 10; $i++) {`
	`echo $i;`
	`}`

foreach - do something for each element in array	`foreach($array as $arrayElement){`
	`var_dump($arrayElement;)`
	`}`

while - do something if cond is true again and again if is still true	`while(true){`
	`echo("infinity!");`
	`}`

break - escape from loop	`break;`
continue - skip current round	`continue;`

PHP Filesystem Functions

check if file exists	`file_exists("/root/file.txt");`
open a file for reading only	`$file = fopen("./../file.txt", "r");`
get size of file in bytes	`filesize("file.txt");`
print all bytes from fopened file	`echo fread($file, filesize("file.txt"));`
open a file for reading only	`$file = fopen("./../file.txt", "w");`
write to fopened file	`fwrite($file, "some text\n");`
close file after all operations	`fclose($file);`
oneline file write	`file_put_contents($filename, $data);`
oneline file read	`$fileContent = file_get_contents($fileName);`
fopen() modes (second argument)	**r** = Read **r+** = Read,write and prepend **w** = Write, truncate **w+** = Read and write, truncate a Write, append **a+** = Read and write, append

Useful variables

$_SERVER	Server and execution environment information
$_SERVER['SERVER_ADDR']	Server IP
$_SERVER['REQUEST_METHOD']	Which request method was used to access the page; i.e. 'GET', 'HEAD', 'POST', 'PUT'
$_SERVER['HTTP_USER_AGENT']	User-Agent: header from the current request
$_SERVER['REMOTE_ADDR']	The IP address from which the user is viewing the current page
__FILE__	The full path and filename of the current script file
__DIR__	The directory of the file
__LINE__	Current line number in script file
$_GET	Associative array of variables passed to the current script via the URL parameters
$_POST	An associative array of variables passed to the current script via the HTTP POST method when using application/x-www-form-urlencoded or multipart/form-data as the HTTP Content-Type in the request
$_FILES	An associative array of items uploaded to the current script via the HTTP POST method

By **davidsykora**
cheatography.com/davidsykora/

Published 6th October, 2017.
Last updated 21st September, 2017.
Page 2 of 2.