

10 Date Time Functions in PHP:

1- date_add:

When using `DateTime::add()` be careful that additions over Summertime changes will not always produce the expected results. For instance, adding a day (interval = P1D) is probably expected to keep the same time when added to a date even over a summertime change. But adding 24 hours (interval = PT24H) does not seem to take into effect the time change.

When then checking the time difference after adding 24 hours after the clocks went forward, the time difference is only 23 hours.

Example:

```
<?php
date_default_timezone_set('Europe/London');

$diff1Day = new DateInterval('P1D');
$diff24Hours = new DateInterval('PT24H');
$diff1440Minutes = new DateInterval('PT1440M');

// Clocks changed at 2014-03-30 02:00:00

$d0 = new DateTime('2014-03-29 08:00:00');

$d1 = new DateTime('2014-03-29 08:00:00');
// Add 1 day - expect time to remain at 08:00
$d1->add($diff1Day);
print_r($d1);

$d2 = new DateTime('2014-03-29 08:00:00');
// Add 24 hours - expect time to be 09:00
$d2->add($diff24Hours);
print_r($d2);

$seconds = $d1->getTimestamp() - $d0->getTimestamp();
echo "Difference in Hours: " . $seconds / (60 * 60) . "\n";
?>
```

2- checkdate:

Checks the validity of the date formed by the arguments. A date is considered valid if each parameter is properly defined.

- Syntax:
checkdate(int \$month, int \$day, int \$year): bool

Example:

```
<?php
var_dump(checkdate(12, 31, 2000));
var_dump(checkdate(2, 29, 2001));
?>
```

3- date_create:

This function is an alias of: DateTime::__construct()

Example:

If you want to create the DateTime object directly from a timestamp use this

```
<?php
$st = 1170288000 // a timestamp
$dt = new DateTime("@$st");
?>
```

4- date_diff:

Powerful Function to get two date difference.

////////////////////////////////////

PARA: Date Should In YYYY-MM-DD Format

RESULT FORMAT:

'%y Year %m Month %d Day %h Hours %i Minute %s Seconds' => 1 Year 3 Month 14 Day
11 Hours 49 Minute 36 Seconds

'%y Year %m Month %d Day' => 1 Year 3 Month 14 Days

'%m Month %d Day' => 3 Month 14 Day

'%d Day %h Hours' => 14 Day 11 Hours

'%d Day' => 14 Days

'%h Hours %i Minute %s Seconds' => 11 Hours 49 Minute 36 Seconds

'%i Minute %s Seconds' => 49 Minute 36 Seconds

'%h Hours' => 11 Hours

'%a Days' => 468 Days

Example:

```
<?php
function dateDifference($date_1 , $date_2 , $differenceFormat = '%a' )
{
    $datetime1 = date_create($date_1);
    $datetime2 = date_create($date_2);

    $interval = date_diff($datetime1, $datetime2);
```

```
        return $interval->format($differenceFormat);
    }
?>
```

5- date_modify:

Alter the timestamp of a DateTime object by incrementing or decrementing in a format accepted by DateTimeImmutable::__construct().

Example:

\$cday - specified day of the week (0-6 where 0 is Sunday)

\$currentDate - date of start

\$endDate - date of end

We need dates of next couple of days, that day of week match defined.

```
<?php
```

```
    if($currentDate->format('w')!= $cday){
        switch ($cday){
            case 0 : $cdays="Sunday"; break;
            case 1 : $cdays="Monday"; break;
            case 2 : $cdays="Tuesday"; break;
            case 3 : $cdays="Wednesday"; break;
            case 4 : $cdays="Thursday"; break;
            case 5 : $cdays="Friday"; break;
            case 6 : $cdays="Saturday";
        }
        date_modify($currentDate,"+1 {$cdays}");
    }
```

```
    while($currentDate < $endDate) {
        echo $currentDate -> format('Y-m-d H:i:s');
        $currentDate -> modify('+1 week');
    }
```

```
?>
```

6- date_format:

Returns date formatted according to given format

Example:

A simple way to get eBay, or Amazon, web service timestamps is as follows:

```
<?php
$current_time = urlencode(substr(date("c"), 0, 19)."Z");

?>
```

In other words, take the date/time of now (in ISO 8601 format), discard the trailing Daylight Savings Time specifier, add a "Z" where the DST was and urlencode the whole thing to convert the time's colons for REST requests (required for amazon, not sure about eBay).

Another way might be to create your own timestamp:

```
<?php
$current_time = urlencode(date("Y-m-d")."T".date("H:i:s")."Z");

?>
```

This way however takes a little more coding on the line.

As far as performance goes, I'm not sure which may be quicker. I just like things to work and work well, don't much care for how fast they are as long as they get the job done :)

A much simpler way to get the eBay, or Amazon, web service timestamp is as follows:

```
<?php
$current_date = gmdate("Y-m-d\TH:i:sZ");

?>
```

- 7- date_parse:
date_parse — Returns associative array with detailed info about given date/time
syntax: date_parse(string \$datetime): array

Example:

```
<?php
print_r(date_parse("2006-12-12 10:00:00.5"));

?>
```

- 8- date_sub:

Subtracts an amount of days, months, years, hours, minutes and seconds from a DateTime object

Example:

```
<?php
$today = date_create(date('Y-m-d'));
$yesterday = date_sub($today, date_interval_create_from_date_string("1 days"));

    echo var_dump($today);
    echo var_dump($yesterday)
?>
```

9- date:

Format a local time/date

Syntax: date(string \$format, ?int \$timestamp = null): string

Example:

```
<?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));
?>
```

10- time:

Return current Unix timestamp

Syntax: time(): int

Example:

```
<?php
$nextWeek = time() + (7 * 24 * 60 * 60);
            // 7 days; 24 hours; 60 mins; 60 secs
echo 'Now:      '. date('Y-m-d') ."\n";
echo 'Next Week: '. date('Y-m-d', $nextWeek) ."\n";
// or using strtotime():
echo 'Next Week: '. date('Y-m-d', strtotime('+1 week')) ."\n";
?>
```