



APRENDERAPROGRAMAR.COM

FUNCIONES DE TIEMPO,
FECHA Y HORA PHP: TIME,
DATE, FORMATOS DE
FECHA. MKTIME Y
GMMKTIME. EJEMPLOS.
(CU00830B)

Sección: Cursos

Categoría: Tutorial básico del programador web: PHP desde cero

Fecha revisión: 2029

Resumen: Entrega nº30 del Tutorial básico "PHP desde cero".

Autor: Enrique González Gutiérrez y César Krall

FUNCIONES PARA MANEJO DEL TIEMPO, FECHA Y HORAS PHP

Con frecuencia cuando creamos webs o apps tenemos que trabajar con fechas y calendarios. Por ejemplo en la página web de un hotel o un restaurante es posible que trabajemos con fechas y horas de comienzo de reserva, de fin de reserva, etc. PHP dispone de funciones nativas para facilitar el trabajo con fechas, horas y tiempos.



El manejo del "tiempo" es un aspecto controvertido en la programación. La primera dificultad y más obvia es que existen cientos de dispositivos electrónicos y no todos manejan la misma fecha. Por ejemplo mi computador personal puede indicar que hoy es lunes 25 de enero de 2048 y la hora es las 12:35:44 mientras que el servidor al que estoy conectado, suponiendo que está en mi misma ciudad, puede indicar que hoy es lunes 25 de enero de 2048 y la hora es las 12:33:21. Otro servidor, en otro país, puede indicar que la fecha es domingo 24 de enero de 2048 y la hora es las 23:51:15 debido a la diferencia horaria entre países. Aquí, claro está, nos estaríamos refiriendo a horas locales. Para poder disponer de horas de referencia globales para todo el mundo se crearon estándares como:

a) GMT o tiempo medio de Greenwich: ha sido un estándar ampliamente usado. GMT era el tiempo medido en el observatorio británico de Greenwich, que se definió como meridiano cero de la tierra. Así cada país podía expresar su hora en función de la hora del meridiano de Greenwich. Por ejemplo GMT+0 indicaba que la hora era la misma que la hora oficial en el meridiano de Greenwich, mientras que GMT+3 indicaba que la hora eran 3 horas más que en el meridiano de Greenwich (es decir, si es Greenwich eran las 09:00 en un país cuya hora fuera GMT+3 serían las 12:00).

b) UTC o tiempo universal coordinado: es el estándar que se ha adoptado como referencia que indica un tiempo único independientemente de en qué lugar del planeta nos encontremos, basado en las mediciones de relojes atómicos distribuidos por distintos países. Es el estándar que se está imponiendo en los sistemas informáticos.

La circunferencia terrestre se dividió en 24 husos horarios de modo que cada huso quedaba referenciado a un tiempo común (lo que se denominaba el tiempo de Greenwich).



Aunque desde el punto de vista de su definición técnica UTC y GMT no son lo mismo, a efectos prácticos hablar de GMT+3 es lo mismo que hablar de UTC+3.

A pesar de este gran avance, el trabajo con el tiempo sigue presentando grandes dificultades en los sistemas informáticos, tanto por la falta de sincronización entre dispositivos como por la disparidad en cuanto a cómo medir el tiempo y con qué grado de precisión, existencia de horarios de verano e invierno, etc. Por ello encontrarás que es relativamente frecuente encontrar que a medida que los lenguajes de programación evolucionan vayan introduciendo cambios en la forma de manejar el tiempo.

Una cuestión a tener en cuenta es que el tiempo oficial de Greenwich, tiempo GMT ó UTC, no se corresponde con el tiempo local en Greenwich debido a la existencia del horario de verano. Por ejemplo, Lisboa se encuentra en el huso horario de Greenwich, pero en verano la hora local está adelantada una hora respecto al tiempo oficial de Greenwich por motivos de ahorro energético. Esto da lugar a que el 5 de agosto a las 21:00 en Lisboa se corresponda con el 5 de agosto 20:00 UTC debido al horario de verano, a pesar de que Lisboa se encuentre en el mismo huso horaria que Greenwich.

FUNCIONES TIME Y DATE

Estas dos funciones nos permitirán mostrar o capturar la fecha y hora, lo cual tiene una gran cantidad de aplicaciones. Por ejemplo, supón que tienes una tienda on-line y para cada operación de venta tienes que registrar la fecha y hora de la venta. Estas funciones resultarán útiles para ello.

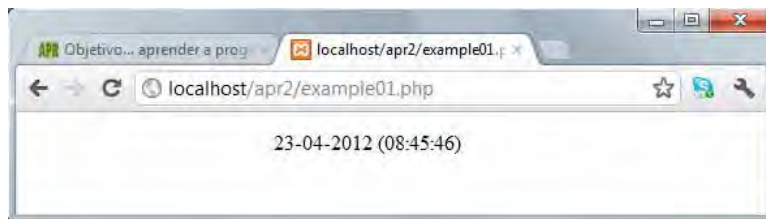
La función `time` devuelve la hora GMT actual medida como el número de segundos desde el 1 de Enero de 1970 00:00:00 GMT (hora del meridiano de Greenwich) obtenidos a partir de la hora actual local del servidor. Cuando hablamos de "hora actual del servidor" nos referimos a una hora local, que es distinta según el país donde nos encontremos. En cambio la hora GMT es única y no depende del país. Hay que tener cuidado con esta circunstancia. Por ejemplo, si trabajas con un servidor localizado en Estados Unidos lo más probable es que el servidor trabaje con la hora de Estados Unidos. Si quieres obtener la hora local de tu país, tendrás que tenerlo en cuenta para restarle o sumarle cierto número de horas a la hora que te devuelva el servidor.

Por otro lado, la función `date` muestra la fecha en el formato que le indiquemos.

En general, te recomendamos que hagas pruebas para comprobar que los resultados de fecha y hora obtenidos son los adecuados, ya que la configuración de cada servidor y la versión de PHP empleada puede afectar a los resultados.

Escribe ahora este código y guárdalo con un nombre de archivo como `ejemplo2.php`. A continuación, sube el fichero al servidor y visualiza el resultado.

```
<?php //Ejemplo curso PHP aprenderaprogramar.com
$time = time();
echo date("d-m-Y (H:i:s)", $time);
?>
```



En este ejemplo hemos especificado como formato para mostrar la fecha d-m-Y (H:i:s) (donde d representa día, m representa mes, Y representa año, H representa hora, i representa minutos y s representa segundos) desde el valor de time u otra fecha dada en segundos desde el 1 de Enero de 1970 00:00:00 GMT.

Como podemos observar, la función date da formato a la variable en segundos \$time. Hay distintos formatos para la función date, a continuación indicaremos los más importantes.

La función time() devuelve un valor numérico entero largo, por ejemplo 1335169779. Este número representa el número de segundos transcurridos desde el 1 de enero de 1970 00:00:00 GMT.

Para transformar ese número en una fecha "entendible por las personas" usamos la función date, cuya sintaxis general es: date ("formato de salida", valorTimeValido)

En cuanto a valorTimeValido, será un número entero, generalmente contenido en una variable.

En cuanto a "formato de salida", disponemos de las siguientes equivalencias:

Para representar el día de la semana:

Carácter de formato	Significado aprenderaprogramar.com	Valores que devuelve
d (minúscula)	Día del mes, con 2 dígitos. Si es menor de 10, lleva un 0 inicial.	01 a 31
D (mayúscula)	Texto representativo del día, con 3 letras y en inglés	Mon (lunes) hasta Sun (domingo)
j (minúscula)	Día del mes con un dígito (sin ceros iniciales)	1 a 31
l (letra ele minúscula)	Texto representativo del día, con todas las letras y en inglés	Monday (lunes) hasta Sunday (domingo)
N (mayúscula)	Representación numérica del día de la semana según ISO-8601, siendo 1 el lunes y 7 el domingo	1 (lunes) a 7 (domingo)
S (mayúscula)	Sufijo para el día del mes en inglés	st, nd, rd ó th
w (minúscula)	Representación numérica del día de la semana siendo 0 domingo y 6 el sábado	0 a 6
z (minúscula)	Día del año desde 0 hasta 365 (un año normal irá de 0 a 364 y uno bisiesto de 0 a 365)	0 a 365

Para representar el mes:

Carácter de formato	Significado aprenderaprogramar.com	Valores que devuelve
F (mayúscula)	Texto representativo del mes, con todas las letras y en inglés	January (enero) a December (diciembre)
m (minúscula)	Mes en formato numérico con dos dígitos, incluyendo cero inicial si es menor que 10	01 hasta 12
M (mayúscula)	Texto representativo del mes, con tres letras iniciales en inglés	Jan (enero) hasta Dec (diciembre)
n (minúscula)	Mes en formato numérico con un dígito, sin ceros iniciales	1 hasta 12
t (minúscula)	Número de días que tiene el mes	28 hasta 31

Para representar el año:

Carácter de formato	Significado aprenderaprogramar.com	Valores que devuelve
L (mayúscula)	Indica si un año es bisiesto o no (1 si es bisiesto, 0 si no lo es)	0 ó 1
Y (mayúscula)	Año en formato numérico de 4 dígitos	1000 a 9999
y (minúscula)	Año en formato numérico de 2 dígitos (los dos últimos del año)	El que corresponda, p.ej. 78
o (minúscula)	Número de año según ISO-8601. En general devuelve el mismo resultado que Y mayúscula	1000 a 9999

Para representar la hora:

Carácter de formato	Significado aprenderaprogramar.com	Valores que devuelve
a (minúscula)	Incluye am ó pm en minúsculas detrás de la hora para indicar antes del meridiano o pasado el meridiano	am ó pm
A (mayúscula)	Igual que a pero en mayúsculas	AM ó PM
B (mayúscula)	Hora internet, también llamada Swatch Internet Time o Biel Mean Time (BMT), poco usada.	000 a 999
g (minúscula)	Número de hora desde 1 hasta 12 sin ceros iniciales	1 a 12
G (mayúscula)	Número de hora desde 0 hasta 23 sin ceros iniciales	0 a 23
h (minúscula)	Número de hora desde 01 hasta 12, con ceros iniciales	01 a 12
H (mayúscula)	Número de hora desde 00 hasta 23, con ceros iniciales	00 hasta 23
i (minúscula)	Minutos desde 00 hasta 59, con ceros iniciales	00 a 59
s (minúscula)	Segundos desde 00 hasta 59, con ceros iniciales	00 a 59

Para representar la zona horaria:

Carácter de formato	Significado aprenderaprogramar.com	Valores que devuelve
e (minúscula)	Representa la zona horaria en inglés (p.ej. GMT, Atlantic/Azores, America/Denver)	Según corresponda
I (letra i mayúscula)	Indica si se está en horario de verano (0 indica que no se está en horario de verano, 1 que sí se está en horario de verano).	0 ó 1
O (letra o mayúscula)	Indica la diferencia respecto a la zona horaria de Greenwich u hora GMT, en horas con cuatro dígitos y símbolo.	Por ejemplo +0300 ó -0500
P (mayúscula)	Indica la diferencia respecto a la zona horaria de Greenwich u hora GMT, en horas con cuatro dígitos con separador dos puntos y símbolo.	Por ejemplo +03:00 ó -05:00
T (mayúscula)	Abreviatura de la zona horaria (abreviatura en inglés)	Según corresponda
Z (mayúscula)	Diferencia en segundos entre la hora utilizada y la hora UTC, con signo positivo o negativo	-43200 hasta 50400

Además disponemos de:

La letra W mayúscula para representar la semana del año según ISO-8601, considerando que las semanas comienzan en lunes. Por ejemplo, un año tiene 52 semanas y con este formato podemos obtener que estamos en la semana 24 del año.

La letra c minúscula para representar la fecha y hora completa en formato ISO-8601 (tipo 2089-03-12T12:19:21+03:00).

La letra r minúscula para representar la fecha y hora completa con formato RFC-2822 (tipo Mon, 21 Jan 2089 11:41:05 -0300).

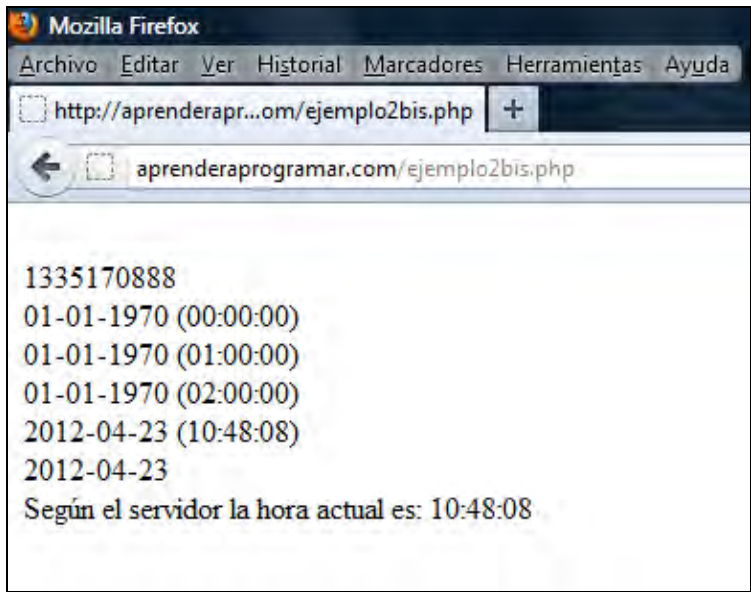
Escribe ahora este código y guárdalo con un nombre de archivo como ejemplo2bis.php. A continuación, sube el fichero al servidor y visualiza el resultado.

```

<?php //Ejemplo curso PHP aprenderaprogramar.com
$time = time();
echo "<br/>";
echo $time;
echo "<br/>";
echo date("d-m-Y (H:i:s)", -3600);
echo "<br/>";
echo date("d-m-Y (H:i:s)", 0);
echo "<br/>";
    
```

```

echo date("d-m-Y (H:i:s)", 3600);
echo "<br/>";
echo date("Y-m-d (H:i:s)", $time);
echo "<br/>";
echo date("Y-m-d ", $time);
echo "<br/>";
echo ("Según el servidor la hora actual es: ". date("H:i:s", $time));
?>
?>
    
```



Con este ejemplo podemos comprobar algunas cosas. En primer lugar, que la función time() devuelve un número, mientras que la función date devuelve un String o cadena de texto.

En segundo lugar, que para obtener fechas anteriores al 1 de enero de 1970 debemos usar números negativos.

En tercer lugar, que el momento 0 que es el 01-01-1970 00:00:00 GMT, cuando es mostrado por un servidor concreto, es adaptado a su horario local. En el servidor del ejemplo, el horario local es GMT + 01:00, es decir, su hora es la hora GMT +01:00, por eso nos devuelve como hora cero la una de la mañana del 1 de enero de 1970. No te preocupes si te resulta un poco confuso, simplemente ten en cuenta que debes hacer pruebas para ver cómo responde el servidor con el que estás trabajando.

En cuarto lugar, que podemos hacer que la fecha se muestre en el formato u orden que queramos, usando los términos clave " d, m, Y, H, i, s" en el orden que queramos.

En quinto lugar, que podemos concatenar cómo se muestra una fecha (o almacenarla en otra variable si quisiéramos) concatenando el resultado de la función date con una cadena de texto cualquiera.

OTRAS FUNCIONES PARA FECHA, HORA Y TIEMPO PHP

PHP define otras funciones nativas para manejo de fechas, horas y tiempo. Vamos a ver algunas de ellas:

FUNCIÓN	UTILIDAD	EJEMPLOS aprenderaprogramar.com
mktime(hora,min,seg,mes,dia,ano) hora: 0 a 23 min y seg: 0 a 59 mes: 1 a 12 dia: 1 a 31 año: cuatro dígitos	Devuelve la hora UTC (GMT) para una fecha local indicada, expresada esta hora GMT en segundos (el número de segundos transcurridos desde el 1 de enero de 1970 00:00:00 GMT, valores negativos indican momentos anteriores a esta fecha). Si faltan uno o más parámetros se tomarán los de la fecha local actual de derecha a izquierda. El valor obtenido depende de la hora local del servidor con el que estemos trabajando.	<pre>echo "Tiempo Unix para el 15-Enero-2089 a las 12h 0m 0s = " . mktime(12,0,0,1,15,2089); // Tiempo Unix para el 15-Enero-2089 a las 12h 0m 0s = 3753975600 (el resultado variará según dónde esté localizado el servidor, porque las 12 h del 15 de enero de 2089 corresponden a distintas fechas-horas GMT según dónde se encuentre el servidor.</pre>
gmmktime(hora,min,seg,mes,dia,ano) hora: 0 a 23 min y seg: 0 a 59 mes: 1 a 12 dia: 1 a 31 año: cuatro dígitos	Devuelve la hora UTC (GMT) expresada en segundos para una fecha GMT indicada en los argumentos recibidos. El valor devuelto es el número de segundos transcurridos desde el 1 de enero de 1970 00:00:00 GMT, valores negativos indican momentos anteriores a esta fecha. Si faltan uno o más parámetros se tomarán los de la fecha GMT actual de derecha a izquierda. El valor obtenido no depende del servidor, ya que la hora GMT es única.	<pre>gmmktime(12,0,0,1,15,2089) // 3756628800 independientemente de la hora local del servidor</pre>

Nota: tener en cuenta que usamos UTC ó GMT como si fueran equivalentes.

Para entender cómo las funciones time y mktime trabajan con tiempo local hemos realizado la siguiente prueba. Hemos colocado el siguiente código en un servidor en Denver (USA), y en un servidor en Madrid (España):

```
echo "Tiempo Unix para el 15-Enero-2089 a las 12h 0m 0s = " . mktime(12,0,0,1,15,2089);
```

El resultado obtenido en Denver (USA) es **3756654000**

El resultado obtenido en Madrid (España) es **3756625200**

¿Por qué obtenemos un resultado diferente?

La diferencia entre uno y otro tiempo es de 28800 segundos, que equivalen a 8 horas.

Hemos comprobado la hora en Denver en el momento de realizar el test y eran las 06:20:00 de la mañana, mientras que en Madrid eran las 14:20:00 horas. Como vemos, tenemos una diferencia horaria de 8 horas entre Denver y Madrid.

Cuando en Madrid son las 12:00 horas del 15 de enero de 2089, en Denver serán las 04:00 horas del 15 de enero de 2089. Las 12:00 horas de un día tienen lugar en distintos instantes de tiempo en Denver y en Madrid, por eso los resultados indican que cuando son las 12 horas en Denver ha transcurrido más tiempo desde las 0 horas del 1 de enero de 1970 GMT que cuando son las 12 horas en Madrid, o dicho de otra manera, las 12 horas de un día tienen lugar antes en Madrid que en Denver. El tiempo Unix sí será igual si pedimos el valor `mktime(4,0,0,1,15,2089)` en Denver: obtendremos 3756625200, el mismo resultado que en Madrid a las 12 del mediodía. Esto indica que ese instante de tiempo (4 de la mañana de 15 de enero de 2089 en Denver y 12 de la mañana de la misma fecha en Madrid) es el mismo instante, es decir, podríamos estar hablando por teléfono dos personas en ese momento, siendo las 4 de la mañana en Denver y las 12 en Madrid.

Si pedimos la hora en Madrid para el 1 de enero de 1970 a las 00:00:00 de 1970 nos devuelve <<Tiempo Unix para el 1 de enero de 1970 a las 00:00:00 = -3600>> ¿Qué significa -3600? Este valor está referido al tiempo UTC e indica qué adelanto o atraso (en segundos) tiene la hora local respecto al tiempo UTC. O dicho de otro modo, cuánto tiempo ha transcurrido desde las 00:00 horas del 1 de enero de 1970 UTC dada una fecha local. Cuando en Madrid son las 00:00:00 del 1 de enero de 1970 en el tiempo UTC de referencia (podemos pensar que es la hora en Greenwich, meridiano de referencia, aunque esto no es exacto) decimos que faltan 3600 segundos (1 hora) para llegar al 1 de enero de 1970. El valor negativo indica que la fecha es anterior al 1 de enero de 1970 00:00:00 GMT, y en concreto en este caso serían las 23 horas del 31 de diciembre de 1969 GMT cuando en Madrid son las 00:00:00 horas del 1 de enero de 1970.

Si pedimos la hora en Denver para el 1 de enero de 1970 a las 00:00:00 de 1970 nos devuelve <<Tiempo Unix para el 1 de enero de 1970 a las 00:00:00 = 25200>> ¿Qué significa 25200? Este valor está referido al tiempo UTC e indica que cuando en Denver son las 00:00:00 del 1 de enero de 1970 el tiempo UTC de referencia está 7 horas por delante, es decir, a las 0 horas en Denver son las 7 horas UTC.

-3600 segundos equivalen a 1 hora por delante de GMT. En invierno en Madrid la hora local es GMT+0100

25200 segundos equivalen a 7 horas por detrás de GMT. En invierno en Denver la hora local es GMT-0700

Tener en cuenta que la hora UTC (GMT) es única (la indicada por los organismos internacionales encargados), mientras que las hora locales varían según el país, e incluso según la zona de un país en que nos encontremos.

Existen más funciones para fechas y tiempo en PHP pero los programadores no las suelen conocer todas, ya que algunas son para tareas muy específicas. No te preocupes por memorizar las funciones ni por conocerlas todas, simplemente sé consciente de su existencia y aprende a buscarlas cuando te resulten necesarias. Si en algún momento necesitas realizar tareas muy específicas con tiempo y fechas en PHP puedes conocer todas las funciones existentes sobre esta materia acudiendo a la documentación oficial de PHP sobre fecha y hora, disponible en <http://php.net/manual/es/ref.datetime.php>

EJERCICIO

Crea un documento HTML que conste de un título h1 con el texto <<Calendario>>. En la página deberá mostrarse el mes y año actuales (en el momento en que hagas el ejercicio) y para cada día del mes, indicar si es lunes, martes, miércoles, jueves, viernes, sábado o domingo.

Ejemplo: suponiendo que hagas el ejercicio en mayo de 2050, debería mostrarse:

Día 1: Domingo, Día 2: Lunes, Día 3: Martes, Día 4: Miércoles, ... etc. hasta Día 31: Martes.

Ampliación voluntaria del ejercicio: si tienes conocimiento de maquetación con HTML y CSS te proponemos que cambies el aspecto para que en lugar de mostrarse como texto, el calendario se muestre con aspecto de calendario habitual. Para la maquetación puedes usar una tabla de 7 columnas por 7 filas (total 49 celdas) con ancho de tabla 300 píxeles y tamaño de fuente en la tabla 24 píxeles. La primera columna corresponderá a lunes y la última a domingo como mostramos en el siguiente ejemplo.

El aspecto, suponiendo que te encuentras en el mes de mayo de 2050, sería el siguiente:

MAYO DE 2050

Lu	Ma	Mi	Ju	Vi	Sa	Do
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Ten en cuenta que debe generarse el calendario del mes en que te encuentres según la hora local del sistema (de tu servidor).

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU00831B

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=70&Itemid=193