



PHP Echo Statement :

Print Message or Variable value

```
echo "Yahoo Baba";
```

```
echo 'Yahoo Baba';
```

```
echo "Yahoo", " Baba";
```

```
echo "Yahoo" . " Baba";
```

```
echo "<b>Yahoo Baba</b>";
```

```
echo 502.25;
```



PHP Print Statement :

Print Message or Variable value

```
print "Yahoo Baba";
```

```
print "Yahoo" . " Baba";
```

```
print "<b>Yahoo Baba</b>";
```

```
print 502.25;
```



Difference between Echo & Print :

`echo "Yahoo", " Baba";`

`print "Yahoo", " Baba";`



`echo` is faster than `print`





PHP Variables :

```
$a = "Yahoo Baba";
```

```
echo $a;
```

```
$num = 25.87;
```

```
echo $num;
```



How to write a Variable Name :

Write Way

- \$firstname
- \$_firstname
- \$first_name
- \$first-name
- \$firstName
- \$firstname99

Wrong Way

- \$first name
- \$99firstname
- \$first%name

\$age \$AGE

Not same





What are Variables ?

`$x = "Hello World";` → String

`$x = 25;` → Integer

`$x = 30.50;` → Float

`$x = true;` → Boolean

`$x = array("HTML","CSS","JS");` → Array

`$x = new MyClass();` → Object

`$x = null;` → Null





Single Line Comment

```
<?php  
    // $x = "Yahoo Baba";  
    echo $x;          # here is another comment  
?>
```



Multi Line Comment

```
<?php  
$x = "Yahoo Baba";  
echo $x; /* here is comment  
which can be on multiple lines */  
?>
```





```
define(name, value, case-insensitive)
```





```
define(name, value, case-insensitive)
```

```
define(num, 500)
```

```
echo num;
```



```
define(name, value, case-insensitive)
```

```
define(num, 500, true)
```

```
echo NUM;
```



```
define(num, 500, true);
```

```
define(_num, 500, true);
```

Can't use \$ sign with constant variable name.

Constant Variables are Global Variables.



Arithmetic Operators

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation
/	Division
%	Modulus (Remainder)
++	Increment Operator
--	Decrement Operator



Assignment Operators

Operator	Example	Same As
=	$x = y$	$x = y$
+=	$x += y$	$x = x + y$
-=	$x -= y$	$x = x - y$
*=	$x *= y$	$x = x * y$
/=	$x /= y$	$x = x / y$
%=	$x \%= y$	$x = x \% y$
**=	$x **= y$	$x = x ** y$





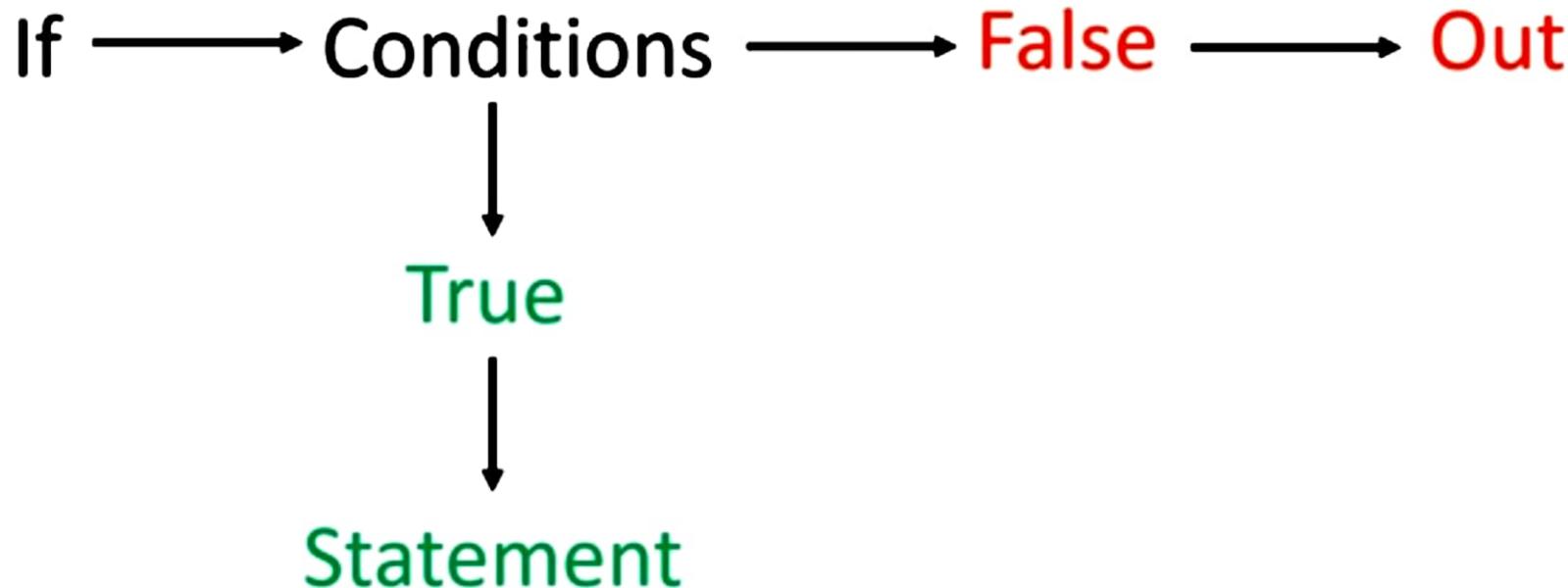
Comparison Operators

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





What is If Statement ?





If Statement in PHP

```
If(Condition){
```

```
}
```



Different Type of Logical Operators :

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



If Else Statement in PHP :

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



If Elseif Statement in PHP

```
If(Condition 1){  
    Statement 1  
} elseif (Condition 2){  
    Statement 2  
} else {  
    Default Statement  
}
```



Switch Statement in PHP

```
switch (expression) {  
    case condition 1: statement(s)  
        break;  
  
    case condition 2: statement(s)  
        break;  
  
    case condition 3: statement(s)  
        break;  
  
    default: statement(s)  
}
```



Conditional Ternary Operator Syntax :

(Condition) ? True Statement : False Statement



String Operator

```
$a = "Hello ";
$b = $a . "World!";
echo $b;
```

Hello World





String Operator

```
$a = "Hello ";
$a .= "this is ";
$a .= "my world.";
echo $a;
```

Hello this is my world.



While loop in PHP

```
$a = 1;  
  
while($a <= 10){  
    echo "Yahoo Baba";  
    $a = $a + 1;  
}
```





Do While Loop in PHP

```
$a = 1;  
do{  
    echo "Yahoo Baba";  
    $a = $a + 1;  
} while($a <= 10)
```



For Loop Syntax in PHP

```
for(Initialization; Condition; Increment / Decrement){  
          
}  
        
```





Continue Statement in PHP :

```
for($a = 1; $a <= 10; $a++){  
    If ($a == 3){  
        continue;  
    }  
    echo $a;  
}
```

1
2
3
4
5
6
7
8
9
10





Break Statement in PHP :

```
for($a = 1; $a <= 10; $a++){
```

1
2
3

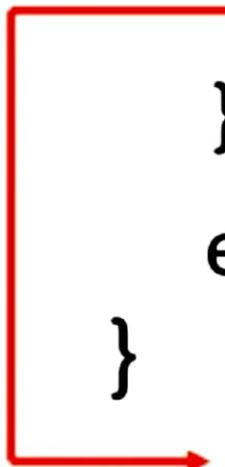
```
    If ($a == 3){
```

```
        break;
```

```
    }
```

```
    echo $a;
```

```
}
```





Goto Statement in PHP :

```
for($a = 1; $a <= 10; $a++){  
    if ($a == 3){  
        goto abc;  
    }  
    echo $a;  
}  
  
echo "Hello";  
abc:  
echo "This is new code";
```

An orange arrow points from the word "Label" to the label "abc:" in the code.



Function Syntax in PHP :

```
function functionName(){ ← Function Definition  
    Statement  
}
```

```
functionName(); ← Calling a Function
```



```
$a = 10;  
$b = 20;  
echo $a + $b;  
....  
....  
....  
  
$a = 30;  
$b = 40;  
echo $a + $b;  
....  
....  
....  
  
$a = 25;  
$b = 55;  
echo $a + $b;
```

```
function sum($a, $b){  
    echo $a + $b;  
}  
....
```

```
sum(10,20);  
....  
....  
sum(30,40);
```



Function with Parameters in PHP :

```
function functionName(parameter1, parameter2){  
}  
}
```



Function with Return Value in PHP :

```
function functionName(parameter1, parameter2){  
    Statements  
    return value;  
}
```



Different Function Arguments in PHP :

Passing Arguments by Value

```
function wow($a){  
    $a = "Hey";  
}  
  
$str = "Hello";  
  
wow($str);  
  
echo $str; → Hello
```

Passing Arguments by Reference

```
function wow(&$a){  
    $a = "Hey";  
}  
  
$str = "Hello";  
  
wow($str);  
  
echo $str; → Hey
```



Variable Function in PHP :

```
function wow(){  
}  
  
$func = "wow";  
  
$func();
```



Recursive Function in PHP :

```
function test(){  
    Statement  
    test();  
}
```

```
test();
```



Variable Scope in PHP :

```
function test(){  
    $a = 10; ← Local Variable  
}  
  
test();  
  
echo $a;
```





Variable Scope in PHP :

```
$a = 10;  
  
function test(){  
    echo $a;  
}  
  
test();
```

```
function.php
1 <?php
2     $x = 10;
3     function test(){
4         global $x;
5         echo "Variable X inside function : $x";
6     }
7
8     test();
9     echo "Variable X outside function : $x"
10
11 ?>
12
```

Notice: Undefined variable: x in
C:\xampp\htdocs\LearnPHP\function
on line 5
Variable X inside function : Variable
X outside function : 10

```
1 <?php
2     $x = 10;
3     $y = 5;
4
5     function test(){
6         global $x , $y;
7         $x = $x + $y;
8     }
9
10    test();
11    echo $x;
12
13    ?>
14
```

15



What is Array ?

`$a = 10;`

`$a = 10,20,30;`  Error

`$a = "10,20,30";`  String

`$a = array(10,20,30);`



How Array Stores Value ?

0 1 2 ← Index
\$a = array(10,20,30);

Indexed arrays



What is Associative Array ?

0 1 2 ← Index

```
$a = array(10,20,30);
```

```
$a = array("Bill"=>10, "Joe"=> 20, "Peter"=> 30);
```





PHP : Associative Array

```
$a = array(  
    Key → "Bill" => 10, ← Value  
        "Joe" => 20,  
        "Peter" => 30  
);
```



Foreach Loop Syntax in PHP

```
foreach($array as $value){
```

►

```
}
```





Multidimensional Array or Nested Array :

Emp No.	Name	Designation	Salary
1	Krishana	Manager	50000
2	Salman	Salesman	20000
3	Mohan	Computer Operator	12000
4	Amir	Driver	5000

```
$emp = array (
    array(1,"Krishana","Manager",50000),
    array(2,"Salman","Salesman",20000),
    array(3,"Mohan","Computer Operator",12000),
    array(4,"Amir","Driver",5000)
);
```



Multidimensional Array in PHP :

```
$emp = [  
    [ 1 , "Krishana" , "Manager" , 50000 ],  
    [ 2 , "Salman" , "Salesman" , 20000 ],  
    [ 3 , "Mohan" , "Computer Operator" , 12000 ],  
    [ 4 , "Amir" , "Driver" , 5000 ]  
];
```





Multidimensional Array in PHP :

```
$emp = [  
    0 [ 1 , "Krishna" , "Manager" , 50000 ],  
    1 [ 2 , "Salman" , "Salesman" , 20000 ],  
    2 [ 3 , "Mohan" , "Computer Operator" , 12000 ],  
    3 [ 4 , "Amir" , "Driver" , 5000 ]  
];
```



Multidimensional Array in PHP :

```
$emp = [  
    0 [ 1 , "Krishna" , "Manager" , 50000 ],  
    1 [ 2 , "Salman" , "Salesman" , 20000 ],  
    2 [ 3 , "Mohan" , "Computer Operator" , 12000 ],  
    3 [ 4 , "Amir" , "Driver" , 5000 ]  
];  
echo $emp[1][1];
```

```
array.php
1 <?php
2 $emp = [
3     [1,"Krishana","Manager",50000],
4     [2,"Salman","Salesman",20000],
5     [3,"Mohan","Computer Operator",12000]
6     [4,"Amir","Driver",5000]
7 ];
8
9 print_r($emp);
10
11 ?>
12
```

localhost/LearnPHP/array.php
Paused

```
Array ( [0] => Array ( [0] => 1 [1]
=> Krishana [2] => Manager [3] =>
50000 ) [1] => Array ( [0] => 2 [1]
=> Salman [2] => Salesman [3] =>
20000 ) [2] => Array ( [0] => 3 [1]
=> Mohan [2] => Computer
Operator [3] => 12000 ) [3] =>
Array ( [0] => 4 [1] => Amir [2] =>
Driver [3] => 5000 ) )
```

```
$marks = [  
    "Krishana" => [  
        "physics" => 85,  
        "maths" => 78,  
        "chemistry" => 89  
    ],  
    "Salman" => [  
        "physics" => 68,  
        "maths" => 73,  
        "chemistry" => 79  
    ],  
    "Mohan" => [  
        "physics" => 62,  
        "maths" => 67,  
        "chemistry" => 92  
    ]  
];
```



Multidimensional Associative Array :

Name	Physics	Chemistry	Math
Krishana	85	89	78
Salman	68	79	73
Mohan	62	92	67

```
$marks = array (  
    "Krishana" => array("physics" => 85, "chemistry" => 89, "math" => 78),  
    "Salman" => array("physics" => 68, "chemistry" => 79, "math" => 73),  
    "Mohan" => array("physics" => 62, "chemistry" => 92, "math" => 67)  
,
```

```
$marks = [  
    "Krishana" => [  
        "physics" => 85,  
        "maths" => 78,  
        "chemistry" => 89  
    ],  
    "Salman" => [  
        "physics" => 68,  
        "maths" => 73,  
        "chemistry" => 79  
    ],  
    "Mohan" => [  
        "physics" => 62,  
        "maths" => 67,  
        "chemistry" => 92  
    ]  
];
```



```
foreach($array as list()){
```

Statement

```
}
```



Foreach Loop with List() in PHP

```
$array = [
    [1,2],
    [30,40]
];
```

```
foreach($array as list($a, $b)){
}
```



Foreach Loop with List() in PHP

```
$array = [
    [1,2],
    [30,40]
];
foreach($array as list($a, $b)){
    echo $a . $b;
}
```



PHP Array : Count() & Sizeof()

```
$a = ["Sanjay","Aman","Rehman","Karan"];
```

count()

sizeof()

```
1 <?php  
2  
3 // $food = array('orange', 'banana', 'apple', 'grapes');  
4  
5 $food = array(  
6     'fruits' => array('orange', 'banana', 'apple'),  
7     'veggie' => array('carrot', 'collard', 'pea')  
8 );  
9  
10 echo count($food, 1);  
11  
12 ?>  
13
```

```
1 <?php  
2  
3 // $food = array('orange', 'banana', 'apple', 'grapes');  
4  
5 $food = array(  
6     'fruits' => array('orange', 'banana', 'apple'),  
7     'veggie' => array('carrot', 'collard', 'pea')  
8 );  
9  
10 echo count($food['fruits'],1);  
11  
12 ?>  
13
```



PHP Array : in_array() & array_search()

```
$a = ["Sanjay","Aman","Rehman","Karan"];
```

Aman

in_array()



1 / 0

array_search()



index / key

```
1 <?php  
2  
3 $food = array('orange', 'banana', 'apple', 'grapes');  
4  
5 echo in_array('apple', $food);  
6  
7  
8 // $a = array(array('p', 'h'), array('p', 'r'), 'o');  
9  
10 // $food = array('a' => 'orange', 'b' => 'banana', 'c' => 'apple',  
11  
12  
13 ?>  
14
```



PHP Array Replace Functions :

```
$a = ["Sanjay", "Aman", "Rehman", "Karan"];
```

```
$b = ["Gagan", "Salman"];
```

array_replace()



Index or **Associative** Array

array_replace_recursive()



Multidemensional **Associative** Array

```
1 <?php  
2  
3 $fruit = [ 'orange', 'banana', 'apple', 'grapes' ];  
4  
5 $veggie = [ 'carrot', 'pea' ];  
6  
7 array_replace($fruit, $veggie);|  
8  
9  
10  
11 // $color = [ 'red', 'green', 'blue' ];  
12 // $array1 = array("a"=>array("red"), "b"=>array("green", "pink"));  
13 // $array2 = array("a"=>array("yellow"), "b"=>array("black"));  
14
```

```
1 <?php  
2  
3 $fruit = ['orange', 'banana', 'apple', 'grapes'];  
4  
5 $veggie = ['carrot', 'pea'];  
6  
7 $newArray = array_replace($fruit, $veggie);  
8  
9  
10  
11 // $color = ['red', 'green', 'blue'];  
12 // $array1 = array("a"=>array("red"), "b"=>array("green", "pink"));  
13 // $array2 = array("a"=>array("yellow"), "b"=>array("black"));  
14
```

```
1 <?php  
2  
3 $fruit = ['orange', 'banana', 'apple', 'grapes'];  
4  
5 $veggie = ['carrot', 'pea'];  
6  
7 $color = ['red', 'green', 'blue'];  
8  
9 $newArray = array_replace($fruit, $veggie, $color);  
10  
11 echo "<pre>";  
12 print_r($newArray);  
13 echo "</pre>";  
14  
15
```

```
1 <?php
2
3 $array1 = array("a"=>array("red"), "b"=>array("green", "pink"));
4
5 $array2 = array("a"=>array("yellow"), "b"=>array("black"));
6
7 $newArray = array_replace_recursive($array1, $array2);
8
9 echo "<pre>";
10 print_r($newArray);
11 echo "</pre>";
12
13
14
15
```

```
1 <?php  
2  
3 $fruit = ["orange", "banana", "apple", "grapes"];  
4  
5 array_pop($fruit);  
6  
7 ?>  
8
```

```
1 <?php  
2  
3 $fruit = ["orange", "banana", "grapes"];  
4  
5 array_push($fruit, "Apple");  
6  
7 echo "<pre>";  
8 print_r($fruit);  
9 echo "</pre>";  
10  
11 ?>  
12
```

```
1 <?php  
2  
3 $fruit = ["orange", "banana", "grapes"];  
4  
5 array_shift($fruit);  
6  
7  
8  
9 ??  
10
```

```
1 <?php  
2  
3 $fruit = ["orange", "banana", "grapes"];  
4  
5 array_unshift($fruit, "apple");  
6  
7 echo "<pre>";  
8 print_r($fruit);  
9 echo "</pre>";  
10  
11 ?>  
12
```



PHP Array – Merge & Combine Functions :

```
$a = ["Sanjay", "Aman", "Rehman"];
```

```
$b = ["Karan"];
```

```
["Sanjay", "Aman", "Rehman", "Karan"]
```



- **array_merge()** Index or Associative Array
- **array_merge_recursive()** Multidimensional Associative Array
- **array_combine()** Index Array



```
$a = ["Sanjay","Aman","Rehman", "Karan"];
```

```
["Aman","Rehman"]
```

```
array_slice(array , start , length)
```



PHP Array – Array_Splice() Function :

```
$a = ["Sanjay", "Aman", "Rehman", "Karan"];
```

```
$b = ["Ram", "Mohan"];
```

```
$a = ["Sanjay", "Ram", "Mohan", "Karan"]
```

array_splice(array , start , length, array-II)



PHP Array – Key Functions :

- `array_keys()`
- `array_key_first()`
- `array_key_last()`
- `array_key_exists()`
- `key_exists()`



PHP Array : Intersect Functions

- `array_intersect`
- `array_intersect_key`
- `array_intersect_assoc`
- `array_intersect_uassoc`
- `array_intersect_ukey`
- `array_uintersect`
- `array_uintersect_assoc`
- `array_uintersect_uassoc`



PHP Array : Difference Functions

- `array_diff`
- `array_diff_key`
- `array_diff_assoc`
- `array_diff_uassoc`
- `array_diff_ukey`
- `array_udiff_assoc`
- `array_udiff_uassoc`
- `array_udiff`



PHP Array Values Function :

```
$a = ["a" => "Sanjay", "b" => "Aman", "c" => "Mohan"];
```

```
["Sanjay", "Aman", "Mohan"];
```

array_values



PHP Array Unique Function :

```
$a = ["a"=>"red", "b"=>"green", "c"=>"red"];
```



```
["a"=>"red", "b"=>"green"];
```

array_unique

```
$array = array(  
    array(  
        'id' => 2201,  
        'first_name' => 'Anil',  
        'last_name' => 'Kapoor',  
    ),  
    array(  
        'id' => 2202,  
        'first_name' => 'Salman',  
        'last_name' => 'Khan',  
    ),  
    array(  
        'id' => 2203,  
        'first_name' => 'John',  
        'last_name' => 'Abraham',  
    )  
);
```

array_column

```
Array  
(  
    [0] => Anil  
    [1] => Salman  
    [2] => John  
)
```



```
$a = ["Red", "Green", "Blue", "Orange", "Brown"];
```

```
Array
(
    [0] => Array(
        [0] => Red
        [1] => Green
    ),
    [1] => Array(
        [2] => Blue
        [3] => Orange
    ),
    [2] => Array (
        [4] => Brown
    )
)
```

array_chunk (array , size)



PHP Array – Array_Flip() Function :

```
$a = array(  
    "Bill" => 10,  
    "Joe" => 20,  
    "Peter" => 30  
)
```



```
$a = array(  
    10 => "Bill",  
    20 => "Joe",  
    30 => "Peter"  
)
```

array_flip(array)



PHP Array – Array_change_key_case () Function :

```
$a = array(  
    "Bill" => 10,  
    "Joe" => 20,  
    "Peter" => 30  
,
```



```
$a = array(  
    "BILL" => 10,  
    "JOE" => 20,  
    "PETER" => 30  
,
```

array_change_key_case(array , case)



PHP Array_Sum() Function :

```
$a = [10, 20, 30, 40, 50];
```

$$10 + 20 + 30 + 40 + 50 = 150$$



```
array_sum(array)
```



PHP Array_Product() Function :

```
$a = [10, 20, 30, 40, 50];
```

$$10 \times 20 \times 30 \times 40 \times 50 = 12,000,000$$

array_product(array)



PHP Array_Rand() Function :

```
$a = ["Red", "Green", "Blue", "Orange", "Brown"];
```

Red

Blue

Brown

array_rand(array , number)



PHP Array_fill_keys() Function :

```
$a = ["a", "b", "c", "d"];
```

```
Array(  
    "a" => "test",  
    "b" => "test",  
    "c" => "test",  
    "d" => "test"  
)
```

array_fill_keys(array , value)



PHP Array_fill() Function :

```
Array(  
    [0] => "test",  
    [1] => "test",  
    [2] => "test",  
    [3] => "test"  
);
```

```
Array(  
    [3] => "test",  
    [4] => "test",  
    [5] => "test"  
);
```

array_fill(index , number , value)



PHP Array – Array_Walk() Function :

```
$a = array(  
    "Bill" => 10,  
    "Joe" => 20,  
    "Peter" => 30  
)  
  
function myFunction(){  
}  
}
```

array_walk(array , function , parameter)



PHP Array – Array_Walk_Recursive() Function :

```
$a = array(  
    "Bill" => array(  
        "rollno" => 12,  
        "age" => 30,  
    ),  
    "Joe" => 20,  
    "Peter" => 30  
,
```

```
function myFunction(){  
}
```

array_walk_recursive(array , function , parameter)



PHP Array – Array_Map() Function :

```
$a = array(  
    "Bill" => 10,  
    "Joe" => 20,  
    "Peter" => 30  
);  
  
function myFunction(){  
}  
}  
  
Return Array
```

array_map(function , array , array2 , array3)



PHP Array – Array_Reduce() Function :

```
$a = array(  
    10, → function myFunction(){  
    20, →  
    30 }  
);
```

Return Single Value ←

array_reduce(array , function , initial)



PHP Array Sorting :

```
$a = ["Sanjay","Aman","Rehman","Karan"];
```

Aman , Karan , Rehman , Sanjay → Ascending Order

Sanjay , Rehman , Karan , Aman → Descending Order

Karan , Rehman , Aman , Sanjay → Reverse Order



- sort
- rsort
- arsort
- asort
- krsort
- ksort
- natcasesort
- natsort
- array_multisort
- array_reverse





PHP Array : Traversing Functions

- next
- prev
- end
- each
- pos
- current
- key
- reset





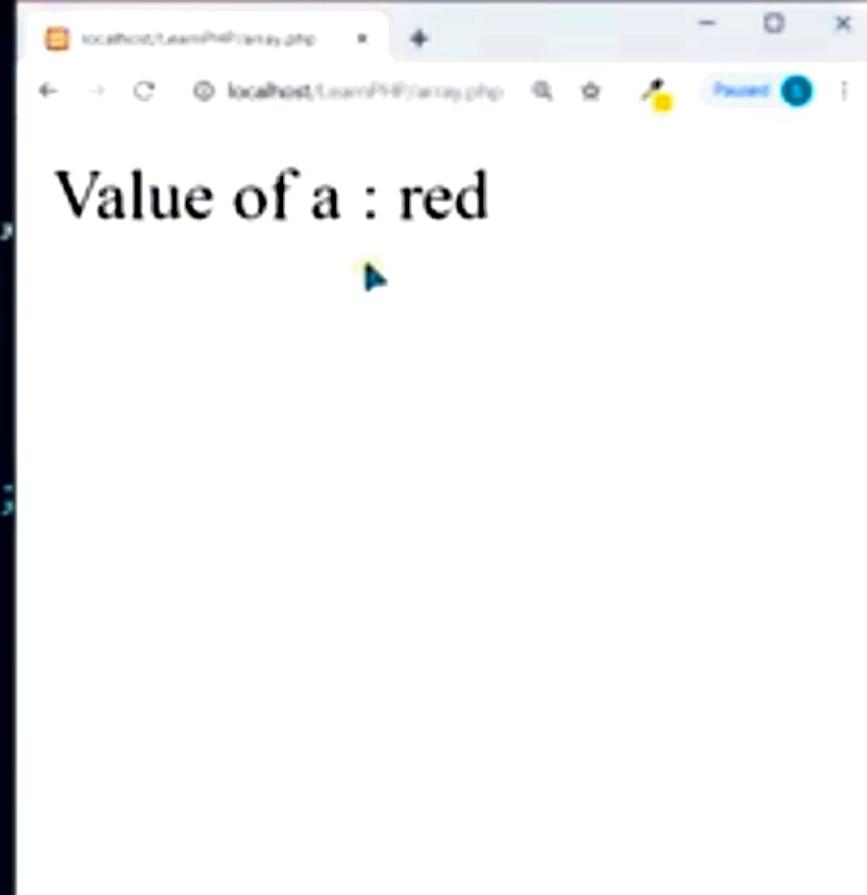
PHP Array : Extract() Function

```
$color = array('a' => 'red', 'b' => 'green', 'c' => 'blue');
```



```
extract(array, extract_rules, prefix)
```

```
array.php
1 <?php
2
3 $color = array('a' => 'red',
4
5 extract($color);
6
7 echo "Value of a : $a <br>";
8
9
10
11
12 ?>
13
```



A screenshot of a Microsoft Edge browser window. The address bar shows "localhost/LearnPHP/array.php". The main content area displays the text "Value of a : red" in a large, black, sans-serif font. There is a small blue navigation icon below the text.



`extract(array, extract_rules, prefix)`

- EXTR_OVERWRITE
- EXTR_SKIP
- EXTR_PREFIX_SAME
- EXTR_PREFIX_ALL



PHP Array : Compact Function

```
$first = "red";
```

```
$second = "green";
```

```
$third = "blue";
```



Array(

"first" => "red",

"second" => "green",

"third" => "blue",

)

compact(var1, var2, var3.....)



PHP Array : Range() Function

1 5

array(1, 2, 3, 4 ,5)

range(start, end, step)



PHP Array : Explode Function

```
$a = "Hello how are you.;"
```

```
array(
```

```
    [0] => Hello,
```

```
    [1] => how,
```

```
    [2] => are,
```

```
    [3] => you.,
```

```
);
```

explode(separator , string , limit)





PHP Array : implode Function

```
$a = ["This", "is", "an", "example"];
```

This is an example



implode(separator , array)



PHP String : Str_split() Function

```
$a = "Hello";
array(
    [0] => H,
    [1] => e,
    [2] => l,
    [3] => l,
    [4] => o,
);
```

str_split(string , length)



PHP String : Chunk_split() Function

```
$a = "Hello";
```

```
He..ll..o..
```

chunk_split(string , length , end)



- `strtolower`
- `strtoupper`
- `ucfirst`
- `ucwords`
- `lcfirst`



PHP String Length & Count Functions

`$a = "Hello World";` → 11

`strlen(string)`

`$a = "Hello World";` → 2

`str_word_count(string, return)`

`$a = "Hello World";` → World

`substr_count(string, substring, start, length)`



- `strpos`
- `strrpos`
- `stripos`
- `strripos`



PHP String : Find Position Functions

\$a = "I love php, I love php too!";

strpos(string, find, start) → 7

strrpos(string, find, start) → 19

case-sensitive

php Php



PHP String : Find Position Functions

`stripos(string, find, start)`

`strripos(string, find, start)`

case-insensitive



- strstr
- striistr
- strchr
- strrchr
- strpbrk





PHP String : Search String Functions

\$a = "I love php, I love php too!";

`strstr(string, search, before_search)` → , I love php too!

`strchr(string, search, before_search)`

`strrchr(string, search)` → too!

case-sensitive

php PHP



PHP String : Search String Functions

`stristr(string, search, before_search)`

case-insensitive

```
$a = "I love php, I love PHP too!";
```

PHP
↓



`strpbrk(string, charlist)`

```
$a = "I love php, I love PHP too!";
```

o



PHP String : Substr() Function

\$a = "I love php, I love php too!";



php, I love

substr(string, start, length)





- str_replace
- str_ireplace
- substr_replace
- strstr



PHP String : Str_replace() Function

```
$a = "I love php, I love php too!";
```

I love python, I love python too!

str_replace(find, replace, string) case-sensitive
PHP

str_ireplace(find, replace, string) case-insensitive





PHP String : Substr_replace() Function

\$a = "I love php, I love php too!";

python and

I love python and php too!

substr_replace (string, replacement, start, length)



PHP String : Strtr() Function

\$a = "I love php, I love PHP too!";

oe → ik

strtr(string, from, to)

strtr(string, array)

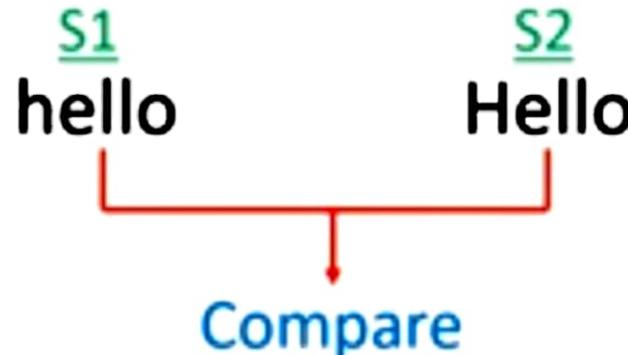


PHP String : Compare Functions

- `strcmp`
- `strncmp`
- `strcasecmp`
- `strncasecmp`
- `strnatcmp`
- `strnatcasecmp`
- `substr_compare`
- `similar_text`



PHP String : Compare



0 → Equal

<0 → S_1 is less than S_2

>0 → S_1 is Greater than S_2



PHP String : Compare Functions

hello

Hello

`strcmp(string1, string2)`

case-sensitive

`strncmp(string1, string2, length)`

`strcasecmp(string1, string2)`

case-insensitive

`strncasecmp(string1, string2 , length)`



PHP String : Sub String Compare Function

Hello world

Hello WORLD



`substr_compare(string1, string2, start, length, case)`

5

False / True



PHP String : Similar Text Function

Hello world

HEllo e^{arth}

`similar_text(string1, string2, percent)`

7



PHP String Reverse Function

hello → olleh

`strrev(string)`



PHP String Shuffle Function

hello → llohe

eohll

lheloo

`str_shuffle(string)`



PHP String Pad Function

hello → hello.....

`str_pad(string, length, pad_string, pad_type)`

- `STR_PAD_BOTH`
- `STR_PAD_LEFT`
- `STR_PAD_RIGHT`



PHP String Repeat Function

WoW → WoWWoWWoW

`str_repeat(string, repeat)`





hello → ell

ho

Yahoo Baba



`trim(string, charlist)`

`rtrim(string, charlist)`

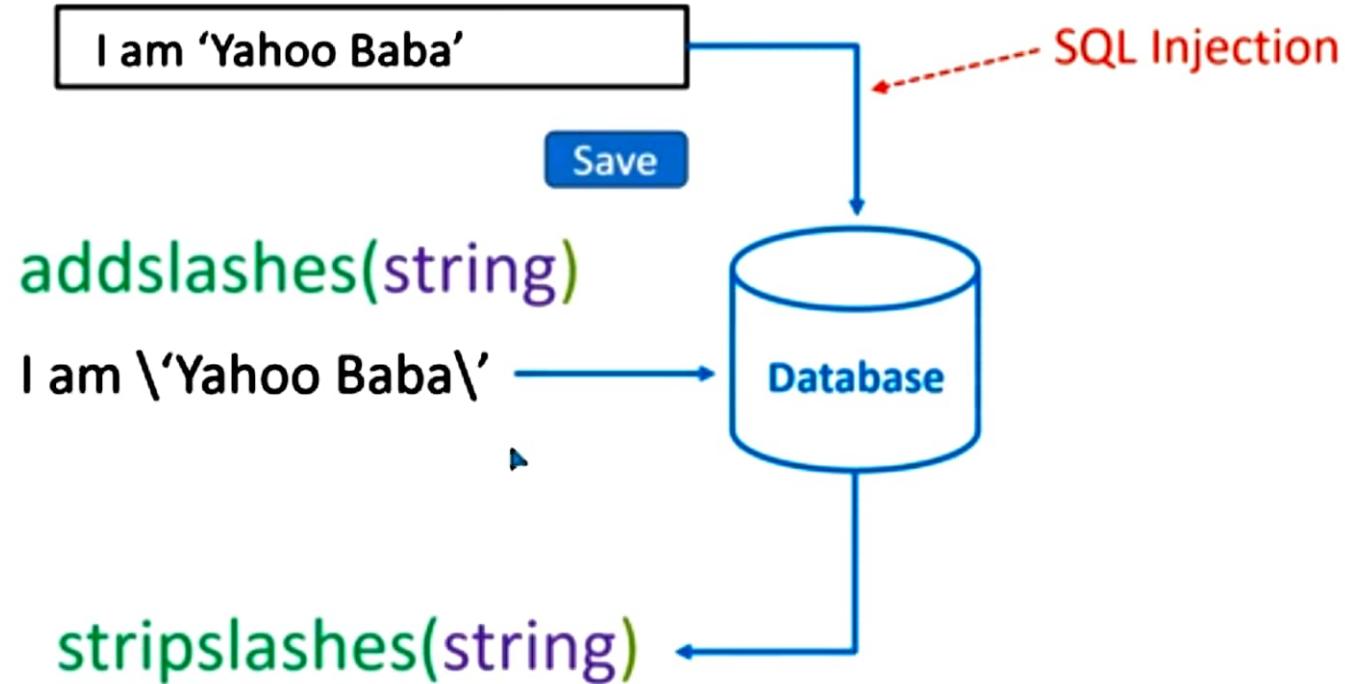
`ltrim(string, charlist)`

`chop(string, charlist)`





PHP String : addslashes & stripslashes Functions





PHP String Addcslashes & Stripcslashes Functions

I am Yahoo Baba

I am Yahoo \Baba

Save

`addcslashes(string, characters)`

`stripcslashes(string)`



PHP String : htmlentities Function

```
<a href="https://www.yahoo巴巴.net">Yahoo Baba</a>
```

----- SQL Injection

Save

htmlentities(string, flags)

```
&lt;a href=&quot;https://www.yahoo巴巴.net&quot;&gt;  
Yahoo Baba Website&lt;/a&gt;
```



html_entity_decode(string, flags)



`htmlentities(string, flags)`



- **ENT_COMPAT** Default. Encodes only double quotes
- **ENT_QUOTES** Encodes double and single quotes
- **ENT_NOQUOTES** Does not encode any quotes

`htmlspecialchars(string, flags)`

`htmlspecialchars_decode(string, flags)`





PHP String : Html Entities

& → &

" → "

' → '

< → <

> → >

- **HTML_ENTITIES**

- **HTML_SPECIALCHARS**

`get_html_translation_table(function, flags)`



PHP String : md5 Function

MD5 Message-Digest Algorithm

`md5(string, raw)`

8b1a9953c4611296a827abf8c47804d7

Password

hello

Login



raw

TRUE Raw 16 character binary format
?♦?S♦a?♦?'?♦?x?♦?

FALSE Default. 32 character hex number

UserName	Password
admin	hello
user	wow





PHP String : sha1 Function

US Secure Hash Algorithm 1

sha1(string, raw)



- **TRUE** Raw 20 character binary format
- **FALSE** Default. 40 character hex number





PHP String : convert_uuencode Function

```
$a = "Hello World";
```

uuencode algorithm

```
convert_uuencode(string)
```

```
+2&5L;&\@5V]R;&0` `
```

```
convert_uudecode(string)
```



PHP String : bin2hex & hex2bin Function

```
$a = "Hello World";
```

Convert ASCII characters to hexadecimal values

bin2hex(string)

48656c6c6f20576f726c64

hex2bin(string)





PHP String : chr() Function



Char	Dec	Oct	Hex		Char	Dec	Oct	Hex		Char	Dec	Oct	Hex
(sp)	32	0040	0x20		@	64	0100	0x40		'	96	0140	0x60
!	33	0041	0x21		A	65	0101	0x41		a	97	0141	0x61
.	34	0042	0x22		B	66	0102	0x42		b	98	0142	0x62
#	35	0043	0x23		C	67	0103	0x43		c	99	0143	0x63
\$	36	0044	0x24		D	68	0104	0x44		d	100	0144	0x64
%	37	0045	0x25		E	69	0105	0x45		e	101	0145	0x65
&	38	0046	0x26		F	70	0106	0x46		f	102	0146	0x66
,	39	0047	0x27		G	71	0107	0x47		g	103	0147	0x67
(40	0050	0x28		H	72	0110	0x48		h	104	0150	0x68
)	41	0051	0x29		I	73	0111	0x49		i	105	0151	0x69
*	42	0052	0x2a		J	74	0112	0x4a		j	106	0152	0x6a
*	43	0053	0x2b		K	75	0113	0x4b		k	107	0153	0x6b
*	44	0054	0x2c		L	76	0114	0x4c		l	108	0154	0x6c
*	45	0055	0x2d		M	77	0115	0x4d		m	109	0155	0x6d
*	46	0056	0x2e		N	78	0116	0x4e		n	110	0156	0x6e
/	47	0057	0x2f		O	79	0117	0x4f		o	111	0157	0x6f
0	48	0060	0x30		P	80	0120	0x50		p	112	0160	0x70
1	49	0061	0x31		Q	81	0121	0x51		q	113	0161	0x71
2	50	0062	0x32		R	82	0122	0x52		r	114	0162	0x72
3	51	0063	0x33		S	83	0123	0x53		s	115	0163	0x73
4	52	0064	0x34		T	84	0124	0x54		t	116	0164	0x74
5	53	0065	0x35		U	85	0125	0x55		u	117	0165	0x75
6	54	0066	0x36		V	86	0126	0x56		v	118	0166	0x76
7	55	0067	0x37		W	87	0127	0x57		w	119	0167	0x77
8	56	0070	0x38		X	88	0130	0x58		x	120	0170	0x78
9	57	0071	0x39		Y	89	0131	0x59		y	121	0171	0x79
:	58	0072	0x3a		Z	90	0132	0x5a		z	122	0172	0x7a
:	59	0073	0x3b		[91	0133	0x5b		{	123	0173	0x7b
<	60	0074	0x3c		\	92	0134	0x5c			124	0174	0x7c
=	61	0075	0x3d]	93	0135	0x5d)	125	0175	0x7d
>	62	0076	0x3e		^	94	0136	0x5e		-	126	0176	0x7e
?	63	0077	0x3f		-	95	0137	0x5f					



PHP String : ord() Function

Hello
↓
`ord(string)` → ASCII Value of First Character



PHP String : strip_tags() Function

```
$a = "Hello <b>World</b>, Hello <i>Earth</i>";
```

Hello World, Hello Earth



`strip_tags(string, allow)`



PHP String : wordwrap() Function

```
$a = "This world is beautiful.;"
```

```
This  
worl  
d is  
beau  
tifu  
l.
```

TRUE - Wrap

FALSE - No-wrap

wordwrap(string, width, break, cut)



PHP Math : min() & max() Function

2, 4, 6, 8, 10

Minimum

2

Maximum

10

`min(value1, value2,...)`

`max(value1, value2,...)`

`min(array_values)`

`max(array_values)`





abs()

-6.2 6.2



6.2

floor()

4.3 4.8



4

ceil()

4.3 4.8



5

round()

4.3 4.8



4



5





PHP Math : intdiv() Function

$$\begin{array}{r} 2 \\ \hline 4 | 8 \\ 8 \\ \hline 0 \end{array}$$

intdiv(dividend, divisor)



PHP Math : Pow() Function

pow(base, exp)

(4)³

$$4 \times 4 \times 4 = 12$$





PHP Math : Sqrt() Function

$$\sqrt{4} = 2$$

`sqrt(number)`



PHP Math : Random Value Functions

`rand(min, max)`

10 20



`lcg_value()`

0 1

Mersenne Twister algorithm

`mt_rand(min, max)`

4 Times faster than `rand()` function.





PHP String : date() Function

25/03/2020

date(format)

Day

d (01 to 31)

j (1 to 31)

S (st, nd, rd or th)

Month

F (January)

m (01)

M (Jan)

Year

Y (2019)

y (19)

n (1)

Week

D (Mon)

I (Monday)

N (1)

w (1)





02:30:27pm

date(format)

Hour

h (01 to 12)
H (00 to 23)

Minutes

i (00 to 59)

Seconds

s (00 to 59)

Meridiem

a (am or pm)
A (AM or PM)

g (1 to 12)

G (0 to 23)



PHP : Date & Time Functions

`date()` → Current Date & Time

`mktime(hour, minute, second, month, day, year)`

`gmmktime(hour, minute, second, month, day, year)`

↓
GMT date (Greenwich Mean Time)



`mktime() → timestamp`

`date_create(time, timezone)`



`date_format(object, format)`



PHP : Check Date Function

15-02-2013

35-02-2013

`checkdate(month, day, year)`



PHP : Date Difference Function

2013-03-15

2013-12-12

272 Days

`date_diff(datetime1, datetime2, absolute)`

TRUE

FALSE





PHP : Date Add & Sub Function

15-03-2013 + 30 Days = 14-04-2013

`date_add(object, interval)`

15-03-2013 - 10 Days = 05-03-2013

`date_sub(object, interval)`

`date_modify(object, modify)`



getdate(timestamp)

- [seconds] - seconds
- [minutes] - minutes
- [hours] - hours
- [mday] - day of the month
- [wday] - day of the week (0=Sunday, 1=Monday,...)
- [mon] - month
- [year] - year
- [yday] - day of the year
- [weekday] - name of the weekday
- [month] - name of the month
- [0] - seconds since Unix Epoch



`gettimeofday(return_float)`



`TRUE` Float value of Time

`FALSE` Return Array

- `[sec]` - seconds since the Unix Epoch
- `[usec]` - microseconds
- `[minuteswest]` - minutes west of Greenwich
- `[dsttime]` - type of dst correction



localtime(timestamp, is_assoc)

↓
time()

↓
TRUE Associative array
FALSE Indexed array

- [tm_sec] - seconds
- [tm_min] - minutes
- [tm_hour] - hour
- [tm_mday] - day of the month
- [tm_mon] - month of the year (January=0)
- [tm_year] - Years since 1900
- [tm_wday] - Day of the week (Sunday=0)
- [tm_yday] - Day of the year
- [tm_isdst] - Is daylight savings time in effect



date_parse_from_format(format, date)

- [seconds] - seconds
- [minutes] - minutes
- [hours] - hours
- [mon] - month
- [year] - year
- [month] - name of the month
- [day] - day of the month



date_parse(date)



- [seconds] - seconds
- [minutes] - minutes
- [hours] - hours
- [mon] - month
- [year] - year
- [month] - name of the month
- [day] - day of the month



now

next Monday

+1 week

First day of last month

strtotime(time)



strftime(format, timestamp)

%a - abbreviated weekday name

%A - full weekday name

%b - abbreviated month name

%B - full month name

%c - preferred date and time representation

%C - century number (the year divided by 100, range 00 to 99)

%d - day of the month (01 to 31)

%D - same as %m/%d/%y

%e - day of the month (1 to 31)

%g - like %G, but without the century

%G - 4-digit year corresponding to the ISO week number (see %V).

%h - same as %b

%H - hour, using a 24-hour clock (00 to 23)



PHP : String Format Time Function

%I - hour, using a 12-hour clock (01 to 12)

%j - day of the year (001 to 366)

%m - month (01 to 12)

%M - minute

%n - newline character

%p - either am or pm according to the given time value

%r - time in a.m. and p.m. notation

%R - time in 24 hour notation

%S - second

%t - tab character

%T - current time, equal to %H:%M:%S

%u - weekday as a number (1 to 7), Monday=1. Warning: In Sun Solaris Sunday=1

%U - week number of the current year, starting with the first Sunday as the first day of the first week





PHP : String Format Time Function

%I - hour, using a 12-hour clock (01 to 12)

%j - day of the year (001 to 366)

%m - month (01 to 12)

%M - minute

%n - newline character

%p - either am or pm according to the given time value

%r - time in a.m. and p.m. notation

%R - time in 24 hour notation

%S - second

%t - tab character

%T - current time, equal to %H:%M:%S

%u - weekday as a number (1 to 7), Monday=1. Warning: In Sun Solaris Sunday=1

%U - week number of the current year, starting with the first Sunday as the first day of the first week



PHP : String Format Time Function

%V - The ISO 8601 week number of the current year (01 to 53), where week 1 is the first week that has at least 4 days in the current year, and with Monday as the first day of the week

%W - week number of the current year, starting with the first Monday as the first day of the first week

%w - day of the week as a decimal, Sunday=0

%x - preferred date representation without the time

%X - preferred time representation without the date

%y - year without a century (range 00 to 99)

%Y - year including the century

%Z or %z - time zone or name or abbreviation

%% - a literal % character



PHP : Date Time Set Function

Date

2015-05-15

Time

3:30

`date_time_set(object, hour, minute, second, microseconds)`



- `date_default_timezone_get()`
- `date_default_timezone_set()`
- `timezone_open()`
- `timezone_name_get()`
- `timezone_location_get()`
- `timezone_identifiers_list()`



PHP : Include & Require Statements

header.php

Company Name

Home | About Us | Products

Welcome to our website

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam eget semper mi, sit amet egestas augue. Aliquam semper suscipit orci, sit amet accumsan ex lacinia dapibus. Quisque convallis sit amet ligula non pretium. Aliquam vel dignissim nunc.

- Conditions
- Our Policy
- Contact Us

sidebar.php

footer.php

@copyright companyname.





PHP : Include & Require Statements

include 'filename'

require 'filename'

header.php

Welcome to our website

 Lorem ipsum dolor sit amet,
 consectetur adipiscing elit. Nullam
 egest semper mi, sit amet egestas
 augue. Aliquam semper suscipit orci,
 sit amet accumsan ex lacinia
 dapibus. Quisque convallis sit amet
 ligula non pretium. Aliquam vel
 dignissim nunc.

sidebar.php

footer.php

home.php



PHP : Include_once & Require_once Statements

header.php

include_once

require_once

footer.php

Company Name

Home | About Us | Products

Welcome to our website

Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Nullam
eget semper mi, sit amet egestas
augue. Aliquam semper suscipit orci,
sit amet accumsan ex lacinia
dapibus. Quisque convallis sit amet
ligula non pretium. Aliquam vel
dignissim nunc.

- Conditions
- Our Policy
- Contact Us

sidebar.php





- `$_GET`
- `$_POST`
- `$_REQUEST`
- `$_SERVER`
- `$_SESSION`
- `$_COOKIE`
- `$_FILES`



PHP : Super Global Variables ?

```
$a = 10;
```

File1.php

```
echo $a;
```

File2.php

Super Global Variables

```
  form.php • testform.php
2 <html>
3   <head>
4     <title>Form Page</title>
5   </head>
6   <body>
7     <form action="testform.php" method="get">
8
9       Name : <input type="text" name="fname"><br><br>
10
11      Age : <input type="text" name="age"><br><br>
12
13      <input type="submit" name="save">
14
15    </form>
16  </body>
17 </html>
```

2

3

4

5

6

7

8

9

10

11

12

13

localhost/LearnPHP/testform.php?name=Yahoo+Baba&age=30&save=Submit



```
1 <?php  
2 echo "<pre>";  
3 print_r($_GET);  
4 echo "</pre>";  
5  
6 echo $_GET['fname'];|  
7  
8  
9  
10
```





- HTTP Connection
- SERVER Information
- HOST Information
- URL Information



PHP : \$_SERVER

Name

Age

Gender Male Female

Submit

File1.php

- Database
- Print
- Conditional Page Content

```
1 <?php
2 echo "<pre>";
3 print_r($_GET);
4 echo "</pre>";
5
6 echo "<pre>";
7 print_r($_SERVER);
8 echo "</pre>";
9
10 echo $_SERVER['PHP_SELF'];
11 echo $_SERVER['HTTP_HOST'];
12
13
14 ?>
```

```
form.php • testform.php
2 <html>
3   <head>
4     <title>Form Page</title>
5   </head>
6   <body>
7     <form action="<?php echo $_SERVER['PHP_SELF'] ?>" method="get">
8
9       Name : <input type="text" name="fname"><br><br>
10
11      Age : <input type="text" name="age"><br><br>
12
13      <input type="submit" name="save">
14
15    </form>
16  </body>
17 </html>
```

```
form.php
```

```
8
9     Name : <input type="text" name="fname"><br><br>
10
11    Age : <input type="text" name="age"><br><br>
12
13    <input type="submit" name="save">
14
15  </form>
16
17  <?php
18      if(isset($_POST['save'])){
19          echo $_POST['fname'] . "<br>";
20          echo $_POST['age'] . "<br>";
21      }
22
23  ?>
```



Company Name

[Home](#) | [About Us](#) | [Products](#)

Welcome to our website

Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Nullam
egestas semper mi, sit amet egestas
augue. Aliquam semper suscipit orci,
sit amet accumsan ex lacinia
dapibus. Quisque convallis sit amet
ligula non pretium. Aliquam vel
dignissim nunc.

- [Conditions](#)
- [Our Policy](#)
- [Contact Us](#)

@copyright companyname.

COOKIES



Temporary information in visitor computer.





1) Create Cookie

```
setcookie(name, value, expire, path, domain, secure, httponly)
```

www.abc.com → test.abc.com

/Some-path

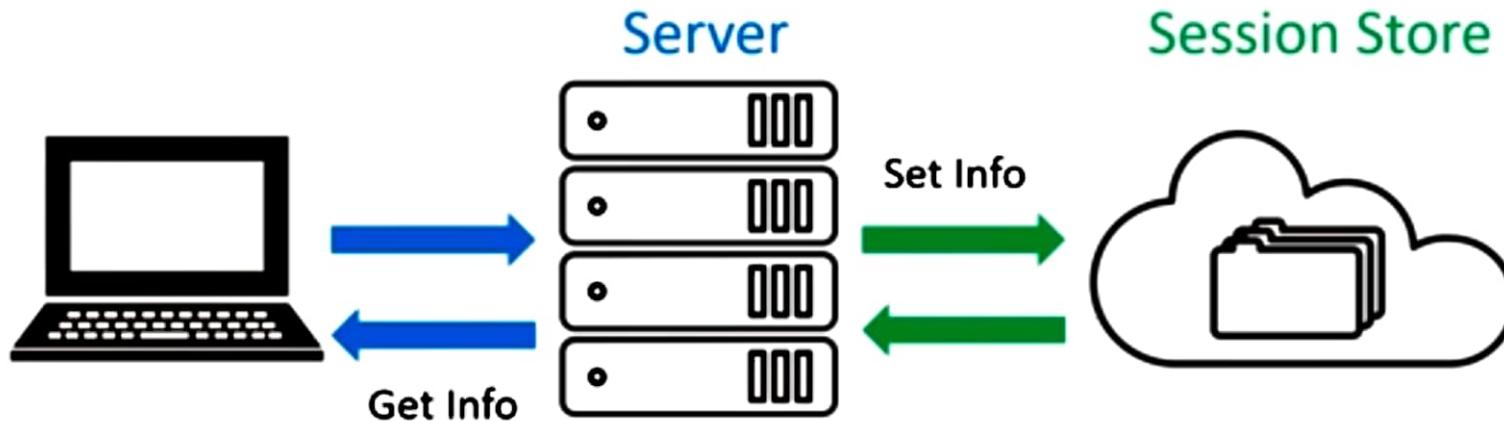
True / false

2) View Cookie Value

```
$_COOKIE[name]
```



PHP : \$_SESSION





PHP : Three Steps to Set & Get SESSION Value

Step 1 : `session_start();`

Step 2 : `$_SESSION[name] = value;` Set Session name & value

Step 3 : `echo $_SESSION[name];` Get Session value

*

Delete Session

Step 1 : `session_unset();` Remove all session variables

Step 2 : `session_destroy();` Destroy the session



Choose File

<input type="file" name="image" />

`$_FILES['image']`



- name
- size
- tmp_name
- type JPG / PNG / GIF

`move_uploaded_file(file, dest)`

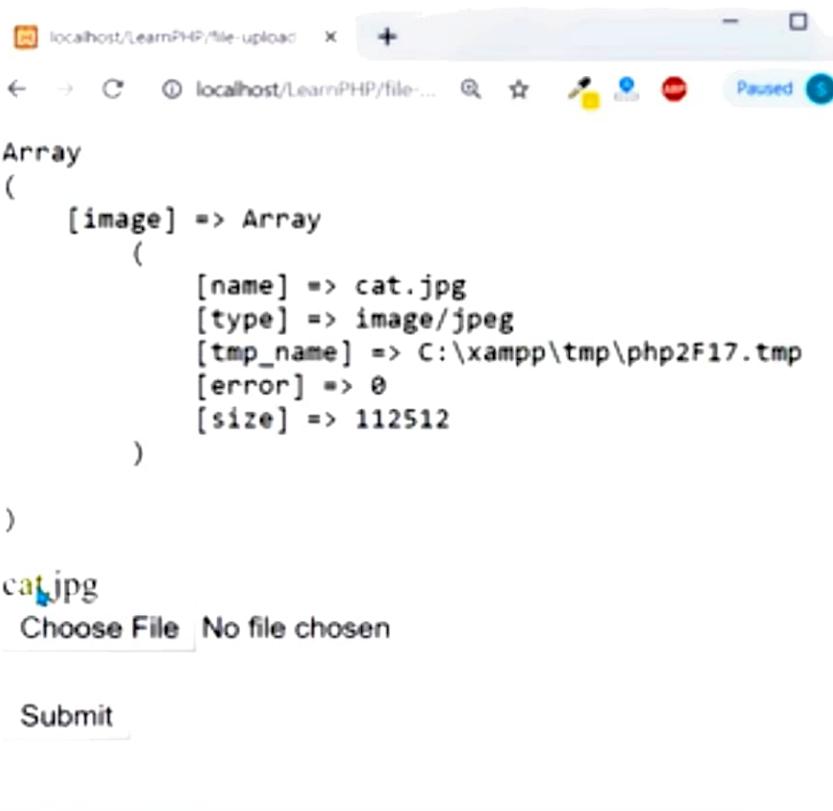


```
1
2 <html>
3   <body>
4     <form action="" method="POST" enctype="multipart/form-data">
5       <input type="file" name="image" /><br><br>
6       <input type="submit"/>
7     </form>
8   </body>
9 </html>
10
```

```
1 <?php
2 if(isset($_FILES['image'])){
3     echo "<pre>";
4     print_r($_FILES);
5     echo "</pre>";
6 }
7
8 ?>
9 <html>
10 <body>
11     <form action="" method="POST" enctype="multipart/form-data">
12         <input type="file" name="image" /><br><br>
13         <input type="submit"/>
14     </form>
15     </body>
16 </html>
```

```
1 <?php
2 if(isset($_FILES['image'])){
3     echo "<pre>";
4     print_r($_FILES);
5     echo "</pre>";
6
7     echo $file_name = $_FILES['image']['name'];
8 }
9
10 ?>
11 <html>
12     <body>
13         <form action="" method="POST" enctype="multipart/form-data">
14             <input type="file" name="image" /><br><br>
15             <input type="submit"/>
16         </form>
17     </body>
```

```
File-upload.php
1 <?php
2 if(isset($_FILES['image'])){
3     echo "<pre>";
4     print_r($_FILES);
5     echo "</pre>";
6
7     echo $file_name = $_FILES['image'][
8 }
9
10?>
11<html>
12    <body>
13        <form action="" method="POST" en
14            <input type="file" name="image" /><br><br>
15            <input type="submit"/>
16        </form>
17    </body>
```



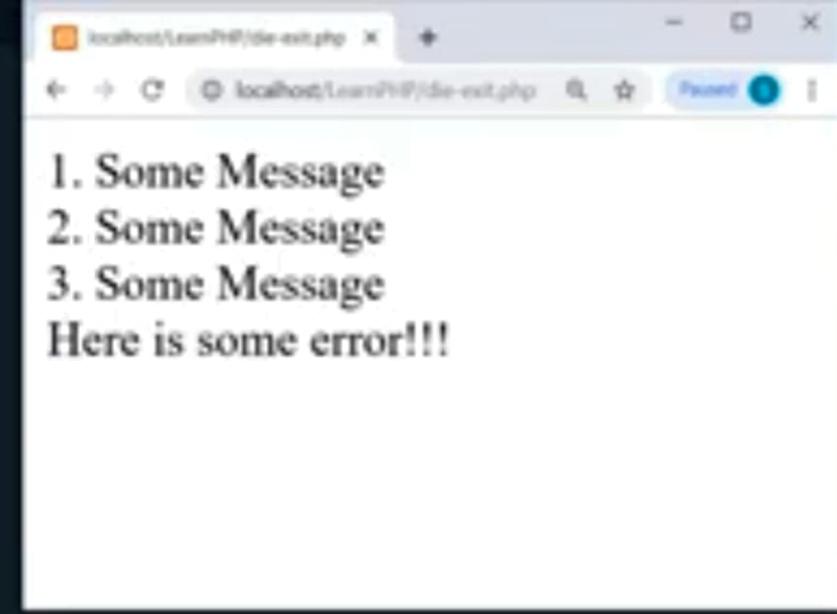
```
1 <?php  
2  
3 echo "1. Some Message <br>";  
4 echo "2. Some Message <br>";  
5 echo "3. Some Message <br>";  
6 die();  
7 echo "4. Some Message <br>";  
8 echo "5. Some Message <br>";  
9  
10  
11 ?>
```

```
die-exit.php  
1 <?php  
2  
3 echo "1. Some Message <br>";  
4 echo "2. Some Message <br>";  
5 echo "3. Some Message <br>";  
6 die("Here is some error!!!");  
7 echo "4. Some Message <br>";  
8 echo "5. Some Message <br>";  
9  
10 ?>  
11
```

localhost/LearnPHP/die-exit.php

1. Some Message
2. Some Message
3. Some Message
Here is some error!!!

```
die-exit.php  
1 <?php  
2  
3 echo "1. Some Message <br>";  
4 echo "2. Some Message <br>";  
5 echo "3. Some Message <br>";  
6 exit("Here is some error!!!");  
7 echo "4. Some Message <br>";  
8 echo "5. Some Message <br>";  
9  
10 ?>  
11
```





PHP MySQLi Fetch Functions :

- `mysqli_fetch_assoc()`
- `mysqli_fetch_array()`
- `mysqli_fetch_row()`
- `mysqli_fetch_all()`
- `mysqli_fetch_field()`

```
mysqli-affected-rows.php
1 <?php
2
3 $conn = mysqli_connect("localhost","root","","test") or die("Connection Failed");
4
5 $sql = "SELECT * FROM students";
6 $result = mysqli_query($conn, $sql) or die("Query Failed");
7
8 echo "Total Rows : " . mysqli_affected_rows($conn);
9
10
11 ?>
12
```



PHP MySQLi Error Functions :

- `mysqli_connect_error()`
- `mysqli_connect_errno()`
- `mysqli_error()`
- `mysqli_error_list()`



PHP Is Functions :

- `is_int`
- `is_integer`
- `is_long`
- `is_float`
- `is_double`
- `is_real`
- `is_numeric`
- `is_string`
- `is_bool`
- `is_array`
- `is_countable`
- `is_null`
- `is_callable`
- `is_scalar`





PHP Filter Validation :

Name	<input type="text" value="Ram Kumar"/>	Complex Validations	Filters
Age	<input type="text" value="25"/>	→ 25.50	
Email	<input type="text" value="ram@example.com"/>	→ ram@example/com	
Website	<input type="text" value="www.ram.com"/>	→ www.ra m.com	
<input type="button" value="Submit"/>			



PHP Filter Validation :

`filter_var(var, filtername, options/flag)`



- FILTER_VALIDATE_INT
- FILTER_VALIDATE_FLOAT
- FILTER_VALIDATE_BOOLEAN
- FILTER_VALIDATE_EMAIL
- FILTER_VALIDATE_URL
- FILTER_VALIDATE_IP
- FILTER_VALIDATE_MAC
- FILTER_VALIDATE_REGEXP





PHP Filter Validation :

Sanitize Filters	
Name	Ram Kumar
Age	25
Email	ram@example.com
Website	www.ram.com

Submit

```
graph LR; A[Name: Ram Kumar] --> B[Age: 25]; A --> C[Email: ram@example.com]; A --> D[Website: www.ram.com]; B --> E[Age: 2.5]; C --> F[Email: ram@example/com]; D --> G[Website: www.ra m.com]
```





PHP Filter Sanitization :

`filter_var(var, filtername, flag)`



- FILTER_SANITIZE_EMAIL
- FILTER_SANITIZE_URL
- FILTER_SANITIZE_NUMBER_INT
- FILTER_SANITIZE_NUMBER_FLOAT
- FILTER_SANITIZE_MAGIC_QUOTES
- FILTER_SANITIZE_STRING
- FILTER_SANITIZE_ENCODED
- FILTER_SANITIZE_SPECIAL_CHARS



`filter_var_array(data array , filter array)`

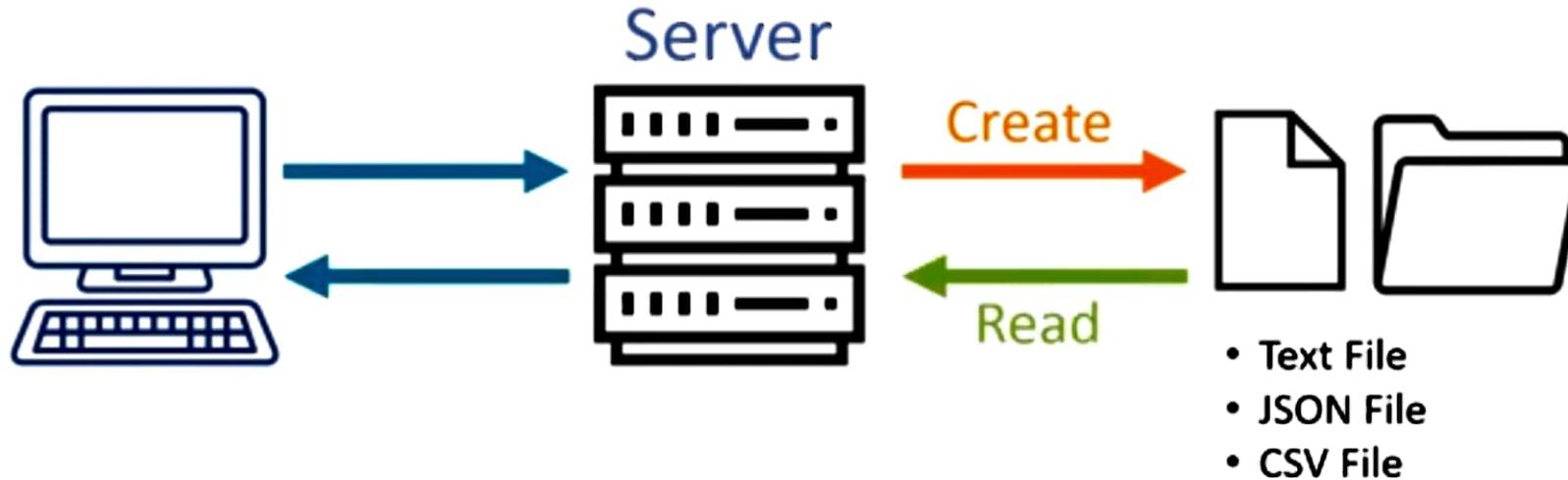


`filter_input(type, variable, filter, options)`



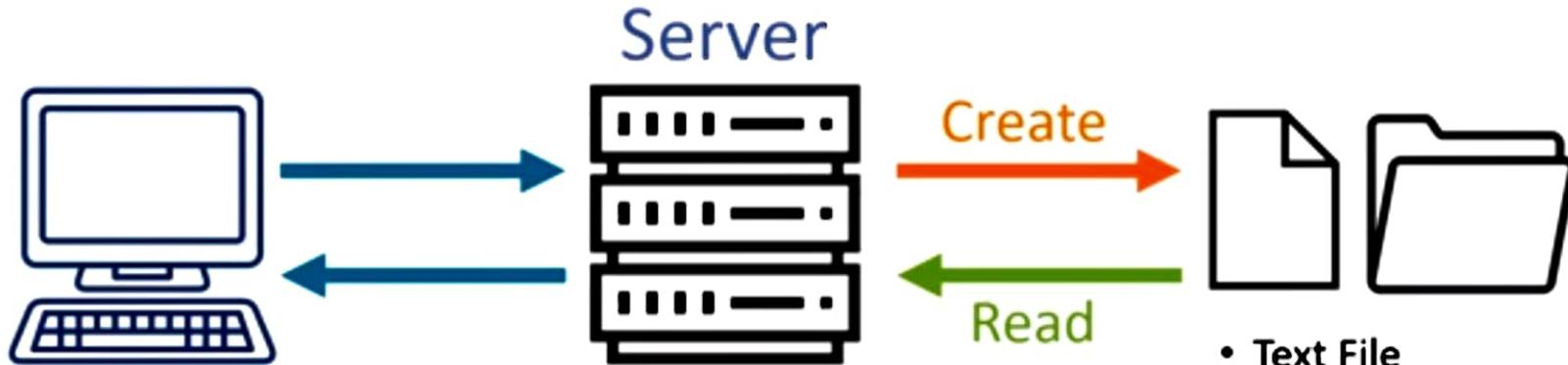


PHP : File System





PHP : File System



File System Functions

- Create Files & Folders
- Read Files
- Get File Information
- Change Permission

- Text File
- JSON File
- CSV File





PHP : File System Functions

- `readfile()`
- `file_exists()`
- `copy()`
- `rename()`
- `mkdir()`
- `rmdir()`
- `delete()`
- `unlink()`
- `filesize()`
- `filetype()`
- `realpath()`
- `pathinfo()`
- `dirname()`
- `basename()`





PHP : File System Functions

- **fopen()**
- **fread()**
- **fgets()**
- **ftell()**
- **fseek()**
- **fpassthru()**
- **rewind()**
- **feof()**
- **file()**
- **fgetc()**
- **fwrite()**
- **fputs()**
- **fclose()**
- **ftruncate()**



- `is_dir()`
- `is_file()`
- `is_readable()`
- `is_writable()`
- `is_writeable()`
- `is_executable()`



PHP : File Permissions



Files



Folder

3 Types of Permissions

- Read
- Write
- Execute

`Chmod(file name, mode)`

Works only with Linux / Unix / Mac OS

`fileperms()`

Works properly with Linux / Unix / Mac OS





PHP : Write in the File



Files

← Data Write

`fopen(file name, mode)`

`fwrite(file, string data)`

`fclose(file)`

`file_put_contents(filename, data, mode, context)`

- `FILE_APPEND`
- `LOCK_EX`



PHP : Read from the File



Read Data

Files

`fopen(file name, mode)`

`fread(file, length)`

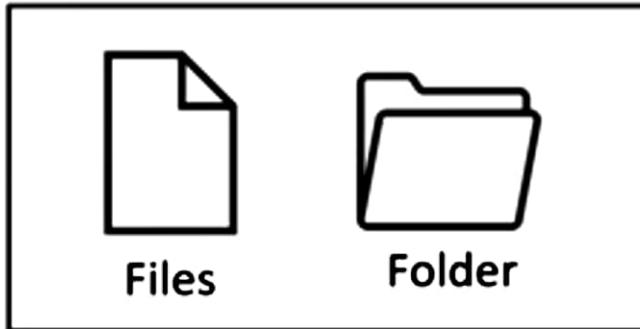
`fclose(file)`

`file_get_contents(filename, include_path, context, start, max_length)`

↓
FALSE



PHP : File System Searching



- Search Files & Folders
- Search in subfolders
- Particular Format files
- Search with Regular Expressions



Sa*

Sa

glob(pattern, flags)

- GLOB_MARK
- GLOB_NOSORT
- GLOB_NOCHECK
- GLOB_BRACE
- GLOB_ONLYDIR





PHP : Directory Functions



Folder



Change Folder



Folder 2

- `getcwd()`
- `chdir()`
- `scandir()`



Files



Scan folder



File Handling Functions

- `fopen()`
- `fread()`
- `fclose()`

Directory Functions

- `opendir()`
- `readdir()`
- `closedir()`



What is Regular Expressions ?

Regular Expressions



Technique for parsing and extracting data from text.

JavaScript, Python, PHP, Java, Ruby, Perl etc.

Regex Regexp Rational Expression



- preg_match()
- preg_match_all()

`preg_match(pattern, string, array)`



/pattern/modifiers;



PHP : Functions for Regular Expressions

`preg_replace(pattern, replacement, string, limit)`

`preg_split(pattern, string, limit, flags)`



PHP : How to Send Email

`mail(to, subject, message, headers)`

`abc@email.com`

Receiver

`From Sender`

`Cc Carbon Copy`

`Bcc Blind Carbon Copy`





PHP & MySQL Programming Steps :

1) Connection

```
$conn = new mysqli(Server Name, User Name, Password, Database Name)
```

2) Run SQL Query

```
$conn->query(SQL Query) ← SQL Injection
```



3) Close Connection

```
$conn->close()
```



SQL Injection

Username

ram

Password

' or '='

Login

```
$username = $_POST["username"];
```

```
$password = $_POST["password"];
```

```
$sql = "SELECT * FROM users WHERE user = '$username' AND pass = '$password'";
```

```
SELECT * FROM users WHERE user = 'ram' AND pass = '' or '' = ''
```



SQL Injection

Username

ram

Password

' or '='

Login

```
$username = $_POST["username"];
```

```
$password = $_POST["password"];
```

```
$username = $conn->real_escape_string($_POST["username"]);
```

```
$password = $conn->real_escape_string($_POST["password"]);
```

```
$sql = "SELECT * FROM users WHERE user = '$username' AND pass = '$password'";
```

```
SELECT * FROM users WHERE user = 'ram' AND pass = '' or ''=''
```



PHP & MySQL Programming Steps :

1) Connection

```
$conn = new mysqli(Server Name, User Name, Password, Database Name)
```

2) Run SQL Query

```
$sql = $conn->prepare("SELECT * FROM users WHERE user = ? AND pass = ?");  
$sql->bind_param("ss", $username, $password);  
$sql->execute();
```

3) Close Connection

```
$conn->close()
```



Fetch Functions with Prepare() Statement

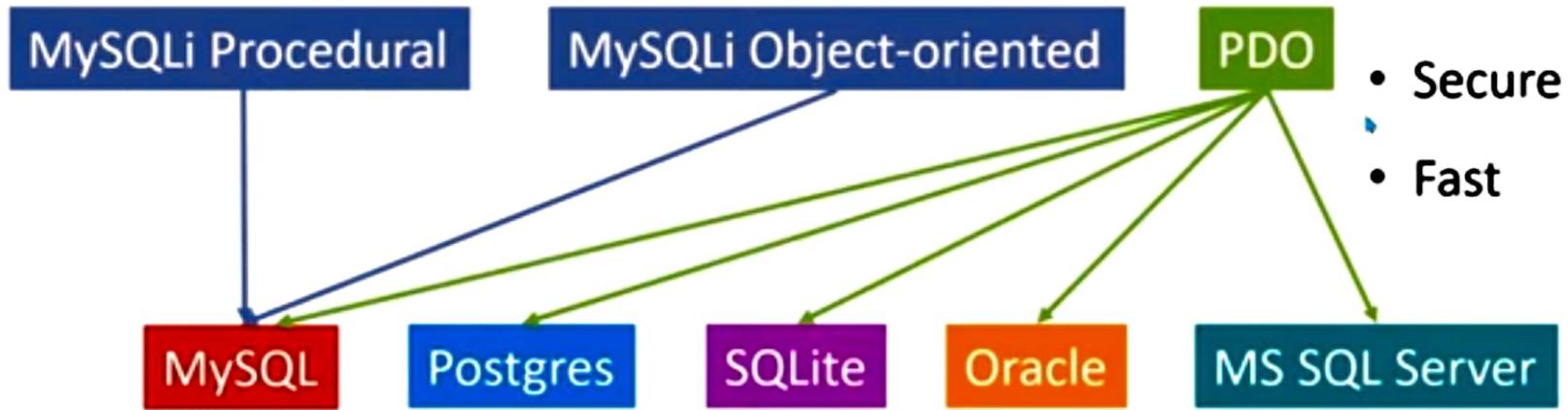
- `fetch()`
- `fetch_assoc()`
- `fetch_row()`
- `fetch_object()`
- `fetch_all(MYSQLI_ASSOC)`
- `fetch_all(MYSQLI_NUM)`
- `get_result()`
- `bind_result()`



- MySQLi Procedural
- MySQLi Object-oriented
- PDO ([PHP Data Objects](#))



Difference between these Methods :





PHP & MySQL Programming Steps :

1) Connection

```
$db_name = "mysql:host=localhost;dbname=Database";  
$conn = new PDO($db_name, User Name, Password);
```

2) Run SQL Query

```
$conn->query(SQL Query)
```

or

```
$conn->prepare(SQL Query)
```

3) Close Connection

```
$conn = null;
```





PHP & MySQL Programming Steps :

1) Connection

```
$db_name = "mysql:host=localhost;dbname=Database";  
$conn = new PDO($db_name, User Name, Password);
```

2) Run SQL Query

```
$conn->prepare("SELECT * FROM users WHERE user = ? AND pass = ?")  
$sql->bindParam(1, $username);  
$sql->bindParam(2, $password);  
$sql->execute();  
$conn = null;
```



PHP PDO Advance Fetch Styles

- PDO::FETCH_COLUMN
- PDO::FETCH_GROUP
- PDO::FETCH_UNIQUE
- PDO::FETCH_KEY_PAIR
- PDO::FETCH_CLASS

PDO Methods

- rowCount()
- exec()
- lastInsertId()

`$sql->fetchAll(PDO::FETCH_COLUMN)`



What is Exception Handling in PHP ?

```
if(condition){  
    // When condition is TRUE  
}else{  
    // When condition is FALSE  
}
```

```
try{  
    // Some Code Goes Here  
}catch(){  
    // When any error comes in try code  
}
```





PHP Exception Handling Syntax

```
try{  
    if(condition){  
  
    }else{  
        throw new Exception("Some Message Goes Here");  
    }  
}  
catch(Exception $e){  
    echo $e->getMessage();  
}
```

\$e->**getLine()**;

\$e->**getCode()**;

\$e->**getFile()**;