

Generación de código

HTML:

Formas de que php
escriba dentro de HTML

Formas de que PHP escriba dentro de HTML

Es posible la intercalación de órdenes en lenguaje PHP, **alternándolas** dentro de una página escrita en lenguaje HTML.

Pueden abrirse y cerrarse las etiquetas de PHP **tantas veces como sea necesario**

Formas de que PHP escriba dentro de HTML

Por ejemplo:

```
<html>
<head>
    <title>Hola</title>
</head>
<body>
    <h1>Esto fue escrito estáticamente, en
HTML</h1>
```

Formas de que PHP escriba dentro de HTML

```
<?php  
print (“<h2>Hola mundo! Esto lo  
escribió el intérprete de PHP</h2>”);  
?>
```

```
<p>Esto ya estaba escrito en código  
HTML.</p>
```

```
<?php  
print (“<p>Esto también lo escribió el  
software intérprete de PHP.</p>”);  
?>
```

Formas de que PHP escriba dentro de HTML

```
<p>  
  <a href="index.php">  
    <?php  
      print ("Volver al Home del sitio, escrito por PHP");  
    ?>  
  </a>  
</p>  
</body>  
</html>
```

Formas de que PHP escriba dentro de HTML

Apertura y cierre de las etiquetas PHP

Las etiquetas PHP:

- Puede abrirse y cerrarse en la misma línea en que abrió, o puede cerrarse en otra línea diferente. Es indistinto.
- **Puede intercalarse dentro de etiquetas HTML preexistentes.**

Formas de que PHP escriba dentro de HTML

- 1. Apertura y cierre estándar:

```
<?php xxxx ?>
```

- O también

```
<?php  
xxxx  
?>
```

- Ésta es la única sintaxis universal: funciona siempre.
- Es la única forma recomendada y la que vamos a usar

Formas de que PHP escriba dentro de HTML

2. Apertura y cierre corto:

- `<? xxxx ?>`
- O también

```
<?  
xxxx  
?>
```

- Esta sintaxis se conoce como **short tags** (etiquetas cortas).
- Fue muy usada en los primeros años de PHP, **pero no es estándar**.

Formas de que PHP escriba dentro de HTML

No todas las configuraciones del intérprete de PHP habilitan su uso, por lo que un código que utilice esta sintaxis **puede dejar de funcionar al ser ubicado en un servidor con una configuración más estricta.**

Por ese motivo, **no la utilizaremos.**

Formas de que PHP escriba dentro de HTML

3. Apertura y cierre mediante etiqueta script:

```
<script language="php">xxxx</script>
```

- O también

```
<script language="php">  
xxxx  
</script>
```

Formas de que PHP escriba dentro de HTML

- Esta sintaxis, ya no es soportada en PHP 7, es innecesariamente larga y es rarísimo encontrar algún código que la emplee.
- Por lo tanto, al no tener ninguna otra ventaja añadida, no se recomienda su uso.

Formas de que PHP escriba dentro de HTML

- Se pueden generar etiquetas HTML mediante:
 - `echo`
 - `print`

Formas de que PHP escriba dentro de HTML

Escribir en el código con la función `print()`

- El lenguaje PHP posee una función que es una de las más utilizadas de todas.
- Hablamos de la **función `print()`**, que le indica al software intérprete de PHP que “escriba” en el código fuente de la página que devolverá al navegador del usuario aquello que pongamos entre sus paréntesis.

Formas de que PHP escriba dentro de HTML

- Ya hemos utilizado intuitivamente esta función en los ejemplos anteriores.
- Si lo que deseamos es que se escriba en el código de la página un texto, literalmente, debemos escribirlo entre comillas dentro de sus paréntesis.

Formas de
que PHP
escriba
dentro de
HTML

Ejemplo:

```
<?php  
    print ("hola");  
?>
```

Formas de que PHP escriba dentro de HTML

- Si sólo tuviéramos que escribir texto y nunca código HTML, no tendríamos más problemas.
- Pero como debemos encerrar **entre comillas** el texto a mostrar, se nos planteará un problema a la hora de escribir código HTML que, a su vez **tenga comillas dentro**.

Formas de que PHP escriba dentro de HTML

En el siguiente ejemplo, veremos por qué:

```
<?php  
print ("<h1 class="portada"> Bienvenidos</h1>");  
?>
```

Formas de que PHP escriba dentro de HTML

- Este ejemplo **generará un error**, pues la comilla ubicada después del signo = está cumpliendo, sin querer, la función de **cerrar la primera de las comillas** –la que se abrió al inicio del print después del paréntesis inicial-.
- Por lo tanto, el tramo de texto se da por concluido y al resto, el que sigue a esa comilla, el intérprete de PHP no sabe cómo tratarlo, y lo advierte mostrando un mensaje de error en la pantalla.

Formas de que PHP escriba dentro de HTML

- Una posible solución al problema de las comillas es desactivarlas (esto se denomina “**escapar**”) **todas las comillas dobles intermedias, una por una**, para que no den por concluida la cadena de texto antes de que lleguemos a la última comilla doble que indica el término de la función print.

Formas de que PHP escriba dentro de HTML

- El carácter de escape es la **barra invertida ** y sirve para no ejecutar el carácter que le sigue inmediatamente como si fuera parte de una orden del lenguaje PHP, sino que lo considera como una letra más que debe ser escrita literalmente.

Formas de que PHP escriba dentro de HTML

- Por esta razón, el ejemplo anterior quedará así:

```
<?php  
print ("<h1 class=\"portada\">Bienvenidos</h1>");  
?>
```

Formas de que PHP escriba dentro de HTML

Esto funciona muy bien en frases cortas, pero el mayor inconveniente o molestia que nos puede causar surge cuando tenemos que **imprimir largos bloques de código** HTML, ya que es muy probable que esos bloques (tal vez páginas enteras) ya los tengamos escritos previamente, generados por nuestro editor de código HTML, y casi es seguro que poseerán **numerosas comillas dobles**.

Formas de que PHP escriba dentro de HTML

- En estos casos, estaríamos obligados a la tediosa tarea de encontrar las comillas una por una, y “escaparlas” anteponiéndoles una barra invertida o, en su defecto, podríamos utilizar las herramientas de búsqueda y reemplazo de caracteres de alguno de los editores HTML para buscar una comilla y reemplazarla por la barra de escape más la comilla.
- Pero, ambos casos, sería una larga y aburrida tarea.

Formas de que PHP escriba dentro de HTML

- Mucho mejor que esto, sería utilizar comillas simples para delimitar el inicio y final del bloque de texto a imprimir:

```
<?php  
print('<h1 class="portada">Bienvenidos</h1>');  
?>
```

- ¡Y problema solucionado, de momento!

Formas de que PHP escriba dentro de HTML

Cómo funciona el comando “echo”

- Este comando (no es una función) también puede utilizar optativamente comillas simples o dobles para delimitar lo que **va a imprimir**, de la misma manera que print.
- Pero, a diferencia de print, no es habitual envolver entre paréntesis lo que escribirá.

Formas de que PHP escriba dentro de HTML

- Ejemplo:

```
<?php  
echo "Hola Mundo entre comillas dobles!";  
echo '<html>  
<head>  
<title>Envuelvo entre comillas  
simples</title>  
</head>  
<body>"Esto tiene comillas dobles,  
"muchas comillas", y no importa"  
</body>  
</html>';  
?>
```

Formas de que PHP escriba dentro de HTML

- Notemos de paso que el código que escribirá puede estar dividido en múltiples líneas (PHP ignora tanto los saltos de línea como los espacios en blanco), y también señalemos otro detalle al que todavía no habíamos prestado atención:

*para dar por terminada una sentencia u orden, **se agrega un punto y coma al final de la línea.***

Formas de que PHP escriba dentro de HTML

Entidades HTML

Cuando estemos insertando texto en documentos HTML puede darse el caso de que necesitemos insertar símbolos.

Para ello HTML nos ofrece las entidades. Las entidades son unas estructuras que, mediante el uso de una codificación, nos permiten representar un símbolo.

La estructura de la entidad HTML es un ampersand(&) seguido del código o nombre de la entidad y terminado en un punto y coma.

```
<?php  
echo "varios &nbsp; &nbsp; &nbsp; espacios";  
?>
```

Formas de que PHP escriba dentro de HTML **heredoc**

Grandes bloques: heredoc

- Cuando tenemos necesidad de escribir largos bloques de código HTML, incluso con variables intercaladas, podemos usar la construcción **heredoc** que nos permite escribir grandes cantidades de texto, sin necesidad de escapar caracteres en su interior.

Formas de que PHP escriba dentro de HTML **heredoc**

- Su uso es muy simple:
 1. Al inicio del bloque de texto, debemos colocar tres veces el signo “menor que”, de esta manera: <<< seguido de varios caracteres alfanuméricos (en el ejemplo que sigue hemos elegido EOT, pero puede ser cualquier otra combinación de letras).
 2. Después pegamos el bloque de texto y código HTML que deseemos.
 3. Para finalizar, repetimos los mismos tres caracteres que indicaron el inicio del bloque.

Formas de que PHP escriba dentro de HTML

heredoc

```
<?php  
echo <<< EOT  
<p>Este texto puede tener dentro  
“comillas” sin necesidad de  
escaparlas.</p>  
<p>También procesa (reemplaza por su  
valor) las $variables que hubiera  
dentro del código</p>  
<p>Esta construcción del lenguaje  
llamada heredoc es ideal para incluir  
largos bloques de código HTML.</p>  
EOT;  
?>
```

Formas de que PHP escriba dentro de HTML **heredoc**

También podemos almacenarlo dentro de una variable:

```
<?php
$codigo =<<<EOT
<p> Este texto puede tener
dentro “comillas” sin necesidad de
escaparlas.</p>
<p>También procesa (reemplaza por su
valor) las $variables que hubiera
dentro del código. </p>
EOT;
echo $codigo;
?>
```


Formas de que PHP escriba dentro de HTML **heredoc**

- Importante: los caracteres indicadores del inicio y fin del bloque, deben incluirse al comienzo del renglón (sin dejar ni un solo espacio en blanco, ni tabulaciones, sangrado de código, ni comentarios), y tampoco deben contener un salto de línea).

Formas de que PHP escriba dentro de HTML

heredoc

- Podemos hacer la prueba y, si vemos que al terminar la orden no se interrumpe la escritura del texto, deberíamos verificar que el renglón final esté completamente vacío de cualquier carácter extraño.
- Hay que probarlo, pues los distintos editores de texto incluyen caracteres ocultos que nos generan errores.

Formas de que PHP escriba dentro de HTML

heredoc

- Además de la facilidad de lectura y pegado, en lugar de los largos bloques de código, esta sintaxis incrementa la velocidad de interpretación del código en el servidor si la comparamos con el uso de varios echo o print seguidos.

Formas de que PHP escriba dentro de HTML

Nowdoc

- ✓ El texto se procesa de igual forma que si fuera una cadena entre comillas dobles, sustituyendo variables y secuencias de escape.
- ✓ Si no quisieras que se realizara ninguna sustitución, debes poner el identificador de apertura entre comillas simples. Esta construcción recibe el nombre de **nowdoc**

```
$a = <<<'MICADENA'
```

```
...
```

```
MICADENA;
```

```
// Devuelve el texto  
literalmente sin interpretar las  
variables // Si se quiere que las  
interprete, se usa heredoc
```

Formas de que PHP escriba dentro de HTML

nowdoc

- ✓ El texto se procesa de igual forma que si fuera una cadena entre comillas dobles, sustituyendo variables y secuencias de escape.
- ✓ Si no quisieras que se realizara ninguna sustitución, debes poner el identificador de apertura entre comillas simples. Esta construcción recibe el nombre de **nowdoc**

```
$a = <<<'MICADENA'
```

```
...
```

```
MICADENA;
```

```
// Devuelve el texto  
literalmente sin interpretar las  
variables // Si se quiere que las  
interprete, se usa heredoc
```

Formas de que PHP escriba dentro de HTML

nowdoc

Nowdoc

Devuelve el texto literalmente, sin interpretar
las variables

El problema de las comillas en PHP

¿Cuándo usar comillas simples y cuándo comillas dobles?

- Los valores **alfanuméricos deben envolverse entre comillas**, ya sean simples o dobles; y en cambio, los valores **numéricos no deben envolverse** entre comillas
- Ejemplo
 - \$entero=5;
 - \$nombre="Alumno";

El problema de las comillas en PHP

- Para almacenar un dato alfanumérico dentro de una variable, deberemos comprender cómo afecta a esos datos el ser delimitado su inicio y su fin por los dos tipos de comillas existentes:
- **comillas dobles y comillas simples**

El problema
de las
comillas en
PHP

La interpretación de
variables dentro de
comillas

El problema de las comillas en PHP

- Ya hemos visto cómo se comporta el comando echo y la función print (ambos son iguales en esto):
- 1. Cuando deseamos que el intérprete de PHP escriba un **texto literalmente** usaremos **comillas simples** para delimitar el inicio y el final de ese texto.
 - La única limitación, es que **no podremos incluir comillas simples dentro de ese texto** (deberemos escaparlas), y cualquier variable que incluyamos dentro de algo envuelto entre comillas simples, **no será reemplazada por su valor**.

El problema de las comillas en PHP

2. En cambio cuando queremos que se interpreten y **reemplacen las variables por su valor, usaremos comillas dobles** para delimitar el inicio y el final del bloque de texto.

El problema de las comillas en PHP

Veamos un ejemplo:

```
<?php  
$nombre="Alumno";  
$numero=7;
```

\$comillasDobles = "Texto entre comillas
dobles, puede contener 'comillas simples'
dentro sin problemas";

El problema de las comillas en PHP

- **\$comillasSimples** = 'Texto entre comillas simples, puede contener "comillas dobles" pero sin variables dentro, porque usa comillas simples para delimitar el inicio y fin del bloque';
- **\$escapeDoble** = "Texto con \"comillas\" dobles escapadas";
- **\$espapeSencillo** = 'Texto con \'comillas\' simples escapadas';

El problema de las comillas en PHP

\$variablesDobles = "Texto con variables como \$nombre y \$numero intercaladas entre comillas dobles, que se reemplazarán por su valor";

\$variablesSimples = 'Texto con variables como \$nombre y \$numero intercaladas entre comillas simples, que **no** se reemplazarán por su valor, quedará escrito \$nombre y \$numero tal cual';

?>

Formas de incluir un salto de línea en HTML

- Si escribimos código en un editor de texto bajo Windows, podemos tener problemas al incluir un salto de línea (Enter) al final del renglón, ya que el salto de línea en Windows no es el carácter de nueva línea `\n` sino que espera encontrar `\r\n`.
- Pero `\r\n` sólo generará un salto de línea para el código fuente, no para HTML.
- La combinación `\r\n` para el intérprete de PHP no se traduce en una etiqueta `
` y por lo tanto el navegador no pondrá un salto de línea.

Formas de incluir un salto de línea en HTML

- La forma de generar un salto de línea es utilizar `
` que es la etiqueta HTML.

- Ejemplos:

```
echo 'Hola<br>';  
echo "Hola<br>";  
echo "Hola\r\n";  
echo 'Hola\r\n';
```


Formas de incluir un salto de línea en HTML

- El navegador mostrará:

Hola

Hola

Hola Hola\r\n

- El código del archivo html generado por el intérprete PHP es el siguiente:

Hola
Hola
Hola

Hola\r\n

Diferencia entre echo y print

- Tanto print como echo no son realmente funciones sino construcciones del lenguaje como (if, while,...), por lo que no es obligatorio que pongas paréntesis cuando las utilices.
- Una diferencia es que print puede comportarse como una función, puesto que devuelve el valor 1 siempre.
- Ello nos lleva a que print es algo más lento que echo, puesto que pierde un tiempo en ocuparse de devolver el valor.

Diferencias en echo y print

- La principal diferencia entre echo y print, es que echo admite múltiples parámetros mientras que print solo admite uno:

- Ejemplo:

```
<?php  
    echo $a,"<br>",$a,"<br>";  
    print $a;  
    print "<br>";  
    print $a;  
?>
```

Diferencias en echo y print

- En PHP tienes dos operadores exclusivos para trabajar con cadenas de texto:
- Con el operador de concatenación punto (.) puedes unir las dos cadenas de texto que le pases como operandos.
- El operador de asignación y concatenación (.=) concatena al argumento del lado izquierdo la cadena del lado derecho.

Diferencias en echo y print

```
<?php  
$a = "Módulo ";  
$b = $a . "DWES"; // ahora $b contiene "Módulo DWES"  
$a .= "DWES"; // ahora $a también contiene "Módulo DWES"  
?>
```

Diferencias en echo y print

- Tanto echo como print admiten en su sintaxis el operador (.)

- Ejemplo:

```
<?php  
$a="hola";  
echo $a."<br>".$a."<br>";  
print $a."<br>".$a."<br>";  
?>
```

Echo y las short tags

`<?=$variable ?>`

Esto es idéntico a escribir:

`<?php echo $variable; ?>`

- A lo largo del curso discutiremos las ventajas e inconvenientes de su utilización.

Historial de cambios

Versión	Descripción
7.0.0	Se eliminaron de PHP las etiquetas de ASP <code><%, %></code> , <code><%=</code> , y la etiqueta de script <code><script language="php"></code> .
5.4.0	La etiqueta <code><?=</code> siempre está disponible independientemente del ajuste en el fichero <code>php.ini</code> de la directiva <code>short_open_tag</code> .

Comentarios en PHP

En una página web los comentarios al HTML van entre los delimitadores:

`<!--` y `-->`

Dentro del código PHP, hay tres formas de poner comentarios:

1. **Comentarios de una línea** utilizando `//` Son comentarios al estilo del lenguaje C. Cuando una línea comienza por los símbolos `//`, toda ella se considera que contiene un comentario, hasta la siguiente línea.
2. **Comentarios de una línea** utilizando `#` Son similares a los anteriores, pero utilizando la sintaxis de los scripts de Linux.
3. **Comentarios de varias líneas.** También iguales a los del lenguaje C. Cuando en una línea aparezcan los caracteres `/*`, se considera que ahí comienza un comentario. El comentario puede extenderse varias líneas, y acabará cuando escribas la combinación de caracteres opuesta: `*/`

Comentarios en PHP

- Recuerda que cuando pongas comentarios al código PHP, éstos no figurarán en ningún caso en la página web que se envía al navegador (justo al contrario de lo que sucede con los comentarios a las etiquetas HTML).