

format

The format of the outputted date [string](#). See the formatting options below. There are also several [predefined date constants](#) that may be used instead, so for example DATE\_RSS contains the format string *'D, d M Y H:i:s'*.

The following characters are recognized in the *format* parameter string

<i>format</i> character	Description	Example returned values
<i>Day</i>	---	---
<i>d</i>	Day of the month, 2 digits with leading zeros	<b>01</b> to <b>31</b>
<i>D</i>	A textual representation of a day, three letters	<b>Mon</b> through <b>Sun</b>
<i>j</i>	Day of the month without leading zeros	<b>1</b> to <b>31</b>
<i>l</i> (lowercase 'L')	A full textual representation of the day of the week	<b>Sunday</b> through <b>Saturday</b>
<i>N</i>	ISO-8601 numeric representation of the day of the week (added in PHP 5.1.0)	<b>1</b> (for Monday) through <b>7</b> (for Sunday)
<i>S</i>	English ordinal suffix for the day of the month, 2 characters	<b>st</b> , <b>nd</b> , <b>rd</b> or <b>th</b> . Works well with <i>j</i>
<i>w</i>	Numeric representation of the day of the week	<b>0</b> (for Sunday) through <b>6</b> (for Saturday)
<i>z</i>	The day of the year (starting from 0)	<b>0</b> through <b>365</b>
<i>Week</i>	---	---
<i>W</i>	ISO-8601 week number of year, weeks starting on Monday (added in PHP 4.1.0)	Example: <b>42</b> (the 42nd week in the year)
<i>Month</i>	---	---
<i>F</i>	A full textual representation of a month, such as January or March	<b>January</b> through <b>December</b>
<i>m</i>	Numeric representation of a month, with leading zeros	<b>01</b> through <b>12</b>
<i>M</i>	A short textual representation of a month, three letters	<b>Jan</b> through <b>Dec</b>
<i>n</i>	Numeric representation of a month, without leading zeros	<b>1</b> through <b>12</b>
<i>t</i>	Number of days in the given month	<b>28</b> through <b>31</b>
<i>Year</i>	---	---
<i>L</i>	Whether it's a leap year	<b>1</b> if it is a leap year, <b>0</b> otherwise.
<i>o</i>	ISO-8601 year number. This has the same value as <b>Y</b> , except that if the ISO week number ( <b>W</b> ) belongs to the previous or next year, that year is used instead. (added in PHP 5.1.0)	Examples: <b>1999</b> or <b>2003</b>
<i>Y</i>	A full numeric representation of a year, 4 digits	Examples: <b>1999</b> or <b>2003</b>
<i>y</i>	A two digit representation of a year	Examples: <b>99</b> or <b>03</b>
<i>Time</i>	---	---
<i>a</i>	Lowercase Ante meridiem and Post meridiem	<b>am</b> or <b>pm</b>
<i>A</i>	Uppercase Ante meridiem and Post meridiem	<b>AM</b> or <b>PM</b>
<i>B</i>	Swatch Internet time	<b>000</b> through <b>999</b>
<i>g</i>	12-hour format of an hour without leading zeros	<b>1</b> through <b>12</b>
<i>G</i>	24-hour format of an hour without leading zeros	<b>0</b> through <b>23</b>
<i>h</i>	12-hour format of an hour with leading zeros	<b>01</b> through <b>12</b>
<i>H</i>	24-hour format of an hour with leading zeros	<b>00</b> through <b>23</b>
<i>i</i>	Minutes with leading zeros	<b>00</b> to <b>59</b>
<i>s</i>	Seconds, with leading zeros	<b>00</b> through <b>59</b>
<i>u</i>	Microseconds (added in PHP 5.2.2). Note that <b>date()</b> will always generate <b>000000</b> since it takes an <a href="#">integer</a> parameter, whereas <a href="#">DateTime::format()</a> does support microseconds.	Example: <b>654321</b>
<i>Timezone</i>	---	---
<i>e</i>	Timezone identifier (added in PHP 5.1.0)	Examples: <b>UTC</b> , <b>GMT</b> , <b>Atlantic/Azores</b>
<i>I</i> (capital i)	Whether or not the date is in daylight saving time	<b>1</b> if Daylight Saving Time, <b>0</b> otherwise.
<i>O</i>	Difference to Greenwich time (GMT) in hours	Example: <b>+0200</b>
<i>P</i>	Difference to Greenwich time (GMT) with colon between hours and minutes (added in PHP 5.1.3)	Example: <b>+02:00</b>
<i>T</i>	Timezone abbreviation	Examples: <b>EST</b> , <b>MDT</b> ...
<i>Z</i>	Timezone offset in seconds. The offset for timezones west of UTC is always negative, and for those east of UTC is always positive.	<b>-43200</b> through <b>50400</b>
<i>Full Date/Time</i>	---	---
<i>c</i>	ISO 8601 date (added in PHP 5)	2004-02-12T15:19:21+00:00
<i>r</i>	<a href="#">» RFC 2822</a> formatted date	Example: <b>Thu, 21 Dec 2000 16:01:07 +0200</b>
<i>U</i>	Seconds since the Unix Epoch (January 1 1970 00:00:00 GMT)	See also <a href="#">time()</a>

Unrecognized characters in the format string will be printed as-is. The **Z** format will always return **0** when using [gmdate\(\)](#).

**Note:**  
Since this function only accepts [integer](#) timestamps the **u** format character is only useful when using the [date\\_format\(\)](#) function with user based timestamps created with [date\\_create\(\)](#).

timestamp

The optional *timestamp* parameter is an [integer](#) Unix timestamp that defaults to the current local time if a *timestamp* is not given. In other words, it defaults to the value of [time\(\)](#).

Return Values

[Report a bug](#)

Returns a formatted date string. If a non-numeric value is used for *timestamp*, FALSE is returned and an E\_WARNING level error is emitted.

Errors/Exceptions

[Report a bug](#)

Every call to a date/time function will generate a E\_NOTICE if the time zone is not valid, and/or a E\_STRICT or E\_WARNING message if using the system settings or the TZ environment variable. See also [date\\_default\\_timezone\\_set\(\)](#)

Changelog

[Report a bug](#)

Version	Description
5.1.0	The valid range of a timestamp is typically from Fri, 13 Dec 1901 20:45:54 GMT to Tue, 19 Jan 2038 03:14:07 GMT. (These are the dates that correspond to the minimum and maximum values for a 32-bit signed integer). However, before PHP 5.1.0 this range was limited from 01-01-1970 to 19-01-2038 on some systems (e.g. Windows).
5.1.0	Now issues the E_STRICT and E_NOTICE time zone errors.
5.1.1	There are useful <a href="#">constants</a> of standard date/time formats that can be used to specify the <i>format</i> parameter.

Examples

[Report a bug](#)

Example #1 date() examples

```
<?php
// set the default timezone to use. Available since PHP 5.1
date_default_timezone_set('UTC');

// Prints something like: Monday
echo date("l");

// Prints something like: Monday 8th of August 2005 03:12:46 PM
echo date('l jS \of F Y h:i:s A');

// Prints: July 1, 2000 is on a Saturday
echo "July 1, 2000 is on a " . date("l", mktime(0, 0, 0, 7, 1, 2000));

/* use the constants in the format parameter */
// prints something like: Wed, 25 Sep 2013 15:28:57 -0700
echo date(DATE_RFC2822);

// prints something like: 2000-07-01T00:00:00+00:00
echo date(DATE_ATOM, mktime(0, 0, 0, 7, 1, 2000));
?>
```

You can prevent a recognized character in the format string from being expanded by escaping it with a preceding backslash. If the character with a backslash is already a special sequence, you may need to also escape the backslash.

Example #2 Escaping characters in date()

```
<?php
// prints something like: Wednesday the 15th
echo date('l \t\h\e jS');
?>
```

It is possible to use **date()** and [mktime\(\)](#) together to find dates in the future or the past.

Example #3 date() and mktime() example

```
<?php
$tomorrow = mktime(0, 0, 0, date("m") , date("d")+1, date("Y"));
$lastmonth = mktime(0, 0, 0, date("m")-1, date("d"), date("Y"));
$nextyear = mktime(0, 0, 0, date("m"), date("d"), date("Y")+1);
?>
```

**Note:**  
This can be more reliable than simply adding or subtracting the number of seconds in a day or month to a timestamp because of daylight saving time.

Some examples of **date()** formatting. Note that you should escape any other characters, as any which currently have a special meaning will produce undesirable results, and other characters may be assigned meaning in future PHP versions. When escaping, be sure to use single quotes to prevent characters like \n from becoming newlines.

Example #4 date() Formatting

```
<?php
// Assuming today is March 10th, 2001, 5:16:18 pm, and that we are in the
// Mountain Standard Time (MST) Time Zone

$today = date("F j, Y, g:i a");           // March 10, 2001, 5:16 pm
$today = date("m.d.y");                  // 03.10.01
$today = date("j, n, Y");                 // 10, 3, 2001
$today = date("Ymd");                     // 20010310
$today = date('h-i-s, j-m-y, it is w Day'); // 05-16-18, 10-03-01, 1631 1618 6 Satpm01
$today = date('\i\t \i\s \t\h\e jS \d\ay.'); // it is the 10th day.
$today = date("D M j G:i:s T Y");         // Sat Mar 10 17:16:18 MST 2001
$today = date('H:m:s \m \i\s\ \m\o\n\t\h'); // 17:03:18 m is month
$today = date("H:i:s");                   // 17:16:18
$today = date("Y-m-d H:i:s");             // 2001-03-10 17:16:18 (the MySQL DATETIME format)
?>
```