

# *Manejo de fechas con PHP*

---

## Índice

---

Índice .....	2
1. MANEJO DE FECHAS CON PHP .....	3
Funciones más usadas en PHP .....	3
time() .....	3
date() .....	3
mktime(hora, min, seg, mes, día, año) .....	4
bool checkdate ( int \$month , int \$day , int \$year ) .....	4
array getdate ( [ int \$timestamp = time() ] ) .....	4
EJEMPLOS COMUNES DE MANEJO DE FECHAS CON PHP .....	5
Calcular una fecha después de un plazo .....	5
Calcular una edad .....	5

## 1. MANEJO DE FECHAS CON PHP

Trabajar con horas y fechas en PHP es necesario-. Hay que tener en cuenta que PHP funciona con el formato de fecha UNIX por lo que tendremos que realizar transformaciones para podernos ajustar a las fechas que nos viene desde los formularios o bases de datos.

### Funciones más usadas en PHP

#### time()

Lo más adecuado en todos los casos suele ser almacenar la fecha en formato unix en una variable:

```
$fecha = time();
```

```
echo $fecha;
```

De este modo guardamos en la variable \$fecha todos los segundos que han transcurrido desde las doce de la noche del 1 de enero de 1970. \$fecha será, por tanto, un número entero.

#### date()

Para sacar partido a los datos recogidos con la función time() tenemos la función "date()" para que muestre la fecha u hora en el formato que nos interese:

```
$fecha_con_formato = date("formato",$fecha);
```

```
echo $fecha_con_formato;
```

Los parámetros más importantes que podemos introducir en "formato" son los siguientes:

Parámetro	Descripción	Parámetro	Descripción
d	Día del mes con ceros iniciales (De 01 a 31).	h	Hora en formato 12-horas con ceros iniciales.
j	Día del mes sin ceros iniciales (De 1 a 31).	H	Hora en formato 24-horas con ceros iniciales.
m	Número del mes con ceros iniciales (De 01 a 12).	i	Minutos con ceros iniciales.
n	Número del mes sin ceros iniciales (De 1 a 12).	s	Segundos con ceros iniciales.
Y	Año con cuatro dígitos (Ejemplo: 2007).	A	AM o PM.
y	Año con dos dígitos (Ejemplo: 07).	T	Configuración de zona horaria del servidor.

Puedes ver [aquí](#) una lista de todos los disponibles).

Ejemplo: Si es el 19 de diciembre de 2012 y son las 12:25, el siguiente código:

```
$fecha_con_formato = date("d/m/Y -- H:i:s",$fecha);
```

nos devolverá:

```
19/12/2012-- 12:25:33
```

## mktime(hora, min, seg, mes, día, año)

Devuelve la marca de tiempo Unix correspondiente a los argumentos dados. Esta marca de tiempo es un entero que contiene el número de segundos entre la Época Unix (1 de Enero del 1970 00:00:00 GMT) y el instante especificado.

Los argumentos pueden omitirse de derecha a izquierda; cualquier argumento que se omita será establecido al valor actual según la fecha y hora locales.

```
<?php
echo date("M-d-Y", mktime(0, 0, 0, 12, 32, 1997));
echo date("M-d-Y", mktime(0, 0, 0, 13, 1, 1997));
echo date("M-d-Y", mktime(0, 0, 0, 1, 1, 1998));
echo date("M-d-Y", mktime(0, 0, 0, 1, 1, 98));
?>
```

Devolverá:

Jan-01-1998

Jan-01-1998

Jan-01-1998

Jan-01-1998

## bool checkdate ( int \$month , int \$day , int \$year )

Comprueba la validez de una fecha formada por los argumentos. Una fecha se considera válida si cada parámetro está propiamente definido.

```
<?php
$fecha_valida = checkdate(2, 29, 2012);
if ($fecha_valida=='true')
    echo 'ok';
else
    echo 'no';
?>
```

Devolverá ok porque 2012 fue bisiesto

## array getdate ([ int \$timestamp = time() ] )

Devuelve un array asociativo que contiene la información de la fecha de *timestamp*, o el momento local actual si no se da *timestamp*.

"seconds"	Representación numérica de los segundos	0 a 59
"minutes"	Representación numérica de los minutos	0 a 59
"hours"	Representación numérica de las horas	0 a 23
"mday"	Representación numérica del día del mes	1 a 31
"wday"	Representación numérica del día de la semana	0 (para Domingo) hasta 6 (para Sábado)
"mon"	Representación numérica de un mes	1 hasta 12
"year"	Una representación numérica completa de un año, 4 dígitos	Ejemplos: 1999 o 2003
"yday"	Representación numérica del día del año	0 hasta 365
"weekday"	Una representación textual completa del día de la semana	Sunday hasta Saturday

"month"	Una representación textual completa de un mes, como January o March	January hasta December
0	Los segundos desde la Época Unix, similar a los valores devueltos por <a href="#">time()</a> y usados por <a href="#">date()</a> .	Dependiente del Sistema, típicamente -2147483648 hasta 2147483647.

```
<?php
$hoy = getdate();
print_r($hoy);
?>
```

Devolverá

Array ( [seconds] => 25 [minutes] => 12 [hours] => 23 [mday] => 19 [wday] => 6 [mon] => 1 [year] => 2013 [yday] => 18 [weekday] => Saturday [month] => January [0] => 1358633545 )

## EJEMPLOS COMUNES DE MANEJO DE FECHAS CON PHP

### Calcular una fecha después de un plazo

Vamos a calcular cuál será la fecha en un plazo de 2 meses de hoy. Los pasos son los siguientes:

```
1 <?php
   $hoy = getdate();
2   $dos_meses_mas_tarde = mktime(12, 0, 0, $hoy['mon']+2, $hoy['mday'], $hoy['year']);
3   echo (date ("d-m-y", $dos_meses_mas_tarde));
4   ?>
```

Devolverá una fecha dos meses después de la actual.

Vamos a calcular la fecha actual 90 días después de la fecha actual:

```
<?php

$today = getdate();

$month = $today['mon'];
$day = $today['mday'];
$year = $today['year'];

$garantia_expira = mktime(12, 0, 0, $month, $day+90, $year);

echo (date ("d-m-Y", $garantia_expira));
?>
```

Vamos a tomar hoy, por ejemplo, es 20 enero de 2012. La operación aritmética que se llevará a cabo es de 20 + 90, que será igual a 112. La función tiene en cuenta esto y devolverá: 20-04-2013

### Calcular una edad

Un ejemplo de cómo calcular las fechas entre 2 fechas son las siguientes:

```
<?php
```

```
$month = 28;
$day = 8;
$year = 1985;

$creation_ftime = mktime(0,0,0, $month, $day, $year);
$now_ftime = time();

$age_ftime = $now_ftime - $creation_ftime;
$product_age = floor($age_ftime / (365 * 24 * 60 * 60));

echo 'El producto fue hecho hace '. $product_age .' año(s) atrás';
```

?>

## Ejercicios

---

1. Realiza una función que acepte una fecha como cadena con el formato dd-mm-aaaa compruebe si la fecha es correcta y nos devuelva la fecha en formato UNIX.
2. Realiza una función que acepte una fecha como cadena con el formato aaaa-mm-dd compruebe si la fecha es correcta y nos devuelva la fecha en formato UNIX.
3. Realiza una función que reciba la fecha en formato UNIX y devuelva la fecha en formato dd-mm-aaaa y aaaa-mm-dd según un segundo argumento que le pasemos a la función.
4. Realiza una función que nos devuelva el nº de días que han pasado entre dos fechas. Hay que tener en cuenta que tendremos que validar las fechas antes o durante la función.
5. Realiza una función que muestre una imagen diferente según la estación del año. Para facilitarlo podemos tener en cuenta como primavera (marzo, abril, mayo) y así sucesivamente.