PHP settype() Function

The settype() function converts a variable to a specific type.

## **Syntax**

settype(variable, type);

## **Parameter Values**

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| variable | Required. Specifies the variable to convert |
| type | Required. Specifies the type to convert variable to. The possible types are: boolean, bool, integer, int, float, double, string, array, object, null |

<!DOCTYPE html>

<html>

<body>

<?php

$a = "32"; // string

settype($a, "integer"); // $a is now integer

$b = 32; // integer

settype($b, "string"); // $b is now string

$c = true; // boolean

settype($c, "integer"); // $c is now integer (1)

?>

</body>

</html>

# PHP gettype() Function

Return the type of different variables:

<?php  
$a = 3;  
echo gettype($a) . "<br>";  
  
$b = 3.2;  
echo gettype($b) . "<br>";  
  
$c = "Hello";  
echo gettype($c) . "<br>";  
  
$d = array();  
echo gettype($d) . "<br>";  
  
$e = array("red", "green", "blue");  
echo gettype($e) . "<br>";  
  
$f = NULL;  
echo gettype($f) . "<br>";  
  
$g = false;  
echo gettype($g) . "<br>";  
?>

# PHP isset() Function

Check whether a variable is empty. Also check whether the variable is set/declared:

<!DOCTYPE html>

<html>

<body>

<?php

$a = 0;

// True because $a is set

if (isset($a)) {

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

}

$b = null;

// False because $b is NULL

if (isset($b)) {

echo "Variable 'b' is set.";

}

?>

</body>

</html>

# PHP unset() Function

The unset() function unsets a variable.

<!DOCTYPE html>

<html>

<body>

<?php

$a = "Hello world!";

echo "The value of variable 'a' before unset: " . $a . "<br>";

unset($a);

echo "The value of variable 'a' after unset: " . $a;

?>

</body>

</html>

# PHP implode() Function

Join array elements with a string:

<!DOCTYPE html>

<html>

<body>

<?php

$arr = array('Hello','World!','Beautiful','Day!');

echo implode(" ",$arr);

?>

</body>

</html>

Hello World! Beautiful Day!

Separate the array elements with different characters:

<!DOCTYPE html>

<html>

<body>

<?php

$arr = array('Hello','World!','Beautiful','Day!');

echo implode(" ",$arr)."<br>";

echo implode("+",$arr)."<br>";

echo implode("-",$arr)."<br>";

echo implode("X",$arr);

?>

</body>

</html>

Hello World! Beautiful Day!  
Hello+World!+Beautiful+Day!  
Hello-World!-Beautiful-Day!  
HelloXWorld!XBeautifulXDay!

# PHP explode() Function

Break a string into an array:

<!DOCTYPE html>

<html>

<body>

<?php

$str = "Hello world. It's a beautiful day.";

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

?>

</body>

</html>

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

# PHP compact() Function

Create an array from variables and their values:

<!DOCTYPE html>

<html>

<body>

<?php

$firstname = "Peter";

$lastname = "Griffin";

$age = "41";

$result = compact("firstname", "lastname", "age");

print\_r($result);

?>

</body>

</html>

Array ( [firstname] => Peter [lastname] => Griffin [age] => 41 )

# PHP extract() Function

The extract() function imports variables into the local symbol table from an array.

This function uses array keys as variable names and values as variable values. For each element it will create a variable in the current symbol table.

This function returns the number of variables extracted on success.

<!DOCTYPE html>

<html>

<body>

<?php

$a = "Original";

$my\_array = array("a" => "Cat","b" => "Dog", "c" => "Horse");

extract($my\_array);

echo "\$a = $a; \$b = $b; \$c = $c";

?>

</body>

</html>

$a = Cat; $b = Dog; $c = Horse