



Introduction to server side scripting Language / PHP

What is PHP ?

PHP stands for **H**ypertext **P**reprocessor

PHP is a server-side scripting language, scripts are executed on the server

PHP supports many databases (MySQL, Informix, Oracle, Sybase , Solid, PostgreSQL,

PHP is free to download and use open source software

PHP files may contain text, HTML tags and scripts

PHP files are returned to the browser as plain HTML

PHP files have a file extension of ‘.php‘, ‘.php3‘, or ‘.phtml‘

PHP runs on different platforms (Windows, Linux, Unix, etc.)

PHP is compatible with almost all servers used today (Apache, IIS, etc.)

PHP Syntax

Syntax is a way to representation of PHP script, which gives the primary idea to specify the code format.

There are three ways to start PHP environment

1. The most commonly and effective PHP syntax:

```
<?php  
echo 'Welcome to the world of php';  
?|
```

2. Short or short-open tags look like this:

```
<?  
echo 'welcome to the world of php';  
?|
```

3. HTML script tags:

```
<script language='PHP' |  
    echo 'welcome to the world of php';  
</script|
```

Variables in PHP

Variable is nothing it is just name of the memory location.

A Variable is simply a container i.e used to store both numeric and non-numeric information contain strings, numbers, or arrays.

All variables in PHP start with a `$` sign symbol.

Assigning values to a variables

```
<html| <body|  
<?php  
    txt='Hello World';  
echo    txt;  
?|  
</body| </html|
```

```
<html| <body|  
<?php  
    txt1='Hello World';  
    txt2='1234';  
echo    txt1 . ' ' . txt2 ;  
?|  
</body| </html|
```

Variable names in PHP are case-sensitive

< ?php

```
name='rexx';
```

```
NAME='rahul';
```

```
echo name.<br/!';
```

```
echo NAME;
```

?|

Output

rexx

rahul

Sum of two numbers

< ?php

```
first = 100;
```

```
second = 200;
```

third = first +

second;

```
echo 'Sum =
```

```
        ' third;
```

?

Destroying PHP Variables

- ↳ To destroy a variable, pass the variable to PHP's **unset()** function.

< ?php

```
first = 100;
```

```
second = 200;
```

```
third = first + second;
```

```
echo 'Sum = ' . third;
```

```
unset(  third);
```

```
//after delete the variable call it again to test
```

```
echo 'Sum = ' . third;
```

?

Output

Sum = 300

Sum= Notice error undefined third variable

Inspecting Variable Contents(Variable Property) (variable X-rays)

```
<?php
```

```
    name = 'Fiona';
```

```
    age=25;
```

```
//display variable contents
```

```
var_dump ( name);
```

```
var_dump( age);
```

```
?|
```

Output: string 'Fiona' (length=5)

int 25

var and

var in PHP

PHP **var** uses the value of the variable whose name is the value of **var**.

It means **var** is known as reference **variable** where as **var** is normal variable.

It allows you to have a “variable’s variable”

```
<?php
```

```
    name='Rajeev';
```

```
    name='Sanjeev';
```

```
echo    name.<br/>';
```

```
echo    name.<br/>';
```

```
echo    Rajeev;
```

```
?|
```


Displaying to Browser

PHP echo and print both are PHP Statement, used to display the output in PHP.

Echo	Print
used with parentheses echo or without parentheses echo	can be used with parentheses print() or without parentheses print
can pass multiple string separated as (,)	doesn't pass multiple argument
doesn't return any value	always return 1
faster then print	slower than echo

Example

Echo

```
< ?php
echo "Enter text here";
    name='John';
echo    name; //or echo (    name);
?|
```

Output

Enter text here John

```
< ?php
    name = 'John';
    profile = 'PHP Developer';
    age = 25;
echo    name ,    profile ,    age, ' years
old';
?|
```

echo statement
separated by comm or
dot(, or .)

Output

John PHP Developer 25
years old

```
< ?php
    name = 'John';
    ret = echo    name;
?|
```

Output

**Parse error: syntax
error, unexpected
T_ECHO**

Example

Print

```
< ?php
    name='John';
print  name; //or print (  name);
?|
```

Output
John

```
< ?php
    name = 'John';
    profile = 'PHP Developer';
    age = 25;
print  name ,  profile ,  age, ' years
old';
?|
```

Output
**Parse error: syntax
error**

```
< ?php
    name = 'John';
    ret = print  name;
//To test it returns or not
echo  ret;
?|
```

Output
John

Comments in PHP

A comment is non-executable lines used to write description for your own understanding.

Browser doesn't read the comments.

1. Single line comments :

Single line comment used for short explanations.

Either Begin with(#) Or backslash(//)

2. Multi-lines comments :

bulk comments are enclosed within (/*.....*/)

Example

Single Line

```
< ?php
# This is the single line comment
# This is the next line comment
// This is also a single line
comment.

?|
```

Another example

```
< ?php
    str= 'welcome ';
//  str. =' student';
echo    str;

?|
```

Multi line

```
< ?php
/* This is multiline comment
Developer : sanjeev rai
view : Comments Demo */

?|
```

Another example

```
< ?php
/*      str = 'welcome ';
        str.= 'users ';    */
echo 'Hello user how are you ? ';

?|
```

Output: Hello user how are you ?

Send Data to the Web Browser

1. PUTTING EVERYTHING ON ONE PAGE

```
<?php
```

```
NAME='ENTER NAME HERE';
```

```
AGE='ENTER AGE HERE';
```

```
HOME='ENTER HOME TOWN HERE';
```

```
echo <<<TEXT
```

```
<p>Hello, my name is NAME, I am AGE years  
old and live in HOME.</p>
```

```
TEXT;
```

```
?|
```

Continued

2. PASSING INFORMATION FROM ONE PAGE TO ANOTHER

PAGE

```
<html| <head|</head|  
<body|  
<form id='Form' action='page-two.php' method='post' |  
  
<p|Enter Name: <input type='text' name='Name' /|</p|  
  
<p|Enter Age: <input type='text' name='Age' /|</p|  
  
<p|Enter Hometown: <input type='text' name='Town'  
/|</p|  
  
<p|<input type='Submit' name='Form_Submit'  
value='Send' /|</p|  
</form| </body| </html|
```

PHP file

```
< ?php
```

```
NAME= $_REQUEST['Name']; AGE= $_REQUEST['Age'];
```

```
HOME= $_REQUEST['Town'];
```

```
echo <<<TEXT <p|Hello, my name is <b| NAME</b|, I am  
<b| AGE</b| years old and live in <b| HOME</b|.</p|
```

```
TEXT;
```

```
?|
```


Constant in PHP

Constants are PHP container that remain constant and never change

Constants are used for data that is unchanged at multiple place within our program.

Variables are temporary storage while Constants are permanent.

Use Constants for values that remain fixed and referenced multiple times.

Rules for defining constant

1. Constants are defined using PHP's **define()** function, which accepts two arguments: The name of the constant, and its value.
2. Constant name must follow the same rules as variable names, with one exception the “ ” prefix is not required for constant names.

Syntax and example

```
<?php
```

```
define('ConstName',  
'value');
```

```
?|
```

Eg1.

Create a constant and assign your name

```
<?php
```

```
define('NAME', 'Rexx');  
echo 'Hello '.NAME;
```

```
?|
```

Output : Hello Rexx

Eg2.

Sum of two numbers using constant

```
<?php
```

```
define('ONE', 100);  
define('TWO', 100);  
define('SUM', ONE+TWO);
```

```
print 'Sum of two constant='.SUM;
```

```
?|
```

Output: Sum of two constant = 200

OR

```
<?php
```

```
define('ONE', 100);  
define('TWO', 100);  
res= ONE+TWO;
```

```
print 'Sum of two constant='. res;
```

```
?|
```



Magic Constant

Predefined constants available to your scripts

<code>__LINE__</code>	The current line number of the file.
<code>__FILE__</code>	The full path and filename of the file.
<code>__FUNCTION__</code>	The function name
<code>__CLASS__</code>	The class name
<code>__METHOD__</code>	The class method name
<code>PHP_VERSION</code>	The PHP version
<code>PHP_INT_MAX</code>	The PHP integer value limit

Example

```
<?php
```

```
echo 'The Line number : ' . __LINE__;
```

```
echo 'Current PHP Version you are using : ' . PHP_VERSION;
```

```
?|
```

Output

The Line number : 2

Current PHP Version you are using : 5.3.1

```
<?php
```

```
echo 'Your file name : ' . __FILE__;
```

```
echo 'Integer Maximum Value :
```

```
' . PHP_INT_MAX;
```

```
?|
```

Output

Your file name :

C:\xampp\lite\htdocs\magic_constantfile.php

Integer Maximum Value : 2147483647

```
<?php
```

```
class demo {
```

```
function test() {
```

```
echo 'Function of demo class : ' .
```

```
__FUNCTION__ . '<br/>';
```

```
}
```

```
function testme() {
```

```
echo 'Method of demo class : ' . __METHOD__ . '<br/>';
```

```
echo 'Class : ' . __CLASS__;
```

```
} }
```

```
object=new demo();
```

```
object->test();
```

```
object->testme();
```

```
?|
```

Output

Function of demo class :

test

Method of demo class :

demo::testme

Class : demo



Thank You !!