

date() function:

- Syntax:
`date(string $format, ?int $timestamp = null): string`
- Returns a string formatted according to the given format string using the given integer timestamp or the current time if no timestamp is given. In other words, timestamp is optional and defaults to the value of `time()`.
- 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'`.
- Timestamp
The optional timestamp parameter is an int Unix timestamp that defaults to the current local time if timestamp is omitted or null. In other words, it defaults to the value of `time()`.
- Return Values
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
Every call to a date/time function will generate a `E_WARNING` if the time zone is not valid. See also `date_default_timezone_set()`
- **Examples**
 - Example #1 date() examples

```
<?php
// set the default timezone to use.
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);
?>

```

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\a\y."); // 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)  
?>
```