

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	<code>\$variableName;</code>
string declaration	<code>\$string = "some text";</code>
integer declaration	<code>\$int = 123;</code>
double declaration	<code>\$doble = 1.2</code>
empty array declaration	<code>\$array = array();</code>
array with elements declaration	<code>\$a = array("text",1,2,3);</code>
boolean declaration	<code>\$bool = true;</code>
null declaration	<code>\$empty = null;</code>

Variable operations

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

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

Flow control

simple condition	<code>if (true){ echo("condition pass"); }</code>
is var1 equal to var2?	<code>if (\$var1 == \$var2){ echo("it is."); } else { echo("it ins't.") }</code>
two vars comperation	<code>(\$var1 == \$var2)</code>
is var equal to int?	<code>(\$var1 == 4)</code>
is var lower than int?	<code>(\$int < 4)</code>
is var bigger than int?	<code>(\$int > 4)</code>
is var lower than 5 and not equal to zero	<code>(\$int < 5 && \$int != 0)</code>
is string equal to string?	<code>(\$str == "dangerous")</code>
is num 1 or 4?	<code>(\$num == 1 \$num == 4)</code>

Array operations

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

Output

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



Output (cont)

conditional printing	<code>echo(\$var1 == 1 ? "its 1!" : "its not 1!");</code>
----------------------	---

Loops

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

foreach - do something for each element in array	<code>foreach(\$array as \$arrayElement) { var_dump(\$arrayElement); }</code>
---	---

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

break - escape from loop	<code>break;</code>
---------------------------------	---------------------

continue - skip current round	<code>continue;</code>
--------------------------------------	------------------------

PHP Filesystem Functions

check if file exists	<code>file_exists("/root/file.txt");</code>
----------------------	---

open a file for reading only	<code>\$file = fopen("../file.txt", "r");</code>
------------------------------	--

get size of file in bytes	<code>filesize("file.txt");</code>
---------------------------	------------------------------------

print all bytes from fopened file	<code>echo fread(\$file, filesize("file.txt"));</code>
-----------------------------------	--

open a file for reading only	<code>\$file = fopen("../file.txt", "w");</code>
------------------------------	--

write to fopened file	<code>fwrite(\$file, "some text\n");</code>
-----------------------	---

close file after all operations	<code>fclose(\$file);</code>
---------------------------------	------------------------------

oneline file write	<code>file_put_contents(\$filename, \$data);</code>
--------------------	---

oneline file read	<code>\$fileContent = file_get_contents(\$fileName);</code>
-------------------	---

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
-----------------	---

