



Funciones de fecha y hora

- [Huso horario: función date_default_timezone_set\(\)](#)
- [Generar cadena con fecha y hora](#)
 - [Función date\(\)](#)
 - [Función strftime\(\)](#)
- [Fecha y hora actual: función localtime\(\)](#)
- [Validación de fecha: función checkdate\(\)](#)
- [Convertir cadenas de fechas](#)

Huso horario: función date_default_timezone_set()

La función [date_default_timezone_set\(\\$identificador_de_zona_horaria\)](#) establece el huso horario que utilizan las funciones de fecha y hora. El manual de PHP contiene la lista de [identificadores de zonas horarias válidas](#).

```
<?php
date_default_timezone_set("Europe/Madrid");
print "<p>" . date("Y") . "</p>\n";
?>
```

```
<p>2007</p>
```

Cuando la directiva `error_reporting` tiene establecido el valor `E_STRICT`, PHP genera un aviso si no se ha establecido el huso horario antes de utilizar cualquier función de fecha y hora.

```
<?php
print "<p>" . date("Y") . "</p>\n";
?>
```

Strict Standards: date() [function.date]: It is not safe to rely on the system's timezone settings. Please use the date.timezone setting, the TZ environment variable or the date_default_timezone_set() function. In case you used any of those methods and you are still getting this warning, you most likely misspelled the timezone identifier. We selected 'Europe/Paris' for '1.0/no DST' instead in **ejemplo.php** on line 2
2007

Date:

(PHP 4, PHP 5)

date — Dar formato a la fecha/hora local

Descripción

```
string date ( string $format [, int $timestamp = time() ] )
```

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\(\)](#).

Parámetros

format

El formato de la fecha de salida tipo [string](#). Vea las opciones de formato más abajo. También hay varias [constantes de fecha predefinidas](#) que pueden usarse en su lugar, así por ejemplo `DATE_RSS` contiene la cadena de formato `'D, d M Y H:i:s'`.

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

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

| Carácter de format | Descripción | Ejemplo de valores devueltos |
|---------------------------|--|--|
| <i>Día</i> | --- | --- |
| <i>d</i> | Día del mes, 2 dígitos con ceros iniciales | 01 a 31 |
| <i>D</i> | Una representación textual de un día, tres letras | <i>Mon</i> hasta <i>Sun</i> |
| <i>j</i> | Día del mes sin ceros iniciales | 1 a 31 |
| <i>l</i> ('L' minúscula) | Una representación textual completa del día de la semana | <i>Sunday</i> hasta <i>Saturday</i> |
| <i>N</i> | Representación numérica ISO-8601 del día de la semana (añadido en PHP 5.1.0) | 1 (para lunes) hasta 7 (para domingo) |
| <i>S</i> | Sufijo ordinal inglés para el día del mes, 2 caracteres | <i>st</i> , <i>nd</i> , <i>rd</i> o <i>th</i> . Funciona bien con <i>j</i> |
| <i>w</i> | Representación numérica del día de la semana | 0 (para domingo) hasta 6 (para sábado) |
| <i>z</i> | El día del año (comenzando por 0) | 0 hasta 365 |

| | | |
|---------------|---|--------------------------------------|
| <i>Semana</i> | --- | --- |
| <i>W</i> | Número de la semana del año ISO-8601, las semanas comienzan en lunes (añadido en PHP 4.1.0) | Ejemplo: 42 (la 42ª semana del año) |
| <i>Mes</i> | --- | --- |
| <i>F</i> | Una representación textual completa de un mes, como <i>January</i> o <i>March</i> | <i>January</i> hasta <i>December</i> |
| <i>m</i> | Representación numérica de una mes, con ceros iniciales | 01 hasta 12 |
| <i>M</i> | Una representación textual corta de un mes, tres letras | <i>Jan</i> hasta <i>Dec</i> |
| <i>n</i> | Representación numérica de un mes, sin ceros iniciales | 1 hasta 12 |
| <i>t</i> | Número de días del mes dado | 28 hasta 31 |

| | | |
|----------|---|---------------|
| <i>B</i> | Hora Internet | 000 hasta 999 |
| <i>g</i> | Formato de 12 horas de una hora sin ceros iniciales | 1 hasta 12 |
| <i>G</i> | Formato de 24 horas de una hora sin ceros iniciales | 0 hasta 23 |
| <i>h</i> | Formato de 12 horas de una hora con ceros iniciales | 01 hasta 12 |
| <i>H</i> | Formato de 24 horas de una hora con ceros iniciales | 00 hasta 23 |
| <i>i</i> | Minutos, con ceros iniciales | 00 hasta 59 |
| <i>s</i> | Segundos, con ceros iniciales | 00 hasta 59 |

| | | |
|----------|---|---------------|
| <i>A</i> | Ante meridiem y Post meridiem en mayúsculas | AM o PM |
| <i>B</i> | Hora Internet | 000 hasta 999 |
| <i>g</i> | Formato de 12 horas de una hora sin ceros iniciales | 1 hasta 12 |
| <i>G</i> | Formato de 24 horas de una hora sin ceros iniciales | 0 hasta 23 |
| <i>h</i> | Formato de 12 horas de una hora con ceros iniciales | 01 hasta 12 |
| <i>H</i> | Formato de 24 horas de una hora con ceros iniciales | 00 hasta 23 |
| <i>i</i> | Minutos, con ceros iniciales | 00 hasta 59 |

| | | |
|------------------------|--|--|
| <i>u</i> | Microsegundos (añadido en PHP 5.2.2). Observe que date() siempre generará 000000 ya que toma un parámetro de tipo integer , mientras que DateTime::format() admite microsegundos. | Ejemplo: 654321 |
| <i>Zona Horaria</i> | --- | --- |
| <i>e</i> | Identificador de zona horaria (añadido en PHP 5.1.0) | Ejemplos: <i>UTC, GMT, Atlantic/Azores</i> |
| <i>l</i> (i mayúscula) | Si la fecha está en horario de verano o no | 1 si está en horario de verano, 0 si no. |
| <i>O</i> | Diferencia de la hora de Greenwich (GMT) en horas | Ejemplo: +0200 |
| <i>P</i> | Diferencia con la hora de Greenwich (GMT) con dos puntos entre horas y minutos (añadido en PHP 5.1.3) | Ejemplo: +02:00 |

| | | |
|----------------------------|---|---|
| <i>T</i> | Abreviatura de la zona horaria | Ejemplos: <i>EST, MDT ...</i> |
| <i>Z</i> | Índice de la zona horaria en segundos. El índice para zonas horarias al oeste de UTC siempre es negativo, y para aquellas al este de UTC es siempre positivo. | -43200 hasta 50400 |
| <i>Fecha/Hora Completa</i> | --- | --- |
| <i>c</i> | Fecha ISO 8601 (añadido en PHP 5) | 2004-02-12T15:19:21+00:00 |
| <i>r</i> | Fecha con formato » RFC 2822 | Ejemplo: <i>Thu, 21 Dec 2000 16:01:07 +0200</i> |
| <i>U</i> | Segundos desde la Época Unix (1 de Enero del 1970 00:00:00 GMT) | Vea también time() |

Los caracteres no reconocidos en la cadena de formato serán impresos tal cual. El formato Z siempre devolverá 0 cuando se usa `gmdate()`.

Nota:

Ya que esta función sólo acepta marcas de tiempo de tipo integer el carácter de formato u sólo es útil cuando se usa la función `date_format()` con marcas de tiempo basadas en usuario creadas con `date_create()`.

timestamp

El parámetro opcional timestamp es una marca de tiempo Unix de tipo integer que por defecto es la hora local si no se proporciona ningún valor a timestamp. En otras palabras, es por defecto el valor de la función time().

Valores devueltos :

Devuelve una cadena de fecha formateada. Si se usa un valor no numérico para timestamp, se devuelve FALSE y se emite un error de nivel E_WARNING.

Errores/Excepciones

Cada vez que se llame a la función date/time se generará un E_NOTICE si la zona horaria no es válida, y/o un mensaje E_STRICT o E_WARNING si se usa la configuración del sistema o la variable global TZ. Vea también date_default_timezone_set()

Historial de cambios

Historial de cambios:

| Versión | Descripción |
|---------|---|
| 5.1.0 | El rango válido de una marca de tiempo es típicamente desde Fri, 13 Dec 1901 20:45:54 GMT a Thu, 19 Jan 2038: 03:14:07 GMT (estas son las fechas que corresponden a los valores mínimos y máximos de un entero con signo de 32bits). En algunos sistemas (Windows), antes de PHP 5.1.0; este rango estaba limitado desde 01-01-1970 a 19-01-2038. |
| | |
| 5.1.0 | Ahora muestra E_STRING y E_NOTICE cuando ocurren errores con zonas horarias |
| 5.1.1 | Hay constantes útiles de formatos de fecha/ hora estándares que se pueden usar para especificar el parámetro format . |

Ejemplos:

Ejemplo #1 Ejemplo de date()

```
<?php
```

```
// Establecer la zona horaria predeterminada a usar. Disponible desde PHP 5.1
```

```
date_default_timezone_set('UTC');
```

```
// Imprime algo como: Monday
```

```
echo date("l");
```

```
// Imprime algo como: Monday 8th of August 2005 03:12:46 PM
```

```
echo date('l jS \of F Y h:i:s A');
```

```
// Imprime: July 1, 2000 is on a Saturday
```

```
echo "July 1, 2000 is on a " . date("l", mktime(0, 0, 0, 7, 1, 2000));
```

```

/* Usar las constantes en el parámetro de formato */
// Imprime algo como: Wed, 25 Sep 2013 15:28:57 -0700
echo date(DATE_RFC2822);

// Imprime algo como: 2000-07-01T00:00:00+00:00
echo date(DATE_ATOM, mktime(0, 0, 0, 7, 1, 2000));
?>

```

Puede prevenir que un carácter reconocido en la cadena de formato sea expandido escapándolo con una barra invertida precedente. Si el carácter con una barra invertida es ya una secuencia especial, necesitará escapar también la barra invertida.

Ejemplo #2 Escapar caracteres en date()

```

<?php
// imprime algo como: Wednesday the 15th
echo date('l \t\h\e jS');
?>

```

Es posible usar date() y mktime() juntos para buscar fechas en el futuro o en el pasado.

Ejemplo #3 Ejemplo de date() y mktime()

```

<?php
$mañana = mktime(0, 0, 0, date("m") , date("d")+1, date("Y"));
$mes_anterior = mktime(0, 0, 0, date("m")-1, date("d"), date("Y"));
$año_siguiente = mktime(0, 0, 0, date("m"), date("d"), date("Y")+1);
?>

```

Nota:

Esto puede ser más fiable que añadir o sustraer simplemente el número de segundos de un día o mes a una marca de tiempo debido al horario de verano.

Algunos ejemplos de formatear date(). Observe que debería escapar cualesquiera otros caracteres, ya que cualquiera que tenga actualmente un significado especial producirá resultados no deseados, y a otros caracteres se les pueden asignar significado en futuras versiones de PHP. Cuando se escapa un carácter, asegúrese de usar comillas simples para prevenir que caracteres como \n se conviertan en nuevas líneas.

Ejemplo #4 date() Formatting

```

<?php
// Se asume que hoy es March 10th, 2001, 5:16:18 pm, y que estamos en la
// zona horaria Mountain Standard Time (MST)

$hoy = date("F j, Y, g:i a"); // March 10, 2001, 5:16 pm
$hoy = date("m.d.y"); // 03.10.01
$hoy = date("j, n, Y"); // 10, 3, 2001
$hoy = date("Ymd"); // 20010310
$hoy = date('h-i-s, j-m-y, it is w Day'); // 05-16-18, 10-03-01, 1631 1618 6 Satpm01
$hoy = date("\i\t \i\s \t\h\e jS \d\a\y."); // it is the 10th day.

```

```
$hoy = date("D M j G:i:s T Y");          // Sat Mar 10 17:16:18 MST 2001
$hoy = date('H:m:s \m \i\s\ \m\o\n\t\h'); // 17:03:18 m is month
$hoy = date("H:i:s");                    // 17:16:18
$hoy = date("Y-m-d H:i:s");              // 2001-03-10 17:16:18 (el formato DATETIME de MySQL)
?>
```

Para formatear fechas en otros lenguajes debería usar las funciones `setlocale()` y `strftime()` en vez de `date()`.

Notas ¶

Nota:

Para generar una marca de tiempo desde una cadena que representa la fecha, puede usar `strtotime()`. Adicionalmente, algunas bases de datos tienen funciones para convertir formatos de fecha en marcas de tiempo (como la función » `UNIX_TIMESTAMP` de MySQL).

Sugerencia:

La marca de tiempo del inicio de una petición está disponible en `$_SERVER['REQUEST_TIME']` desde PHP 5.1.

Ver también ¶

- `gmdate()` - Formatea una fecha/hora GMT/UTC
- `idate()` - Formatea una fecha/hora local como un entero
- `getdate()` - Obtener información de la fecha/hora
- `getlastmod()` - Obtiene la hora de la última modificación de la página
- `mktime()` - Obtener la marca de tiempo Unix de una fecha
- `strftime()` - Formatea una fecha/hora local según la configuración regional
- `time()` - Devuelve la fecha Unix actual
- `strtotime()` - Convierte una descripción de fecha/hora textual en Inglés a una fecha Unix

Constantes Predefinidas de DateTime

add a note add a note

`date_default_timezone_set`

(PHP5>=5.1.0)

`date_default_timezone_set()` //Establece la zona horaria predeterminada usada por todas las funciones de fecha y hora en un script.

Descripción:

`bool date_default_timezone_set()`

Nota:

A partir de PHP 5.1.0 (cuando las funciones de fecha y hora fueron reescritas), cada llamada a la función de fecha/ hora generará un **E_NOTICE** si la zona horaria no es válida, o un mensaje **E_WARNING** si se usa la configuración del sistema o la variable de entorno **TZ**.

En vez de usar esta función para establecer la zona horaria predeterminada en su script, también puede utilizar la configuración INI `date.timezone` para establecer la zona horaria predeterminada.

Parámetros:

timezone_indentifier

El identificador de la zona horaria, como UTC o Europe/ Lisbon. La lista de los identificadores válidos está disponible en el listado de las zonas horarias soportadas (<http://php.net/manual/es/timezones.php>).

Ejemplos:

```
<?php
date_default_timezone_set('America/Los Angeles');
$script_tz = date_default_timezone_get();

if (strcmp($script_tz, ini_get('date.timezone'))){
    echo 'La zona horaria del script difiere de la zona horaria de la configuración ini.';
}else{
    echo 'La zona horaria del script y la zona horaria de la configuración ini coinciden.';
}
?>
```

Historial de cambios:

| Versión | Descripción |
|---------|--|
| 5.3.0 | Ahora se lanza el E_WARNING en lugar de E_SCRIPT |
| 5.1.2 | La función valida el parámetro timezone_indentifier |

El siguiente ejemplo tiene resultados similares bajo Linux y Windows:

```
<?php
# uncomment to see differences
# date_default_timezone_set(date_default_timezone_get());

function getmicrotime(){
    list($usec, $sec) = explode(' ', microtime());
    return ($sec.substr($usec,1,7));
}

$start_time=getmicrotime();
for($i=0;$i<&nr;$i++){
    # $value=getdate(rand($start,$end));
    date("H:i:s d.m.Y", rand($start,$end));
}
$end_time=getmicrotime();
echo "Time: " . ($end_time-$start_time);

// With date_default_timezone_set(): "Time: 0.379343986511"
// Without date_default_timezone_set(): "Time: 7.491370697"

}
?>
```



```

<?php
function date_system_timezone_set(){
    $shell = new COM("WScript.Shell") or die("Requires Windows Scripting Host");
    $time_bias = -($shell->RegRead(
"HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Control
\\TimeZoneInformation\\Bias"))/60;
    $ab = -($shell->RegRead(
"HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Control
\\TimeZoneInformation\\ActiveTimeBias"))/60;
    $sc = $shell->RegRead(
"HKEY_USERS\\.DEFAULT\\Control Panel\\International\\sCountry");
    foreach(timezone_abbreviations_list() as $tza) foreach($tza as $entry){
        $country = strtok($entry['timezone_id'],'/');
        $locale = strtok('|');
        if($country==$sc && $ab==($entry['offset']/60/60) && ($ds = $time_bias!=$ab)==$entry['dst']){
            date_default_timezone_set($timezone_identifier = $country."/".$locale);
            return sprintf("%.1f",$ab)."/".($ds?"':'no ').'DST'." ".$timezone_identifier;
        }
    }
    return false;
}
?>

```

Cuando la directiva `error_reporting` tiene establecido el valor `E_STRICT`, PHP genera un aviso si no se ha establecido el huso horario antes de utilizar cualquier función de fecha y hora.

```

<?php
print "<p>" . date("Y") . "</p>\n";
?>

```

Strict Standards: date() [function.date]: It is not safe to rely on the system's timezone settings. Please use the date.timezone setting, the TZ environment variable or the date_default_timezone_set() function. In case you used any of those methods and you are still getting this warning, you most likely misspelled the timezone identifier. We selected 'Europe/Paris' for '1.0/no DST' instead in **ejemplo.php** on line 2
2007

Generar cadena con fecha y hora

Función date()

La función `date($formato)` permite generar una cadena que incluye la fecha y/o la hora del sistema. La función `date()` espera uno o dos argumentos. El primer argumento define el formato de la cadena y el segundo el instante que se quiere convertir (si no se indica el segundo argumento, la función trabaja con la fecha y hora actual del sistema).

Para definir el formato de la cadena se utilizan, entre otros, los siguientes caracteres:

| | día | mes | año |
|---------------------------|-----|-----|------------------------------|
| números con ceros delante | d | m | y (2 cifras) Y (4 cifras) |
| números sin ceros delante | j | n | |
| nombre corto (3 letras) | D | M | |
| nombre largo (completo) | l | F | |

| | hora (12h) | hora (24 h) | minutos | segundos |
|---------------------------|------------|-------------|---------|----------|
| números con ceros delante | h | H | i | s |
| números sin ceros delante | g | G | | |

Además, se pueden utilizar los siguientes valores:

| | carácter |
|--|----------|
| fecha completa en formato RFC2822 | r |
| fecha completa en formato ISO 8061 | c |
| número de días total del mes actual | t |
| número del día de la semana (domingo: 7, lunes: 1, etc.) | N |
| número del día de la semana (domingo: 0, lunes: 1, etc.) | w |
| Número de día del año | Z |

```
<?php
date_default_timezone_set("Europe/Madrid");
print "<p>Ejemplo 1: " . date("d/m/y H:i:s") . "</p>\n";
print "<p>Ejemplo 2: " . date("l, F jS Y ") . "</p>\n";
?>
```

```
<p>Ejemplo 1: 03/03/05 13:43:48</p>
<p>Ejemplo 2: Thursday, March 3rd 2005</p>
```

Función strftime()

La función `strftime($formato)` permite generar una cadena que incluye la fecha y/o la hora del sistema en el idioma deseado. La función `strftime()` espera uno o dos argumentos. El primer argumento define el formato de la cadena y el segundo el instante que se quiere convertir (si no se indica el segundo argumento, la función trabaja con la fecha y hora actual del sistema).

Para definir el formato de la cadena se utilizan, entre otros, los siguientes caracteres:

| | día | mes | año |
|--------------------------------|-----|-----|--------------------------------|
| números con ceros delante | %d | %m | %y (2 cifras) %Y (4 cifras) |
| nombre corto (3 letras) | %a | %b | |
| nombre largo (nombre completo) | %A | %B | |

| | hora (12h) | hora (24 h) | minutos | segundos |
|---------------------------|------------|-------------|---------|----------|
| números con ceros delante | %l | %H | %M | %S |

Además, se pueden utilizar los siguientes valores:

| | carácter |
|------------------------------------|----------|
| fecha y hora (dd/mm/aaaa hh:mm:ss) | %c |
| fecha (dd/mm/aaaa) | %x |
| hora (hh:mm:ss) | %X |

Además, se pueden utilizar los siguientes valores:

| | carácter |
|------------------------------------|----------|
| fecha y hora (dd/mm/aaaa hh:mm:ss) | %c |
| fecha (dd/mm/aaaa) | %x |
| hora (hh:mm:ss) | %X |

```
<?php
date_default_timezone_set("Europe/Madrid");
setlocale(LC_TIME, "spanish");
print "<p>" . strftime("Ejemplo 1: %A, %d de %B de %Y") . "</p>\n";
?>
```

```
<p>Ejemplo 1: jueves, 30 de abril de 2009</p>
```

[Volver al principio de la página](#)

Fecha y hora actual: función localtime()

La función `localtime()` devuelve una matriz con información detallada de la fecha y hora actual. Para que devuelva esa información de la hora actual, se pasa como argumento la función `time()`. El segundo argumento indica si la matriz tiene índices asociativos.

```
<?php
$fecha_actual = localtime(time(), 1);
$año_actual = $fecha_actual["tm_year"] + 1900;
$mes_actual = $fecha_actual["tm_mon"] + 1;
$día_actual = $fecha_actual["tm_mday"];
print "<p>Hoy es el $día_actual/$mes_actual/$año_actual.</p>";
?>
```

```
<p>Hoy es el 10/3/2005.</p>
```

Validación de fecha: función checkdate()

La función `checkdate($mes, $dia, $anyo)` devuelve el valor `true` si la fecha enviada es correcta en el calendario Gregoriano y `false` si no lo es.

```
<?php
$dia = 29;
$mes = 2;
$anyo = 2008;
if (checkdate($mes, $dia, $anyo)) {
    print "<p>El día $dia/$mes/$anyo existe.</p>\n";
} else {
    print "<p>El día $dia/$mes/$anyo no existe.</p>\n";
}
?>
```

<p>El día 29/2/2008 existe.</p>

```
<?php
$dia = 29;
$mes = 2;
$anyo = 2007;
if (checkdate($mes, $dia, $anyo)) {
    print "<p>El día $dia/$mes/$anyo existe.</p>\n";
} else {
    print "<p>El día $dia/$mes/$anyo no existe.</p>\n";
}
?>
```

<p>El día 29/2/2007 no existe.</p>

Hay que tener en cuenta que el [calendario Gregoriano](#) se estableció (en algunos países) a partir del 15 de octubre de 1582, por lo que no tiene sentido aplicar esta función a fechas anteriores a esa fecha. Por ejemplo, la función `checkdate()` no tiene en cuenta que hay unos días (del 5 al 14 de octubre de 1582) que no existieron en el calendario Gregoriano. Por otra parte, si el año es negativo o cero, la función `checkdate()` devuelve siempre `false`.

```
<?php
$dia = 12;
$mes = 10;
$anyo = 1582;
if (checkdate($mes, $dia, $anyo)) {
    print "<p>El día $dia/$mes/$anyo existe.</p>\n";
} else {
    print "<p>El día $dia/$mes/$anyo no existe.</p>\n";
}
?>
```

<p>El día 12/10/1582 existe.</p>

```
<?php
$dia = 1;
$mes = 1;
$anyo = 0;
if (checkdate($mes, $dia, $anyo)) {
    print "<p>El día $dia/$mes/$anyo existe.</p>\n";
} else {
    print "<p>El día $dia/$mes/$anyo no existe.</p>\n";
}
?>
```

<p>El día 1/1/0 no existe.</p>

Convertir cadenas de fechas

Para convertir las cadenas de fechas del formato dd-mm-aaaa al formato aaaa-mm-dd y viceversa se pueden definir estas funciones:

```
function fechaDma($amd)
{
    return substr($amd, 8, 2) . "-" . substr($amd, 5, 2) . "-" . substr($amd, 0, 4);
}

function fechaAmd($dma)
{
    return substr($dma, 7, 4) . "-" . substr($dma, 4, 2) . "-" . substr($dma, 1, 2);
}
```