Chapter 10

How to work with dates

Objectives

Applied

1. Use any of the functions, methods, and techniques that are presented in this chapter to work with dates, timestamps, and date intervals.

Objectives (continued)

Knowledge

1. Describe a timestamp.
2. Describe the Y2K38 problem, and explain how the use of DateTime objects solves this problem.
3. Describe the use of the PHP functions for working with timestamps.
4. Describe the use of the DateTime and DateInterval objects.

The date function

```
date($format[, $ts])
```

Returns the current date/time as string in the format specified by the code used for $format.

```
echo date('Y');          //2012
```

The optional second parameter is the timestamp for a date other than the current date. Note that this is a signed integer representing the number of seconds since 12:00am January 1, 1970 GMT

```
echo date('Y-m-d', 1331843400);   //2012-03-15
```

Common format codes for the date function

Character	Description
`D`	Day of week – three letters
`l`	Day of week – full name
`n`	Month – no leading zero
`m`	Month – leading zero
`M`	Month – three letters
`F`	Month – full name
`j`	Day of month – no leading zero
`d`	Day of month – leading zero
`Y`	Year – four digits
`L`	Leap year (1) or not (0)
`g`	Hours – 12-hour format, no leading zero
`h`	Hours – 24-hour format, no leading zero

Common codes for the date function (cont.)

Character	Description
`G`	Hours – 12-hour format, leading zero
`H`	Hours – 24-hour format, leading zero
`i`	Minutes – leading zero
`s`	Seconds – leading zero
`a`	am/pm – lowercase
`A`	AM/PM – uppercase
`T`	Time zone abbreviation
`U`	Seconds since Unix epoch

How to format a timestamp

```
$date1 = date('n/j/Y');                    // 3/15/2011
$date2 = date('Y-m-d');                    // 2011-03-15
$date3 = date('l, F d, Y');  // Monday, March 15, 2011
$date4 = date('g:i a');                    // 1:30 pm
$date5 = date('H:i:s');                    // 13:30:00
$date6 = date('Y-m-d \a\t H:i:s');
                                  // 2011-03-15 at 13:30:00
$date7 = date('Y-m-d', 1331843400);   // 2012-03-15
```

URL for a list of all PHP timestamp functions

```
http://www.php.net/manual/en/ref.datetime.php
```

Functions for working with timestamps

```
time()  //the current date/time as a timestamp

mktime([$h[, $m[, $s[, $M[, $D[, $Y]]]]]]) //returns a
timestamp based on the date/time given through the
parameters

checkdate($M, $D, $Y)

getdate([$ts])  //returns an array containing the parts
of the current date or the timestamp parameter
```

How to create a timestamp

```
$now = time();

$expires = mktime(13, 30, 0, 3, 15, 2012);

$expires = mktime(13, 30, 0, 3, 15);

$expires = mktime(13, 30, 0, 3);
```

How to validate a date

```
$valid_date = checkdate(11, 31, 2012);
```

How to validate a time

A custom function for validating time

```
function checktime($h, $m, $s) {
    return $h >= 0 && $h < 24 && $m >= 0 && $m < 60
           && $s >= 0 && $s < 60;
}
```

A statement that calls the custom function

```
$valid_time = checktime(12, 30, 0);
```

How to get the parts of a timestamp

```
$expires = mktime(13, 30, 0, 3, 15, 2012);
$parts = getdate($expires);
$year    = $parts['year'];     // 2012
$mon     = $parts['mon'];      // Month number - 3
$month   = $parts['month'];    // Month name - 'March'
$mday    = $parts['mday'];     // Day of month - 15
$weekday = $parts['weekday'];  // Weekday - 'Monday'
$wday    = $parts['wday'];     // Weekday as number - 1
$hours   = $parts['hours'];    // Hours - 13
$minutes = $parts['minutes'];  // Minutes - 30
$seconds = $parts['seconds'];  // Seconds - 0
```

The strtotime function

```
strtotime($str[, $ts])  //generates a timestamp from a
string which specifies a date.
```

Types of templates used in strtotime

- Absolute: the given date or time. If time is omitted, the default time is midnight.
- Relative: the date or time offset from the base

Generating a timestamp with an absolute template

```
// Current time of Sun 04/08/2012 1:30:00 pm
$date1 = strtotime('2013-06-01'); // Sat 06/01/2013 00:00
$date2 = strtotime('6/1/2013');   // Sat 06/01/2013 00:00
$date3 = strtotime('Jun 1');      // Fri 06/01/2012 00:00
$date4 = strtotime('8:45');       // Sun 04/08/2012 08:45
$date5 = strtotime('8am');        // Sun 04/08/2012 08:00
$date6 = strtotime('2013-02-29 8:45am');
                                  // Fri 03/01/2013 08:45
```

Generating a timestamp with a relative template

```
// Current time of Sun 04/08/2012 1:30:00 pm
$date1 = strtotime('+1 hour');   // Sun 04/08/2012 14:30
$date2 = strtotime('-2 days');   // Sat 04/06/2012 13:30
$date3 = strtotime('tomorrow');  // Mon 04/09/2012 00:00
$date4 = strtotime('tomorrow 10:15am');
                                 // Mon 04/09/2012 10:15
$date5 = strtotime('next sunday');
                                 // Sun 04/15/2012 00:00
$date6 = strtotime('last day of');
                                 // Mon 04/30/2012 13:30
$date7 = strtotime('first day of next month');
                                 // Tue 05/01/2012 13:30
$date8 = strtotime('third wednesday of');
                                 // Wed 04/18/2012 00:00
$date9 = strtotime('nov second tue of 8am');
                                 // Tue 11/13/2012 08:00
```

How to modify a timestamp

```
$checkout = mktime(13, 30, 0, 4, 8, 2012);

$due_date = strtotime('+3 weeks 6pm', $checkout);
```

Determining if a year is a leap year

```
function is_leapyear($ts) {
    return (date('L', $ts) == '1');
}
$year_2010 = is_leapyear(strtotime('2010-1-1'));
$year_2012 = is_leapyear(strtotime('2012-1-1'));
```

Displaying a message about an expiration date

```
$now = time();
$exp = strtotime('2012-4 first day of next month midnight');
if ($exp < $now) {
    echo 'Your card has expired.';
} else {
    echo 'Your card has not expired.';
}
```

Displaying a message about an expiration date

```
$now = time();
$exp = '04/2012';    // Typical expiration date format

// Change exp format from mm/yyyy to yyyy-mm
$month = substr($exp, 0, 2);
$year  = substr($exp, 3, 4);
$exp = $year . '-' . $month;

// Set exp date; calculate number of days from current date
$exp = strtotime($exp .
                 ' first day of next month midnight');
$days = floor(($exp - $now) / 86400);  //864000 seconds/day

// Display a message
if ($days < 0) {
    echo 'Your card expired ' . abs($days) . ' days ago.';
} else if ($days > 0) {
    echo 'Your card expires in ' . $days . ' days.';
} else {
    echo 'Your card expires at midnight.';
}
```

Displaying a countdown until the New Year

```
$now = time();
$new_year = strtotime('next year Jan 1st', $now);

// Calculate the days, hours, minutes, and seconds
$seconds = $new_year - $now;
$days = floor($seconds / 86400);
$seconds -= $days * 86400;
$hours = floor($seconds / 3600);
$seconds -= $hours * 3600;
$minutes = floor($seconds / 60);
$seconds -= $minutes * 60;

// Display the countdown
echo "$days days and $hours:$minutes:$seconds to New Year's.";
```

The DateTime class provides an object oriented way to work with dates and times.

Each part of the date and time is stored as a separate signed integer.

How to create a DateTime object

Using the current date and time

```
$now = new DateTime();
```

Using a strtotime format string

```
$expires = new DateTime('2012-03-15 13:30:00');
$tomorrow = new DateTime('+1 day');
$due_date = new DateTime('+3 weeks');
$appointment = new DateTime('next Friday +1 week 13:30');
```

Methods of a DateTime object

```
format($format) //same codes as the date function

setTime($h, $m, $s)

setDate($y, $m, $d)

modify($str)  //similar to strtotime relative template

getTimestamp()

setTimestamp($ts)
```

How to use the methods of a DateTime object

How to copy a DateTime object

```
$invoice_date = new DateTime('2012-03-15 13:30:00');
$due_date = clone $invoice_date;  //clone is a keyword
```

How to set the time and date of a DateTime object

```
$due_date->setTime(22, 30, 0);
$due_date->setDate(2012, 3, 15);
```

How to modify a DateTime object

```
$due_date->modify('+3 weeks');
```

How to display a DateTime object

```
echo 'Payment Due: ' .
     $due_date->format('M. j, Y \a\t g:i a');
```

How to convert a timestamp to a DateTime object

```
$tomorrow = strtotime('tomorrow 8am');
$nextday = new DateTime();
$nextday->setTimestamp($tomorrow);
```

A DateInterval is a span of time rather than a point in time. To create a DateInterval object:

```
$interval = new DateInterval('P30D');
```

The parts of the interval string

Part	Description
P	Begins the interval code
***n*Y**	Specifies the number of years
***n*M**	Specifies the number of months
***n*W**	Specifies the number of weeks
***n*D**	Specifies the number of days
T	Starts the time portion of the interval code
***n*H**	Specifies the number of hours
***n*M**	Specifies the number of minutes
***n*S**	Specifies the number of seconds

How to use interval strings

```
$interval_1 = new DateInterval('P1Y2M10D');
$interval_2 = new DateInterval('PT1H2M3S');
$interval_3 = new DateInterval('P1Y2M3DT1H2M3S');
```

The format method of a DateInterval object

```
format($format)
```

Format codes for the DateInterval format method

Code	Description
`%R`	Sign of the interval (+ or -)
`%y`	Years
`%m`	Months
`%d`	Days
`%h`	Hours
`%i`	Minutes
`%s`	Seconds

How to display a date interval

```
echo $interval_1->format('%m months, %d days');
echo $interval_1->format('%R %M months');
echo $interval_1->format('%R %y %m %d %h %i %s');
echo $interval_1->format('%R%yy %mm %dd %H:%I:%S');
```

Methods of a DateTime object that use DateInterval objects

```
add($interval)
sub($interval)
diff($date)
```

Adding a DateInterval object to a DateTime object

```
$checkout_length = new DateInterval('P3W');
$due_date = new DateTime();
$due_date->add($checkout_length);
```

Subtracting a DateInterval object from a DateTime object

```
$voting_age = new DateInterval('P18Y');
$dob = new DateTime();
$dob->sub($voting_age);
echo 'You can vote if you were born on or before ' .
      $dob->format('n/j/Y');
```

How to determine the time between two dates

```
$now = new DateTime('2012-05-15 12:45:00');
$due = new DateTime('2012-04 last day of midnight');
$time_span = $now->diff($due);
echo $time_span->format('%R%dd %H:%I:%Sh');
```

Determining if a year is a leap year

```
function is_leap($date) {
    return ($date->format('L') == '1');
}
$year_2010 = is_leap(new DateTime('2010-1-1'));
$year_2012 = is_leap(new DateTime('2012-1-1'));
```

Displaying a message about an expiration date

```
$now = new DateTime();
$exp = new DateTime(
       '2012-4 first day of next month midnight');
if ($exp < $now) {
    echo 'Your card has expired.';
} else {
    echo 'Your card has not expired.';
}
```

Displaying a message about an expiration date

```
$now = new DateTime();
$exp = '04/2012';

$month = substr($exp, 0, 2);
$year  = substr($exp, 3, 4);
$exp = $year . '-' . $month;

$exp = new DateTime($exp .
       ' first day of next month midnight');
$span = $now->diff($exp);

// Display a message
$span_text  = $span->format(
              '%y years, %m months, and %d days');
if ($span->format('%R') == '-') {
    echo 'Your card expired ' . $span_text . ' ago.';
} else {
    echo 'Your card expires in ' . $span_text . '.';
}
```

Displaying a countdown until the New Year

```
$now = new DateTime();
$new_year = new DateTime('next year Jan 1st');

// Calculate and format the time left until the new year
$span = $now->diff($new_year);
$md_left = $span->format('%m months, %d days');
$hms_left = $span->format('%h:%I:%S');

// Display a message
if ($now->format('MD') == '0101') {
    echo 'Happy New Year!';
} else if ($now->format('MD') == '1231') {
    echo "$hms_left remaining to the New Year.";
} else {
    echo "$md_left, and $hms_left remaining to the New Year.";
}
```