fprintf »
« echo

- Manual de PHP
- Referencia de funciones
- Procesamiento de texto
- Strings
- Funciones de strings

Change language: Spanish

Submit a Pull Request Report a Bug

explode

(PHP 4, PHP 5, PHP 7, PHP 8)

explode — Divide un string en varios string

Descripción ¶

explode(string `$delimiter`, string `$string`, int `$limit` = PHP_INT_MAX): array

Devuelve un array de string, siendo cada uno un substring del parámetro `string` formado por la división realizada por los delimitadores indicados en el parámetro string `separator`.

Parámetros ¶

`separator`

El string delimitador.

`string`

El string de entrada.

`limit`

Si el parámetro `limit` es positivo, el array devuelto contendrá el máximo de `limit` elementos, y el último elemento contendrá el resto del `string`.

Si el parámetro `limit` es negativo, se devolverán todos los componentes a excepción del último `-limit`.

Si el parámetro `limit` es cero, se tratará como 1.

Nota:

Antes de PHP 8.0, implode() aceptaba los parámetros en cualquier orden. **explode()** nunca ha soportado esto: Debe asegurarse de que el argumento `separator` vaya antes que el argumento `string`.

Valores devueltos ¶

Devuelve un array de string creados por la división del parámetro `string` usando los delimitadores indicados en el parámetro `delimiter`.

Si `separator` es un string vacío (""), **explode()** devolverá un **ValueError**. Si `separator` contiene un valor que no forma parte del parámetro `string` y se utiliza un `limit` negativo, entonces devolverá un array vacío, en caso contrario se devolverá un array que contiene el valor de `string`. Si el valor de `separator` aparece al comienzo o al final del `string`, dichos valores se añadirán como un valor de array vacío en la primera o última posición a devolver respectivamente.

Historial de cambios ¶

Versión	Descripción
8.0.0	**explode()** ahora lanzará **ValueError** cuando el argumento `separator` reciba un string vacío (""). Anteriormente, **explode()** devolvía `false` en su lugar.

Ejemplos ¶

Ejemplo #1 Ejemplo de explode()

```php
<?php
// Ejemplo 1
$pizza  = "porción1 porción2 porción3 porción4 porción5 porción6";
$porciones = explode(" ", $pizza);
echo $porciones[0]; // porción1
echo $porciones[1]; // porción2

// Ejemplo 2
$datos = "foo:*:1023:1000::/home/foo:/bin/sh";
list($user, $pass, $uid, $gid, $gecos, $home, $shell) = explode(":", $datos);
echo $user; // foo
echo $pass; // *

?>
```

Ejemplo #2 Ejemplos de la devolución de explode()

```php
<?php
/*
   Un string que no contiene el delimitador simplemente
   devolverá un array de longitud uno con el string original.
*/
$entrada1 = "hola";
$entrada2 = "hola,qué tal";
var_dump( explode( ',', $entrada1 ) );
var_dump( explode( ',', $entrada2 ) );

?>
```

El resultado del ejemplo sería:

```
array(1)
(
    [0] => string(4) "hola"
)
array(2)
(
    [0] => string(4) "hola"
    [1] => string(8) "qué tal"
)
```

Ejemplo #3 Ejemplos del parámetro `limit`

```php
<?php
$str = 'uno|dos|tres|cuatro';

// límite positivo
print_r(explode('|', $str, 2));

// límite negativo
print_r(explode('|', $str, -1));
?>
```

El resultado del ejemplo sería:

```
Array
(
    [0] => uno
    [1] => dos|tres|cuatro
)
Array
(
    [0] => uno
    [1] => dos
    [2] => tres
)
```

Notas ¶

> **Nota**: Esta función es segura binariamente.

Ver también ¶

- preg_split() - Divide un string mediante una expresión regular
- str_split() - Convierte un string en un array
- mb_split() - Divide cadenas de caracteres multibyte usando una expresión regular
- str_word_count() - Devuelve información sobre las palabras utilizadas en un string
- strtok() - Tokeniza string
- implode() - Une elementos de un array en un string

+ add a note

User Contributed Notes 7 notes

up
down
16
Gerben ¶
7 months ago

Note that an empty input string will still result in one element in the output array. This is something to remember when you are processing unknown input.

For example, maybe you are splitting part of a URI by forward slashes (like "articles/42/show" => ["articles", "42", "show"]). And maybe you expect that an empty URI will result in an empty array ("" => []). Instead, it will contain one element, with an empty string:

```php
<?php

$uri = '';
$parts = explode('/', $uri));
var_dump($parts);
```

```
?>

Will output:

array(1) {
  [0]=>
  string(0) ""
}

And not:

array(0) {
}
```

up
down
11
bocoroth ¶
1 year ago

Be careful, while most non-alphanumeric data types as input strings return an array with an empty string when used with a valid separator, true returns an array with the string "1"!

```
var_dump(explode(',', null)); //array(1) { [0]=> string(0) "" }
var_dump(explode(',', false)); //array(1) { [0]=> string(0) "" }

var_dump(explode(',', true)); //array(1) { [0]=> string(1) "1" }
```

up
down
3
henrik Schmidt ¶
1 year ago

"Return value" text needs updating for php 8, an empty delimiter now throws an Exception.

up
down
0
Alejandro-Ihuit ¶
1 month ago

If you want to directly take a specific value without having to store it in another variable, you can implement the following:

```
$status = 'Missing-1';

echo $status_only = explode('-', $status)[0];

// Missing
```

up
down
-10
Emilio Bravo ¶
1 year ago

```
$string = "PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION";
$exploded = explode("::",$string);
/*

explode('::',$string) = eliminate every :: and for each division of ::make an array element

Example:

PDO::ERRMODE_EXCEPTION (exploded) = array     (
```

```
    [0] => string PDO
    [1] => string ERRMODE_EXCEPTION
)
Example:

$exploded[0] = "PDO";
*/
foreach ($exploded as $index) {
    echo $index . "\n";
}
/*

Output:

PDO
ATTR_ERRMODE => PDO
ERRMODE_EXCEPTION

*/
```

up
down
-13
joshuachidiebere17217 at gmail dot com ¶
9 months ago

```
<?php
/*Array to split*/
$task =["Teach","Assess","Record ","Examine","Investigate"];
/*use Implode() function to convert
$task into arrays of string
*/
$string=implode(",", $task);
/*use explode() function to seperate $string into different
$task into arrays of string
*/
explode(",", $string);
foreach ($task as $variable => $tk) {
$variable>0;
$variable++;
echo '$variable_'.$variable.' is ' .$tk.'<br>';
}
echo min(3,6);
?>
```

up
down
-27
David Spector ¶
1 year ago

When using 'explode' to create an array of strings from a user-specified string that contains newline characters, you may wish the resulting array to correctly reflect the user's intentions by ignoring any final empty line (many users like to end multi-line input with a final newline, for clarity).

Here is a function to call after 'explode' to support this effect:

```
// When using explode, delete the last line in the array of lines when it is empty
function IgnoreEmptyLastLine(&$linesArr)
    {
    $last=count($linesArr)-1;
```

```
    if ($last>=0 && !$linesArr[$last])
        unset($linesArr[$last]);
    } // IgnoreEmptyLastLine
```

+ add a note

- str_ireplace
- str_pad
- str_repeat
- str_replace
- str_rot13
- str_shuffle
- str_split
- str_starts_with
- str_word_count
- strcasecmp
- strchr
- strcmp
- strcoll
- strcspn
- strip_tags
- stripcslashes
- stripos
- stripslashes
- stristr
- strlen
- strnatcasecmp
- strnatcmp
- strncasecmp
- strncmp
- strpbrk
- strpos
- strrchr
- strrev
- strripos
- strrpos
- strspn
- strstr
- strtok
- strtolower
- strtoupper
- strtr
- substr_compare
- substr_count
- substr_replace
- substr
- trim
- ucfirst
- ucwords
- utf8_decode
- utf8_encode
- vfprintf
- vprintf
- vsprintf
- wordwrap

- Deprecated
  - convert_cyr_string
  - hebrevc

- View Source