



APRENDERAPROGRAMAR.COM

MOSTRAR FECHA EN
ESPAÑOL U OTRO IDIOMA
CON PHP. SETLOCALE Y
STRFTIME. FORMATO.
DATE_ DEFAULT
_TIMEZONE _SET
EJEMPLOS. (CU00831B)

Sección: Cursos

Categoría: Tutorial básico del programador web: PHP desde cero

Fecha revisión: 2029

Resumen: Entrega nº31 del Tutorial básico “PHP desde cero”.

Autor: Enrique González Gutiérrez y César Krall

FUNCIONES PARA EXPRESAR FECHAS EN UN LENGUAJE

PHP es un lenguaje con marcada influencia anglosajona debido a sus orígenes y desarrollo. Pero cuando estamos creando páginas web es frecuente que queremos obtener fechas expresadas en el lenguaje con el que estamos desarrollando la web. Vamos a estudiar cómo hacerlo.



SETLOCALE

La función `setlocale` es una función PHP utilizada para indicar la configuración regional que debe ser usada para realizar determinadas tareas. Su sintaxis es la siguiente:

```
setlocale ( $procesosAfectados, $codConfiguracion1, $ codConfiguracion2, ... , $ codConfiguracionN);
```

```
También se admite: setlocale ( $procesosAfectados, $arrayDeCodigosDeConfiguracion);
```

Esta función pretende que por ejemplo en lugar de 2089 July, Mon 21st pueda aparecer algo como "21 de julio de 2089" si lo deseamos. Para ello indicaríamos que el proceso afectado es la fecha y que la configuración es el español: `setlocale (LC_TIME, "es_ES");`

Esta idea, aparentemente buena, se topa con numerosas dificultades en la práctica:

- a) No todos los servidores admiten todo tipo de cambios.
- b) Según el servidor que estemos utilizando, será posible usar unos lenguajes o no.
- c) La comunicación entre las aplicaciones web y otros entornos como bases de datos se puede ver afectada y pueden surgir problemas.
- d) No todos los servidores admiten el mismo nombre – código de configuración
- e) Más problemas que no vamos a citar aquí.

En resumen, debes tener presente que el cambio de la configuración local del servidor puede ser compleja o no responder como cabría esperar: podrás ver miles de consultas en foros sobre este tipo de problemas, y muchas de ellas sin respuesta. Es recomendable hacer pruebas y si es necesario programar pequeños scripts o funciones para lograr el funcionamiento que nosotros deseamos. Con esto queremos decir que puede resultar más sencillo trabajar del modo natural tal y como esté configurado el servidor, en lugar de tratar de alterar esta configuración para que el servidor trabaje en nuestro idioma.

Los valores posibles para \$procesosAfectados son los siguientes:

Valor \$procesosAfectados	Aplicación
LC_ALL	Para todos los procesos
LC_COLLATE	Para la comparación de strings
LC_CTYPE	Para la clasificación y conversión de caracteres, por ejemplo cuando se use strtoupper()
LC_MONETARY	Afecta a aspectos como el símbolo de moneda, separador de miles, etc.
LC_NUMERIC	Para el separador decimal y otros aspectos como signos.
LC_TIME	Para el formato de fecha y hora que se mostrará al usar la función strftime()
LC_MESSAGES	Para las respuestas del sistema.

Nosotros en este curso vamos a limitarnos a estudiar y ver un ejemplo de uso para mostrar fechas usando la función de php strftime que estudiaremos a continuación combinada con el uso de setlocale con LC_TIME. El uso de setlocale puede tener efectos colaterales (afectar a otros procesos) por lo que recomendamos no usarlo sin hacer pruebas exhaustivas para verificar la respuesta esperada.

Los códigos de países dependen del sistema operativo con el que estemos trabajando. La función setLocale admite que se introduzca una lista de códigos de países con la esperanza de que algunos de ellos sea reconocido por el sistema (lo cual habla ya de las dificultades que se encuentran para realizar este tipo de operaciones).

A continuación señalamos algunas formas de códigos de algunos países:

País	Código (*)	Otro código (*)
México	es_MX.UTF-8	es_MX
España	es_ES.UTF-8	es_ES
Argentina	es_RA.UTF-8	es_RA
Colombia	es_CO.UTF-8	es_CO
Venezuela	es_VE.UTF-8	es_VE
Perú	es_PE.UTF-8	es_PE
Chile	es_CL.UTF-8	es_CL
Guatemala	es_GT.UTF-8	es_GT
Ecuador	es_EC.UTF-8	es_EC
Cuba	es_CU.UTF-8	es_CU
Bolivia	es_RB.UTF-8	es_RB

País	Código (*)	Otro código (*)
República Dominicana	es_DO.UTF-8	es_DO
Honduras	es_HN.UTF-8	es_HN
El Salvador	es_SV.UTF-8	es_SV
Paraguay	es_PY.UTF-8	es_PY
Nicaragua	es_NI.UTF-8	es_NI
Costa Rica	es_CR.UTF-8	es_CR
Puerto Rico	es_PR.UTF-8	es_PR
Panamá	es_PA.UTF-8	es_PA
Uruguay	es_UY.UTF-8	es_UY
Guinea Ecuatorial	es_GQ.UTF-8	es_GQ

(*) Tener en cuenta que no todos los códigos funcionarán, ya que depende del servidor con el que estemos trabajando el que sean reconocidos o no. Recomendamos usar es_ES.UTF-8. Hay más códigos aparte de los aquí indicados.

El ejemplo de uso de setlocale lo veremos a continuación junto al de la función strftime.

FUNCIÓN STRFTIME PHP

La función strftime es una función PHP utilizada para obtener una cadena de texto expresada en el lenguaje de la configuración local del servidor (por defecto o establecida a través de setlocale). Su sintaxis es la siguiente:

```
strftime ($cadenaConMarcas, $valorTiempoEnSegundos)
```

\$valorTiempoEnSegundos indica los segundos transcurridos desde el 1 de enero de 1970 00:00:00 GMT.

Si se omite \$valorTiempoEnSegundos se toma el valor correspondiente a la hora actual local del servidor.

El valor \$cadenaConMarcas hace referencia al uso de códigos especiales que devuelven un determinado resultado. Por ejemplo echo strftime (Hoy es %A); nos devolverá "Hoy es lunes" si la configuración de idioma del servidor es español, ó "Hoy es monday" si la configuración de idioma del servidor es inglés, ya que el código %A equivale a "día de la semana".

Otro ejemplo: echo 'Fecha actual: '.strftime("%A, %d de %B de %Y").'
'; devolverá algo similar a <<jueves, 04 de septiembre de 2089>> si la configuración de idioma del servidor es español.

En la siguiente tabla se resumen los códigos disponibles:

Uso para	Código	Descripción	Ejemplo valores devueltos
Día	%a	Día, en forma abreviada	Dom hasta Sab
Día	%A	Día, completo	Domingo hasta Sábado
Día	%d	El día del mes con dos dígitos (con ceros iniciales)	01 a 31
Día	%e	El día del mes, con un espacio precediendo si < 10.	1 a 31
Día	%j	Día del año, 3 dígitos con ceros iniciales	001 a 366
Día	%u	Representación numérica ISO-8601 del día de semana	1 (para Lunes) hasta 7 (para Domingo)
Día	%w	Representación numérica del día de la semana	0 (para Domingo) hasta 6 (para Sábado)
Semana	%U	Número de semana del año dado, comenzando con el primer Domingo como primera semana	13 (para la 13ª semana del año)
Semana	%V	Número de semana según ISO-8601	01 hasta 53
Semana	%W	Número de semana del año, comenzando con el primer Lunes como la primera semana	46 (para la 46ª semana del año)
Mes	%b	Nombre del mes abreviado	Ene hasta Dic
Mes	%B	Nombre del mes completo	Enero hasta Diciembre
Mes	%h	Nombre del mes abreviado, igual que %b	Ene hasta Dic
Mes	%m	Representación de dos dígitos del mes	01 (para Enero) hasta 12 (para Diciembre)
Año	%C	Representación de dos dígitos del siglo (año dividido entre 100, truncado a un entero)	19 para el Siglo XX
Año	%g	Representación de dos dígitos del año según ISO-8601	Ejemplo: 89 para 6 de Enero de 2089
Año	%G	La versión de cuatro dígitos completa de %g	Ejemplo: 2089 para 13 de Enero de 2089
Año	%y	Representación de dos dígitos del año	Ejemplo: 09 para 2009, 79 para 1979
Año	%Y	Representación de cuatro dígitos del año	Ejemplo: 2038
Hora	%H	Representación de dos dígitos de la hora	00 hasta 23
Hora	%k	Representación de dos dígitos de la hora, con un espacio precediendo a los dígitos simples	0 hasta 23
Hora	%I	Representación de dos dígitos de la hora	01 hasta 12
Hora	%l (ele)	La hora en formato de 12 horas, con un espacio si < 10	1 hasta 12
Hora	%M	Representación de dos dígitos de los minutos	00 hasta 59
Hora	%p	'AM' o 'PM' en MAYÚSCULAS basados en la hora dada	Ejemplo: AM para 00:31, PM para 22:23
Hora	%P	'am' o 'pm' en minúsculas basados en la hora dada	Ejemplo: am para 00:31, pm para 22:23
Hora	%r	Lo mismo que "%I:%M:%S %p"	Ejemplo: 09:34:17 PM para 21:34:17
Hora	%R	Lo mismo que "%H:%M"	Ejemplo: 16:44 para 4:44 PM
Hora	%S	Representación de dos dígitos de los segundos	00 hasta 59
Hora	%T	Lo mismo que "%H:%M:%S"	Ejemplo: 21:34:17 para 09:34:17 PM
Hora	%X	Representación preferida de la hora basada en la configuración regional, sin la fecha	Ejemplo: 03:59:16 o 15:59:16
Hora	%z	El índice de la zona horaria.	Ejemplo: -0500 para Hora Este de EEUU
Hora	%Z	La abreviatura de la zona horaria	Ejemplo: EST para Hora del Este de EEUU
Fecha y hora	%c	Marca preferida de la fecha y hora basadas en la configuración regional	Ejemplo: Tue Feb 5 00:45:10 2009 para el 5 de Febrero de 2009 a las 12:45:10 AM
Fecha y hora	%D	Lo mismo que "%m/%d/%y"	Ejemplo: 02/05/89, 5 de Febrero de 2089
Fecha y hora	%F	Lo mismo que "%Y-%m-%d" (normalmente usado en las marcas de fecha de bases de datos)	Ejemplo: 2089-02-05 para 5 Febrero 2089
Fecha y hora	%x	Representación preferida de la fecha basada en la configuración local, sin la hora	Ejemplo: 02/05/09 para el 5 de Febrero de 2009

Además de las marcas indicadas existen otras como %n para insertar un salto de línea ó %t para insertar una tabulación.

Ten en cuenta que no todos los servidores disponen de todos los códigos ni todos los servidores responderán como se espera cuando trates de hacer que muestren una fecha en un formato dado. Por ejemplo algunos servidores no admiten %p ni %P, en este caso si intentas que se muestre AM/PM no se mostrará nada. Tendrás que resolverlo escribiendo código específico para lograrlo.

Para probar el funcionamiento de esta función, en combinación con setlocale, escribe este código y comprueba qué resultados obtienes:

```
<meta charset='utf-8'>
<?php
$miFecha= gmmktime(12,0,0,1,15,2089);
echo 'Antes de setlocale strftime devuelve: '.strftime("%A, %d de %B de %Y",
$miFecha). '<br/>';
echo 'Antes de setlocale date devuelve: '.date("l, d-m-Y (H:i:s)", $miFecha). '<br/>';
setlocale(LC_TIME, "es_ES");
echo 'Después de setlocale es_ES date devuelve: '.date("l, d-m-Y (H:i:s)",
$miFecha). '<br/>';
echo 'Después de setlocale es_ES strftime devuelve: '.strftime("%A, %d de %B de %Y",
$miFecha). '<br/>';
setlocale(LC_TIME, 'es_ES.UTF-8');
echo 'Después de setlocale es_ES.UTF-8 date devuelve: '.date("l, d-m-Y (H:i:s)",
$miFecha). '<br/>';
echo 'Después de setlocale es_ES.UTF-8 strftime devuelve: '.strftime("%A, %d de %B de %Y",
$miFecha). '<br/>';
setlocale(LC_TIME, 'de_DE.UTF-8');
echo 'Después de setlocale de_DE.UTF-8 date devuelve: '.date("l, d-m-Y (H:i:s)",
$miFecha). '<br/>';
echo 'Después de setlocale de_DE.UTF-8 strftime devuelve: '.strftime("%A, %d de %B de %Y",
$miFecha). '<br/>';
?>
```

El resultado que nosotros hemos obtenido con un servidor ubicado en Denver (USA) es el siguiente:

```
Antes de setlocale strftime devuelve: Saturday, 15 de January de 2089
Antes de setlocale date devuelve: Saturday, 15-01-2089 (05:00:00)
Después de setlocale es_ES date devuelve: Saturday, 15-01-2089 (05:00:00)
Después de setlocale es_ES strftime devuelve: sábado, 15 de enero de 2089
Después de setlocale es_ES.UTF-8 date devuelve: Saturday, 15-01-2089 (05:00:00)
Después de setlocale es_ES.UTF-8 strftime devuelve: sábado, 15 de enero de 2089
Después de setlocale de_DE.UTF-8 date devuelve: Saturday, 15-01-2089 (05:00:00)
Después de setlocale de_DE.UTF-8 strftime devuelve: Samstag, 15 de Januar de 2089
```

Nota: comprueba que usas como codificación para tu archivo php "Codificación UTF-8 sin BOM". Ten en cuenta que los resultados que obtengas pueden ser distintos a estos ya que dependen de la configuración que tenga establecida tu servidor.

Vamos a comentar los resultados obtenidos:

Hemos creado la variable \$miFecha cuyo contenido es el número de segundos desde el 1 de enero de 1970 00:00:00 GMT para la fecha 15 de enero de 2089 12:00:00 GMT. Como la hora local tiene un retraso de 7 horas respecto de la hora GMT, al pedir que se muestre por pantalla la fecha y hora con strftime se nos muestra <<Saturday, 15 de January de 2089>>. Este valor nos viene en inglés porque la configuración local del servidor es "idioma inglés".

Al cambiar la configuración local de idioma con setlocale a es_ES obtenemos la fecha en español, pero en lugar de sábado se nos muestra s?bado. La función date sigue devolviendo la fecha en inglés porque esta función no es sensible a la configuración local de idioma.

Al cambiar la configuración local de idioma con setlocale a 'es_ES.UTF-8' ya obtenemos la fecha correctamente escrita en español.

A modo de ejemplo, hemos cambiado la configuración a alemán para comprobar lo que se visualiza (la fecha escrita en alemán).

Podríamos haber escrito setlocale(LC_TIME, 'es_ES','es_ES.UTF-8'); ¿Qué ocurrirá en este caso? Se tomará el primer código reconocido dentro de la lista de códigos. En este caso es_ES es un código reconocido, aunque no es el que nos da los resultados deseados.

Como vemos, el trabajo relacionado con fecha, hora e idioma puede resultar problemático y no tenemos más remedio que realizar pruebas hasta obtener el resultado deseado.

FUNCIÓN DATE_DEFAULT_TIMEZONE_SET PHP

La función date_default_timezone_set es una función PHP utilizada para establecer una zona horaria específica como configuración local para el servidor. De esta manera, podemos hacer que al pedir la fecha y hora actual nos muestre la fecha y hora de un país (o región de un país) concreto. Su sintaxis es la siguiente:

```
date_default_timezone_set ($valorDeTimezone);
```

La configuración así establecida afectará a todas las funciones de fecha y hora.

Los valores de \$valorDeTimezone han de escogerse entre los valores admitidos. En la siguiente tabla se muestran algunos de ellos (nota: algunos valores pueden no ser reconocidos, recomendamos hacer pruebas para comprobarlo):

Zonas horarias para México, España y Argentina:

País	Zona	Zona	Zona
México	America/Mexico_City	America/Hermosillo	America/Matamoros
	America/Cancun	America/Tijuana	America/Mazatlan
	America/Merida	America/Santa_Isabel	America/Chihuahua
	America/Monterrey	America/Bahia_Banderas	America/Ojinaga
España	Europe/Madrid	Atlantic/Canary	
Argentina	America/Argentina/Buenos_Aires	America/Argentina/Tucuman	America/Argentina/Mendoza
	America/Argentina/Cordoba	America/Argentina/Catamarca	America/Argentina/San_Luis
	America/Argentina/Salta	America/Argentina/La_Rioja	America/Argentina/Rio_Gallegos
	America/Argentina/Jujuy	America/Argentina/San_Juan	America/Argentina/Ushuaia

Zonas horarias para otros países:

País	Zona	País	Zona
Colombia	America/Bogota	Honduras	America/Tegucigalpa
Venezuela	America/Caracas	El Salvador	America/El_Salvador
Perú	America/Lima	Paraguay	America/Asuncion
Chile	America/Santiago ó Pacific/Easter	Nicaragua	America/Managua
Guatemala	America/Guatemala	Costa Rica	America/Costa_Rica
Ecuador	America/Guayaquil ó Pacific/Galapagos	Puerto Rico	America/Puerto_Rico
Cuba	America/Havana	Panamá	America/Panama
Bolivia	America/La_Paz	Uruguay	America/Montevideo
República Dominicana	America/Santo_Domingo	Guinea Ecuatorial	Africa/Malabo

Para probar el funcionamiento de esta función, en combinación con setlocale, escribe este código y comprueba qué resultados obtienes:

```
<meta charset='utf-8'>
<?php
$miFecha= gmmktime(12,0,0,1,15,2089);
setlocale(LC_TIME, 'es_ES.UTF-8');
echo 'Después de setlocale es_ES.UTF-8 strftime devuelve: '.strftime("%A, %d de %B de %Y
%H:%M", $miFecha).'';
```



```
echo 'Fecha actual: '.strftime("%A, %d de %B de %Y %H:%M").' <br/>';
date_default_timezone_set ('Europe/Madrid');
echo 'Establecida zona horaria Europe/Madrid obtenemos: '.strftime("%A, %d de %B de %Y %H:%M", $miFecha).' <br/>';
echo 'Ahora fecha actual es: '.strftime("%A, %d de %B de %Y %H:%M").' <br/>';
?>
```

El resultado que nosotros hemos obtenido con un servidor ubicado en Denver (USA) es el siguiente:

```
Después de setlocale es_ES.UTF-8 strftime devuelve: sábado, 15 de enero de 2089 05:00
Fecha actual: jueves, 04 de septiembre de 2044 04:31
Establecida zona horaria Europe/Madrid obtenemos: sábado, 15 de enero de 2089 13:00
Ahora fecha actual es: jueves, 04 de septiembre de 2044 12:31
```

Fijate cómo de obtener la hora local en Denver hemos pasado a obtener la hora local en Madrid (8 horas adelantada respecto a la de Denver) gracias al uso de date_default_timezone_set.

No siempre se obtienen los resultados deseados, por lo que es imprescindible realizar pruebas.

EJERCICIO 1

Crea una función php de nombre mostrarDiasMes que reciba como argumentos un mes (valor numérico de 1 a 12) y un año (valor numérico de 4 dígitos) y muestre por pantalla el texto “extendido” correspondiente a la fecha en idioma español. Por ejemplo mostrarDiasMes(1, 2089) deberá devolver: 1 de enero de 2089, 2 de enero de 2089, 3 de enero de 2089, 4 de enero de 2089... hasta 31 de enero de 2089”.

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

EJERCICIO 2

Repite el ejercicio anterior, pero en esta ocasión mostrando la fecha en formato de hora tal y como se usa en Holanda y en idioma holandés.

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

Próxima entrega: CU00832B

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