

Unidad 2

Manejo de fechas

FUNCIONES PARA EL MANEJO DE FECHA Y HORA

Introducción

Estas funciones le permiten obtener la fecha y hora del servidor en donde están siendo ejecutados los scripts PHP. Podemos usar estas funciones para dar formato a las fechas y horas en muchas maneras diferentes.

Veremos algunas de las mas importantes con ejemplos en este apunte. Pueden encontrar la lista completa de funciones de fecha y hora en el manual de PHP

<http://php.net/manual/es/ref.datetime.php>

Función date()

El manejo de fechas es un tema muy importante. Manejar esta función es bastante sencillo, podemos usarla añadiendo su valor a una variable y además especificando el tiempo que queremos manejar, es decir, año, mes, día etc.

Ejemplo:

```
<?php
$prueba = date("Y/m/d");
Echo "Hoy es: ".date("d/m/Y");
echo "<br/>";
if ($prueba < "2020/11/01")
{
echo "Sitio web en mantenimiento hasta el 01/11/2020";
}
else
{
echo "Bienvenido a nuestro sitio!";
}
?>
```

Como verán esto es un simple programa que efectúa un condicional, si la fecha es menor al año 2020, mes de Noviembre, día 1, mostrará un mensaje por pantalla: "Sitio web en mantenimiento",

de lo contrario muestra un mensaje de bienvenida. Enumeramos a continuación los parámetros de dicha función:

- A - AM/PM
- B - Hora swatch de internet
- d - Día del mes 01 a 31
- g - Hora de 1 a 12
- G - Hora de 0 a 23
- h - Hora de 01 a 12
- H - Hora de 00 a 23
- i - Minutos 00 a 59
- j - Día del mes de 1 a 31
- m - Numero de mes de 01 a 12
- n - Numero de mes de 1 a 12
- s - Segundos de 00 a 59
- t - Días del mes de 28 a 31
- U - Fecha Unix
- w - Día de la semana de 0 a 6 empezando por Domingo
- W - Semana del año ej: 42
- y - Año ej: 99
- Y - Año ej: 1999
- z - Día del año de 0 a 366

Por ejemplo, para mostrar la fecha actual en formato día (del 1 al 31), mes (del 1 al 12) y año completo, sería este código:

```
<?php echo date ( "j - n - Y" ); ?>
```

Función time()

Para mostrar la hora en PHP además de la propia función date(), podemos utilizar la función time().

La función time() nos devolverá la hora de la fecha actual en el formato timestamp.

El término timestamp en PHP sirve para llevar un registro de la fecha y hora. Y lo que hace exactamente para ello es calcular el tiempo transcurrido en segundos desde la fecha que toma

por referencia, el 1 de Enero de 1970. Así que por ejemplo el numero 86400, correspondería al día 2 de Enero de 1970 a las 00:00, ya que 86400 son los segundos que tiene un día.

Ejemplo:

```
<?php
$hora = time();
echo ($hora);
// mostraría en el horario en que se ejecute mostraría algo como: "1306124335"
?>
```

Si ahora le aplicamos la función `getdate()`, podremos ver la hora de una forma más sencilla.

```
<?php
$hora = getdate(time());
echo( $hora['hours'] . ":" . $hora['minutes'] . ":" . $hora['seconds'] );
?>
```

Este sería el nuevo resultado: 1:21:21

Explicemos entonces esta función:

Función `getdate()`

Una vez aclarado el concepto timestamp, vamos a explicar la función `getdate()`, mencionada anteriormente.

La función `getdate()` nos devolverá un array con la fecha en formato timestamp con los siguientes elementos:

“seconds” = segundos

“minutes” = minutos

“hours” = horas

“mday” = día del mes

“wday” = día de la semana en número

“mon” = número del mes

“year” = año

“yday” = número de día del año, por ejemplo “125”

“weekday” = día de la semana en letra (ingles), por ejemplo “Monday”

“month” = mes en letra (ingles), por ejemplo “November”

Ejemplo:

```
<?php
$fecha = getdate();
echo ("Día: " . $fecha["mday"]."<br/>");
echo ("Mes: " . $fecha["mon"]."<br/>");
echo ("Año: " . $fecha["year"]."<br/>");
echo ("Hora: " . $fecha["hours"]."<br/>");
echo ("Minutos: " . $fecha["minutes"]."<br/>");
echo ("Segundos: " . $fecha["seconds"]."<br/>");
?>
```

Este sería el nuevo resultado:

Día: 23

Mes: 5

Año: 2011

Hora: 2

Minutos: 21

Segundos: 11

función `date_default_timezone_set()`

Establece el huso horario que utilizan las funciones de fecha y hora.

El manual de PHP contiene la lista de identificadores de zonas horarias válidas.

Ver: <http://www.php.net/manual/es/timezones.php>

Ejemplo:

```
<?php
date_default_timezone_set("America/Argentina/Buenos_Aires");
echo "<p>".date('h:m:i')."</p>";
?>
```

Función `strftime()`

La función `strftime($formato)` permite generar una cadena que incluye la fecha y/o la hora del sistema en el idioma deseado. La función `strftime()` espera uno o dos argumentos. El primer

argumento define el formato de la cadena y el segundo el instante que se quiere convertir (si no se indica el segundo argumento, la función trabaja con la fecha y hora actual del sistema).

Para definir el formato de la cadena se utilizan, entre otros, los siguientes caracteres:

Día Mes Año

números con ceros delante %d %m %y (2 cifras) %Y (4 cifras)

nombre corto (3 letras) %a %b

nombre largo (nombre completo) %A %B

Hora(12h) hora (24 h) minutos

Segundos números con ceros delante %l %H %M %S

Además, se pueden utilizar los siguientes valores:

caracter

fecha y hora (dd/mm/aaaa hh:mm:ss) %c

fecha (dd/mm/aaaa) %x

hora (hh:mm:ss) %X

```
<?php
date_default_timezone_set('America/Argentina/Buenos_Aires');
setlocale(LC_TIME, 'spanish');
echo "<h1>".strftime("Ejemplo 1: %A, %d de %B de %Y")."</h1>";
?>
```

Deberíamos obtener como resultado:

Ejemplo 1: jueves, 04 de Julio de 2019

Lista de las zonas horarias admitidas: <https://www.php.net/manual/es/timezones.php>

Calcular fechas fácilmente con php

Podemos calcular fechas en php fácilmente utilizando literales que nos ofrece php. Mejor lo vemos con un ejemplo:

```
<?php
echo $date=date("d-m-Y H:i:s", strtotime ("-12hours")); echo "<br/>";
echo $date2=date("d-m-Y H:i:s", strtotime ("-2days")); echo "<br/>";
echo $date3=date("d-m-Y H:i:s", strtotime ("-1years"));
```



```
echo "<br/>";  
echo $date4=date("d-m-Y H:i:s", strtotime ("next Thursday")); echo "<br/>";  
echo $date5=date("d-m-Y H:i:s", strtotime ("last Monday"));  
?>
```

Descripción

int strtotime (string \$hora [, int \$ahora])

La función espera que se le pase una cadena conteniendo una fecha en formato Inglés e intentará procesarla y convertirla a una timestamp (muestra de tiempo) de UNIX relativa a la timestamp proporcionada en ahora, o la hora actual si no se indica ninguna.

Si falla, devolverá -1.

Otro ejemplo con strtotime()

```
<?php  
echo strtotime ("now"), "\n";  
echo strtotime ("10 September 2000"), "\n"; echo strtotime ("+1 day"), "\n";  
echo strtotime ("+1 week"), "\n";  
echo strtotime ("+1 week 2 days 4 hours 2 seconds"), "\n"; echo strtotime ("next  
Thursday"), "\n";  
echo strtotime ("last Monday"), "\n";  
?>
```

Diferencia entre dos fechas

Sencillo método para calcular la diferencia entre dos fechas en días y horas.

```
<?php  
function dias_restantes($fecha_final) {  
    $fecha_actual = date("Y-m-d");  
    $s = strtotime($fecha_final)-strtotime($fecha_actual);  
    $d = intval($s/86400);  
    echo "Días restantes hasta la fecha $fecha_final: $d";  
}  
//para usarlo: dias_restantes("2020-11-01");  
?>
```

Fecha y hora actual: función localtime() La función localtime() devuelve un vector con información detallada de la fecha y hora actual. Para que devuelva esa información de la hora actual, se pasa

como argumento la función `time()`. El segundo argumento indica si el vector tiene índices asociativos.

```
<?php
$fecha_actual = localtime(time(),1);
$anio_actual = $fecha_actual['tm_year'] + 1900;
$mes_actual = $fecha_actual['tm_mon'] + 1;
$dia_actual = $fecha_actual['tm_mday'];
echo "<p>Hoy es el $dia_actual/$mes_actual/$anio_actual.</p>";
?>
```

Mostrar fecha actual en español con php Con esta sencilla función obtendremos la fecha actual y la mostrará en el formato "Jueves, 1 de Noviembre de 2020".

```
<?php
function fecha_actual()
{
    $week_days = array ("Domingo", "Lunes", "Martes", "Miercoles", "Jueves", "Viernes",
    "Sabado");
    $months = array ("", "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto",
    "Septiembre", "Octubre", "Noviembre", "Diciembre");
    $year_now = date ("Y");
    $month_now = date ("n");
    $day_now = date ("j");
    $week_day_now = date ("w");
    $date = $week_days[$week_day_now] . ", " . $day_now . " de " .
    $months[$month_now] . " de " . $year_now;
    echo $date;
}
fecha_actual();
?>
```

Validación de fecha: función `checkdate()`

La función `checkdate($mes, $dia, $anio)` devuelve el valor `true` si la fecha enviada es correcta en el calendario Gregoriano y `false` si no lo es.

```
<?php
$dia = 29;
$mes = 10;
$anio = 2020;
```



```
if (checkdate($mes, $dia, $anio)) {  
    echo "<p>El día $dia/$mes/$anio existe.</p>";  
} else {  
    echo "<p>El día $dia/$mes/$anio no existe.</p>";  
}  
$dia = 29;  
$mes = 10;  
$anio = 2019;  
if (checkdate($mes, $dia, $anio)) {  
    echo "<p>El día $dia/$mes/$anio existe.</p>";  
} else {  
    echo "<p>El día $dia/$mes/$anio no existe.</p>";  
}  
?>
```

Hay que tener en cuenta que el calendario Gregoriano se estableció (en algunos países) a partir del 15 de octubre de 1582, por lo que no tiene sentido aplicar esta función a fechas anteriores a esa fecha. Por ejemplo, la función checkdate() no tiene en cuenta que hay unos días (del 5 al 14 de octubre de 1582) que no existieron en el calendario Gregoriano. Por otra parte, si el año es negativo o cero, la función checkdate() devuelve siempre false.
