PHP

Inventer of PHP : Rasmus Lerdorf

Early name of pHp : Personal home page / Form handling (pHp/fi)

- Introduced in 1995 .
- pHp is a server-side scripting language .

Present name of pHp : Hypertext preprocessor .

Html ke content ko dynamic bnata hai. **Output:**

only browser

php code run : firstly php ka code ko server par run karta hai then output browser ko send krta hai .

(phle php phir html)

All OS and browser support pHp

Requirement to run a pHp code -

1. Install php
2. MySQL
3. Apache

Wamp (Windows Apache MySQL PHP) server

Xamp (Cross-platform Apache MYSQL PHP) server

Syntax :

```
<?php
    // code
?>
```

Comment :

single line comment : // , #

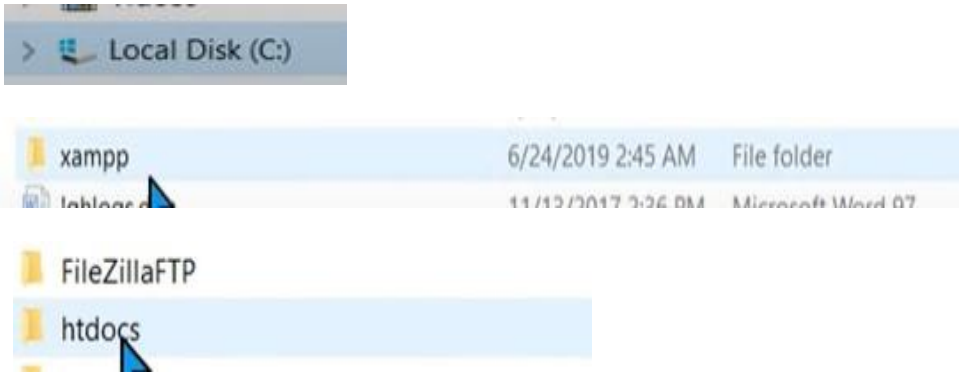
multiline comment :- / ..... */* **Note :**

Php ke code ko hum server ki files (jaha server ka path hota hai) hoti hai vhi likhte hai .

In XAMP Server :

C drive ❼ xamp folder ❼ htdocs folder ❼ own file **htdocs**

folder (main – jiske ander hme kaam karna hai)

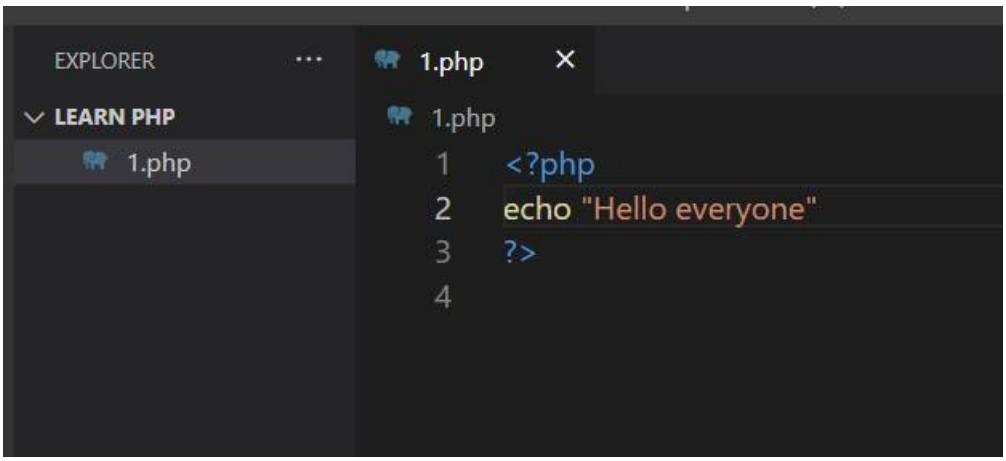


❼

❼

❼

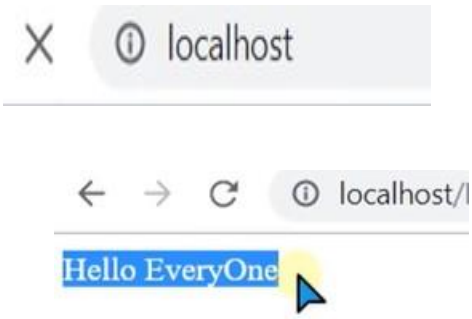
CODE :



Output :

Sabse phle SERVER Open karna hai

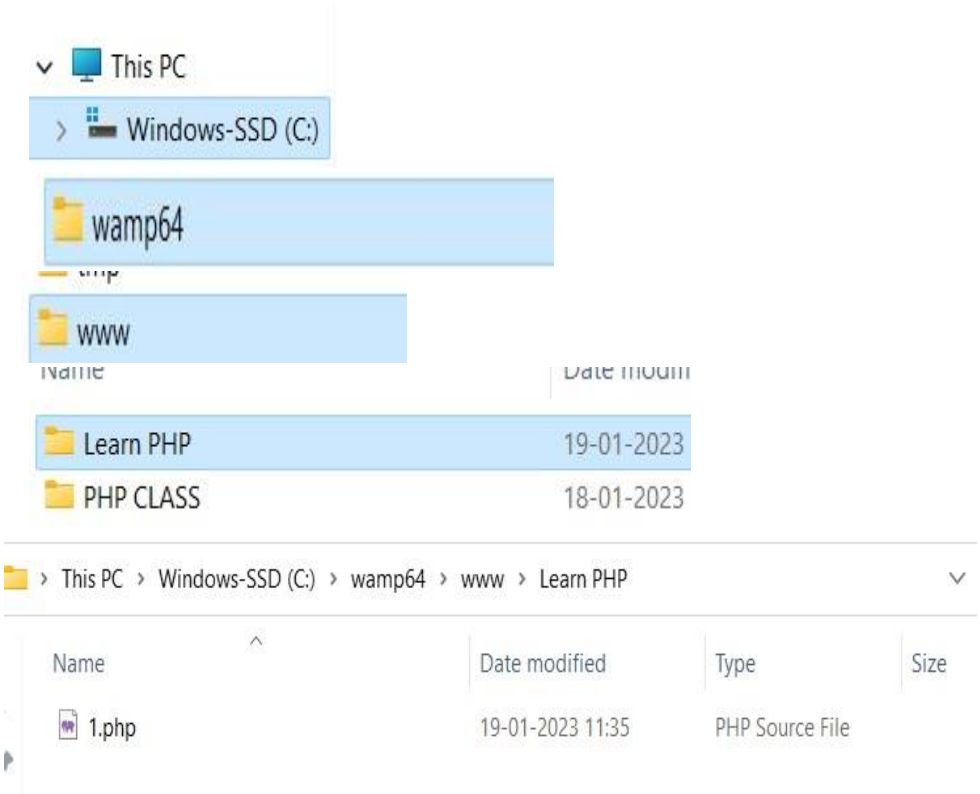
Opn Brwoser Localhost/ own folder name / file name



In WAMP Server :

C drive ❼ wamp folder ❼ www folder ❼own folder

1.)



❾

❾

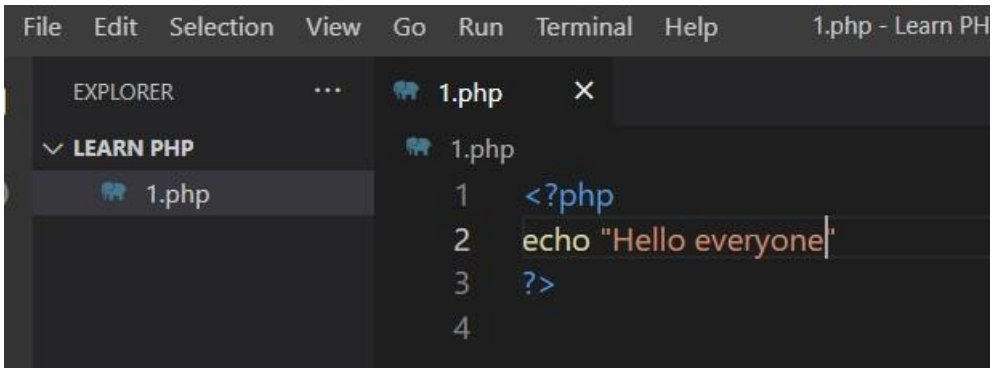
❾

❾

❾

Php ke code ko hum Keval HOST (localhost) par hi chala sakte hai

2.) CODE :

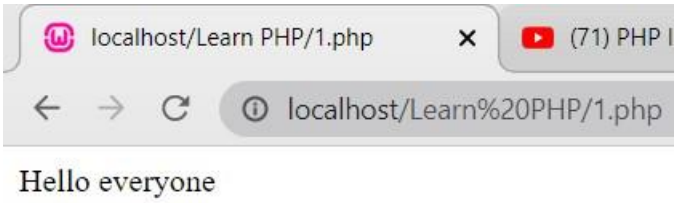


3.) OUTPUT :

Localhost/ folder name ❼ Browser

❾

❾



❾

_____ **PHP**

ECHO STATEMENT :

Print message or Variable value.

Example :

echo " learn php";

echo ' learn php'; echo "

Soni " , " Nishad "; echo "

Soni " . " Nishad " ;

echo " Soni Nishad "; ❼ php me html code

echo 502.25; echo " 502 "; ❼ String

. ❼ CONCATENATE (**join two or more text strings into one string**) .

Echo is faster than print.

Print Statement : print "

Soni Nishad "; print

502.25;

print " Soni " . " Nishad " ;

print "Soni " , "Nishad"; ❼ that is not run using print X

Syntax in Variable :

$x = " This is a variable" ;

echo $x ;

❾

PHP is a Case Sensitive language.

Ex : Test , test ---❼ both are not a **Same**

How to write a Variable Name :

Write Way	Wrong Way
$firstname	$first name
$_firstname	$99firstname
$first_name	$first%name
$var-name	
$firstName	$age $AGE ----|
$firstname99	**Not Same** ←|

Data Type :

Variable	Data Type
$x = " Hello World"; ----------❼	String

$x = 25; ----------❼ Integer

$x = 30.25; ----------❼ Float

$x = true; ----------❼ Boolean

$x = Array("HTML","CSS","JS"); ----❼ Array

$x = new MyClass(); ----------❼ Object

$x = null; ----------❼ Null

Comment :

1. *Single line comment .*

   // This is single line commnt .

   # This is single line commnt .

2. *Multiple line comment .*

   / * This is multiple line comment */

Constant Variable :

- Define (name , value , case-insensitive)

  Ex : define (num , 500 , true) ;
  define(_num , 500 , true);

  Can't use $ sign with constant variable name .
  Constant Variables are Global Variables .

Airthmetic Operators :

Operator	*Description*
+	Addition
-	Subtraction
*	Multiplication

**	Exponentiation
/	Division
%	Modulus (Remainder)
++	Increment Operator
--	Decrement Operator

Assignment Operators :

Operator	Example	Same As
=	x = y	x = y;
+=	x += y	x = x + y;
-=	x -= y	x = x – y;
*=	x *= y	x = x * y;
/=	x /= y	x = x / y ;
%=	x %= y	x = x % y ;
**=	x **= y	x = x ** y ;

Comparison Operators :-

Operator	Description	
<=>	Spaceship	$x <=> $y
It returns -1, 0 or 1 respectively less than, equal to, or greater than		

Operator	Description	
==	Equal to	$x == $y
===	Equal value and equal type	$x === $y
!=	Not equal	$x != $y
<>	Not equal	$x <> $y
!==	Not equal value or not equal type	$x !== $y
>	Greater than	$x > $y
<	Less than	$x < $y
>=	Greater than or equal to	$x >= $y
<=	Less than or equal to	$x <= $y

<u>If Statement</u> : [Comparision statement]

If ----------➐ Condition -----➐ False ---➐ Out

|

True

|

Statement

Syntax :

```
if( Condition true)
    {
        Statement;
    }
```

- If(Condition) :

endif;

Logical Operators :

If ----------➐ Condition1 -----➐ False ---➐ Out

Condition2 -----➐ False ---➐ Out

|

True

|

Statement

Types of Logical Operators

Operator	Name
&&	Logical AND
||	Logical OR
!	Logical NOT
and	Logical AND
or	Logical OR
xor	Exclusive OR

If Statement With Logical And :

If(Condition1 && Condition2)

-

{

Statement

}

- Run only when both conditions must be TRUE.

If Statement With Logical OR :

If(Condition1 || Condition2)

{

Statement

}

- Run only when either one conditions must be TRUE.

If Statement With Logical Not :

If !(Condition) {

| |

True False

| |

False True

}

If Statement With Exclusive OR :

- If(Condition1 xor Condition2) {

| |

| |

False False

| |

------ False-------

If(Condition1 xor Condition2) {

| |

| |
True False

| |

------- True -------

IF Else Statement :

If -------➐ Conditions ------➐ Flase ------➐ Statement 2

|
|
True
|
Statement

Syntax :

```
If (Condition True){
   Statement for True
    }
else
   {
    }
```

If Elseif Statement :

```
 ➲   If( Condition 1 )
     {
       Statement 1
      }
 elseif ( Condition 2 )
  {
```

-

Statement 2

}

else {

Default Statement

}

- **Type 2 :**

If(Conditio 1) :

elseif (Condition 2) :

else

:

endif ;

Switch Statement :

- switch (expression) { case condition 1 : Statement(s)

break;

case condition 2 : Statement(s)

break;

case condition 3 : Statement(s)

break; default : statement (s)

}

Type 2 : switch (expression) :

case condition 1 : Statement(s)

break;

case condition 2 : Statement(s)

break;

case condition 3 : Statement(s)

break;

default : statement (s)

endswitch;

Conditional Ternary Operator :

Conditions --------❼False ------❼ Statement 2

|

True

|

Statement

⇨ (Condition) ? True Statement : False Statement

String Operator :

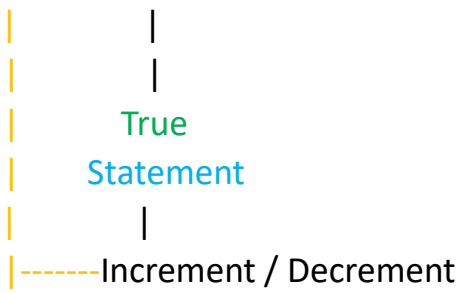
Loop :

Initialization

|

|----- Condition ------❼ false ------❼ Out of Loop

o



Types of Loop :

- While loop
- For loop
- Do-While loop
- For Each loop

Example :

```
$a = 1;  <----------- Initialization
While ( $a <= 10 ) {   <------- Condition
echo " Soni  Nishad ";
$a = $a + 1 ;  <---------- Increment
}
```

Math Function :

- abs — Absolute value
- acos — Arc cosine
- acosh — Inverse hyperbolic cosine
- asin — Arc sine
- asinh — Inverse hyperbolic sine
- atan2 — Arc tangent of two variables
- atan — Arc tangent
- atanh — Inverse hyperbolic tangent

- base_convert — Convert a number between arbitrary bases
- bindec — Binary to decimal
- ceil — Round fractions up
- cos — Cosine
- cosh — Hyperbolic cosine
- decbin — Decimal to binary
- dechex — Decimal to hexadecimal
- decoct — Decimal to octal
- deg2rad — Converts the number in degrees to the radian equivalent

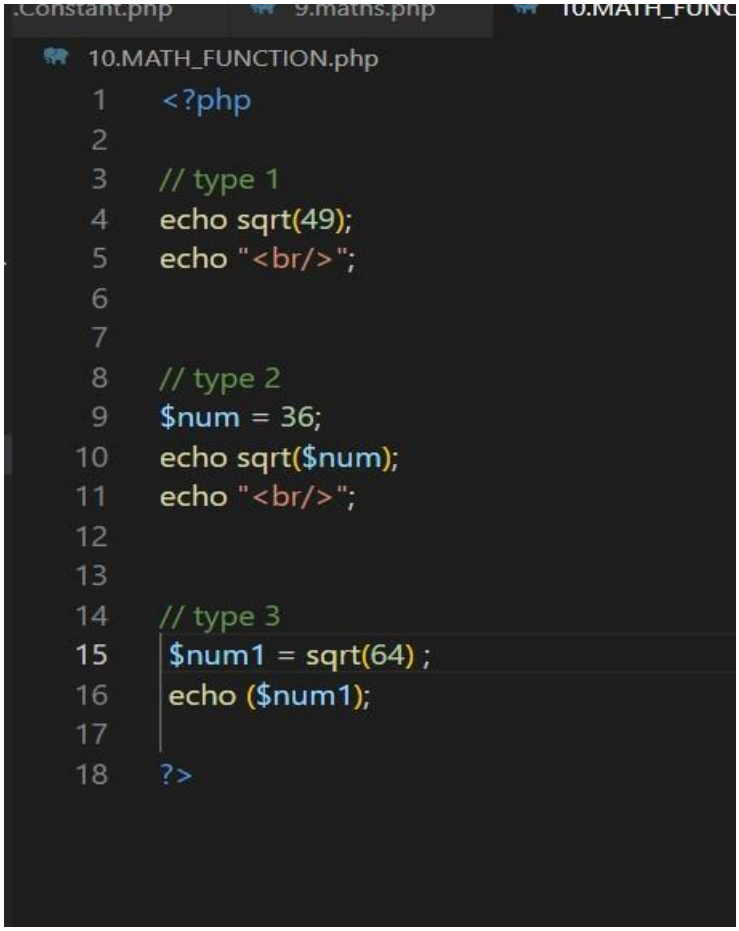
- exp — Calculates the exponent of e
- expm1 — Returns exp(number) - 1, computed in a way that is accurate even when the value of number is close to zero
- floor — Round fractions down
- fmod — Returns the floating point remainder (modulo) of the division of the arguments
- getrandmax — Show largest possible random value
- hexdec — Hexadecimal to decimal
- hypot — Calculate the length of the hypotenuse of a right-angle triangle
- intdiv — Integer division

- is_finite — Finds whether a value is a legal finite number
- is_infinite — Finds whether a value is infinite
- is_nan — Finds whether a value is not a number
- lcg_value — Combined linear congruential generator
- log10 — Base-10 logarithm
- log1p — Returns log(1 + number), computed in a way that is accurate even when the value of number is close to zero
- log — Natural logarithm

- max — Find highest value
- min — Find lowest value
- mt_getrandmax — Show largest possible random value
- mt_rand — Generate a random value via the Mersenne Twister Random Number Generator
- mt_srand — Seeds the Mersenne Twister Random Number Generator
- octdec — Octal to decimal
- pi — Get value of pi
- pow — Exponential expression

- rad2deg — Converts the radian number to the equivalent number in degrees
- rand — Generate a random integer
- round — Rounds a float
- sin — Sine
- sinh — Hyperbolic sine
- sqrt — Square root
- srand — Seed the random number generator
- tan — Tangent
- tanh — Hyperbolic tangent

FOR EXAMPLE :



Output :

Assignment Operator :

If-Statement :

If … Statement is used when a different sequence of instructions is to be excuted depending on logical value (True/Flase) of the condition evaluated.

Syntax :- If(Condition)

{

// Block of code ;

}

Type 2 :

If(Condition) : //

Block of code;

endif;

Nested IF Statement :

If(condition) :

// block of code;

If(condition) :

// Code 2;

endif;

If(condition) :

// Code 3;

endif; endif;

If-else Statement :

If …else statement I sused when a different sequence of instructions is to be excuted depending on the logical value (True/False) of the condition evaluated .

Syntax :

Type 1 :

```
if(condition)
{
   Statement_1;
}
else
{
    Statement_2;
```

```
}
    Statement_3;
```

Type 2 :-

```
If(condition) :
// Code 1;
endif; else :
    // else block;
endif;
```

Nested If-Else Statement :

In nested if-else statement , an entire if else construct is written within either the body of the if Statement or the body of an else Statement.

Syntax :

```
If(condition_1)
{
    If(condition_2 {
                    Statement_1_block;
                    }
else{
                Statement_2_block;
              }
    }
else
    {
     Statement_3_block;
      }
```

Statement_4_block;

Type 2 :

If(condition_1) :

If(condition_2):

Statement_1_block;

else :

Statement_2_block;

endif; else :

Statement_3_block;

endif;

Note :

STRING

strlen() ---> that is used for string lenght .

strrev () ---> string reverse str_word_count(); --> THat function is used to count the word.

strpos (<location> , “string”) ;

str_replace($var , “….” , “….”) **Relational**

Operator :

Single / Double / Triple :

Not Equal / Not Identical :

Spaceship :

Logical Operator :

Increment and Decrement :

String Operator :

Operator Precedence :

The Precedence of an Operator specifies how “tightly” to binds two expressions together .

Associativity	Operators
non-associative	clone new
left	[
right	**
right	++ -- ~ (int) (float) (string) (array) (object) (bool) @
non-associative	instanceof
right	!
left	* / %
left	+ - .
left	<< >>
non-associative	< <= > >=
non-associative	== != === !== <> <=>
left	&
left	^
left	|
left	&&
left	||
right	??
left	? :
right	= += -= *= **= /= .= %= &= |= ^= <<= >>=
left	and
left	xor
left	or

Else if Statement :

To show a multi-way decision based on several conditions, we use else if statement .

Syntax :

```
If(condition)
    {
Statement 1;
    }
   elseif(condition2)
```

```
        {
Statement 2 ;
        }
        elseif(condition n)
        {
Statement n;
        }
else
    Statement x;
```

Type 2 :

```
If(condition):
Statement 1;
elseif(condition2) :
Statement 2 ;
elseif(condition n)
: Statement n; else
: Statement x;
endif ;
```

Ternary Or Conditional Operator :

It works similar as if else statement .

Syntax :

```
Variable = Condition ? Expression1 : Expression2 ;
```

- If condition is true then return Expression1
- If condition false then return Expression2

Ex :-

```
$result = (5>1) ? “Greater” : “Less” ; echo
$result;
```

Switch Statement :

Check several possible constant values for an expression.

Syntax :---

```
Switch(expression)
  {
     Case expression 1:
Block of code;
break;
     Case expression 1:
          Block of code;
break;          .
          . default
:
    default block of instructions
  }
```

Type 2 :

```
 switch(expression) :
          Case expression :
Block of statement ;
          break;
          Case expression 1 :
Block of code 1 ;
          break;
.
          .
  default;
          default block of instructions
;    endswitch;
```

While loop :

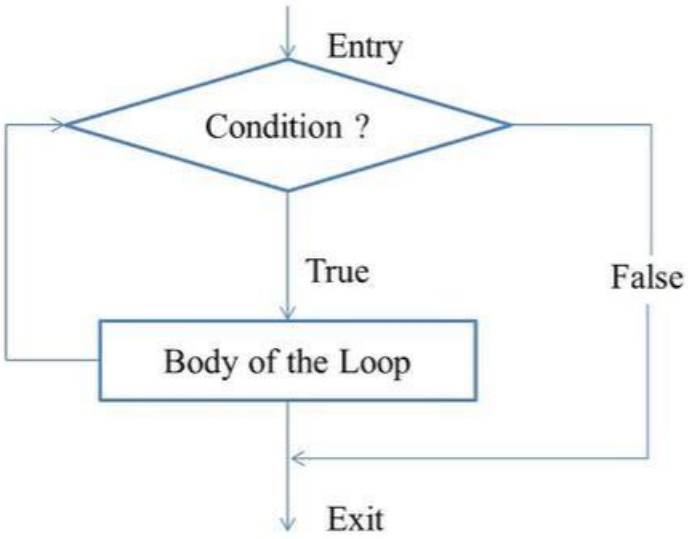
The while loop keeps repeating action until an associated condition returns false .

Syntax :

```
        Initialization;
while (condition) {
            Block of statement;
           Increment/decrement;
            }
```

Type 2 :

```
 Initialization;
while (condition) :
        block of statement ;
increment/decrement ;  endwhile;
```



Nested While loop :

```
while(condition) :
    while(condition) :
        statement(s) ;
```

```
endwhile;
statement;     endwhile;
```

Type 2 :

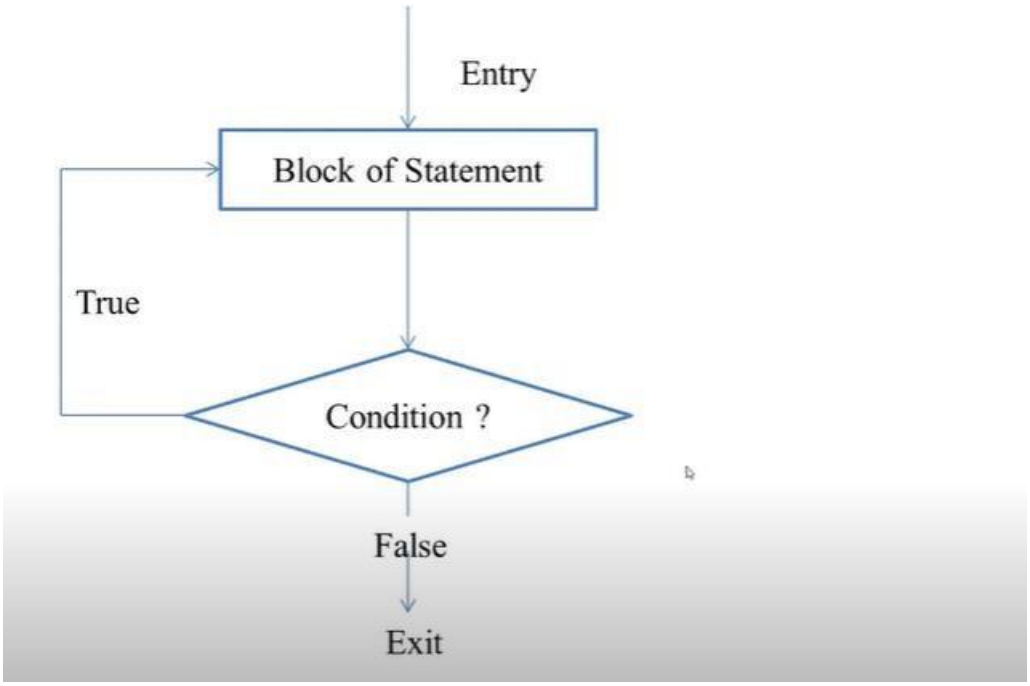
```
while(condition)
{
        while(condition)
            {
              statement(s) ;
            }
      statement(s);
}
```

Do-while Loop :

The do while loop is similar to while loop, but the condition is checked after the loop body is excuted. This ensure that the loop body is run at least once.

Syntax :

```
      initialization ;
do
       {
        block of statement;
        Increment/decrement;
       } while(condition);
```



For Loop :

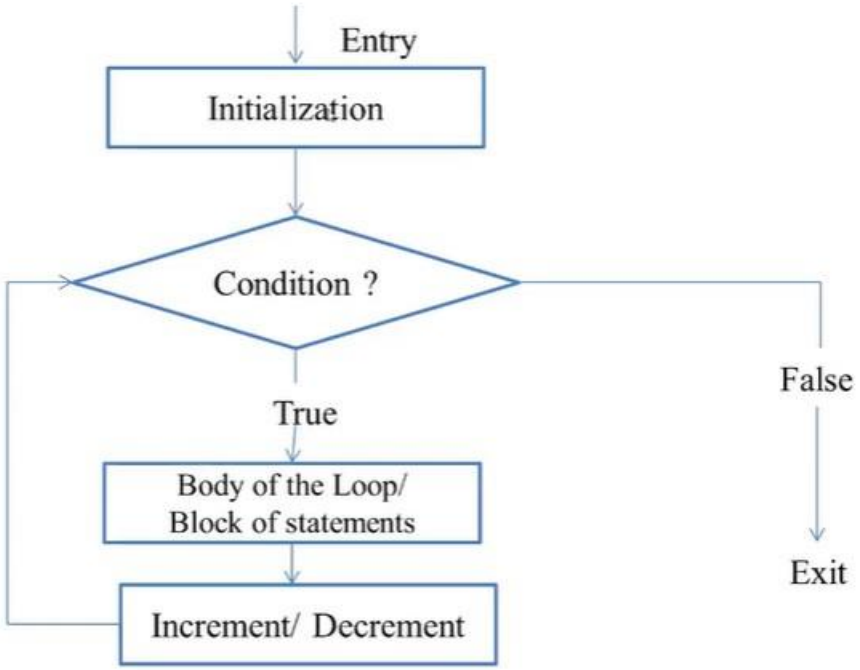
The for loop is frequently used , usually where the loop will be traversed a fixed number of times.

Syntax :

```
For(initialization; test condition; increment/decrement)
{
    Block of statement;
}
```

Type 2

```
For(initialization;test condition; increment/decrement):
Block of statement;
Endfor;
```



Nested For Loop :

The statement block of a loop lies completely inside the block of another loop.

Syntax :

```
For(initialization; test condition; increment/decrement)
{
      For(initialization; test condition; increment/decrement)
      {                          ----------|
       Block of statement;                 | ❼Inner loop
      }                          ----------|
 Block of statement;
}  ---------❼ Outer loop Break
```

Statement :

This statement is used to stop loop or switch statement at any time.

Continue Statement :

This statement is used to SKIP an iteration of loop.

Array

An Array is a data structure that stores one or more similar types of values in a single value.

Array in PHP is a type of data structure that allows us to store multiple elements of similar data type under a single variable thereby saving us the effort of creating a different variable thereby saving us the effort of creating a different variable for every data.
The arrays are helpful to create a list of elements of similar types, which can be accessed using their index or key.
An array is created using an array () function in PHP.

There are basically three types of Arrays in PHP –

1. **Indexed Array**
2. **Associative Array**
3. **Multidimensional Array**

1. **Numeric/indexed Array –** In this array index will be represented by a number.

   By default numeric array index start from 0.

   Ex: $num [0] = “Soni”;

```
n_default_parameter.php   23. function_returning_value.php
24. Array.php
1   <?php
2
3   $names = array("vipul" , "rohit", "musu");
4   // echo $names;
5
6   print_r ($names);
7   ?>
```

Output –

```
Array ( [0] => vipul [1] => rohit [2] => musu )
```

❾

```
<?php

$names = array("vipul" , "rohit", "musu");
// echo $names;

echo "<pre>";
print_r ($names);
?>
```

Output –

```
Array
(
    [0] => vipul
    [1] => rohit
    [2] => musu
)
```

2. **Associative Array –** In this Array index/key will be represented by a string.
   Ex **:** $fee[“soni”] = 500;

3. **Multidimensional Array** – Array of Arrays is known as multidimensional arrays.

Syntax:

$array_name [0] = value;

EX:-- $name [0] = " news24 " ;

$name [0] = 25;

Type 2

$array_name [ ] = value;

EX:-- $name [ ] = " news24";

$name[ ] = 25;

Note: By default, array starts with index 0.

Declaration and initialization of Array

$name[0] = "Rahul" ; $name[ ] = "Rahul";

$name[1] = "Sonam"; $name[ ] = "Sonam";

$name[2] = "Sumit"; $name[ ] = "Sumit";

Array Function:

Array () function is used to create array.

Syntax :-

$array_name = array("Value1" , "Value2" , "Value3" ,.....);

Ex:-

$name = array("Rahul" , "Sonam" , "Sumit");

Operator (=>) :-

The => Operator lets you create key/index – value pairs in arras . The item on the left of the => is the key/index and the item on the right is the value .

$name = array (1 => "Rahul" , "Sonam" , "Sumit");

Associative Array

Syntax:-

$array_name ["Key"] = value;

EX:

$fees["Rahul"] = 500;

Array Function

Syntax:-

$array_name = array("key1"=> Value1,"key2"=> Value2 ,……);

Modifing and Deleting:

Deleting:

***unset()** function is used to delete an Array element .*

Syntax:

unset($array_name[]);

$name = " " ; -----❼ Space khali kr deta hai .

Example:

unset($name[2]);

Copy An Array in Array

We can copy entire array using assignment operator.

$name[0] = "Soni";

$name[1] = “Sonam”;

$name[2] = “Soumya”;

$name[3] = “Anchal”;

$student = $name; echo

$student[3];

Count () Function

The **count () function** returns the number of elements in array.

Syntax:- **count(array , mode).**

Array ❼ Specifies the array.

Mode ❼ Specifies the mode. Possible values.

0 – Default. Does not count all elements of multidimensional arrays.

1 – Counts the array recursively (counts all the elements of multidimensional arrays).

Array_keys() Function :-

The array_keys() function returns an array containing the keys .

Array – Specifies an array *

Value – You can specify a value, then only the keys with this value are returned.

Strict - Used with the value parameter. Possible values:

True – Returns the keys with the specifies value, depending on type : the number 8 is not the same as the string “8” .

False – Default value. Not depending on type,

the number 8 is the same as string “8”.

Foreach loop with Array

The foreach loop works only on arrays, and is used to loop through each key.value pair in an array.

PHP foreach loop is mainly used for looping through the values of an array.
It loops over the array, and each value for the current array element is assigned to $value, and the array pointer is advanced by one to go the next element in the array.
Loops through a block of code for each element in an array.

Syntax :-

- foreach($array_name as $value)
  {
  Block of Statement;
  }

- foreach($array_name as $value) : Block of Statement;
  endforeach;

- foreach ($array_name as $key => $value)
  {
  Block of Statement;
  }

- foreach($array_name as $key => $value) : Block of statement; endforeach;

  Where , $value is array's value.
  $key is array's key/index .

```php
<?php

$students = array(
    'bittu' , 'soni' , 'muskan'
);

foreach($students as $names)
{
    echo $names;
}

?>
```

Output –

```
bittusonimuskan
```

1.) foreach($array_name as $value)

{

Block of Statement;

}

The ***first form loops*** over the array given by *array_expression*. On each iteration, the value of the current element is assigned to **$value** and the internal array pointer is advanced by one (so on the next iteration, you'll be looking at the next element) .

1. foreach ($array_name as $key => $value)

{

Block of Statement;

}

The ***second form*** will additionally assign the current element's key to the **$key** variable on each iteration.

Print_r () Function :- This function is used to display information in a way that's readable by humans .

Syntax:

Print_r($array_name , Bool_Return) ;

When Return can be TRUE or FALSE .

By default there is False.

When Return is set to TRUE , Print_r() will return the information rather than print it.

EXAMPLE:-

$name[0] = “Rahul” ;

$name[1] = “Soni” ;

Print_r($name);

$results = print_r ($name, true) ;

Echo $results;

sort and reverse the data –

To sort the data in an array in PHP, we have PHP **sort()** Function and to reverse the array data in PHP we have PHP **rsort()** Function.

The **sort()** function sorts an indexed array in ascending order.

Tip: Use the **rsort()** function to **sort** an indexed array in descending order.

For example –

SORT() –

```php
<?php

$students = array('soni', 'priya' , 'ritesh');
sort ($students);

foreach($students as $names)
{
   echo "<li>" . $names ."<br/>";
}
echo "</ol>";

?>
```

RSORT () –

```php
<?php

$students = array('soni', 'priya' , 'ritesh');
rsort ($students);

foreach($students as $names)
{
   echo "<li>" . $names ."<br/>";
}
echo "</ol>";

?>
```

Arrays push() pop() shift() and unshift() function explained in PHP – What are Array Function in PHP – **array_pop($arr)** : This function removes an element from the end of an array.

array_push($arr, $val): This function adds an element to the end of an array.

array_shift($arr): This function removes an element from the beginning of an array.

array_unshift($arr, $val): This function adds an element to the beginning of an array.

Array_pop($arr) –

```php
<?php

$student = array('mahek' , 'neha' , 'neelam' , 'janvi');

echo "<pre>";
print_r($student);


// After using pop
array_pop($student);

echo "<pre>";
print_r($student);



?>
```

Output –

```
Array
(
    [0] => mahek
    [1] => neha
    [2] => neelam
    [3] => janvi
)

Array
(
    [0] => mahek
    [1] => neha
    [2] => neelam
)
```

Array_pust($arr) –

```php
<?php

$student = array('mahek' , 'neha' , 'neelam' , 'janvi');

echo "<pre>";
print_r($student);

// After using push
array_push($student, 'priya' );

echo "<pre>";
print_r($student);
?>
```

Output –

```
Array
(
    [0] => mahek
    [1] => neha
    [2] => neelam
    [3] => janvi
)

Array
(
    [0] => mahek
    [1] => neha
    [2] => neelam
    [3] => janvi
    [4] => priya
)
```

array_shift(arr)–

```php
<?php

$student = array('mahek' , 'neha' , 'neelam' , 'janvi');

echo "<pre>";
print_r($student);

// array_shift(arr)
array_shift($student);

echo "<pre>";
print_r($student);

?>
```

Output –

```
Array
(
    [0] => mahek
    [1] => neha
    [2] => neelam
    [3] => janvi
)

Array
(
    [0] => neha
    [1] => neelam
    [2] => janvi
)
```

Array_unshift($arr) –

```php
<?php

$student = array('mahek' , 'neha' , 'neelam' , 'janvi');

echo "<pre>";
print_r($student);



// array_shift(arr)
array_unshift($student, 'nikki');

echo "<pre>";
print_r($student);

?>
```

Output –

```
Array
(
    [0] => mahek
    [1] => neha
    [2] => neelam
    [3] => janvi
)

Array
(
    [0] => nikki
    [1] => mahek
    [2] => neha
    [3] => neelam
    [4] => janvi
)
```

What is PHP implode and explode function – implode and explode function in PHP.

Difference between implode and explode in PHP

PHP **Implode()** Function The implode function in PHP is used to "join elements of an array with a string".

The **implode()** function returns a string from elements of an array.

It takes an array of strings and joins them together into one string using a delimiter (string to be used between the pieces) of your choice.

The implode function in PHP is easily remembered as "array to string", which simply means that it takes an array and returns a string.

It rejoins any array elements and returns the resulting string, which may be put in a variable.

```
<?php

$colors = array('red' , 'green' , 'blue');

print_r($colors);

$res = implode (" ", $colors) ;
echo $res . "";

?>
```

Output –

```
Array ( [0] => red [1] => green [2] => blue ) red green blue
```

PHP **Explode()** Function The explode function is used to "Split a string by a specified string into pieces

i.e. it breaks a string into an array".

The explode function in PHP allows us to break a string into smaller text with each break occurring at the same symbol.

This symbol is known as the delimiter. Using the explode command we will create an array from a string.

The explode() function breaks a string into an array, but the implode function returns a string from the elements of an array.

```php
<?php

$biodata = "my name is soni";

$result = explode(" ", $biodata);

echo "<pre>";
print_r ($result);

foreach($result as $val )
{
   echo $val;
}

?>
```

Output –

```
Array
(
    [0] => my
    [1] => name
    [2] => is
    [3] => soni
)
mynameissoni
```

Array Operators

Example	Name	Result
$a + $b	Union	Union of $a and $b
$a == $b	Equality	TRUE if $a and $b have the same key/value pairs.
$a === $b	Identity	TRUE if $a and $b have the same key/value pairs in the same order and of the same types.
$a != $b	Inequality	TRUE if $a is not equal to $b.
$a <> $b	Inequality	TRUE if $a is not equal to $b.
$a !== $b	Non-identity	TRUE if $a is not identical to $b.

$a + $b – Union :-

The **+ Operator** returns the right-hand array appended to the left-hand array ; **for keys that exist in both arrays,** the elements from the left-hand array will be used, and the matching elements from the ***right-hand*** array will be ignored.

MultiDimensional Array

Multidimensional array is Arrays of Arrays .

Multidimensional array can be 2D, 3D, 4D etc.

Ex :-

2D - $name[ ][ ]

3D - $name[ ][ ][ ]

Rahul	Dell	10
Sonam	Hp	20
Sumit	Zed	30

[0][0] Rahul	[0][1] Dell	[0][2] 10
[1][0] Sonam	[1][1] Hp	[1][2] 20
[2[0] Sumit	[2][1] Zed	[2][2] 30

STRING

String is group of Character.

Ex:

"welcome";

"New learn";

$name = "soni";

A string literal can be specified in four different ways:

- **Single quoted**

Ex : 'Soni Nishad'

- **Double quoted**

Ex : " new learn "

- **Heredoc**
- **nowdoc**

Access Character:

Single Quoted vs Double Quoted:

1.) ***Single quote*** **is said to be literal. It doesn't parse the data.**

   Double quote **is said to be interpreted.**

2. **)** We can't use ***single quote*** within single quote.

   We can't use **do*uble quote*** within double quote.
3.)We can use ***double quote*** within single quote. We can use ***single quote*** within double quote.
4. **)** Use **escape ** to use ***double*** quote within *double quote.* **Use escape ** to use ***single*** quote within *single quote.*
5. **) \' and \\ only these two works with single quote.**
   All escape sequence works with double quotes.

Escape Sequence :

Function –

What is Function in PHP?

PHP User-Defined Functions

Besides the built-in PHP functions, it is possible to create your own functions.

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

A PHP function will not execute automatically when a page loads.

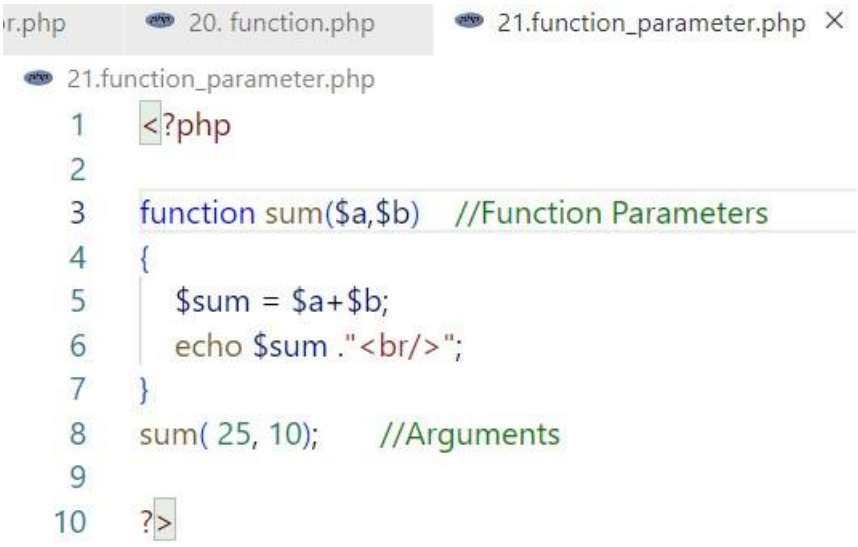
A PHP function will be executed by a call to the function.

Function names follow the same rules as other labels in PHP. A valid function name starts with a letter or underscore, followed by any number of letters, numbers, or underscores.

Syntax :

Function functionName(paramameter)

❾ Difference between function arguments in PHP vs function parameters.



❾ Function Default Arguments ❼ Function Default Parameters –

What is Default Function Parameters in PHP?

When creating functions in PHP it is possible to provide default parameters so that when a parameter is not passed to the function it is still available within the function with a pre-defined value.
These default values can also be called optional parameters because they don't need to be passed to the function.

For example –

```php
<?php

function mult( $a, $b=5 )  // default parameter --> 5 here
{
   $mult = $a*$b;
   echo $mult . "<br/>";
}
mult(10,4);
mult(3, 5);
// mult( 4);    By default you are not passed the values then we use default parameters.


?>
```

❾ PHP Functions Returning Value Explained –

A function can return a value using the return statement in conjunction with a value or object. return stops the execution of the function and sends the value back to the calling code.

You can return more than one value from a function using a return array(1,2,3,4).

For example –

default_parameter.php 23. function_returning_value.php

23. function_returning_value.php

```php
-->


<?php

function mult($a, $b)
{
    $multresult = $a+$b;
    return $multresult;    //return
}

$output = mult(10,4);

$output2 = mult(14, 8);


echo "multiplication " .$output;
echo "<br/>";
echo "multiplication " .$output;
echo "<br/>";
echo "multiplication " .$output;
echo "<br/>";
echo "multiplication " .$output;
echo "<br/>";
echo "multiplication " .$output2;


?>
```

Date & Time Function

https://www.thapatechnical.com/2020/03/php-date-function-explained.html?m=1

❼ The date() function formats a local date and time, and returns the formatted date string.

Syntax – `date(`*`format, timestamp`*`)`

❾

if(isset($_POSTA['submit']))

{

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

}

1. How to Create Connection with Database in MYSQL

Sol ❼

```php
<?php


$username = "roots";
$password = "";
$server = 'localhost';
$db = 'crudyoutube';


$con = mysqli_connect($server,$username,$password,$db);

if($con){
    echo "Connection Successful";
}else{
    // echo "NO connection";
    die("no conencetion" . mysqli_connect_error());
}


?>
```

Example ❼

jobregister.php

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <title>JOB Registration</title>
    <!-- -->
    <?php include 'link.php'; ?>
</head>

<body> ···
</body>

</html>

<!-- connection include -->
<?php  include 'connection.php'; ?>
```

```php
<!-- connection include -->
<?php

    $con=mysqli_connect("localhost","root","","crudpratice");

if($con)
{
?>

<script>
alert("connection successful");
</script>

<?php
}
else
{
?>
<script>
alert("connection not ");
</script>

<?php
}
?>
```

2. How to insert data in database

jobregister.php

```
<!DOCTYPE html>
<html lang="en">
<head> ···
</head>
<body> ···
</body>
</html>
<!-- connection include -->
<?php
  include 'connection.php';
 if(isset($_POST['submit'])) {
    $name = $_POST['name'];
    $degree = $_POST['degree'];
    $mobile = $_POST['mobile'];
    $email = $_POST['email'];
    $refer = $_POST['refer'];
    $jobprofile = $_POST['jobpost'];

    $query = "insert into jobregistration(name, degree, mobile, email, refer, jobpost)
    values( '$name', '$degree',  '$mobile', '$email',  '$refer' , '$jobprofile' )" ;
   $res=mysqli_query($con,$query);
if($res){
   // echo 'data inserted';
   ?>

<script>
   alert ('data inserted properly');
</script>

   <?php
   }else{
   ?>
   <!-- echo 'not inserted'; -->
<script>
   alert ('data not inserted');
</script>
<?php
} }
?>
```

jobregister.php

```
48  <!-- connection include -->
49  <?php
50    include 'connection.php';
51   if(isset($_POST['submit'])) {
52      $name = $_POST['name'];
53      $degree = $_POST['degree'];
54      $mobile = $_POST['mobile'];
55      $email = $_POST['email'];
56      $refer = $_POST['refer'];
57      $jobprofile = $_POST['jobpost'];
58
59      $query = "insert into jobregistration(name, degree, mobile, email, refer, jobpost)
60      values( '$name', '$degree',  '$mobile', '$email',  '$refer' , '$jobprofile' )" ;
61     $res=mysqli_query($con,$query);
62  if($res){
63     // echo 'data inserted';
64     ?>
65
66  <script>
67     alert ('data inserted properly');
68  </script>
69
70     <?php
71     }else{
72     ?>
73     <!-- echo 'not inserted'; -->
74  <script>
75     alert ('data not inserted');
76  </script>
77  <?php
78  } }
79  ?>
```

❾

Type 2 –

```php
<!-- connection include -->
<?php
// include 'connection.php';
if(isset($_POST['submit']))
{
    $name = $_POST['name'];
    $degree = $_POST['degree'];
    $mobile = $_POST['mobile'];
    $email = $_POST['email'];
    $refer = $_POST['refer'];
    $jobprofile = $_POST['jobpost'];

    $con=mysqli_connect("localhost","root","","crudpratice");

    $query = "insert into jobregistration(name, degree, mobile, email, refer, jobpost)
    values( '$name', '$degree',  '$mobile', '$email',  '$refer' , '$jobprofile' )" ;
   $res=mysqli_query($con,$query);

if($res)
{
   // echo 'data inserted';
    ?>

<script>
   alert ('data inserted properly');
</script>

    <?php
   } else {
    ?>
    <!-- echo 'not inserted'; -->
<script>
   alert ('data not inserted');
</script>

    <?php
}
}
?>
```

3. How to Select Data from Database in PHP & Display

Solution –

display.php

```php
<?php
include 'connection.php';
$query = "select * from jobregistration";
$res = mysqli_query($con,$query);

$nums = mysqli_num_rows($res);
// echo $nums;

//  fetch/retrive the data
$value = mysqli_fetch_array($res);
echo $value[1];

?>
```

❾ Type 2

display.php

```php
</head>
<body>
  <div class="main-div">
    <h1>List of candidates for web developer job</h1>
    <div class="center-div">
      <div class="table-responsive">
        <table>
          <thead>
            <tr>
              <th>id</th>
              <th>name</th>
              <th>degree</th>
              <th>mobile</th>
              <th>email</th>
              <th>refer</th>
              <th>post</th>
              <th>Operation</th>
            </tr>
          </thead>
        <?php
          include 'connection.php';
          $query = "select * from jobregistration";
          $res = mysqli_query($con,$query);
          $nums = mysqli_num_rows($res);
          while($value = mysqli_fetch_array($res))
          {
          ?>
          <tbody>
            <tr>
              <td><?php echo $value['id'] ?></td>
              <td><?php echo $value['name'] ?></td>
              <td><?php echo $value['degree'] ?></td>
              <td><?php echo $value['mobile'] ?></td>
              <td><span class="email-style"><?php echo $value['email'] ?></span></td>
              <td><?php echo $value['refer'] ?></td>
              <td><?php echo $value['jobpost'] ?></td>
              <td>
                <a href="update.php?id=<?php echo $value['id'] ?>" title="update">Update</a>
              </td>
              <td>
              <a href="delete.php?id=<?php echo $value['id'] ?>" title="delete">Delete</a>
              </td>
            </tr>
        <?php
          }
          ?>
        </tbody>
```

4. How to Update Data from Database in PHP & Display

Solution –

```php
<!-- connection include -->
<?php
include 'connection.php';
$ids = $_GET['id'];
$showquery = "select *from jobregistration where id={$ids}";
$showdata = mysqli_query($con,$showquery);
$arrdata = mysqli_fetch_array($showdata);
if(isset($_POST['submit']))
{
   $idupdate = $_GET['id'];
$name = $_POST['name'];
$degree = $_POST['degree'];
$mobile = $_POST['mobile'];
$email = $_POST['email'];
$refer = $_POST['refer'];
$jobprofile = $_POST['jobpost'];
// $query = "insert into jobregistration(name, degree, mobile, email, refer, jobpost)
// values( '$name', '$degree',  '$mobile', '$email',  '$refer' , '$jobprofile' )" ;
$upquery = "update jobregistration set id=$idupdate, name= '$name' , degree =  '$degree',  mobile = '$mobile',
email = '$email', refer = '$refer' , jobpost = '$jobprofile' where id = $idupdate ";
$res=mysqli_query($con,$upquery);
if($res){
// echo 'data inserted';
?>
<script>
alert ('data inserted properly');
</script>

<?php
}else{
?>
<!-- echo 'not inserted'; -->
<script>
alert ('data not inserted');
</script>
<?php
} }
?>
```

Type 2 [Full code] –

ate.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Update</title>
    <!--  -->
    <?php include 'link.php'; ?>
     include 'connection.php';

</head>

<body>
    <center>
        <div class="outer">
            <div class="inner">
                <div class="col-sm-6">
                </div>
                                    <!-- connection include -->
                                    <?php
                                    include 'connection.php';
                                    $ids = $_GET['id'];
                                    $showquery = "select *from jobregistration where id={$ids}";
                                    $showdata = mysqli_query($con,$showquery);
                                    $arrdata = mysqli_fetch_array($showdata);
                                    if(isset($_POST['submit']))
                                    {
                                       $idupdate = $_GET['id'];
                                    $name = $_POST['name'];
                                    $degree = $_POST['degree'];
                                    $mobile = $_POST['mobile'];
                                    $email = $_POST['email'];
                                    $refer = $_POST['refer'];
                                    $jobprofile = $_POST['jobpost'];

                                    // $query = "insert into jobregistration(name, degree, mobile, email, refer, jobpost)
                                    // values( '$name', '$degree',  '$mobile', '$email',  '$refer' , '$jobprofile' )" ;

                                    $upquery = "update jobregistration set id=$idupdate, name= '$name' , degree =  '$degree',  mobile = '$mobile',
                                    email = '$email', refer = '$refer' , jobpost = '$jobprofile' where id = $idupdate ";
```

```
                $upquery = "update jobregistration set id=$idupdate, name= '$name' , degree =  '$degree',  mobile = '$mobile',
                email = '$email', refer = '$refer' , jobpost = '$jobprofile' where id = $idupdate ";
                $res=mysqli_query($con,$upquery);
                if($res){
                // echo 'data inserted';
                ?>
                <script>
                alert ('data inserted properly');
                </script>

                <?php
                }else{
                ?>
                <!-- echo 'not inserted'; -->

                <script>
                alert ('data not inserted');
                </script>

                <?php
                } }
                ?>

        <div class="col-sm-6">
          <form action="" method="post">
            <h2 class="mt-4">Job Registration</h2>
            <input type="text" name="name" placeholder="Enter your name" value="<?php echo $arrdata['name'] ?>" class="form-control">
            <br />
            <input type="text" name="degree" placeholder="Enter your Qualification"  value="<?php echo $arrdata['degree'] ?>" class="form-control">
            <br />
            <input type="number" name="mobile" placeholder="Enter your mobile number"  value="<?php echo $arrdata['mobile'] ?>" class="form-control">
            <br />
            <input type="email" name="email" placeholder="Enter your email"   value="<?php echo $arrdata['email'] ?>"  class="form-control">
            <br />
            <input type="text" name="refer" placeholder="Any Refrences"   value="<?php echo $arrdata['refer'] ?>" class="form-control">
            <br />
            <input type="text" name="jobpost" placeholder="Enter jobpost"   value="<?php echo $arrdata['jobpost'] ?>"   class="form-control">
            <br />
            <input type="submit" value="Submit" name="submit">
            <a href="display.php" class="text-dark">Check form</a>
          </form>
        </div>
      </div>
    </div>
  </center>
</body>
</html>
```

5. How to Delete Data from Database in PHP & Display

Solution –

```
    <td>
      <a href="update.php?id=<?php echo $value['id'] ?>" title="update">Update</a>
    </td>
    <td>
    <a href="delete.php?id=<?php echo $value['id'] ?>" title="delete">Delete</a>
    </td>
</tr>
```

delete.php

```
<?php
  include 'connection.php';

  $id=$_GET['id'];

  $query = " delete from jobregistration where id=$id ";
  $res =mysqli_query($con, $query);

?>
```

Type 2 –

delete.php

```
<?php
  include 'connection.php';

  $id=$_GET['id'];

  $query = " delete from jobregistration where id=$id ";
  $res =mysqli_query($con, $query);

  if($query)
  {
   ?>

   <script>
     alert("Deleted successfully");
   </script>

   <?php
  }
  else
  {
   ?>

<script>
     alert("Not Deleted");
   </script>

   <?php
  }
//   header('location:display.php');
?>
```

2-feb-2023

Pattern Question :

1.
```
    *
   * *
  * * *
 * * * *
* * * * *
```
Sol :

```
for($i=1; $i<=5; $i++)
{
   for($j=1;$j<=$i; $j++)
    {
       echo "*  " ;
    }
       echo "<br/>";
}
```

ASCII values :

A-Z ----------> 65 to 90
A = 65
Z = 90 a-z ----------> 97
to 122 a = 97 z = 122
0-9 ------> 48 to 57
0 = 48
9 = 57

PHP

Programming Language Vs Scripting Language

Programming language	Scripting language
Has all the features needed to develop complete applications.	Mostly used for routine tasks
The code has to be compiled before it can be executed	The code is usually executed without compiling
Does not need to be embedded into other languages	Is usually embedded into other software environments.

Heavy GUI programming is usually done using C++. Photoshop is written completely in C++. It has an MDI or multiple document interface where multiple windows reside within a single parent window.
The first release of Autocad was written in C. Later features written in C++ were added.
Oracle RDBMS is an object-relational database management system written in assembly language, C and C++.

What Can PHP Do?

PHP can generate dynamic page content

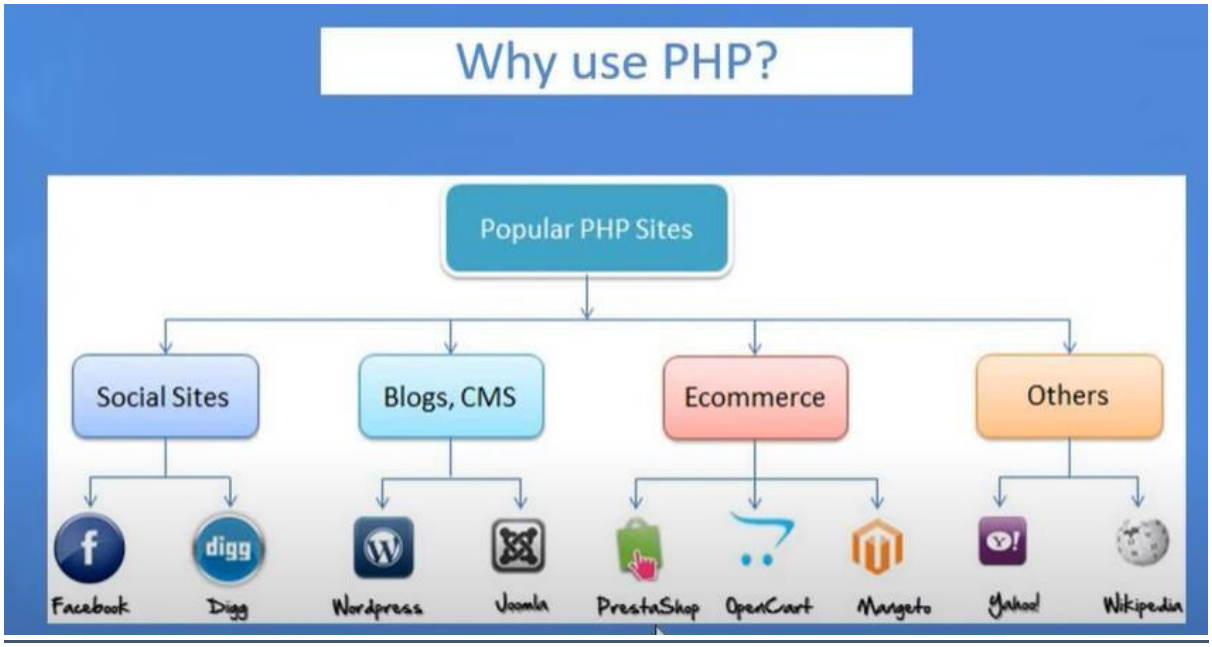
PHP can create, open, read, write, delete, and close files on the server

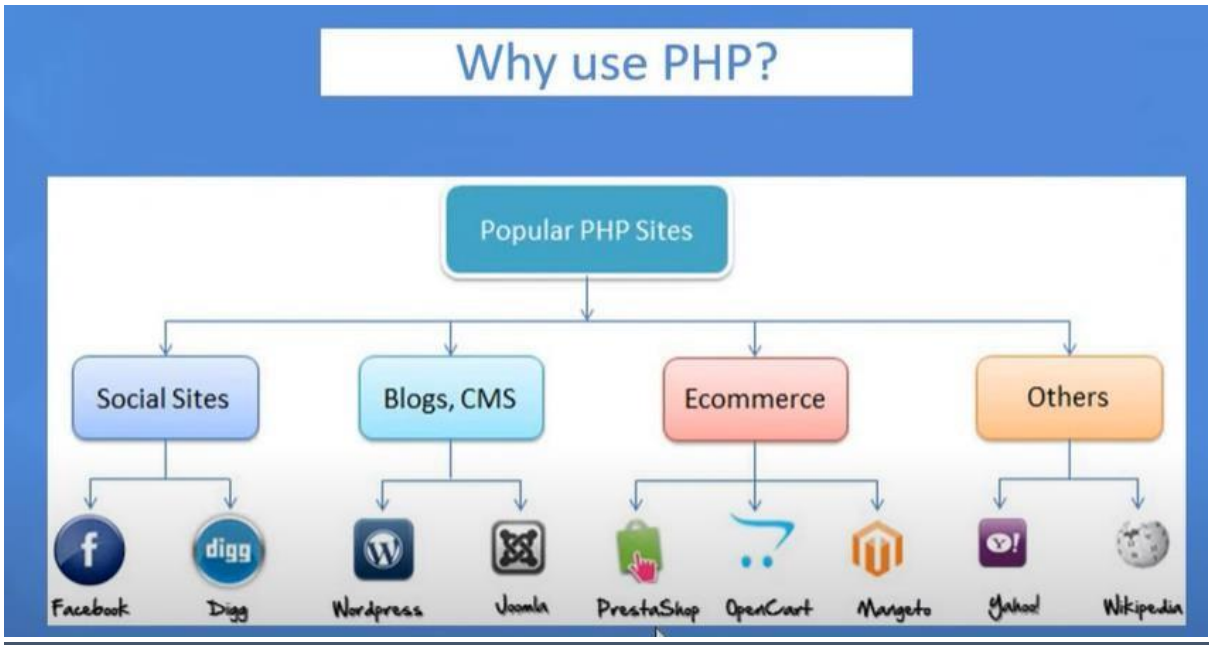
PHP can collect form data

PHP can send and receive cookies

PHP can add, delete, modify data in your database

PHP can be used to control user-access





Database Connectivity

2. **Specify the Sql statement :-**

$query"insert into empinfo values(1001,'soni',8000)";

Now excute sql statement

$res=mysqli_query($con,$query);

3. **Close Connection :-** $con->close();

```
                        <!-- query string ek aisi
string hoti hai jisme hum URL ke duvara ek page se dusre
page par bhejte hai . -->
```

10-02-2023

Two step registration :

Interview Question –

_____

1. What is PHP? Explain its features.
2. What is the difference between java and PHP?
3. What is the difference between print and echo?
4. What is an array in PHP? Explain its types.
5. What is a function? What are the advantages of function?
6. What are class and object?
7. What is abstraction?

8. What is encapsulation?
9. What is Inheritance?
10. What is Polymorphism?
11. What is a constructor? How to create a constructor in PHP?
12. What are exceptions? Explain exception handling.
13. How to create an Object in PHP?
14. What is the work of this keyword?
15. What is a super keyword?
16. How to connect MySQL database with PHP code?
17. What is the difference between mysqli_fetch_array() and mysqli_fetch_assoc() ?
18. What is session?
19. What is difference between get and post?
20. What is framework?
21. How php interact with html?

Task

_____

1.) WAP to create an Array AR with size 10 and store 10 numbers in array. Now copy even numbers in array EAR and odd numbers in array OAR. Display numbers of EAR and OAR.

2.) WAP to create an array with size 10 and store 10 numbers in array. Now display maximum and minimum value of array.

3.) WAP to create an array with size 5 and store 5 names in array. Now display names in ascending and descending order.

Why PHP is called loosely typed language?

A loosely typed language is one that doesn't need you to define the type of variable you are declaring.

That is why PHP is called loosely typed language.

So basically you have to just declare a variable and store any type of value (String, integer, floating, etc) you want. There is no need to externally specify the datatype.

what is an expression in PHP?

Difference between Operand and Operator in PHP explained.

An expression is a bit of PHP that can be evaluated to produce a value.

The simplest expressions are literal values and variables.

A literal value evaluates to itself, while a variable evaluates to the value stored in the variable. More complex expressions can be formed using simple expressions and operators.

An operator takes some values (the operands) and does something (for instance, adds them together).

Operators are written as punctuation symbols—for instance, the + and - familiar to us from math. Some operators modify their operands, while most do not.

Project

Modules of Hospital

- Home page
- About us
- Appointments
- Facilities
- Contact us
- Registration • Login

❾ User

- View their appointments
- Book a service
- Search for doctor
- Take appointment

- Profile

❾ Doctor

- Profile
- View their appointments
- Cancel appointment
- Requirement form

Programs

1.)

```php
<?php
//WAP to check given Number is even or Odd.

$n=10;
if($n%2==0)
{
    echo "Number is even";
}
else
{
    echo "Number is odd";
}

?>
```

2.)

```php
<?php
// Using Ternary Operator ,
// WAP a progrma to check given number is Even or Odd using Ternary Operator.
// (expression1) ? (expression2) : (expression3);

$n=11;
echo ($n%2==0) ? "Number is even" : "Number is Odd";

?>
```

3.)

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Check Even Odd</title>
</head>
<body>
    <h2>Check Even or odd</h2>

    <!--
        isset() is a function in php , which check that variable is
            initialized or not .
    -->
    <?php

if(isset($_POST["submit"]))
    {
        $n=$_POST["n"];
        if($n%2==0)
        {
            echo "Number is Even";
        }
        else
        {
            echo "Number is odd";
        }
    }

     ?>


    <form action="" method="post">
            Enter a number :
            <input type="number" name="n" required/><br/><br/>
            <input type="submit"  value="Submit" name="submit"/>
    </form>

</body>
</html>
```

4.)

p4.php

```php
<?php
// WAP to find greatest number in three unequal numbers.

$n1=100;
$n2=200;
$n3=50;


// && --> when first condition is true or also a secound cond is true then both are ture

if($n1>$n2 && $n1>$n3)
{
    echo $n1. "is greatest";
}
else if($n2>$n1 && $n2>$n3)
{
    echo $n2. " is greatest";
}
else
{
    echo $n3. " is greatest";
}


?>
```

5.)

```
<!-- Using ternary operator
     find greatest number
-->


<?php
$n1=1000;
$n2=500;
$n3=200;

$g = ($n1>$n2) ? (($n1>$n3) ? $n1:$n3) : (($n2>$n3) ? $n3:$n2);
echo $g. " is greatest";
?>
```

6.)

o6.php

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Check Greatest number </title>
</head>
<body>
    <h2>Greatest number taking input from user (using ternary operator)</h2>

    <?php
    if(isset($_POST["submit"]))
    {
        $n1=$_POST["n1"];
        $n2=$_POST["n2"];
        $n3=$_POST["n3"];

        $g=($n1>$n2) ? (($n1>$n3)?$n1:$n3) : (($n2>$n3)?$n2:$n3);
        echo $g." is greatest";

    }
    ?>



    <form action="" method="post">
        Enter first number :
        <input type="number" name="n1" required/><br/><br/>
        Enter second number :
        <input type="number" name="n2" required/><br/><br/>
        Enter third number :
        <input type="number" name="n3" required/><br/><br/>
       <input type="submit" value="Submit" name="submit"/>
    </form>

</body>
</html>
```

7.)

```php
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Greatest number</title>
</head>
<body>
    <h2>Greatest number taken input from User </h2>

    <?php
    if(isset($_POST["save"]))
    {
        $n1=$_POST["n1"];
        $n2=$_POST["n2"];
        $n3=$_POST["n3"];

        if($n1>$n2 && $n1>$n3)
        {
            echo $n1. " is greatest";
        }
        else if ($n2>$n3 && $n3>$n1)
        {
            echo $n2. " is greatest";

        }

    }
    ?>

    <form method="post">
        Enter first number :
        <input type="number" name="$n1" required/><br/><br/>
        Enter two number :
        <input type="number"  name="$n2" required/><br/><br/>
        Enter third number :
        <input type="number"  name="$n3"  required/><br/><br/>
        <input type="submit"  Value="Submit"  name="save"/>

    </form>
</body>
</html>
```

8.)

```php
<?php
//WAP to find sum of digits of given number.

$n=123;
$s=0;

while($n>0)
{
    $r=$n%10;
    $s=$s+$r;
    $n=$n/10;
}
echo "Sum of digits = " .$s;


?>
```

9.)

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sum of digits</title>
</head>
<body>
    <h2>Sum of digits of given number</h2>
     <?php


    if(isset($_POST["submit"]))
    {
      $n=$_POST["n"];
      $s=0;
      while($n>0)
      {
         $r=$n%10;
         $s=$s+$r;
         $n=$n/10;
      }
      echo  "Sum of digits =" .$s;

    }

     ?>



    <form action="" method="post">
      Enter a number :
      <input type="number" name="n" required>
      <br/><br/>
      <input type="submit" value="Submit" name="submit">
    </form>
</body>
</html>
```

10.)

```
<!--
    Prime number
-->


<?php

$n=5;
$c=0;

for($i=1; $i<=$n; $i++)
{
    if($n%$i==0)   //
    {
        $c++;
    }
}
if($c==2)
{
    echo  $n. " number is prime";
}
else
{
    echo $n. " number is not a prime";
}

?>
```

11.)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prime</title>
</head>
<body>
    <h2>Prime number taking input from user </h2>
    <form method="Post">
        Enter a number :
        <input type="number"  name="n" required/><br/><br/>
        <input type="submit"  name="submit"  value="Submit"/>
    </form>
    <br/>

    <?php
    if(isset($_POST["submit"]))
    {
        $n=$_POST["n"];
        $c=0;
        for($i=1; $i<=$n; $i++)
        {
            if($n%$i==0)
            {
                $c++;
            }
        }
        if($c==2)
        {
            echo $n. " number is prime.";
        }
        else
        {
            echo $n. " number is not a prime";
        }
    }
    ?>

</body>
</html>
```

12.)

p13.php

```php
<!--
    Fibonacci series
-->

<?php
// WAP to print fibonacci series up to 10 terms.
// Fibonacci number is that number --> apne pahle 2 terms ka add hota hai.-

$n1=0;  // first number
$n2=1;  // second number
$n3=0;  // Third number
$n=10;  // Number of terms
echo $n1. " " .$n2. " ";
for($i=1; $i<=$n-2; $i++)
{
    $n3=$n1+$n2;
    echo $n3. " ";

    $n1=$n2;
    $n2=$n3;
}
?>
```

13.)

p14.php

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Fibonacci Series</title>
</head>
<body>
    <h2>Fibonacci Series</h2>

    <?php
    if(isset($_POST["save"]))
    {
        $n=$_POST["n"];
        $n1=0;
        $n2=1;
        $n3=0;
        $n=count;
        echo $n1. "" .$n2. "";
        for($i=1; $i<=$count-2; $i++)
        {
            $n3=$n1+$n2;
            echo $n3;
            $n1=$n2;
            $n2=$n3;
        }
    }
    ?>

    <form action="" method="post">
        Enter a number :
        <input type="number" name="n" required>
        <br/><br/>
        <input type="submit" value="Submit" name="save">
    </form>

</body>
</html>
```

14.)

p15.php

```php
<?php
/*
    Array :- Array is a collection of elements. In php there are three types of array.
    array --
    1. Indexed Array :- Arrays with numeric index.
    2. Associative Array :- Array with key and value pair.
    3. Multidimensional Array :-   Arrays containing one or more array.

     How to create array ?
     $arr = array();

     */

    //  Example of Indexed Array
    $arr = array( "C","C++" , "C#" , "JAVA", "Python", "php");
    // echo $arr[0]. " "  .$arr[1] . ""  .$arr[2].  "" .$arr[3]. " " .$arr[4].  " " .$arr[5];

    for($i=1; $i<count($arr); $i++)
    {
        echo $arr[$i]. " <br/>";
    }


?>
```

15.)

p16.php

```php
<?php

/*
sort()  ---  sort() function sorts indexed array in ascending order.
rsort()  -- rsort() function sorts indexed array in descending order.
*/

$arr=array(12,5,8,78,90);
echo "Original array<br/>";

for($i=0; $i<count($arr); $i++)
{
    echo $arr[$i]. "<br/>";
}

//  sort array
sort($arr);
echo "Element of array in ascending order<br/>";

for($i=0; $i<count($arr); $i++)
{
    echo $arr[$i] . "<br/>";
}

//
rsort($arr);
echo "Element of array in decending order<br/>";

for($i=0; $i<count($arr); $i++)
{
    echo  $arr[$i] . "<br/>";
}

?>
```

16.)

p17.php

```php
<?php
//  Associative Array

$arr=array(
    "brijesh"=>"42",
    "ravi" =>"43",
    "shikha"=>"45",
    "nisha"=>"42"
);

foreach($arr as $k=>$v)
{
    echo "Hello " . $k . " Your age is " . $v . "<br/>";
}
?>
```

17.)

```php
<?php
// Multidimension Array

$arr=array(
    array(1,2,3),
    array(4,5,6),
    array(7,8,9)
);
echo "Element of matrix<br/>";
for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        echo $arr[$i][$j]. " ";
    }
    echo "<br/>";
}

?>
```

18.)

```php
<?php
//WAP to find submmation and subtraction of two matrices.

$arr1 = array(
    array(9, 8, 7),
    array(4,5,6),
    array(1,2,3)
);
echo "Element of First Matrix : <br/>";
for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        echo $arr1[$i][$j] ." ";
    }
    echo "<br/>";
}


$arr2= array(
    array(1,2,3),
    array(4,5,6),
    array(7,8,9)
);

echo "Element of second matrix : <br/>";

for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        echo $arr2[$i][$j] ." ";
    }
    echo "<br/>";
}

$sum=array( ) ;
$sub=array( ) ;
```

```php
$sum=array( ) ;
$sub=array( ) ;

// Code for Submmiasion
echo "Sum of Matrix : <br/>";
for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        $sum[$i][$j]=$arr1[$i][$j] + $arr2[$i][$j];
        echo $sum[$i][$j]. " ";
    }
    echo "<br/>";
}

// Code of Subtraction
echo "Subtration of Matrix : <br/>";

for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        $sub[$i][$j]=$arr1[$i][$j] - $arr2[$i][$j];
        echo $sub[$i][$j]. " ";
    }
    echo "<br/>";
}

?>
```

19.)

```php
<?php
//WAP to find submmation and subtraction of two matrices.

$arr1 = array(
    array(9, 8, 7),
    array(4,5,6),
    array(1,2,3)
);

$arr2= array(
    array(1,2,3),
    array(4,5,6),
    array(7,8,9)
);

$sum=array( ) ;
$sub=array( ) ;

//
echo "Element of First Matrix : <br/>";
for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        echo $arr1[$i][$j] ." ";
    }
    echo "<br/>";
}


echo "Element of second matrix : <br/>";
for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        echo $arr2[$i][$j] ." ";
    }
    echo "<br/> ";
}
```

```php
// Code for Submmiasion
for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        $sum[$i][$j]=$arr1[$i][$j] + $arr2[$i][$j];
    }
}


// Code of Subtraction
for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        $sub[$i][$j]=$arr1[$i][$j] - $arr2[$i][$j];
    }
}


// Display sum matrix element
echo "Sum of two matrixces :<br/>";
for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        echo $sum[$i][$j]. " ";
    }
    echo "<br/>";
}


echo "Subtration of Matrix : <br/>";
for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        echo $sub[$i][$j]. " ";
    }
    echo "<br/>";
}
?>
```

20.)

p21.php

```php
<?php
// WAP to Matrix Multiplication

$arr1=array(
    array(1,2,3),
    array(2,3,4),
    array(3,4,5)
);

$arr2=array(
    array(1,0,0),
    array(0,1,0),
    array(0,0,1)
);

$mult=array();

for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        $mult[$i][$j]=0;
        for($k=0; $k<3; $k++)
        {
            $mult[$i][$j]=$mult[$i][$j]+$arr1[$i][$k]*$arr2[$k][$j];
        }
    }
}
echo "Result of matrix multiplication<br/>";
for($i=0; $i<3; $i++)
{
    for($j=0; $j<3; $j++)
    {
        echo $mult[$i][$j]  . " ";
    }
    echo "<br/>";
}
?>
```

21.)

p22.php

```php
<?php

/*

Function :- Function is a named block of code, which perform specific task.

Why use function?
Function is used to avoid repeatition of code.

How to create function in php?
function functionname(parameters)
{
    Function code
}

*/


function add($x, $y)
{
   return $x+$y;
}
$res=add(100,200);
echo "Result = " .$res. "<br/>";

?>
```

22.)

```php
<!--
    Make a Simple Calculator using user-defined functions and take input from user .
-->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Calculator</title>
</head>
<body>
    <h2>Calculator</h2>
        <?php

        function add($a,$b)
        {
            return ($a+$b);
        }
        add($a,$b);

        function sub($a,$b)
        {
            return ($a-$b);
        }
        sub($a,$b);

        function mult($a,$b)
        {
            return ($a*$b);
        }
        mult($a,$b);

        function div($a,$b)
        {
            return ($a+$b);
        }
        div($a,$b);
```

```php
    if(isset($_POST["click"]))
    {
      $n=$_POST["n"];
      $n1=$_POST["n2"];

      $result=$_POST["result"];
      if($result==add)
      {

      }
    }



    ?>


  <form action="" method="post">
    Enter a first number :
    <input type="number" name="n" required><br/><br/>
    Enter second number :
    <input type="number" name="n1" required><br/>
    <input type="submit" value="Add" name="click">
    <input type="submit" value="Sub" name="click">
    <input type="submit" value="Mult" name="click">
    <input type="submit" value="Div" name="click">

  </form>
</body>
</html>
```

23.)

p24.php

```php
<?php
/*
Class :-  Class is a blueprint of Object. Class is a Collection of variable and method.

class ClassName
{
    variables and method
}
obj=new classname();

Object :- Object is a real world entity, which has its properties and functionalities.

*/


// class MyClass
// {
//     public function sayHello()
//     {
//         echo "Hello World";
//     }
// }
// $m=new MyClass();
// $m ->sayHello();




// Using parameter
class MyClass
{
    public function sayHello($name)
    {
        echo "Hello " .$name;
    }
}
$m=new MyClass();
$m ->sayHello("soni nishad");    // ---> method ko call krne ke liye ARRAW Function (->) ka use karte hai.
?>
```

23.)

```php
<?php

class Employee
{
    public $empid;
    public $empname;
    public $salary;

    //class ke variable ko Access krne ke liye this keyword ka use krte hai.
    /*
        this->empid=$empid
                    |       |
                    variable  parameter
    */

    public function setValue($empid, $empname, $salary)
    {
        $this->empid=$empid;
        $this->empname=$empname;
        $this->salary=$salary;
    }

    public function display()
    {
        echo "Employee Id = "     .$this->empid. "<br/>";
        echo "Employee Name = "   .$this->empname.  "<br/>";
        echo  "Employee salary = "   .$this->salary     . "<br/>";
    }
}

$e1=new Employee();
$e1->setValue("1001", "rajat", 50000);

$e2=new Employee();
$e2->setValue("1002","nisha",60000);

$e1->display();
$e2->display();

?>
```

24.)

p26.php

```php
<?php
/*

Constructor :- Constructor is special method, which is used to initialize variables.
               Constructor call automatically as soon as object is created.

    public function  __constructor(parameters)
    {
      code
    }

*/

class Rectangle
{
    public $l;
    public $b;

    public function __construct($l, $b)
    {
        $this->l=$l;
        $this->b=$b;
    }

    public function area()
    {
        return $this->l*$this->b;
    }

}

$r=new Rectangle(10,15);
$a=$r->area();
echo "Area of Rectangle =".$a;
?>
```

25.)

p27.php

```php
<?php
/*
Inheritance  ---  Inheritance is a feature of object oriented programming.
                  In Inheritance you can create a new class by using existing class.

   Existing class is called base class and new created class is called derived class.
   Concept of inheritance is also called "Reusiablitity".

   what is descreator?

Syntax :-
---------

class A
{
   code
}
class B extends A
{
   code
}
*/
class A{
   public function showA()
   {
      echo "This message from class A . <br/>";
   }
}

class B extends A {
    public function showB()
    {
      echo "This is message from class B .<br/>";
    }
}

$b=new B();
$b->showB();
$b->showA();

?>
```

26.)

```php
<?php

// WAP to check given string is palindrome or not

// $str="hello";
$str="madam";
$rev=strrev($str);

if($str==$rev)
{
   echo "String is Palindrome";
}
else {
   echo "String is not a palindrome";
}

?>
```

27.)

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Check Plindrome</title>
</head>
<body>
    <h2>Check Palindrome</h2>


    <?php
    if(isset($_POST["check"]))
    {
        $str=$_POST["str"];
        $rev=strrev($str);

        if($str==$rev)
        {
            echo "String is palindrome .<br/>";
        }
        else
        {
            echo "String is not Palindrome <br/>";
        }
    }
    ?>

    <form action="" method="post">
        Enter a text :
        <input type="text" name="str"  required><br/><br/>
        <input type="submit" value="Check" name="check"/>
    </form>
</body>
</html>
```

❾ **Image Adding** –

code.php

```php
<?php
if(isset($_POST["save"]))
{
    $name=$_POST["name"];
    $gender=$_POST["gender"];
    $address=$_POST["address"];
    $contactno=$_POST["contactno"];
    $profilepic=$_FILES["profilepic"]["name"];
    $profilepic_tmp=$_FILES["profilepic"]["tmp_name"];
    $ext=pathinfo($profilepic,PATHINFO_EXTENSION);
    if($ext=="jpg"||$ext=="jpeg"||$ext=="png"||$ext=="bmp"||$ext=="gif")
    {
        $con=mysqli_connect("localhost","root","","phpdb");
        $query="insert into friendinfo(name, gender, address, contactno, profilepic)
        values('$name','$gender','$address','$contactno','$profilepic')";
        $res=mysqli_query($con,$query);
        if($res>0)
        {
            move_uploaded_file($profilepic_tmp,"upload/$profilepic");
            header("Location:index.php?msg=2");
        }
    }
    else
    {
        header("Location:index.php?msg=1");
    }
}
else
{
    header("Location:index.php");
}
?>
```

9

```php
$contactno=$_POST["contactno"];
$profilepic=$_FILES["profilepic"]["name"];
$profilepic_tmp=$_FILES["profilepic"]["tmp_name"];
$ext=pathinfo($profilepic,PATHINFO_EXTENSION);
if($ext=="jpg"||$ext=="jpeg"||$ext=="png"||$ext=="bmp"||$ext=="gif")
{
    $con=mysqli_connect("localhost","root","","phpdb");
```

```php
$profilepic=$_FILES["profilepic"]["name"];
$profilepic_tmp=$_FILES["profilepic"]["tmp_name"];
$ext=pathinfo($profilepic,PATHINFO_EXTENSION);
if($ext=="jpg"||$ext=="jpeg"||$ext=="png"||$ext=="bmp"||$ext=="gif")
{
    $con=mysqli_connect("localhost","root","","phpdb");
    $query="insert into friendinfo(name, gender, address, contactno, profilepic)
    values('$name','$gender','$address','$contactno','$profilepic')";
    $res=mysqli_query($con,$query);
    if($res>0)
    {
        move_uploaded_file($profilepic_tmp,"upload/$profilepic");
        header("Location:index.php?msg=2");
    }
}
```

❼

```php
move_uploaded_file($profilepic_tmp,"upload/$profilepic");
header("Location:index.php?msg=2");
```

VIEW FILE –

```php
<h1>View all files</h1>
<?php
$con=mysqli_connect("localhost","root","","phpdb");
$query="select * from friendinfo";
$res=mysqli_query($con,$query);
if(mysqli_num_rows($res)>0)
{

?>
```

❾

```php
<?php
while($row=mysqli_fetch_assoc($res))
{
?>

    <tr>
      <td><?php echo $row["id"] ?></td>
```

```php
<?php
while($row=mysqli_fetch_assoc($res))
{
?>

    <tr>
      <td><?php echo $row["id"] ?></td>
      <td><?php echo $row["name"] ?></td>
      <td><?php echo $row["gender"] ?></td>
      <td><?php echo $row["address"] ?></td>
      <td><?php echo $row["contactno"] ?></td>

      <td>
        <img src="upload/<?php echo $row["profilepic"]?>"  width="100" height="100"/>
      </td>

      <td>
        <a href="delete.php?id=<?php echo $row["id"]?>">Delete</a>
      <td>
    </tr>
<?php
}
?>
</table>

   <?php
   }
```

---

❾Delete code –

```php
 <td>
    <a href="delete.php?id=<?php echo $row["id"]?>">Delete</a>
 <td>
</tr>
```

---

❾UPDATE CODE –

```php
<?php
//  agr set nahi hai to  block execute
   if(! isset($_GET["id"]))
   {
       header("Loction:index.php");
   }
   $id=$_GET["id"];
   $con=mysqli_connect("localhost","root","","phpdb");
   // query for record select
   $query="select * from userinfo where id='$id' ";
   $res=mysqli_query($con,$query);
   if(mysqli_num_rows($res)>0)
   {
      $row=mysqli_fetch_assoc($res);
?>
```

❾

```
<form action="updatecode.php" method="post">
Name :
<input type="text" name="name" value="<?php  echo $row["name"]  ?>" />
<br/><br/>

Gender :
<?php
// opening male
    if($row["gender"]=="Male")
    {
?>
       <input type="radio" checked="true"  name="gender" value="Male"/>Male
       <input type="radio"  name="gender" value="Female"/>Female
<?php
    }  // male closing
    else if($row["gender"]=="Female")
    { // Female opening
?>
<input type="radio" name="gender" value="Male"/>Male
<input type="radio" checked="true"  name="gender" value="Female"/>Female
<?php
    // Female closing
   }
?>
<br/><br/>
```

❾

```php
if(isset($_POST["update"]))
{
    $id=$_POST["id"];
    $name=$_POST["name"];
    $gender=$_POST["gender"];
    $address=$_POST["address"];
    $contactno=$_POST["contactno"];
    $emailaddress=$_POST["emailaddress"];
    $con=mysqli_connect("localhost","root","","phpdb");
    $query="update userinfo set name='$name', gender='$gender',
     address='$address', contactno='$contactno',
     emailaddress='$emailaddress'  where id='$id' ";
     mysqli_query($con,$query);
     header("Location:index.php");
}
else
{
   header("Location:index.php");
}
```

➔ What is PDO ---------- php interview ques

➔OOPS USING PDO ------ php

➔CRUD USING PDO ------ php

➔Scope resolution operator (::) ➔ [Nekodotayim] ----- php