1. **Both**, **PHP** is **interpreted, but** PHP is compiled down to an intermediate bytecode that is then interpreted by the runtime zend engine.

The Zend Engine is the open source scripting engine that interprets the PHP programming language.

**PHP compiler is responsible for:**

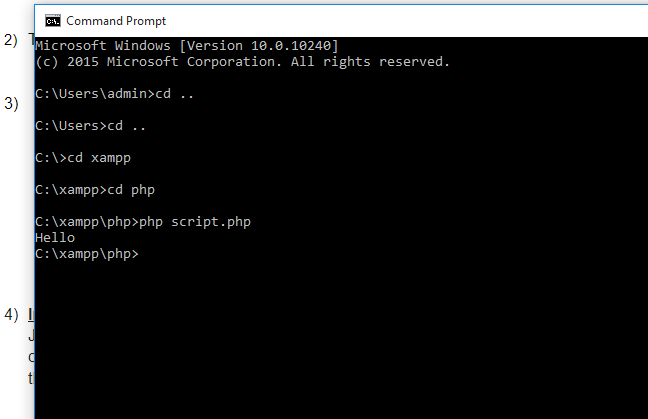
* Converting code to a bytecode that can be used by runtime engine.
* Resolve functions,names and classes names
* Creating symbol table.( The **@symbol** is the error control operator. It makes **PHP** suppress any error messages (notice, warning, fatal, etc) generated by the associated expression. It works just like a unary operator.)

PHP Interpreter does:

* Does goes through the bytecode line by line and executes it.
* Handles runtime exceptions.

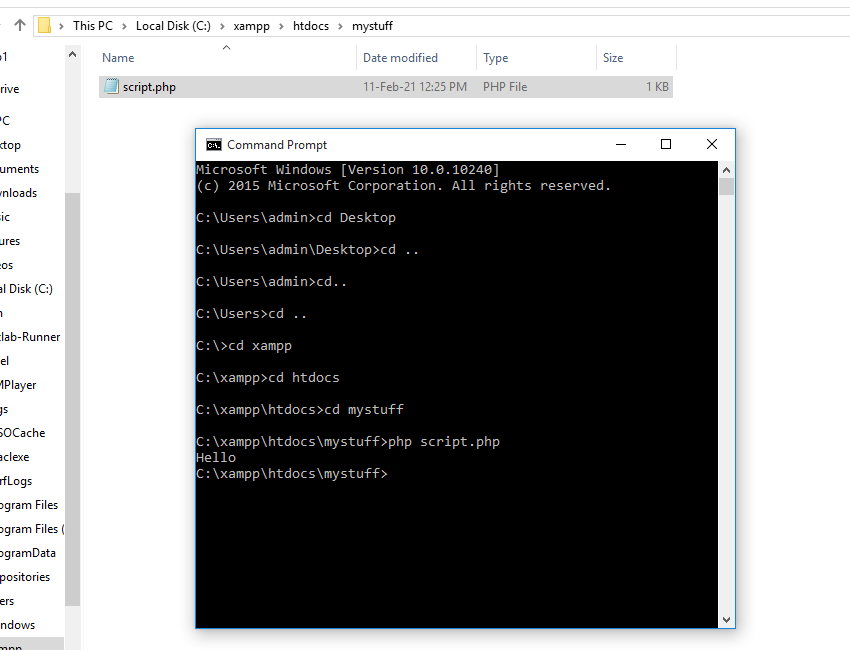
1. The latest stable version of PHP includes 8.0.2 / 4 February 2021.

* PHP is included in the installer package. XAMPP should be installed in the location C:\xampp and it has a folder named php (C:\xampp\php) .
* Open cmd. Go to php folder directory cd c:\xampp\php
* Check version - **php -v**

****

1. **In Command Line-**

Just make a program with .php extension inside the htdocs folder and on command prompt - run as php filename.php going to the htdocs directory where the php file is located with cd .



**In Eclipse-**

*Open Eclipse*

Windows =>Open Perspectives => others =>PHP

Now you should see the XAMPP bar on the top of the window with icons to start/stop XAMPP/MySQL/Apache.

Create a new PHP project. File => Give a proper Project name. IMPORTANT: Uncheck “use default location” and browse this path “C:\xampp\htdocs<Project folder>”. Project folder should be newly created at that location.

So all your project file will go into “C:\xampp\htdocs<Project folder>”.

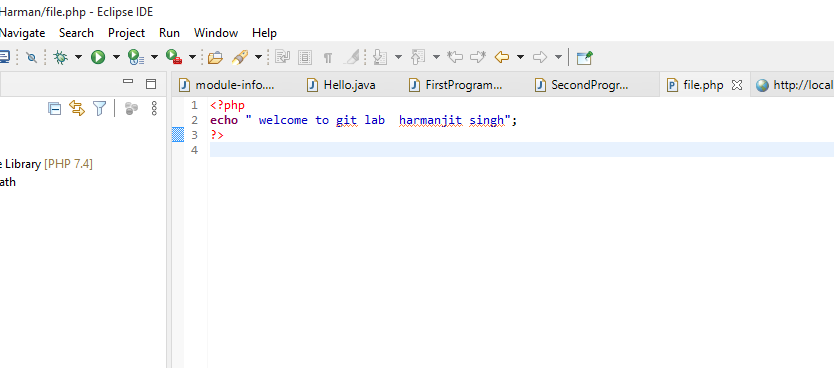
Now create a new HTML page under this project and edit the file.

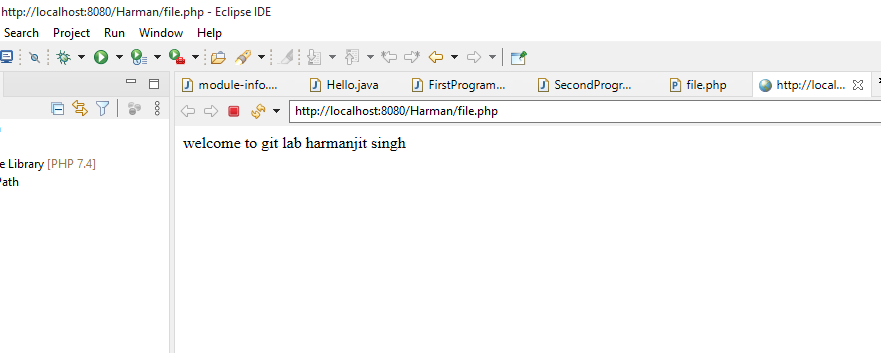
Start XAMPP using the icon in eclipse.

Right click on the HTML page and click “open PHP browser”.

The result would be negative. In the browser path, edit the path to include the created which will be missing in the default browser path.

Eg- <http://localhost:8080/Harman/file.php>





1. There are few guidelines which can be followed while coding in PHP.

* Indenting and Line Length − Use an indent of 4 spaces and don't use any tab because different computers use different settings for tab. It is recommended to keep lines at approximately 75-85 characters long for better code readability.
* Control statements should have one space between the control keyword and opening parenthesis, to distinguish them from function calls.
* C style comments (/\* \*/) and standard C++ comments (//) are both fine. Use of Perl/shell style comments (#) is discouraged.
* Always use <?php ?> to write PHP code.

1. **COMMON STRING FUNCTIONS IN PHP-**
2. Counting of the number of words in a String-

This function is also useful in validation of input fields.

**Syntax -** Str\_word\_count(string)

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

echo str\_word\_count("Harman is smart!"); //3

?>

</body>

</html>

1. Reversing a String-

Strrev() is used for reversing a string.

Syntax - Strev(string)

Example

strrev(“Welcome to Cloudways”);

Output- syawduolC ot emocleW

1. Finding Text Within a String-

Strpos() enables searching particular text within a string. It works simply by matching the specific text in a string.

**Syntax-** Strpos(string,text);

<!DOCTYPE html>

<html>

<body>

<?php

echo strpos("I love php, I love php too!","php"); //7

?>

</body>

</html>

1. Replacing text within a string-

**Str\_replace()** is a built-in function, basically used for replacing specific text within a string.

Syntax- Str\_replace(string to be replaced,text,string)

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

echo str\_replace("world","Harman","Hi world!"); //Hi Harman!

?>

</body>

</html>

1. Converting lowercase into Title Case-

Ucwords() is used to convert the first alphabet of every word into uppercase.

Syntax - Ucwords(string)

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

echo ucwords("old is gold");

?>

</body>

</html>

Output-

Old is Gold

1. Converting a whole string into UPPERCASE-

Strtoupper() is used to convert a whole string into uppercase.

Syntax - Strtoupper(string);

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

echo strtoupper("Hello"); //HELLO

?>

</body>

</html>

1. Converting the whole String to lowercase-

Strtolower() is used to convert a string into lowercase.

Syntax- Strtolower(string)

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

echo strtolower("Work Hard"); //work hard

?>

</body>

</html>

1. Repeating a String-

PHP provides a built-in function for repeating a string a specific number of times.

Syntax- Str\_repeat(string,repeat)

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

echo str\_repeat("Work",10); /*/WorkWorkWorkWorkWorkWorkWorkWorkWorkWork*

?>

</body>

</html>

1. Comparing Strings-

We can compare two strings by using **strcmp()**. It returns output either greater than zero, less than zero or equal to zero.

Syntax- Strcmp(string1,string2)

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

echo strcmp("Hi!","Hi!"); //0

?>

</body>

</html>

1. Displaying part of String Through substr() function - We can display or extract a string from a particular position.

Syntax -substr(string,start,length)

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

echo substr("Hello Webners",6); //Webners

?>

</body>

</html>

1. Removing white spaces from a String- Trim() is dedicated to remove white spaces and predefined characters from both the sides of a string.

Syntax- trim(string,charlist)

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

$str = "Work Hard!";

echo trim($str,"Wrd!"); //ork Ha

?>

</body>

</html>

1. **addcslashes()-**

Returns a string with backslash in front of the specified characters.

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

$str = addcslashes("Webner Solutions","S");

echo($str);

?>

</body>

</html>

Output- Webner \Solutions

1. **addslashes()-**

Returns a string with backslashes in front of predefined characters.

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

$str = addslashes('Winners "never" quit');

echo($str);

?>

</body>

</html>

Output-

Winners \"never\" quit

13) **Strstr() Function-**

strstr(String, search text);

It returns the first occurrence of search text to the end of the String. Is a case sensitive function.

Eg-

<?php

$string = “understanding means transforming some text”;

$txt = “some”

echo strstr($string, $txt);

?>

Output-

some text

1. **Strrev() Function-**

Syntax-

strrev(string)

It returns the reverse of string.

Eg- java is a string, strrev function will return avaj.

Eg-

<?php

Echo strev(“Welcome to PHP”);

?>

Output-

PHP of emociew

1. **Strlen() Function-**

strlen(string);

It returns the number of character in a string. It is used to find the length of the string.

Eg-

<?php

$name=”Harman”;

if(strlen($name) !=7)

{

echo “correct input”;

}

else

{

echo “incorrect input”;

}

?>

Output-

Correct input.

1. **Chr() Function-**

Syntax-

chr(integer);

It returns the single character string representation of the ASCII byte value.

chr() function is used to find the ascii character of any integer.

Eg-

<?php

Echo chr(35);

?>

Output-

#

1. **bin2hex() -**

Converts a string of ASCII characters to hexadecimal values.

Eg-

<?php

$str=”a”;

echo bin2hex($str) . “<br/>”;

echo pack(“H\*”,’61’), “<br />”;

?>

Output-

61

a

1. **chop()-**

Alias of rtrim()

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

$str = "Hello World!";

echo $str . "<br>";

echo chop($str,"World!");

?>

</body>

</html>

Output-

Hello World!

Hello

2) <!DOCTYPE html>

<html>

<body>

<?php

$str = "Hello \n\n";

echo $str;

echo chop($str);

?>

</body>

</html>

Output-

Hello ! Hello !

1. **vfprintf()-**

It returns the length of the outputted string.

vfprintf(stream, format,arg)

1. **explode()-**

Breaks a string into an array.

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

$str = "I work. in webner solutions.";

print\_r (explode(" ",$str));

?>

</body>

</html>

1. **Strlen()-**

Strlen() displays the length of any string.

**Syntax** - Strlen(string);

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

echo strlen("Harman"); //6

?>

</body>

</html>

7) **COMMON DATE AND TIME FUNCTIONS IN PHP INCLUDES-**

1. checkdate()-

Checks a date for validity, which accepts a month,day and year combination and returns true/false indicating date is valid.

Eg-

<?php

if(checkdate(2,30,2008)

{

echo ‘Date is valid’;

}

else

{

echo ‘Date is invalid’;

}

?>

Output-

Date is invalid.

1. strtotime()-

Creates timestamps from English-language descriptions.

Eg-

<?php

$str=’July 7 2008’;

echo date(‘d M y’,strtotime($str));

?>

Output-

‘7 Jul 2008’

1. gmdate()-

Expresses a timestamp in GMT.

gmdate(*format, timestamp)*

*Eg-*

<?php

echo gmdate("l") . "<br>";

echo gmdate("l jS \of F Y h:i:s A") . "<br>";

echo "Feb 12, 2021 was on a ".gmdate("l", mktime(0,0,0,02,12,2021)) . "<br>";

echo gmdate(DATE\_RFC822) . "<br>";

echo gmdate(DATE\_ATOM,mktime(0,0,0,02,12,2021));

?>

1. date\_add() -

Adds days, months, years, hours, minutes, and seconds to a date.

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

$date=date\_create("2021-02-12");

date\_add($date,date\_interval\_create\_from\_date\_string("10 days"));

echo date\_format($date,"Y-m-d");

?>

</body>

</html>

1. date\_format()-

Returns a new DateTime object, and then formats the date.

Eg-

<!DOCTYPE html>

<html>

<body>

<?php

$date=date\_create("2021-02-12");

echo date\_format($date,"Y/m/d H:i:s");

?>

</body>

</html>

8) METHODS-

1. **PHP array() function-**

The array() function is used to create an array.

There are 3 types of arrays-

1. Indexed Arrays
2. Associative Arrays
3. Multidimensional Arrays

Eg-

<html>

<body>

<?php

$dishes=array("Barfi","Gulabjamun","Khoya");

echo "I like " . $dishes[0] . ", " . $dishes[1] . " and " . $dishes[2] . ".";

?>

</body>

</html>

**02. PHP array\_chunk() function-**

The array\_chunk() function splits an array into chunks of new arrays.

Eg-

<html>

<body>

<?php

$bikes=array("Motorola","Harley","hybooza","splendor");

print\_r(array\_chunk($bikes,2)); *//Array ( [0] => Array ( [0] => Motorola [1] => Harley ) [1] => Array ( [0] => hybooza [1] => splendor ) )*

?>

</body>

</html>

2) <html>

<body>

<?php

$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43","Harry"=>"50"); *//Array ( [0] => Array ( [Peter] => 35 [Ben] => 37 ) [1] => Array ( [Joe] => 43 [Harry] => 50 ) )*

print\_r(array\_chunk($age,2,true));

?>

</body>

</html>

**03. PHP array\_combine() function-**

The array\_combine() function creates an array by using the elements from one "keys" array and one "values" array.

Eg-

<html>

<body>

<?php

$fname=array("Harman","Harry","Harish");

$age=array("21","20","25");

$b=array\_combine($fname,$age);

print\_r($b);

?>

</body>

</html>

**Output-**

Array ( [Harman] => 21 [Harry] => 20 [Harish] => 25 )

**04) PHP array\_flip() function-**

The array\_flip() function flips/exchanges all keys with their associated values in an array.

Eg-

<html>

<body>

<?php

$a=array("a"=>"Harman","b"=>"works","c"=>"in","d"=>"WebnerSolutions");

$result=array\_flip($a);

print\_r($result);

?>

</body>

</html>

Output-

Array ( [Harman] => a [works] => b [in] => c [WebnerSolutions] => d )

**05) PHP array\_pop() function-**

Delete the last element of the array.

Eg-

<html>

<body>

<?php

$p=array("apple","orange","litchi"); //Array ( [0] => apple [1] => orange )

array\_pop($p);

print\_r($p);

?>

</body>

</html>

**06) PHP array\_Slice() function-**

Return the rest of the elements in the array.

Eg-

<html>

<body>

<?php

$a=array("red","green","blue","yellow","brown");

print\_r(array\_slice($a,2)); //Array ( [0] => blue [1] => yellow [2] => brown )

?>

</body>

</html>

9) Eg-

SET-

A set is a sequence of unique values.

Eg-

isset() function-

The isset() function checks whether a variable is set, which means that it has to be declared and is not NULL.

Eg-

<?php

$a = 0;

// True because $a is set

if (isset($a)) {

echo "Variable 'a' is set.<br>";

}

$b = null;

// False because $b is NULL

if (isset($b)) {

echo "Variable 'b' is set.";

}

?>

**MAP-**

The array\_map() function sends each value of an array to a user-made function, and returns an array with new values, given by the user-made function.

Eg-

<html>

<body>

<?php

function myfunction($v)

{

if ($v==="Dog")

{

return "Bruno";

}

return $v;

}

$a=array("Horse","Dog","Cat");

print\_r(array\_map("myfunction",$a)); //Array ( [0] => Horse [1] => Bruno [2] => Cat )

?>

</body>

</html>

POINTERS IN PHP-

PHP does not have pointers but it does have references. References are not pointers.

<?php

$a=0;

$b=&a;

echo $a; //0

unset($b);

echo $a;

10) PHP STRINGS-

1. <?php

echo str\_word\_count("Hi everyone"); //2

?>

1. <?php

echo strrev("Harman"); //namraH

?>

1. <?php

echo strpos("My name is Harman!", "Harman"); //11

?>

1. <?php

echo str\_replace("Beautiful", "Handsome", "Hello Beautiful");

?> //Hello Handsome

PHP DATE()-

<?php

echo date("l") . "<br>";

echo date("l jS \of F Y h:i:s A") . "<br>";

echo "Jan 20,2021 was on a ".date("l", mktime(0,0,0,01,20,2021)) . "<br>";

echo date(DATE\_RFC822) . "<br>";

echo date(DATE\_ATOM,mktime(0,0,0,01,20,2021));

?>

PHP TIME()-

<?php

$currentTimeinSeconds = time();

$currentDate = date('Y-m-d', $currentTimeinSeconds);

echo ($currentDate);

?>

PHP ARRAYS()-

<html>

<body>

<?php

$dishes=array("Barfi","Gulabjamun","Khoya");

echo "I like " . $dishes[0] . ", " . $dishes[1] . " and " . $dishes[2] . ".";

?>

</body>

</html>

PHP MAPS()-

<html>

<body>

<?php

function myfunction($v)

{

if ($v==="Dog")

{

return "Bruno";

}

return $v;

}

$a=array("Horse","Dog","Cat");

print\_r(array\_map("myfunction",$a)); //Array ( [0] => Horse [1] => Bruno [2] => Cat )

?>

</body>

</html>

PHP SET()-

<?php

$a = 0;

// True because $a is set

if (isset($a)) {

echo "Variable 'a' is set.<br>";

}

$b = null;

// False because $b is NULL

if (isset($b)) {

echo "Variable 'b' is set.";

}

?>

11) <html>

<body>

<?php

echo substr\_count("W. W is alphabet","W"); //2

?>

</body>

</html>

<?php

echo substr\_count("Toyata Car. Toyata Car is very expensive","Car"); //2

?>

12) <?php

$sentence= ["Hi my name is Harman"];

$vowels = ["a", "e", "i", "o", "u", "A", "E", "I", "O", "U"];

$length = count($sentence);

$xcount = 0;

for($count = 0; $count < $length; $count++) {

$consonants = str\_replace($vowels, "", $sentence[$count]);

echo "{$consonants}<br />";

$xcount++;

}

?>

Output-

H my nm s Hrmn

13) <html>

<body>

<?php

$text = "Hi my name is Harmanjit Singh.";

$newtext = wordwrap($text, 20, "<br />\n");

echo $newtext;

?>

</body>

</html>

14) <!DOCTYPE html>

<html>

<body>

<?php

date\_default\_timezone\_set("America/New\_York");

echo $time1= date("H:i a");

date\_default\_timezone\_set("America/Chicago");

echo $time2= date("H:i a");

date\_default\_timezone\_set("Europe/London");

echo $time3= date("H:i a");

?>

</body>

</html>

->

<html>

<body>

<?php

date\_default\_timezone\_set('Europe/London');

$sTime = date("d-m-Y H:i:s");

print 'The time is: ' . $sTime;

?>

</body>

</html>

15) <?php

$date1 = “2016-06-26 12:00:00”;

$date2 = “2016-06-27 1:00:00”;

$timestamp1=strtotime($date1);

$timestamp2=strtotime($date2);

echo “difference b/w 2 dates “.$hour= abs($timestamp2-$timestamp1)/3600;

?>

OR

<?php

function differenceInHours($startdate,$enddate)

{

$starttimestamp=strtotime($startdate);

$endtimestamp=strtotime($enddate);

$difference=abs($endtimestamp-$starttimestamp)/3600;

return $difference;

}