Php Cheatsheet

Haris Ali Khan • December 25, 2022 • 11 min read

Basics

Hello World

echo function is used to display or print output

```
<?php echo "Hello World!"; ?>
```

Comments

Commets are used to make the code more understandable for programmer, they are not executed by compiler or interpreter.

One Liner

This is a singleline comment

```
// Twinkle Twinkle Little Star
```

Another One Liner

This is a single-line comment

```
# Chocolate dedo mujhe yaar
```

Multiline

This is a multiline comment

```
/* Code With
Harry */
```

Vardump

This function dumps information about one or more variables.

```
<?php var_dump(var1, var2, ...); ?>
```

Variables

Variables are "containers" for storing information.

Defining Variables

```
<?php
$Title = "PHP Cheat Sheet By CodeWithHarry";
?>
```

Datatypes

Datatype is a type of data

String

A string is a sequence of characters, like "Hello world!".

```
<?php
$x = "Harry";
echo $x;
?>
```

Integer

An integer is a number without any decimal part.

```
<?php
$x = 1234;
var_dump($x);
?>
```

Float

A float is a number with a decimal point or a number in exponential form.

```
<?php
$x = 1.2345;
var_dump($x);
?>
```

Array

An array stores multiple values in one single variable

```
<?php
$names = array("Harry","Rohan","Shubham");
var_dump($names);
?>
```

Class

A class is a template for objects

```
<?php
class Harry{
// code goes here...
}
?>
```

Object

An object is an instance of the class.

```
<?php
class Bike {
public $color;
public $model;
public function __construct($color, $model) {
$this->color = $color;
$this->model = $model;
}
public function message() {
return "My bike is a " . $this->color . " " . $this->model . "!";
}
}
```

```
$myBike = new Bike("red", "Honda");
echo $myBike -> message();
?>
```

Escape Characters

Escape sequences are used for escaping a character during string parsing. It is also used for giving special meaning to represent line breaks, tabs, alerts and more.

Line feed

It adds a newline

```
<?php
echo "\n";
?>
```

Carriage return

It inserts a carriage return in the text at this point.

```
<?php
echo "\r";
?>
```

Horizontal tab

It gives a horizontal tab space

```
<?php
echo "\t";
?>
```

Vertical tab

It gives a vertical tab space

```php
<?php
echo "\v";
?>
```

Escape

It is used for escape characters

```php
<?php
echo "\e";
?>
```

Form feed

It is commonly used as page separators but now is also used as section separators.

```php
<?php
echo "\f";
?>
```

Backslash

It adds a backslash

```php
<?php
echo "\\";
?>
```

Dollar sign

Print the next character as a dollar, not as part of a variable

```php
<?php
echo "\$";
?>
```

Single quote

Print the next character as a single quote, not a string closer

```php
<?php
echo "\'";
?>
```

Double quote

Print the next character as a double quote, not a string closer

```php
<?php
echo "\"";
?>
```

Operators

Operators are symbols that tell the compiler or interpreter to perform specific mathematical or logical manipulations. These are of several types.

Arithmetic Operators

Addition

Sum of $x and $y

```
$x + $y
```

Subtraction

Difference of $x and $y

```
$x - $y
```

Multiplication

Product of $x and $y

```
$x * $y
```

Division

Quotient of $x and $y

```
$x / $y
```

Modulus

The remainder of $x divided by $y

```
$x % $y
```

Exponentiation

Result of raising $x to the $y'th power

```
$x ** $y
```

PHP Assignment Operators

The PHP assignment operators are used with numeric values to write a value to a variable.

x = y

The left operand gets set to the value of the expression on the right

```
x = y
```

x += y

Addition

```
x = x + y
```

x -= y

Subtraction

```
x = x - y
```

x *= y

Multiplication

```
x = x * y
```

x /= y

Division

```
x = x / y
```

x %= y

Modulus

```
x = x % y
```

PHP Comparison Operators

Equal

Returns true if $x is equal to $y

```
$x == $y
```

Identical

Returns true if $x is equal to $y, and they are of the same type

```
$x === $y
```

Not equal

Returns true if $x is not equal to $y

```
$x != $y
```

Not equal

Returns true if $x is not equal to $y

```
$x <> $y
```

Not identical

Returns true if $x is not equal to $y, or they are not of the same type

```
$x !== $y
```

Greater than

Returns true if $x is greater than $y

```
$x > $y
```

Less than

Returns true if $x is less than $y

```
$x < $y
```

Greater than or equal to

Returns true if $x is greater than or equal to $y

```
$x >= $y
```

Less than or equal to

Returns true if $x is less than or equal to $y

```
$x <= $y
```

PHP Increment / Decrement Operators

Pre-increment

Increments $x by one, then returns $x

```
=++$x
```

Post-increment

Returns $x, then increments $x by one

```
$x++
```

Pre-decrement

Decrements $x by one, then returns $x

```
--$x
```

Post-decrement

Returns $x, then decrements $x by one

```
$x--
```

PHP Logical Operators

And

True if both $x and $y are true

```
$x and $y
```

Or

True if either $x or $y is true

```
$x or $y
```

Xor

True if either $x or $y is true, but not both

```
$x xor $y
```

And

True if both $x and $y are true

```
$x && $y
```

Or

True if either $x or $y is true

```
$x || $y
```

Not

True if $x is not true

```
!$x
```

B.Com @ PESU Electronic City Campus
Admissions 2024
Commerce for the Real World
APPLY N
PESSA
P
UNI

PHP String Operators

Concatenation

Concatenation of $txt1 and $txt2

```
$txt1 . $txt2
```

Concatenation assignment

Appends $txt2 to $txt1

```
$txt1 .= $txt2
```

PHP Array Operators

Union

Union of $x and $y

```
$x + $y
```

Equality

Returns true if $x and $y have the same key/value pairs

```
$x == $y
```

Identity

Returns true if $x and $y have the same key/value pairs in the same order and of the same types

```
$x === $y
```

Inequality

Returns true if $x is not equal to $y

```
$x != $y
```

Inequality

Returns true if $x is not equal to $y

```
$x <> $y
```

Non-identity

Returns true if $x is not identical to $y

```
$x !== $y
```

PHP Conditional Assignment Operators

Ternary

Returns the value of $x. The value of $x is expr2 if expr1 = TRUE. The value of $x is expr3 if expr1 = FALSE

```
$x = expr1 ? expr2 : expr3
```

Conditional Statements

Conditional statements are used to perform operations based on some condition.

If Statement

if statement checks the condition and if it is True, then the block of if statement executes; otherwise, control skips that block of code.

```
if (condition) {
// code to execute if condition is met
}
```

If..Else

if the condition of if block evaluates to True, then if block executes otherwise else block executes

```
if (condition) {
// code to execute if condition is met
} else {
// code to execute if condition is not met
}
```

If..Elseif..Else

It executes different codes for more than two conditions

```
if (condition) {
// code to execute if condition is met
} elseif (condition) {
// code to execute if this condition is met
} else {
// code to execute if none of the conditions are met
}
```

Switch Statement

It allows a variable to be tested for equality against a list of values (cases).

```
switch (n) {
case x:
code to execute if n=x;
break;
case y:
code to execute if n=y;
break;
case z:
code to execute if n=z;
break;
// add more cases as needed
default:
code to execute if n is neither of the above;
}
```

Loops

Iterative statements or Loops facilitate programmers to execute any block of code lines repeatedly.

For Loop

It is used to iterate the statements several times. It is frequently used to traverse the data structures like the array and linked list.

```
for (starting counter value; ending counter value; increment by which
to increase) {
// code to execute goes here
}
```

Foreach Loop

The foreach loop loops through a block of code for each element in an array.

```
foreach ($InsertYourArrayName as $value) {
// code to execute goes here
}
```

While Loop

It iterate the block of code as long as a specified condition is True or vice versa

```
while (condition that must apply) {
// code to execute goes here
}
```

Do-While Loop

This loop is very similar to the while loop with one difference, i.e., the body of the do-while loop is executed at least once even if the condition is False. It is an exit-controlled loop.

```
do {
// code to execute goes here;
} while (condition that must apply);
```

Predefined Variables

PHP provides a large number of predefined variables to all scripts. The variables represent everything from external variables to built-in environment variables, last error messages etc. All this information is defined in some predefined variables.

$GLOBALS

$GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script.

```
<?php
$a = 10;
$b = 15;

function addition() {
$GLOBALS['c'] = $GLOBALS['a'] + $GLOBALS['b'];
}
addition();
echo $c;
?>
```

$_SERVER

Returns the filename of the currently executing script. $_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.

```
$_SERVER['PHP_SELF']
```

Returns the version of the Common Gateway Interface (CGI) the server is using

```
$_SERVER['GATEWAY_INTERFACE']
```

Returns the IP address of the host server

```
$_SERVER['SERVER_ADDR']
```

Returns the name of the host server (such as www.codewithharry.com)

```
$_SERVER['SERVER_NAME']
```

Returns the server identification string (such as Apache/2.2.24)

```
$_SERVER['SERVER_SOFTWARE']
```

Returns the name and revision of the information protocol (such as HTTP/1.1)

```
$_SERVER['SERVER_PROTOCOL']
```

Returns the request method used to access the page (such as POST)

```
$_SERVER['REQUEST_METHOD']
```

Returns the timestamp of the start of the request (such as 1377687496)

```
$_SERVER['REQUEST_TIME']
```

Returns the query string if the page is accessed via a query string

```
$_SERVER['QUERY_STRING']
```

Returns the Accept header from the current request

```
$_SERVER['HTTP_ACCEPT']
```

Returns the Accept_Charset header from the current request (such as utf-8,ISO-8859-1)

```
$_SERVER['HTTP_ACCEPT_CHARSET']
```

```
$_SERVER['HTTP_REFERER']
```

Is the script queried through a secure HTTP protocol?

```
$_SERVER['HTTPS']
```

Returns the IP address from where the user is viewing the current page

```
$_SERVER['REMOTE_ADDR']
```

Returns the Hostname from where the user is viewing the current page

```
$_SERVER['REMOTE_HOST']
```

Returns the port being used on the user's machine to communicate with the web server

```
$_SERVER['REMOTE_PORT']
```

Returns the absolute pathname of the currently executing script

```
$_SERVER['SCRIPT_FILENAME']
```

Returns the value given to the SERVER_ADMIN directive in the web server configuration file (if your script runs on a virtual host, it will be the value defined for that virtual host) (such as someone@codewithharry.com)

```
$_SERVER['SERVER_ADMIN']
```

Returns the port on the server machine being used by the webserver for communication (such as 80)

Copy

```
$_SERVER['SERVER_PORT']
```

Returns the server version and virtual hostname which are added to server-generated pages

```
$_SERVER['SERVER_SIGNATURE']
```

Returns the file system based path to the current script

```
$_SERVER['PATH_TRANSLATED']
```

Returns the path of the current script

```
$_SERVER['SCRIPT_NAME']
```

Returns the URI of the current page

```
$_SERVER['SCRIPT_URI']
```

$_GET

PHP $_GET is a PHP super global variable which is used to collect form data after submitting an HTML form with method="get".

```
<?php
echo "Hello" . $_GET['Example'] . " at " . $_GET['web'];
?>
```

$_POST

PHP $_POST is a PHP super global variable which is used to collect form data after submitting an HTML form with method="post". $_POST is also widely used to pass variables.

```
<html>
<body>
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
Name: <input type="text" name="fname">
<input type="submit">
</form>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
$name = $_POST['fname'];
if (empty($name)) {
echo "Please Enter your name";
} else {
echo $name;
}
}
```

$_REQUEST

PHP $_REQUEST is a PHP super global variable which is used to collect data after submitting an HTML form.

```
<html>
<body>
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
Name: <input type="text" name="fname">
<input type="submit">
</form>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
$name = $_REQUEST['fname'];
if (empty($name)) {
echo "Name is empty";
} else {
echo $name;
}
}
?>
```

Variable-handling Functions

The PHP variable handling functions are part of the PHP core. No installation is required to use these functions.

boolval

Boolval is used to get the boolean value of a variable

```php
<?php
echo '0: '.(boolval(0) ? 'true' : 'false')."\n";
echo '42: '.(boolval(42) ? 'true' : 'false')."\n";
echo '0.0: '.(boolval(0.0) ? 'true' : 'false')."\n";
echo '4.2: '.(boolval(4.2) ? 'true' : 'false')."\n";
echo '"": '.(boolval("") ? 'true' : 'false')."\n";
echo '"string": '.(boolval("string") ? 'true' : 'false')."\n";
echo '"0": '.(boolval("0") ? 'true' : 'false')."\n";
echo '"1": '.(boolval("1") ? 'true' : 'false')."\n";
echo '[1, 2]: '.(boolval([1, 2]) ? 'true' : 'false')."\n";
echo '[]: '.(boolval([]) ? 'true' : 'false')."\n";
echo 'stdClass: '.(boolval(new stdClass) ? 'true' : 'false')."\n";
?>
```

isset

It is used to check whether a variable is empty. It also checks whether the variable is set/declared:

```
<?php
$x = 0;
// True because $x is set
if (isset($x)) {
echo "Variable 'x' is set";
}
```

unset

It unsets variables.

```
<?php
$a = "Namaste world!";
echo "The value of 'a' before unset: " . $a ;
unset($a);
echo "The value of 'a' after unset: " . $a;
?>
```

debug_zval_dump

debug_zval_dump is used to dump a string representation of an internal zval structure to output

```
<?php
$var1 = 'Hello';
$var1 .= ' World';
```

```
$var2 = $var1;

debug_zval_dump($var1);
?>
```

empty

Empty is used to check whether a variable is empty or not.

```
<?php
$var = 0;

// Evaluates to true because $var is empty
if (empty($var)) {
echo '$var is either 0, empty, or not set at all';
}

// Evaluates as true because $var is set
if (isset($var)) {
echo '$var is set even though it is empty';
}
?>
```

floatval

It returns the float value of different variables:

```
<?php
$var = '122.34343The';
$float_value_of_var = floatval($var);
echo $float_value_of_var; // 122.34343
?>
```

get_defined_vars

It returns all defined variables, as an array:

```
<?php
$b = array(1, 1, 2, 3, 5, 8);

$arr = get_defined_vars();

// print $b
print_r($arr["b"]);

/* print path to the PHP interpreter (if used as a CGI)
* e.g. /usr/local/bin/php */
echo $arr["_"];

// print the command-line parameters if any
print_r($arr["argv"]);

// print all the server vars
```

get_resource_type

It returns the resource type:

```
<?php
// prints: stream
$fp = fopen("foo", "w");
echo get_resource_type($fp) . "\n";

// prints: curl
$c = curl_init ();
echo get_resource_type($c) . "\n"; // works prior to PHP 8.0.0 as since 8.0 curl_init returns a Curl
?>
```

gettype

It returns the type of different variables:

```
<?php
$a = 3;
echo gettype($a) ;
?>
```

intval

It returns the integer value of different variables:

```
<?php
echo intval(42); ?>
```

is_array

To check whether a variable is an array or not:

```
<?php
$a = "Hello";
echo "a is " . is_array($a) ;?>
```

Array

An array stores multiple values in one single variable.

Declaring an Array

```
<?php
$cms = array("Harry", "Lovish", "Rohan");
echo "Who needs chocolate? Is it" . $cms[0] . ", " .
$cms[1] . " or " . $cms[2] . "?";
?>
```

Functions

A function is a block of statements that can be used repeatedly in a program

Defining Functions

```
function NameOfTheFunction() {
//place PHP code here
}
```

MySQLi Functions

These functions allow you to access MySQL database server.

mysqli_connect() Function

It opens a non-persistent MySQL connection

```
mysqli_connect()
```

mysqli_affected_rows() Function

It returns the number of affected rows

```
mysqli_affected_rows()
```

mysqli_connect_error() Function

It shows the Error description for the connection error

```
mysqli_connect_error()
```

mysqli_fetch_all() Function

It fetches all result rows as an array

```
mysqli_fetch_all()
```

mysqli_fetch_array() Function

It fetches a result row as an associative, a numeric array, or both

```
mysqli_fetch_array()
```

mysqli_fetch_assoc() Function

It fetches a result row as an associative array

```
mysqli_fetch_assoc()
```

mysqli_fetch_row() Function

It fetches one row from a result set and returns it as an enumerated array

```
mysqli_fetch_row()
```

mysqli_kill() Function

It kills a MySQL thread

```
mysqli_kill()
```

mysqli_close() Function

It closes a database connection

```
mysqli_close()
```

Download this Cheatsheet

Add a new comment

Type Your Comment

Post Comment

Comments (21)

haseebwajidpersonal_gm 2024-01-02

Harry bhai, thank you so much for all the effort you've put into this. I found your channel when it only had a few thousand subscribers, and today, Mashallah, it has more than 5.2 million. Also, thank you for these notes and cheat sheets. They will be our next 'W3Schools' (and I hope better than that). Believe me, I haven't made any notes this semester, and it's the end of the 'Web Technologies' subject. I came across this and found this stuff.

REPLY

atifaslam.2022aa_gm 2023-12-20

harry bhai lvl !

REPLY

simrandangi9171_gm 2023-10-29

<a href="https://besttopets.com/">besttopets</a>

REPLY

maharhamza200019 2023-10-23

You are amazing Harry Bhai!!

REPLY

arfikhan527_gm 2023-09-21

Thanks Harry Bhai ☺

REPLY

mk.pro.lap.9095_gm 2023-09-19

Thank you so much sir you greatest coder means harry bhai

VIEW ALL REPLIES

REPLY

monty35only_gm 2023-08-19

Salute sir.

REPLY

rohit.msl85 2023-06-11

awesome harry

REPLY

harsharora1816_gm 2023-06-02

see my site i have make this in PHP https://www.getcustomercarenumbers.com

REPLY

harsharora1816_gm 2023-06-02

Need advance <a href="https://www.getcustomercarenumbers.com">php course</a>

REPLY

harsharora1816_gm 2023-06-02

PHP Cheatsheet was really very helpful!

REPLY

pankajpensia107_gm 2023-05-22

Worlds best coding teacher

VIEW ALL REPLIES

REPLY

tounsils_gm 2023-04-17

Interesting. Thank you tn76.com

REPLY

sonukumarsinghsingh502_gm 2022-11-19 thank you so much sir

REPLY

mr.umang10_gm 2022-10-27

thank you so much

REPLY

mehzee92_gm 2022-10-09

Love from Pakistan My all times favorite person.

REPLY

anilcoder567849_gm 2022-09-20

https://www.codewithharry.com is amazing website to learn programming

REPLY

anilcoder567849_gm 2022-09-20

good

REPLY

Souravakar10 2022-07-29

I want to learn javascript & php. Please provide the source code and how to exicute it.

REPLY

shikha.dongre30 2022-07-27

Thank you so much Harry. I was just revising the PHP to restart after a career gap and you made it easy. I just scroll down and all my memory freshen up. It is almost similar to my PHP notes.

REPLY

bhavesh.kanjariya 2022-07-24

Harry sir how are u? Sir i want to meet u if possible Please

VIEW ALL REPLIES

REPLY