

## Un ejemplo

### Dos interpretaciones

Recordemos que cuando un documento tiene extensión **php** es objeto de **dos interpretaciones** antes de ser visualizado en el navegador. En primer lugar es PHP quien ejecuta sus scripts y devuelve al cliente el documento resultante y, más tarde, será el propio navegador quien realice una *segunda interpretación* —del documento recibido— cuyo resultado será la visualización de la página.

### Los saltos de línea

Cualquier *salto de línea* que se inserte en un documento será respetado por PHP y enviado al navegador, pero, como éste solo entiende como tales sus propias etiquetas `<br>`, no serán visualizados, aunque sí estarán en el código fuente de la página visualizada.

Hemos de considerar un nuevo concepto. PHP permite utilizar algunos caracteres especiales que son transformados durante la ejecución del script.

Uno de ellos —no es el único— es `\n` que es interpretado y convertido por el intérprete de PHP en un *salto de línea* cuyo efecto será visible en el código fuente del documento enviado al navegador, pero que —por no ser una etiqueta `<br>`— no producirá efecto alguno en la visualización de la página.

### `nl2br($A)`

Esta función inserta de **forma automática** etiquetas HTML de salto de línea (`<br>`).

Por cada salto de línea que exista en el texto de la cadena inserta una etiqueta `<br>` en la salida HTML.

### `strtr($a,$dicc)`

Busca en la cadena `$a` las *palabras* coincidentes con los **índices** del **array asociativo** (`$dicc`) y las sustituye por los **valores** asociados a esos índices.

### `get_meta_tags($a)`

Devuelve un *array asociativo* cuyos *índices* son los valores de la propiedad `name` de las etiquetas `<meta>` (*escritas siempre en minúsculas, sin comillas, y reemplazando, en el caso de que fuera necesario, los caracteres*

```
<?
/* definamos algunas variables de cadena
   tal como se describe en sus contenidos
   e incluyamos caracteres especiales \n */

$cadena1="Este texto está escrito
         en varias líneas
         y hemos saltado de una a otra
         pulsando enter";

$cadena2="Aquí\nseparamos\nlas\nlíneas\ncon\nsin\npulsar\nenter";

$cadena3=<<<Prueba
Nuevamente texto en varias líneas
ahora usando sintaxis de documento incrustado.
Seguiremos probando
Prueba;

$cadena4=<<<OtraPrueba
Ahora\ninsertaré\nalgo\ncomo\nesto
OtraPrueba;
# definamos una variable conteniendo saltos de línea HTML
# y vayamos construyendo una variable de salida
# en la que uniremos las variables anteriores insertando
# entre ellas saltos de línea para facilitar la visualización
# en el navegador
$saltador="<br><br><br>";

$salida=$cadena1.$saltador;
$salida .= $cadena2.$saltador;
$salida .= $cadena3.$saltador;
$salida .= $cadena4.$saltador;
# visualicemos el resultado
print $salida;
# apliquemos ahora a la variable salida
# la función nl2br y veamos el resultado
print $saltador.nl2br($salida);

?>
```

ejemplo22.php

Si ejecutas el ejemplo y visualizas el código fuente a través del navegador podrás observar como los resultados del primer **print** generan saltos de línea en éste. Y en el caso del código correspondiente al segundo print, podrás visualizar etiquetas `<BR />` que son el resultado de la aplicación de la función **nl2br**

Las cadenas y las etiquetas HTML		
Cadenas	Sintaxis	Resultado
<code>\$a="Esto es\nun texto escrito\n en varias líneas\nsin etiquetas&lt;BR&gt;";</code>	<code>nl2br(\$a)</code>	<b>Esto es un texto escrito en varias líneas sin etiquetas&lt;BR&gt;</b>
<code>\$a="index.php"</code>	<code>\$b=get_meta_tags(\$a); echo \$b[keywords]</code>	Programación, PHP, Apache, MySQL
<code>\$a="index.php"</code>	<code>\$b=get_meta_tags(\$a); echo \$b[description]</code>	Materiales para un curso a distancia

[Ver índice](#)   
 [Búsqueda rápida](#)   
 [Página anterior](#)   
 [Página siguiente](#)

especiales por un guión bajo \_) de la página web cuya dirección (absoluta o relativa) se indica en la cadena \$a. Los valores de los elementos del array son los contenidos de esas etiquetas.

### htmlspecialchars(\$a)

Convierte en entidades de HTML algunos caracteres (los que se indican a la derecha). Con ello se consigue su visualización y se impide que sean interpretados como signos del lenguaje HTML.

### htmlentities(\$a)

Es una función similar a la anterior, pero en este caso afecta a todos los caracteres que tienen equivalentes como entidad HTML utilizando el juego de caracteres ISO-8859-1. Recuerda que son éstos.

\$a="<H1>A</H1>"	echo \$a	A
\$a="<H1>A</H1>"	echo htmlspecialchars(\$a)	<H1>A</H1>
La tabla de sustituciones de htmlspecialchars es esta:  Sustituye & por &amp;  " por &quot;  < por &lt;  > por &gt;		

### Ejercicio nº 16

Modifica el script del ejercicio nº 15 de forma que los saltos de línea introducidos en el textarea del formulario sean sustituidos por etiquetas <BR> de HTML.

[Anterior](#)



[Índice](#)



[Siguiete](#)

