PHP Basics

Prof. Sachin Ponde

Escaping Special Characters

- If you enclose one set of quotation marks within another, PHP will get confused about which quotation marks are to be printed literally, and which ones are simply used to enclose the string value.
- PHP allows you to escape certain characters by preceding them with a backslash (\).

- There so-called escape sequences include:

Sequence	What it represents
\n	A line feed character
\t	A tab
\r	A carriage return
\”	A double quotation mark
\’	A single quotation mark

```
<?php
echo “You said \”Hello\””;
?>
```

output: You said “Hello”

```
<?php
echo “Welcome\nto\nPHP”;
?>
```

Output:

Welcome
to
PHP

```
<?php
echo ‘Welcome\nto\nPHP’;
?>
```

Output:

Welcome\nto\nPHP

- Escape sequences such as those for line feeds (\n), carriage returns (\r) and double quotation marks (\”) can only be understood by PHP parser when they are themselves enclosed in double quotes.
- If these escape sequences are enclosed in single quotes, they will be printed “as it is”.

```
<?php
echo “Welcome\nto\nPHP”;
?>
```

Output:

Welcome
to
PHP

```
<?php
echo ‘Welcome\nto\nPHP’;
?>
```

Output:

Welcome\nto\nPHP

Variable name itself to be variable

```
<?php
$attribute='price';
${$attribute}=678;
echo $price;
?>
```

Output:

678

Destroying Variables

```
<?php
$car='Porshe';
echo "Before unset(), my car is a $car";
unset($car);
echo "After unset(), my car is a $car";
?>
```

Output:

Before unset(), may car is a Porshe

After unset(), my car is a

(will generate 'undefined variable' error)

Inspecting Variable Contents

```
<?php
$name=“Sanjay”;
$age=28;
var_dump($name);
var_dump($age);
print_r($name);
print_r($age);
?>
```

Output;

```
string 'Sanjay' (length=6)
int 28
Sanjay28
```

- print_r() function performs the same function as var_dump(), although it returns less information.

Understanding Data Types

```
<?php
//Boolean
$validUser=true;
//Integer
$size=15;
//Floating point
$temp=98.6;
//String
$cat="Siamese";
//Null
$here=null;
?>
```

Setting & Checking Variable Data Types

```
<?php
$whoami='Sachin';
echo gettype($whoami);
$whoami=99.8;
echo gettype($whoami);
unset($whoami);
echo gettype($whoami);
?>
```

Output:

string
double
NULL (Error for undefined variable)

- PHP automatically determines a variable's data type from the content it holds.
- And if the variable's content changes over the duration of a script, the language will automatically set the variable to the appropriate new data type. (Type Juggling)

PHP functions to test variable data types

Function	Purpose
is_bool()	Tests if a variable holds a boolean value
is_numeric()	Tests if a variable holds a numeric value
is_int()	Tests if a variable holds a integer value
is_float()	Tests if a variable holds a floating-point value
is_string()	Tests if a variable holds a string value
is_null()	Tests if a variable holds a NULL value
is_array()	Tests if a variable holds an array

Using Constatnts

```
<?php
define ('PROGRAM', 'The Matrix');
define('VERSION' 11.7);
echo 'Welcome to '.PROGRAM.' (version'.VERSION.')';
?>
```

Output:

Welcome to The Matrix (version 11.7)

Using String Functions

Function	What it does
empty()	Tests if a string is empty
strlen()	Calculates the number of characters in a string
strrev()	Reverses a string
str_repeat()	Repeats a string
substr()	Retrieves a section of a string
strcmp()	Compares two strings
str_word_count()	Calculates the number of words in a string
str_replace()	Replaces parts of a string
trim()	Removes leading and trailing whitespace from a string
strtolower()	Lowercases a string
strtoupper()	Uppercases a string
ucfirst()	Uppercases the first character of a string
ucwords()	Uppercases the first character of every word of a string

Function	What it does
addslashes()	Escapes special characters in a string with backslashes
stripslashes()	Removes backslashes from a string
htmlspecialchars()	Encodes special HTML characters within a string
strip_tags()	Removes PHP and HTML code from a string

Checking for Empty string

```
<?php
$str='';
echo (boolean)empty($str);          //true
$str=null;
echo (boolean)empty($str);          //true
$str='0';
echo (boolean)empty($str);          //true
unset($str);
echo (boolean)empty($str);          //true
?>
```

Reversing and Repeating strings

```
<?php
$str='Welcome to PHP';
echo strlen($str);
?>
```

Output:

```
14
```

```
<?php
$str="One Small Step";
echo strrev($str);
?>
```

Output:

```
petS llamS enO
```

```
<?php
$str='yo';
echo str_repeat($str, 3);
?>
```

Output:

```
yoyoyo
```

Working with Substrings

```
<?php
$str='Welcome to PHP';
echo substr($str,3,4);
?>
```

Output:

```
come
```

```
<?php
$str='Welcome to PHP';
echo
  substr($str,3,5).substr
  ($str,-3,3);
?>
```

Output:

```
come PHP
```

Comparing, Counting and Replacing Strings

```
<?php
$a="hello";
$b="hello";
$c="hEllo";
echo strcmp($a,$b);
echo strcmp($a,$c);
?>
```

Output:

```
0
1
```

- Strcmp() function performs a case-sensitive comparison of two strings, returning a negative value if the first is less than the second, a positive value if it is the other way around and zero if both strings are equal.
- It calculates ASCII value of the string and then compares both strings to check that they are equal, greater or less from each other.

```php
<?php
$str="The name's bond,
  James Bond";
echo
  str_word_count($str);
?>
```

Output:

5

```php
<?php
$str='john@domain.net';
echo str_replace('@',' 
  at ',$str);
?>
```

Output:

john at domain.net

Formatting Strings

```
<?php
//removing leading and trailing whitespaces
$str='    a  b    c   ';
echo trim($str);
?>
```

Output:

```
a  b    c
```

```
<?php
//change string case
$str='Yabba Dabba Doo';
echo strtolower($str);
echo strtoupper($str);
?>
```

Output:

```
yabba dabba doo
YABBA DABBA DOO
```

```php
<?php
//change string case
$str='the yellow brigands';
echo ucwords($str);
echo ucfirst($str);
?>
```

Output:

```
The Yellow Brigands
The yellow brigands
```

Working with HTML Strings

```
<?php
$str="You're awake, aren't you?";
echo addslashes($str);
?>
```

Output:

```
You\'re awake, aren\'t you?
```

```
<?php
//Remove slashes
$str="John D'Souza says \"Catch you later\".";
echo stripslashes($str);
?>
```

Output:

```
John D'Souza says "Catch you later".
```

Using Numeric Functions

Function	What it does
ceil()	Rounds a number up
floor()	Rounds a number down
abs()	Finds the absolute value of a number
pow()	Raises one number to the power of another
log()	Finds the logarithm of a number
exp()	Finds the exponent of a number
rand()	Generates a random number
bindec()	Converts a number from binary to decimal
decbin()	Converts a number from decimal to binary
decoct()	Converts a number from decimal to octal
hexdec()	Converts a number from hexadecimal to decimal
number_format()	Formats a number with grouped thousands and decimals
printf()	Formats a number using a custom specification

```php
<?php
$num=19.7;
//round number up
echo floor($num);

//round number down
Echo ceil($num);
?>
```

Output:

```
19
20
```

```php
<?php
//return absolute value of no
$num= -19.7;
echo abs($num);
?>
```

Output:

```
19.7
```

```php
<?php
//calculate 4^3
echo pow(4,3);
?>
```

Output:

```
64
```

```php
<?php
//generates random number
echo rand();

//generates random number between 10 and 99
echo rand(10,99);
?>
```

```php
<?php
//Format number (with defaults)
$num=1106482.5843;
echo number_format($num);

//format number (with custom separators)
echo number_format($num,3,'*','?');
?>
```

Output:

```
1,106,483
1?106?482*584
```

```
<?php
//Format as decimal number
printf("%05d",65);

//format as floating-point number
printf("%09.3f",239);

//format number incorporate into string
printf("I see %4d apples and %4.2f
oranges",8,26);
?>
```

Output:

```
00065
00239.000
I see 8 apples and 26.00 oranges
```

Working with Array Functions

Function	What it Does
explode()	Splits a string into array elements
implode()	Joins array elements into a string
range()	Generates a number range as an array
min()	Finds smallest value in an array
max()	Finds the largest value in an array
shuffle()	Randomly rearranges the sequence of elements in an array
array_slice()	Extracts a segment of an array
array_shift()	Removes an element from the beginning of an array
array_unshift()	Adds an element to the beginning of an array
array_pop()	Removes an element from the end of an array
array_push()	Adds an element to the end of an array
array_unique()	Removes duplicate elements from an array

Function	What it Does
array_reverse()	Reverses the sequence of elements in an array
array_merge()	Combines two or more arrays
array_intersect()	Calculates the common elements between two or more arrays
array_diff()	Calculates the difference between two arrays
in_array()	Checks if a particular value exists in an array
array_key_exists()	Checks if a particular key exists in an array
sort()	Sorts an array
asort()	Sorts an associative array by value
ksort()	Sorts an associative array by keys
rsort()	Reverse sorts an array
krsort()	Reverse sorts an associative array by value
arsort()	Reverse sorts an associative array by key
pos()	This function returns the value of the current element in an array.

Function	What it Does
next()	Moves internal pointer to the next element of array
count()	Return the number of elements in an array
array_search()	The array_search() function search an array for a value and returns the key.
array_combine()	creates an array by using the elements from one "keys" array and one "values" array.
array_count_values()	This function counts all the values of an array.
array_fill()	This function fills an array with values.
array_change_key_case()	This function changes all keys in an array to lowercase or uppercase.
array_flip()	This function flips/exchanges all keys with their associated values in an array.
array_product()	This function calculates and returns the product of an array.
array_sum()	This function returns the sum of all the values in the array.
array_replace()	This function replaces the values of the first array with the values from following arrays.

```
<?php
//define string
$str='tinker,tailor,soldier,spy';

//convert string to array
$arr=explode(',', $str);
print_r($arr);

?>
```

Output:

```
('tinker', 'tailor', 'soldier', 'spy')
```

```
<?php
//define string
$arr=array(‘one’, ‘two’, ‘three’, ‘four’);

//convert array to string
$str=implode(‘ and ’, $arr);
print_r($str);

?>
```

Output:

```
One and two and three and four
```

```
<?php
//define array
$arr=range(1,1000);
print_r($arr);

?>
```

Output:

Generates an array containing all the values between 1 and 1000

```
<?php
//define string
$data=array('Monday', 'Tuesday', 'Wednesday');

//get array size
echo 'The array has '.count($data).' elements';

?>
```

Output:

The array has 3 elements

```
<?php
//define array
$arr=array(7, 36, 5, 48, 28, 90, 91, 3, 67, 42);

//get min and max
echo 'Minimum is '.min($arr).' and maximum is '.max($arr);
?>
```

Output:

```
Minimum is 3 and maximum is 91
```

Extracting Array Segment

```
<?php
//define array
$rainbow=array('violet','indigo','blue','green',
'yellow','orange', 'red');

//extract 3 central values
$arr1=array_slice($rainbow, 2, 3);
print_r($arr1);

//extract 3 central values starting from the end
$arr2=array_slice($rainbow, -5, 3);
print_r($arr2);

?>
```

Output:

```
('blue', 'green', 'yellow')
('blue', 'green', 'yellow')
```

Adding and Removing Array Elements

```
<?php
//define array
$movies=array('The Lion King','Super 30','Mission Mangal');
//remove the element from the beginning of array
array_shift($movies);
print_r($movies);
//remove the element from the end of array
array_pop($movies);
print_r($movies);
//add element to the end of array
array_push($movies, 'Batla House');
print_r($movies);
//add element to the beginning of array
array_unshift($movies, 'Uri');
print_r($movies);
?>
```

Output:

```
('Super 30', 'Mission Mangal')
('Super 30')
('Super 30', 'Batla House')
('Uri', 'Super 30', 'Batla House')
```

Removing Duplicate Array

```
<?php
//define array
$duplicates=array('a','b','a','c', 'e','d','e');

//remove duplicates
$uniques=array_unique($duplicates);
print_r($uniques);

?>
```

Output:

```
('a', 'b', 'c' 'e', 'd')
```

Randomizing and Reversing Array

```
<?php
//define array
$rainbow=array('violet','indigo','blue','green',
'yellow','orange', 'red');

//randomize array
shuffle($rainbow);
print_r($rainbow);

//reverse array
$arr=array_reverse($rainbow);
print_r($arr);

?>
```

Output:

```
('red', 'orange', 'yellow', 'green', 'blue', 'indigo',
'violet')
```

Searching Array

```
<?php
//define array
$cities=array('London','Paris','Barcelona','Lisbon',
'Zurich');

//search array for value
echo in_array('Barcelona', $cities);
?>
```

Output:

```
1 (true)
```

```
<?php
//define array
$cities=array("United Kingdom"=>"London","United
States"=>"Washington", "France"=>"Paris","India"
=>"Delhi");

//search array for key
echo array_key_exists('India', $cities);
?>
```

Output:

```
1 (true)
```

Sorting Array

```
<?php
//define array
$data=array(15, 81, 14, 74, 2);

//sort and print array
Sort($data);
print_r($data);
?>
```

Output:

```
(2, 14, 15, 74, 81)
```

```
<?php
//define array
$profile=array("fname"=>"Virat",
"lname"=>"Kohli"
"sex"=>"male"
"sector"=>"Criketer");

//sort by value
asort($profile);
print_r($profile);
ksort($profile);
print_r($profile);
?>
```

Output:

```
("sector"=>"Cricketer",
"lname"=>"Kohli",
"fname"=>"Virat",
"sex"=>"male")

("fname"=>"Virat",
"lname"=>"Kohli",
"sector"=>"Cricketer",
"sex"=>"male")
```

Merging Array

```
<?php
//define array
$dark=array('black', 'brown', 'blue');
$light=array('white', 'silver', 'yellow');

//merge arrays
$colors=array_merge($dark, $light);
print_r($colors);
?>
```

Output:

```
('black', 'brown', 'blue', 'white', 'silver', 'yellow')
```

Comparing Arrays

```
<?php
//define array
$orange=array('red', 'yellow');
$green=array('yellow', 'blue');

//find common elements
$common=array_intersect($orange, $green);
print_r($common);

//find elements in first array but not in second
$unique=array_diff($orange, $green);
print_r($unique);
?>
```

Output:

```
('yellow')
('red')
```

```
<html>
<body>

<?php
$fname=array(“Rohit",“Virat",”Ma
hendra");
$age=array(“33",“35","43");
$c=array_combine($fname,$age);
print_r($c);
?>

</body>
</html>
```

Output:

Array ([Rohitr] => 33 [Virat] => 35 [Mahendra] => 43)

```
<html>
<body>

<?php
$a=array("A","Cat","Dog","A","Dog“, “A”);
print_r(array_count_values($a));
?>

</body>
</html>
```

Output:

Array ([A] => 3 [Cat] => 1 [Dog] => 2)

array_fill(*index, number, value*)

```
<html>
<body>

<?php
$a1=array_fill(3,4,"blue");
$b1=array_fill(0,1,"red");
print_r($a1);
echo "<br>";
print_r($b1);
?>

</body>
</html>
```

Output:

Array ([3] => blue [4] => blue [5] => blue [6] => blue)

Array ([0] => red)

```
<html>
<body>

<?php
$a1=array("a"=>"red","b"=>"green","c"=>"blue","d"=>"yellow");
$result=array_flip($a1);
print_r($result);
?>

</body>
</html>
```

Output:

Array ([red] => a [green] => b [blue] => c [yellow] => d)

```
<html>
<body>

<?php
$a=array(5,5);
echo(array_product($a));
?>

</body>
</html>
```

Output:
25

```
<html>
<body>

<?php
$a=array(5,15,25);
echo array_sum($a);
?>

</body>
</html>
```

Output:
45

array_replace(*array1, array2, array3, ...*)

```
<html>
<body>

<?php
$a1=array("red","green");
$a2=array("blue","yellow");
print_r(array_replace($a1,$a2));
?>

</body>
</html>
```

Output:
Array ([0] => blue [1] => yellow)

```
<html>
<body>

<?php
$a1=array("red","green");
$a2=array("blue","yellow");
$a3=array("orange","burgundy");
print_r(array_replace($a1,$a2,$a3));
?>

</body>
</html>
```

Output:
Array ([0] => orange [1] => burgundy)

```
<html>
<body>

<?php
$age=array(“Rohit"=>"33",“Virat"=>“35",“Mahendra"=>"43");
print_r(array_change_key_case($age,CASE_UPPER));
?>

</body>
</html>
```

Output:

Array ([ROHIT] => 33 [VIRAT] => 35 [MAHENDRA] => 43)

```
<html>
<body>

<?php
$people = array("Peter", "Joe", "Glenn", "Cleveland");

echo pos($people) . "<br>";
?>

</body>
</html>
```

Output:
Peter

current() - returns the value of the current element in an array
end() - moves the internal pointer to, and outputs, the last element in the array
next() - moves the internal pointer to, and outputs, the next element in the array
prev() - moves the internal pointer to, and outputs, the previous element in the array
reset() - moves the internal pointer to the first element of the array
each() - returns the current element key and value, and moves the internal pointer forward

Processing arrays with loops and iterators

```
<?php
//define array
$cities=array('London', 'Paris', 'Madrid', 'Mumbai',
'Jakarta', 'Los Angeles');

//iterate over array-print each value
for($i=0;$i<count($cities);$i++)
{
  echo $cities[$i]."\r\n";
}
?>
```

Output:

```
London
Paris
Madrid
Mumbai
Jakarta
Los Angeles
```

```
<?php
//define array
$cities=array('London', 'Paris', 'Madrid', 'Mumbai',
'Jakarta', 'Los Angeles');

//iterate over array-print each value
foreach($cities as $c)
{
  echo "$c  \r\n";
}
?>
```

Output:

```
London
Paris
Madrid
Mumbai
Jakarta
Los Angeles
```

```
<?php
//define array
$cities=array("United Kingdom"=>"London", "United
States"=>"Washington", "France"=>"Paris",
"India"=>"Delhi");

//iterate over array-print each value
foreach($cities as $key=>$value)
{
  echo "$value is in $key.  <br>";
}
?>
```

Output:

London is in United Kingdom.
Washington is in United States.
Paris is in France.
Delhi is in India.

Array Iterator

```
<?php
//define array
$cities=array("United Kingdom"=>"London", "United
States"=>"Washington", "France"=>"Paris", "India"=>"Delhi");

//create an ArrayIterator object
$iterator=new ArrayIterator($cities);

//rewind to beginning of array
$iterator->rewind();

//iterate over array—print each value
while($iterator->valid())
{
  print $iterator->current()." is in ".$iterator->key().". <br>";
  $iterator->next();
}
?>
```

Output:

```
London is in United Kingdom.
Washington is in United States.
Paris is in France.
Delhi is in India.
```

Date and Time

```
<?php
$now=getdate();
print_r($now);
echo "<br>Today's date is ".$now['mday']." ".$now['mon']." ".$now['year']." and Time
is ".$now['hours'].":".$now['minutes'].":".$now['seconds'];

$now=time();
echo "<br>".$now;
echo "<br>Today's date is ".date("d M Y h:m a", $now);

$str="22 August 2019";
echo "<br>".date("d M Y",strtotime($str));

$str="Aug 22 2019";
echo "<br>".date("d M Y",strtotime($str));
?>
```

Output:

Array ([seconds] => 15 [minutes] => 37 [hours] => 6 [mday] => 22 [wday] => 4 [mon] => 8 [year] => 2019 [yday] => 233 [weekday] => Thursday [month] => August [0] => 1566455835)

Today's date is 22 8 2019 and Time is 6:37:15

1566455835

Today's date is 22 Aug 2019 06:08 am

22 Aug 2019

22 Aug 2019

Characters used in Date()

Character	What it Means
d	Day of the month (numeric)
D	Day of the week (string)
l	Day of week (string)
F	Month (string)
M	Month (string)
m	Month (numeric)
Y	Year
h	Hour (in 12-hour format)
H	Hour (in 24-hour format)
a	AM or PM
i	Minute
s	Second

```
<?php
If(checkdate(2,30,2008))
{
  echo 'Date is Valid';
}
else
{
  echo 'Date is invalid';
}
?>
```

Output:

Date is invalid