PHP Echo

PHP echo is a language construct, not a function. Therefore, you don't need to use parenthesis with it. But if you want to use more than one parameter, it is required to use parenthesis.

The syntax of PHP echo is given below:

1. void echo ( string $arg1 [, string $... ] )

PHP echo statement can be used to print the string, multi-line strings, escaping characters, variable, array, etc. Some important points that you must know about the echo statement are:

* echo is a statement, which is used to display the output.
* echo can be used with or without parentheses: echo(), and echo.
* echo does not return any value.
* We can pass multiple strings separated by a comma (,) in echo.
* echo is faster than the print statement.

PHP echo: printing string

1. **<?php**
2. echo "Hello by PHP echo";
3. **?>**
4. **Output:**
5. Hello by PHP echo

## PHP echo: printing multi line string

1. **<?php**
2. echo "Hello by PHP echo
3. this is multi line
4. text printed by
5. PHP echo statement
6. ";
7. **?>**

**Output:**

Hello by PHP echo this is multi line text printed by PHP echo statement

## PHP echo: printing escaping characters

1. **<?php**
2. echo "Hello escape \"sequence\" characters";
3. **?>**
4. **Output:**
5. Hello escape "sequence" characters

## PHP echo: printing variable value

1. **<?php**
2. $msg="Hello JavaTpoint PHP";
3. echo "Message is: $msg";
4. **?>**
5. **Output:**
6. Message is: Hello JavaTpoint PHP

PHP Print

Like PHP echo, PHP print is a language construct, so you don't need to use parenthesis with the argument list. Print statement can be used with or without parentheses: print and print(). Unlike echo, it always returns 1.

The syntax of PHP print is given below:

1. int print(string $arg)

PHP print statement can be used to print the string, multi-line strings, escaping characters, variable, array, etc. Some important points that you must know about the echo statement are:

* print is a statement, used as an alternative to echo at many times to display the output.
* print can be used with or without parentheses.
* print always returns an integer value, which is 1.
* Using print, we cannot pass multiple arguments.
* print is slower than the echo statement.

PHP print: printing string

1. **<?php**
2. print "Hello by PHP print ";
3. print ("Hello by PHP print()");
4. **?>**

**Output:**

Hello by PHP print Hello by PHP print()

PHP print: printing multi line string

1. **<?php**
2. print "Hello by PHP print
3. this is multi line
4. text printed by
5. PHP print statement
6. ";
7. **?>**

**Output:**

Hello by PHP print this is multi line text printed by PHP print statement

PHP print: printing escaping characters

1. **<?php**
2. print "Hello escape \"sequence\" characters by PHP print";
3. **?>**

**Output:**

Hello escape "sequence" characters by PHP print

PHP print: printing variable value

*File: print4.php*

1. **<?php**
2. $msg="Hello print() in PHP";
3. print "Message is: $msg";
4. **?>**

**Output:**

Message is: Hello print() in PHP

# PHP echo and print Statements

We frequently use the echo statement to display the output. There are two basic ways to get the output in PHP:

* echo
* print

echo and print are language constructs, and they never behave like a function. Therefore, there is no requirement for parentheses. However, both the statements can be used with or without parentheses. We can use these statements to output variables or strings.

## Difference between echo and print

### **echo**

* echo is a statement, which is used to display the output.
* echo can be used with or without parentheses.
* echo does not return any value.
* We can pass multiple strings separated by comma (,) in echo.
* echo is faster than print statement.

### **print**

* print is also a statement, used as an alternative to echo at many times to display the output.
* print can be used with or without parentheses.
* print always returns an integer value, which is 1.
* Using print, we cannot pass multiple arguments.
* print is slower than echo statement.

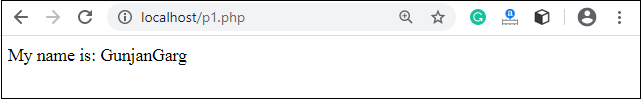
You can see the difference between echo and print statements with the help of the following programs.

### **For Example (Check multiple arguments)**

You can pass multiple arguments separated by a comma (,) in echo. It will not generate any syntax error.

1. **<?php**
2. $fname = "Gunjan";
3. $lname = "Garg";
4. echo "My name is: ".$fname,$lname;
5. **?>**

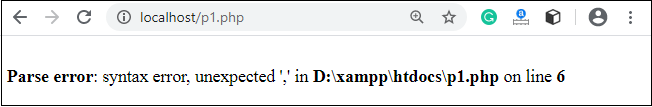
**Output:**



It will generate a **syntax error** because of multiple arguments in a print statement.

1. **<?php**
2. $fname = "Gunjan";
3. $lname = "Garg";
4. print "My name is: ".$fname,$lname;
5. **?>**

**Output:**

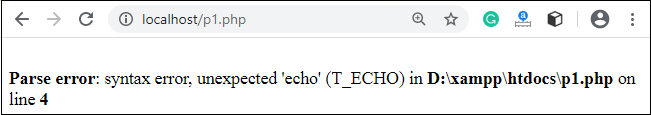


### **For Example (Check Return Value)**

echo statement does not return any value. It will generate an error if you try to display its return value.

1. **<?php**
2. $lang = "PHP";
3. $ret = echo $lang." is a web development language.";
4. echo "**</br>**";
5. echo "Value return by print statement: ".$ret;
6. **?>**

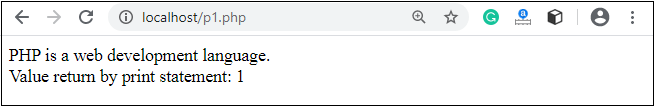
**Output:**



As we already discussed that print returns a value, which is always 1.

1. **<?php**
2. $lang = "PHP";
3. $ret = print $lang." is a web development language.";
4. print "**</br>**";
5. print "Value return by print statement: ".$ret;
6. **?>**

**Output:**



PHP Variables

<="" p="" style="color: rgb(51, 51, 51); font-family: inter-regular, system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Helvetica, Arial, sans-serif; font-size: 16px; font-style: normal; font-variant-ligatures: normal; font-variant-caps: normal; font-weight: 400; letter-spacing: normal; orphans: 2; text-align: justify; text-indent: 0px; text-transform: none; white-space: normal; widows: 2; word-spacing: 0px; -webkit-text-stroke-width: 0px; background-color: rgb(255, 255, 255); text-decoration-thickness: initial; text-decoration-style: initial; text-decoration-color: initial;">

In PHP, a variable is declared using a **$ sign** followed by the variable name. Here, some important points to know about variables:

* As PHP is a loosely typed language, so we do not need to declare the data types of the variables. It automatically analyzes the values and makes conversions to its correct datatype.
* After declaring a variable, it can be reused throughout the code.
* Assignment Operator (=) is used to assign the value to a variable.

Syntax of declaring a variable in PHP is given below:

1. $variablename=value;

Rules for declaring PHP variable:

* A variable must start with a dollar ($) sign, followed by the variable name.
* It can only contain alpha-numeric character and underscore (A-z, 0-9, \_).
* A variable name must start with a letter or underscore (\_) character.
* A PHP variable name cannot contain spaces.
* One thing to be kept in mind that the variable name cannot start with a number or special symbols.
* PHP variables are case-sensitive, so $name and $NAME both are treated as different variable.

PHP Variable: Declaring string, integer, and float

Let's see the example to store string, integer, and float values in PHP variables.

*File: variable1.php*

1. **<?php**
2. $str="hello string";
3. $x=200;
4. $y=44.6;
5. echo "string is: $str **<br/>**";
6. echo "integer is: $x **<br/>**";
7. echo "float is: $y **<br/>**";
8. **?>**

**Output:**

string is: hello string

integer is: 200

float is: 44.6

PHP Variable: Sum of two variables

*File: variable2.php*

1. **<?php**
2. $x=5;
3. $y=6;
4. $z=$x+$y;
5. echo $z;
6. **?>**

**Output:**

11

PHP Variable: case sensitive

In PHP, variable names are case sensitive. So variable name "color" is different from Color, COLOR, COLor etc.

1. **<?php**
2. $color="red";
3. echo "My car is " . $color . "**<br>**";
4. echo "My house is " . $COLOR . "**<br>**";
5. echo "My boat is " . $coLOR . "**<br>**";
6. **?>**

**Output:**

My car is red

Notice: Undefined variable: COLOR in C:\wamp\www\variable.php on line 4

My house is

Notice: Undefined variable: coLOR in C:\wamp\www\variable.php on line 5

My boat is

PHP Variable: Rules

PHP variables must start with letter or underscore only.

PHP variable can't be start with numbers and special symbols.

**<?php**

1. $a="hello";//letter (valid)
2. $\_b="hello";//underscore (valid)
4. echo "$a **<br/>** $\_b";
5. **?>**

**Output:**

hello

hello

1. **<?php**
2. $4c="hello";//number (invalid)
3. $\*d="hello";//special symbol (invalid)
5. echo "$4c **<br/>** $\*d";
6. **?>**

**Output:**

Parse error: syntax error, unexpected '4' (T\_LNUMBER), expecting variable (T\_VARIABLE)

or '$' in C:\wamp\www\variableinvalid.php on line 2

PHP: Loosely typed language

PHP is a loosely typed language, it means PHP automatically converts the variable to its correct data type.