

Contenido

Funciones PHP.....	1
Funciones de fecha.....	1
Date https://www.php.net/manual/es/function.date.php date()	1
Formato de fecha y hora local.....	1
Funciones de cadenas	5
Longitud de cadena, https://www.php.net/manual/es/function.strlen.php	5
Primer carácter a mayúsculas si es alfabético ucfirst.....	5
Extraer porción string.....	6
Funciones arrays.....	6
Unir elementos de un array en un string, https://www.php.net/manual/es/function.implode.php	6
Ordenar array mateniendo la asociación de índices:.....	7
Contar apariciones de cada valor en un array array_count_values	8
Devolver un array con sus valores de un sin duplicados array_unique.....	8
Filtrado	9
Filtrar el valor de una variable con un filtro específico filter_var.....	9
Generar entero aleatorio rand	11

Funciones PHP

Funciones de fecha

Date <https://www.php.net/manual/es/function.date.php> **date()**

Devuelve una cadena formateada según el formato dado usando el parámetro de tipo integer **timestamp** dado o el momento actual si no se da una marca de tiempo. En otras palabras, **timestamp** es opcional y por defecto es el valor de [time\(\)](#).

date([string](#) \$format, [int](#) \$timestamp = time()): [string](#)

Formato de fecha y hora local. <https://www.php.net/manual/es/function.strftime.php>

strftime([string](#) \$format, [int](#) \$timestamp = time()): [string](#)

Formatea una fecha/hora local según la configuración local. Los nombres del mes y del día de la semana y otras cadenas dependientes del lenguaje respetan el localismo establecido con [setlocale\(\)](#).

format

*Los siguientes caracteres están reconocidos en el parámetro de cadena
format*

<i>format</i>	Descripción	Ejemplo de valores devueltos
<i>Día</i>	---	---
%a	Una representación textual abreviada del día	dom hasta sáb
%A	Una representación textual completa del día	domingo hasta sábado
%d	El día del mes con dos dígitos (con ceros iniciales)	01 a 31
%e	El día del mes, con un espacio precediendo a los dígitos simples. No está implementado como está descrito en Windows. Véase más abajo para más información.	1 a 31
%j	Día del año, tres dígitos con ceros iniciales	001 a 366
%u	Representación numérica del día de la semana del ISO-8601	1 (para lunes) hasta 7 (para domingo)
%w	Representación numérica del día de la semana	0 (para domingo) hasta 6 (para sábado)
<i>Semana</i>	---	---
%U	Número de semana del año dado, comenzando con el primer domingo como la primera semana	13 (para la 13ª semana completa del año)
%V	Número de semana del año dado del ISO-8601:1988, comenzando con la primera semana del año con al menos 4 días de semana, con el lunes como principio de la semana	01 hasta 53 (donde 53 cuenta para una semana traslapada)
%W	Una representación numérica de la semana del año,	46 (para la 46ª semana del año)

<i>Mes</i>		comenzando con el primer domingo como la primera semana	comenzando con un lunes)
		---	---
	<i>%b</i>	Nombre del mes abreviado, basado en el localismo	ene hasta dic
	<i>%B</i>	Nombre del mes completo, basado en el localismo	enero hasta diciembre
	<i>%h</i>	Nombre del mes abreviado, basado en el localismo (un alias de <i>%b</i>)	ene hasta dic
<i>Año</i>	<i>%m</i>	Representación de dos dígitos del mes	01 (para enero) hasta 12 (para diciembre)
		---	---
	<i>%C</i>	Representación de dos dígitos del siglo (año dividido entre 100, truncado a un entero)	19 para el Siglo XX
	<i>%g</i>	Representación de dos dígitos del año por el estándar ISO-8601:1988 (véase <i>%V</i>)	Ejemplo: 09 para la semana del 6 de enero de 2009
	<i>%G</i>	La versión completa de cuatro dígitos de <i>%g</i>	Ejemplo: 2008 para la semana del 3 de Enero de 2009
	<i>%y</i>	Representación de dos dígitos del año	Ejemplo: 09 para 2009, 79 para 1979
	<i>%Y</i>	Representación de cuatro dígitos del año	Ejemplo: 2038
<i>Hora</i>		---	---
	<i>%H</i>	Representación de dos dígitos de la hora en formato de 24 horas	00 hasta 23
	<i>%k</i>	La hora en format de 24 horas, con un espacio precediendo a las horas de un dígito	0 hasta 23
	<i>%I</i>	Representación de dos dígitos de la hora en formato de 12 horas	01 hasta 12

<i>%l ('L' minúscula)</i>		La hora en formato de 12 horas, con un espacio precediendo a los dígitos simples	1 hasta 12
	<i>%M</i>	Representación de dos dígitos de los minutos	00 hasta 59
	<i>%p</i>	"AM" o "PM" en MAYÚSCULAS basados en la hora dada	Ejemplo: AM para 00:31, PM para 22:23
	<i>%P</i>	"am" o "pm" en minúsculas basados en la hora dada	Ejemplo: am para 00:31, pm para 22:23
	<i>%r</i>	Lo mismo que "%l:%M:%S %p"	Ejemplo: 09:34:17 PM para 21:34:17
	<i>%R</i>	Lo mismo que "%H:%M"	Ejemplo: 00:35 para 12:35 AM, 16:44 para 4:44 PM
	<i>%S</i>	Representación de dos dígitos de los segundos	00 hasta 59
	<i>%T</i>	Lo mismo que "%H:%M:%S"	Ejemplo: 21:34:17 para 09:34:17 PM
	<i>%X</i>	Representación preferida de la hora basada en el localismo, sin la fecha	Ejemplo: 03:59:16 o 15:59:16
	<i>%Z</i>	El índice de la zona horaria. No implementado como está descrito en Windows. Véase más abajo para más información.	Ejemplo: -0500 para Hora del Este de EEUU
	<i>%Z</i>	La abreviatura de la zona horaria No implementado como está descrito en Windows. Véase más abajo para más información.	Ejemplo: EST para Hora del Este
		---	---
<i>Marcas de fecha y hora</i>			
	<i>%c</i>	Marca preferida de la fecha y hora basadas en el localismo	Ejemplo: Tue Feb 5 00:45:10 2009 para el 5 de febrero de

		2009 a las 12:45:10 AM
%D	Lo mismo que "%m/%d/%y"	Ejemplo: 02/05/09 para el 5 de febrero de 2009
%F	Lo mismo que "%Y-%m-%d" (normalmente usado en las marcas de fecha de bases de datos)	Ejemplo: 2009-02-05 para el 5 de febrero de 2009
%s	Marca temporal del instante de la Época Unix (lo mismo que la función time())	Ejemplo: 305815200 para el 10 de septiembre de 1979 08:40:00 AM
%x	Representación preferida de la fecha basada en la configuración local, sin la hora	Ejemplo: 02/05/09 para el 5 de febrero de 2009
Miscelánea	---	---
%n	Un carácter de nueva línea ("\\n")	---
%t	Un carácter de tabulación ("\\t")	---
%%	Un carácter de porcentaje literal ("%")	---

Funciones de cadenas

Longitud de cadena **strlen()**

, <https://www.php.net/manual/es/function strlen.php>

strlen(string \$string): [int](#)

Primer carácter a mayúsculas si es alfabético [ucfirst](#)

ucfirst([string](#) \$str): [string](#)

Devuelve una cadena con el primer caracter **str** en máyusculas, si el caracter es alfabético.

Extraer porción string, **substr()**

<https://www.php.net/manual/es/function.substr.php>

substr([string](#) \$string, [int](#) \$start, [int](#) \$length = ?): [string](#)

Devuelve la longitud del string dado.

Explode

implode

Funciones arrays

Unir elementos de un array en un string **implode()**

<https://www.php.net/manual/es/function.implode.php>

implode([string](#) \$separator, [array](#) \$array): [string](#) ->alias join

Parámetros

separator

Opcional. Por defecto es un string vacío.

array

El array de strings a ser usados por implode.

Valores devueltos [¶](#)

Devuelve un string que contiene la representación de todos los elementos del array en el mismo orden, con el string 'glue' entre cada elemento.

Ejemplos [¶](#)

Ejemplo #1 Ejemplo de implode()

```
<?php
```

```
$array = ['lastname', 'email', 'phone'];
```

```

var_dump(implode(", ", $array)); // string(20) "lastname,email,phone"

// Devuelve un string vacío si se usa un array vacío:
var_dump(implode('hello', [])); // string(0) ""

// El separador es opcional:
var_dump(implode(['a', 'b', 'c'])); // string(3) "abc"

?>

```

Ordenar array manteniendo la asociación de índices **asort()**:

asort([array](#) &\$array, [int](#) \$sort_flags = SORT_REGULAR): [bool](#)

Esta función ordena un array manteniendo la correlación de los índices del array con los elementos con los que están asociados. Esta función se utiliza principalmente para ordenar arrays asociativos en los que el orden es importante. **Ordena por valores**

Nota:

Si dos miembros se comparan como iguales, su orden relativo en el array ordenado será indefinido.

Parámetros

array

El array de entrada.

sort_flags

Quizá se necesita cambiar el comportamiento del ordenado usando este parámetro opcional sort_flags, para más información ver [sort\(\)](#).

Devuelve [true](#) en caso de éxito o [false](#) en caso de error.

```

<?php
$fruits = array("d" => "lemon", "a" => "orange", "b" => "banana", "c" => "apple");
asort($fruits);
foreach ($fruits as $key => $val) {
    echo "$key = $val\n";
}
?>

```

El resultado del ejemplo sería:

c = apple

b = banana

d = lemon

a = orange

arsort([array](#) &\$array, [int](#) \$sort_flags = SORT_REGULAR): [bool](#)

Hace lo mismo que `asort()`, pero invierte el orden.

a = orange

d = lemon

b = banana

c = apple

Contar apariciones de cada valor en un array `array_count_values`

`array_count_values(array $array): array`

`array_count_values()` retorna un array usando los valores de la matriz array como keys y su frecuencia en array como valores.

Parametros: `$array` Array de valores a contar

Retorna un array asociativo de valores a partir de array como keys y su respectivo recuento como valores.

Arroja un **`E_WARNING`** por cada elemento el cual no sea `string` o `integer`.

```
<?php
$array = array(1, "hello", 1, "world", "hello");
print_r(array_count_values($array));
?>
```

El resultado del ejemplo sería:

Array

```
(
    [1] => 2
    [hello] => 2
    [world] => 1
)
```

Devolver un array con sus valores de un sin duplicados `array_unique`

`array_unique(array $array, int $sort_flags = SORT_STRING): array`

Toma un array y devuelve un nuevo array sin valores duplicados.

Tenga en cuenta que las claves se conservan. Si múltiples elementos se comparan bajo el parámetro `sort_flags`, entonces la clave y el valor del primer elemento igual se conservarán.

Parámetros

array

El array de entrada.

sort_flags

El segundo parámetro opcional `sort_flags` se puede utilizar para modificar el tipo de orden usando estos valores:

Indicadores de tipos de orden:

- [SORT_REGULAR](#) - compara ítems normalmente (no cambia los tipos)
- [SORT_NUMERIC](#) - compara ítems numéricamente
- [SORT_STRING](#) - compara ítems como strings
- [SORT_LOCALE_STRING](#) - compara ítems como strings, basados en la configuración regional en uso.

Devuelve el array filtrado.

Filtrado

Filtrar los elementos de un array usando una función de devolución de llamada `array_filter`

`array_filter(array $array, callable $callback = ?, int $flag = 0): array`

Recorre cada valor de **array**, pasándolos a la función **callback**. Si la función **callback** devuelve [true](#) el valor actual desde **array** es devuelto al [array](#) resultante. Las claves del array se conservan.

Parámetros

array

El array a recorrer.

callback

La función de devolución de llamada a usar.

Si no se proporciona **callback**, todas las entradas de **array** iguales a [false](#) (véase [convertir a boolean](#)) serán eliminadas.

flag

Indicador que determina qué argumentos se envían a **callback**:

- [ARRAY_FILTER_USE_KEY](#) - pasar la clave como el único argumento a **callback** en lugar del valor
- [ARRAY_FILTER_USE_BOTH](#) - pasar el valor y la clave como argumentos a **callback** en lugar del valor

Por omisión es 0 que pasará el valor como el único argumento a **callback** en su lugar.

Valores devueltos

Devuelve el array filtrado.

Filtrar el valor de una variable con un filtro específico filter_var

(PHP 5 >= 5.2.0, PHP 7, PHP 8)

`filter_var` — Filters a variable with a specified filter

`filter_var(mixed $value, int $filter = FILTER_DEFAULT, array|int $options = 0): mixed`

Parameters

value

Value to filter. Note that scalar values are [converted to string](#) internally before they are filtered.

filter

The ID of the filter to apply. The [Types of filters](#) manual page lists the available filters.

If omitted, [FILTER_DEFAULT](#) will be used, which is equivalent to [FILTER_UNSAFE_RAW](#). This will result in no filtering taking place by default.

options

Associative array of options or bitwise disjunction of flags. If filter accepts options, flags can be provided in "flags" field of array. For the "callback" filter, [callable](#) type should be passed. The callback must accept one argument, the value to be filtered, and return the value after filtering/sanitizing it.

Return Values

Returns the filtered data, or [false](#) if the filter fails.

Example #1 A filter_var() example

```
<?php
var_dump(filter_var('bob@example.com', FILTER_VALIDATE_EMAIL));
var_dump(filter_var('http://example.com', FILTER_VALIDATE_URL,
FILTER_FLAG_PATH_REQUIRED));
?>
```

The above example will output:

```
string(15) "bob@example.com"
```

```
bool(false)
```

Example #2 Filter an array example

```
<?php
$emails = [
    "bob@example.com",
    "test@example.local",
    "invalidemail"
];

var_dump(filter_var($emails, FILTER_VALIDATE_EMAIL, FILTER_REQUIRE_ARRAY));
?>
```

The above example will output:

```
array(3) {
    [0]=>
        string(15) "bob@example.com"
    [1]=>
        string(18) "test@example.local"
    [2]=>
        bool(false)
}
```

Generar entero aleatorio rand

rand(): [int](#)

rand([int](#) \$min, [int](#) \$max): [int](#)

If called without the optional **min**, **max** arguments rand() returns a pseudo-random integer between 0 and [getrandmax\(\)](#). If you want a random number between 5 and 15 (inclusive), for example, use rand(5, 15).

Esta función no genera valores criptográficos fiables por lo que no debería emplearse para propósitos criptográficos. Si fuera necesario un valor criptográfico seguro, considérese utilizar [random_int\(\)](#), [random_bytes\(\)](#), o [openssl_random_pseudo_bytes\(\)](#) en su lugar.

min

The lowest value to return (default: 0)

max

The highest value to return (default: [getrandmax\(\)](#))

<https://www.php.net/manual/en/function.ucfirst.php>

<https://www.php.net/manual/es/function.in-array.php>

<https://www.php.net/manual/en/function.strtoupper.php>

<https://www.php.net/manual/es/function.cal-days-in-month.php>

<https://www.php.net/manual/es/function.explode.php>

<https://www.php.net/manual/es/function.intdiv.php>

<https://www.php.net/manual/es/function.explode.php>