1. Addslashes : The addslashes() function returns a string with backslashes in front of predefined characters.

The predefined characters are:

* single quote (')
* double quote (")
* backslash (\)
* NULL

<?php   
$str = addslashes('What does "yolo" mean?');  
echo($str);   
?>

1. explode : The explode() function breaks a string into an array.

<?php  
$str = "Hello world. It's a beautiful day.";  
print\_r (explode(" ",$str));

Out Put : Array ( [0] => Hello [1] => world. [2] => It's [3] => a [4] => beautiful [5] => day. );

?>

1. echo : The echo() function outputs one or more strings.
2. htmlentities : The htmlentities() function converts characters to HTML entities.

<?php  
$str = "<© W3Sçh°°¦§>";  
echo htmlentities($str);

Out put : &lt;&copy; W3S&ccedil;h&deg;&deg;&brvbar;&sect;&gt;  
?>

1. htmlspecialchars : Convert the predefined characters "<" (less than) and ">" (greater than) to HTML entities

<?php  
$str = "This is some <b>bold</b> text.";  
echo htmlspecialchars($str);

Out Put : This is some &lt;b&gt;bold&lt;/b&gt; text.  
?>

1. implode : Join array elements with a string

<?php  
$arr = array('Hello','World!','Beautiful','Day!');  
echo implode(" ",$arr);

Out Put : Hello World! Beautiful Day !  
?>

1. ltrim : Remove characters from the left side of a string

<?php  
$str = "Hello World!";  
echo $str . "<br>";  
echo ltrim($str,"Hello");

Out Put : World!  
?>

8) stripos : Returns the position of the first occurrence of a string inside another string (case-insensitive)

<?php  
echo stripos("I love php, I love php too!","PHP");

Out Put : 7  
?>

1. [strpos()](http://www.w3schools.com/php/func_string_strpos.asp) - Finds the position of the first occurrence of a string inside another string (case-sensitive)

<?php  
echo strpos("I love php, I love php too!","php");

Out Put : 7  
?>

1. strlen : Return the length of the string "Hello"

<?php  
echo strlen("Hello");

Out Put : 5

?>

1. strrev: Reverses a string

<?php  
echo strrev("Hello World!");

!dlroW olleH  
?>

1. substr() : Returns a part of a string

<?php

echo substr("Hello world",6);

Out Put : world

?>

1. ucfirst() : Converts the first character of a string to uppercase

<?php

echo ucfirst("hello world!");

Out put : Hello world!

?>

1. ucwords: Convert the first character of each word to uppercase

<?php

echo ucwords("hello world");

Out Put : Hello World

?>

1. strtoupper : Convert all characters to uppercase:

<?php

echo strtoupper("Hello WORLD!");

Out Put : HELLO WORLD!

?>

1. strtolower() : converts a string to lowercase

<?php

echo strtolower("Hello WORLD.");

Out put : hello world.

?>

16) The split() function splits the string into an array using a regular expression and returns an array.

<?php print\_r(str\_split("Hello")); ?>

OUTPUT :-

Array  
(  
[0] => H  
[1] => e  
[2] => l  
[3] => l  
[4] => o  
)

17) filter\_var() : The following example uses the filter\_var() function to remove all HTML tags from a string:

<?php  
$str = "<h1>Hello<b> World!</b></h1>";  
$newstr = filter\_var($str, FILTER\_SANITIZE\_STRING);

echo $newstr;  
Out Put : Hello World!

?>

**ARRAY FUNCTION :**

1. array\_column() : Returns the values from a single column in the input array
2. array\_combine(): Creates an array it takes elements from one array as keys and takes another array elements as values

<?php

$fname=array("Peter","Ben","Joe");

$age=array("35","37","43");

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

print\_r($c);

Out Put : Array ( [Peter] => 35 [Ben] => 37 [Joe] => 43 )

?>

1. array\_count\_values(): Counts all the values of an array

<?php

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

print\_r(array\_count\_values($a));

Array ( [A] => 2 [Cat] => 1 [Dog] => 2 );

?>

1. Compare arrays, and returns the differences (compare values only)

<?php

$a1=array("a"=>"red","b"=>"green","c"=>"blue","d"=>"yellow");

$a2=array("e"=>"red","f"=>"green","g"=>"blue");

$result=array\_diff($a1,$a2);

print\_r($result);

Out Put : Array ( [d] => yellow )

?>

1. array\_merge() Merges one or more arrays into one array

<?php

$a1=array("red","green");

$a2=array("blue","yellow");

print\_r(array\_merge($a1,$a2));

Out Put: Array ( [0] => red [1] => green [2] => blue [3] => yellow );

?>

1. array\_pad() Inserts a specified number of items, with a specified value, to an array

<?php

$a=array("red","green");

print\_r(array\_pad($a,5,"blue"));

Out Put : Array ( [0] => red [1] => green [2] => blue [3] => blue [4] => blue );

?>

1. array\_push() Inserts one or more elements to the end of an array

<?php

$a=array("red","green");

array\_push($a,"blue","yellow");

print\_r($a);

Out Put : Array ( [0] => red [1] => green [2] => blue [3] => yellow )

?>

1. array\_reverse() Returns an array in the reverse order

<?php

$a=array("a"=>"Volvo","b"=>"BMW","c"=>"Toyota");

print\_r(array\_reverse($a));

Out Put : Array ( [c] => Toyota [b] => BMW [a] => Volvo );

?>

1. \*array\_search(): Searches an array for a given value and returns the key

<?php

$a=array("a"=>"red","b"=>"green","c"=>"blue");

echo array\_search("red",$a);

Out Put : a

$b=array("red","green","blue");

echo array\_search("green",$b);

Out put : 1 (array position )

?>

1. \*array\_shift() Removes the first element from an array, and returns the value of the removed element

<?php

$a=array("a"=>"red","b"=>"green","c"=>"blue");

echo array\_shift($a)."<br>";

print\_r ($a);

Out Put : red

Array ( [b] => green [c] => blue )

?>

1. array\_slice() Returns selected parts of an array

<?php

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

print\_r(array\_slice($a,2));

Out Put :

Array ( [0] => blue [1] => yellow [2] => brown );

?>

1. array\_sum() Returns the sum of the values in an array

<?php

$a=array(5,15,25);

echo array\_sum($a);

Out Put : 45

?>

1. array\_unique() Removes duplicate values from an array

<?php

$a=array("a"=>"red","b"=>"green","c"=>"red");

print\_r(array\_unique($a));

Out Put: Array ( [a] => red [b] => green );

?>

1. array\_values(): Returns all the values of an array

<?php

$a=array("Name"=>"Peter","Age"=>"41","Country"=>"USA");

print\_r(array\_values($a));

Array ( [0] => Peter [1] => 41 [2] => USA )

?>

1. in\_array— Checks if a value exists in an array

<?php

$os = array("Mac", "NT", "Irix", "Linux");

if (in\_array("Irix", $os)) {

echo "Got Irix";

}

1. key — Fetch a key from an array
2. krsort — Sort an array by key in reverse order
3. ksort — Sort an array by key
4. list — Assign variables as if they were an array
5. range — Create an array containing a range of elements
6. reset — Set the internal pointer of an array to its first element
7. rsort — Sort an array in reverse order
8. shuffle — Shuffle an array
9. sizeof — Alias of count
10. sort — Sort an array
11. uasort — Sort an array with a user-defined comparison function and maintain index association
12. uksort — Sort an array by keys using a user-defined comparison function
13. usort — Sort an array by values using a user-defined comparison function

PHP Error Handling :

Die , Exit, Try catch

$\_SERVER :

$\_SERVER['PHP\_SELF'] :- Returns the filename of the currently executing script

$\_SERVER['SERVER\_ADDR'] :- Returns the IP address of the host server

$\_SERVER['REMOTE\_ADDR'] :- Returns the IP address from where the user is viewing the current page

$\_SERVER['SERVER\_NAME'] :- Returns the name of the host server (such as www.w3schools.com)

$\_SERVER['SERVER\_SOFTWARE'] :- Returns the server identification string (such as Apache/2.2.24)

$\_SERVER['REQUEST\_METHOD'] :- Returns the request method used to access the page (such as POST)

$\_SERVER['HTTP\_HOST'] :- Returns the Host header from the current request

$\_SERVER['HTTP\_REFERER'] :- Returns the complete URL of the current page (not reliable because not all user-agents support it)

$\_SERVER['REMOTE\_HOST'] :- Returns the Host name from where the user is viewing the current page

$\_SERVER['SERVER\_PORT'] :- Returns the port on the server machine being used by the web server for communication (such as 80)