



La opción INCLUDE

[Ver índice](#)



Utilización de ficheros externos

PHP dispone de funciones que permiten insertar en un documento una parte o la totalidad de los contenidos de otro. Esta opción resulta muy interesante, tanto desde el punto de vista operativo como en lo relativo a la seguridad.

Estas son algunos de los recursos que permiten ese tipo de inclusiones:

`include("nom. del fichero")`

El parámetro *nom. del fichero* es una cadena que contiene el *path* y el *nombre* del fichero cuyos contenidos pretendemos incluir.

Pueden incluirse ficheros con cualquier extensión aunque es muy habitual utilizar archivos con extensión *.inc.php*.

La primera parte (*inc*) nos permitirá identificar este tipo de ficheros mientras que la extensión *php* obligaría a que (si un usuario malicioso pretende visualizar el contenido del fichero) fuera interpretado por PHP y, como consecuencia de ello, solo devolvería el resultado sin permitir la visualización de informaciones privadas (contraseñas por ejemplo) que pudiera contener.

Este tipo de ficheros pueden contener: *texto*, *etiquetas HTML* y *funciones*.

Si *no contiene funciones se podrá insertar tantas veces como se invoque* y se insertará, además, *todo su contenido* tal como puedes ver en el ejemplo.

Si el fichero *contiene funciones* solo podrá ser invocado **una vez** ya que si se hiciera una segunda llamada se produciría un **error** por duplicidad en los nombres de las funciones.

Como verás en el ejemplo, es posible incluir cualquier tipo de funciones. Tanto las de la forma:

```
<? function nombre { ?>
```

```
.....  
... código HTML ...
```

```
.....  
<? } ?>
```

Los ficheros a incluir

Este primer ejemplo de fichero a incluir contiene únicamente texto y etiquetas HTML pero no contiene ninguna llamada a ninguna función PHP, ni tampoco ningún script de este lenguaje. Le hemos guardado con dos extensiones: **ejemplo1.inc.php** y **ejemplo1.inc**.

Si pulsas sobre las opciones *Ver ejemplo* podrás comprobar que los resultados de visualización son distintos dependiendo de la extensión de cada fichero.

```
<h3><font color="#ff0000">Este sería un texto  
que se incluiría dentro de la página  
mediante las funciones  
include o require</font></h3><br>
```

[Ver ejemplo1.inc](#)

[Ver ejemplo1.inc.php](#)

Este otro fichero –que también hemos incluido con dos extensiones– contiene **funciones PHP** que pueden ser invocadas desde cualquier otro documento PHP.

Dado que las funciones contienen etiquetas HTML al abrir –mediante el navegador– el documento con extensión *.inc* serán interpretadas y se visualizarán parte de los contenidos. Cuando lleva extensión *.php* se visualizará una página en blanco ya que esas etiquetas están contenidas en funciones que no son invocadas desde el propio script.

```
<?  
function Encabezado() { ?>  
    <HTML>  
    <head>  
    <title>  
    Pruebas con la función include  
    </title>  
    </head>  
    <BODY>  
    <center><br>  
    <font size=6 face="Times" color="#0000ff">Pruebas PHP</font><br>  
    <hr width=75%>  
<? } ?>  
  
<? function Pie() { ?>  
    <center><hr width=50%>  
    <font size=2 face="Arial" color="#ff0000">Luchando con PHP</font>  
    <hr width=50%></center>  
    </body>  
    </html>  
<? } ?>  
  
<? function Calcula($a,$b) {  
    return $a*$b;  
} ?>
```

[Ver ejemplo2.inc](#)

[Ver ejemplo2.inc.php](#)

Incluyendo ficheros

Aquí tienes el código de un documento en el que se *invocan* mediante la función **include** los dos documentos anteriores y se ejecutan sus funciones.

porciones de código HTML en cualquier script, como las del tipo:

```
function nombre {
```

```
.....  
... instrucciones PHP ...  
.....
```

```
}
```

que permiten invocar funciones repetitivas a partir de cualquier documento PHP.

Definidas las funciones en el fichero a *incluir* y colocado al comienzo de la página un *script* que contenga el **include** y la ruta de ese fichero, bastará con invocar cualquiera de las funciones, en cualquier punto del documento, para que esa llamada sea sustituida por el resultado de la ejecución de la función.

Como es lógico, solo serán visualizados en el navegador del cliente los resultados de la ejecución de las funciones que hayan sido *invocadas*.

La función require()

Tiene la misma sintaxis que **include** y una funcionalidad *similar*, aunque con algunas diferencias.

Igual que ocurría con aquél, cuando un fichero es invocado por **require** esa llamada lo que hace es **sustituirse a sí misma** por el contenido del fichero especificado.

A diferencia de **include**, la etiqueta **require** lee y escribe –en el documento a partir del que es invocada– el archivo referenciado completo y **no acepta condicionales** que sí son aceptados por **include**.

Evitar errores por duplicidad de llamadas

Tanto en el caso de usar la instrucción **include** como con **require**, si se intenta incluir **dos o más veces** un fichero que contenga funciones, se producirá un **error** (PHP no permite que dos funciones tengan el mismo nombre) y se interrumpirá la ejecución del script.

Los errores de ese tipo puede evitarse usando las funciones:

```
include_once("fichero")  
y  
require_once("fichero")
```

que a diferencia de *include* y *requiere* van a **impedir** que un mismo fichero pueda incluirse dos veces.

En los ejemplos vas a poder comprobar que no aparece el

```
<!-- empezariamos incluyendo el fichero que contiene  
      las funciones. No escribiría nada hasta que las  
      funciones que contiene no fueran invocadas //-->  
<? include("ejemplo2.inc.php") ?>  
<!-- Insertaremos un script PHP que invoque  
      la función encabezado. Debe llevar las etiquetas  
      de apertura y cierre <? y ?> de PHP //-->  
<? Encabezado() ?>  
<!-- Insertaremos código HTML  
      según nuestra conveniencia //-->  
Aquí iría el contenido de la página<br>  
.... esto es texto HTML.....<br>  
.....<br><br><br>  
<!-- Incluimos el fichero ejemplo1.inc.php y dado que no contiene  
      ninguna función, insertará aquí todo su contenidos //-->  
<? include("ejemplo1.inc.php") ?>  
<!-- Insertaremos más código HTML -->  
.....<br>  
.....<br><br><br>  
<!-- Incluimos nuevamente el fichero ejemplo1.inc.php Puede repetirme  
      la inclusión porque no contiene funciones  
      si las contuviera habría un conflicto de duplicidad  
      porque una función no puede estar definida dos veces  
      con el mismo nombre. La instrucción include  
      como en todos los casos deberá ir dentro de un script PHP  
      y por tanto entre <? y ?> //-->  
<? include("ejemplo1.inc.php") ?>  
<!-- Ahora ejecutaremos la función PHP Calcula() pasando como  
      parámetros 7 y 9. El return de la función nos devolverá  
      el resultado que imprimiremos aquí //-->  
  
<? print "Aquí va el resultado de la multiplicación: ".Calcula(7,9); ?>  
<!-- Por último invocaremos la función Pie() -->  
<? Pie() ?>
```

[Ver ejemplo79.php](#)

La función include_once

```
<!-- Repetimos la inclusión de ambos ficheros  
      pero veremos que no aparece mensaje de error  
      por duplicidad de funciones y además  
      pese a hacer una doble inserción de ejemplo1  
      solo se visualiza una vez por efecto del filtro  
      establecido por include_unde //-->  
<? include_once("ejemplo2.inc.php") ?>  
<? Encabezado() ?>  
Aquí iría el contenido de la página<br>  
.....<br>  
.....<br><br><br>  
<? include_once("ejemplo1.inc.php") ?>  
.....<br>  
no aparecerá nada aquí debajo<br><br><br>  
<? include_once("ejemplo1.inc.php") ?>  
<? include_once("ejemplo2.inc.php") ?>
```

[Ver ejemplo80.php](#)

El resultado de este otro ejemplo es idéntico al anterior. Sólo hemos sustituido **include_once** por **require_once**, que tiene una funcionalidad similar a la anterior.

mensaje de error cuando se utiliza esta función y que el texto que *—anteriormente—* era incluido dos veces ahora solo aparece **una vez**.

Comprobando los ficheros incluidos

PHP dispone de dos funciones que permiten recoger en un **array** la lista de ficheros que se han insertado en el documento por medio de las instrucciones **require**, **require_once** y con **include_once** e **include**.

Tales funciones son estas:

\$v=get_included_files()

Recoge en un **array** **escalar** (contenido en la variable \$v) los nombres de los ficheros incluidos en el archivo en uso por **include_once**.

\$v=get_required_files()

Igual que la función anterior recoge en un **array** **escalar** (contenido en la variable \$v) los nombres de los ficheros incluidos en el archivo en uso mediante **require_once**.

Mejorando la seguridad

Hemos hablado de la función **show_source**, que permitía visualizar el código fuente de los scripts —no solo locales sino de cualquier URL— si esta función no estaba desactivada en el php.ini.

Esa posibilidad de ver no sólo permite el *espionaje industrial* (ver la técnica de construcción de los scripts, etcétera) sino que permite ver también las claves y contraseñas de acceso a las bases de datos que pudieran contener los scripts.

Aparte de simplificar el trabajo la opción de incluir ficheros externos permite **guardar** la *información confidencial* fuera del **root** del servidor y usarla mediante estas llamadas.

De ese modo *—show_source—* permitiría visualizar el nombre de ese fichero externo pero no su contenido.

Si creáramos un directorio —por ejemplo como *subdirectorio* de **c:\Apache** (fuera del root del servidor)— y le ponemos como nombre **sg**, podríamos guardar allí los ficheros *ejemplo2* y *ejemplo1*, con lo cual sus contenidos no serían visibles con *show_source*.

En este caso la instrucción **include** ha de contener el *path* y sería la siguiente:

```
include("C:\Apache\sg\ fichero")
```

```
<? require_once("ejemplo2.inc.php") ?>
<? Encabezado() ?>
Aquí iría el contenido de la página<br>
.....<br>
.....<br><br><br>
<? require_once("ejemplo1.inc.php") ?>
.....<br>
....nada de aquí en adelante ...<br><br><br>
<? require_once("ejemplo2.inc.php") ?>
<? require_once("ejemplo1.inc.php") ?>
```

[Ver ejemplo81.php](#)

La función get_included_files()

```
<? include_once("ejemplo2.inc.php") ?>
<? Encabezado() ?>
<? include_once("ejemplo1.inc.php") ?>
<? Pie() ?>
```

Lista de fichero utilizados por include

```
<?
$z= get_included_files();
foreach($z as $clave=>$valor) {
echo "Clave: ", $clave, " Valor: ", $valor, "<br>";
};
?>
```

[Ver ejemplo82.php](#)

¡Cuidado!

La manera de escribir los **path** difiere de un sistema operativo a otro. Bajo *Windows* debemos usar **** como separador, mientras que otros S.O. (Unix, Linux, etcétera) requieren utilizar **/**.

Para publicar tus páginas en un **hosting no Windows** tendrías que modificar tus scripts. Tenlo en cuenta

Ejercicio nº 27

Crea un subdirectorio con nombre **seguridad** en **C:/ServidoresLocales/Apache/**. Después crea un documento similar a *ejemplo2.inc.php* y guárdalo allí. Crea otro documento *—ejercicio27.php—* en que incluyas algunas de aquellas funciones. Presta atención de incluir el path correcto y comprueba su funcionamiento.

Utilizando include para gestión de fechas

Las funciones que incluye PHP para el manejo de fechas solo contemplan periodos posteriores a 1970. Para el caso de fechas anteriores a esta, existen un funciones que pueden descargarse desde: <http://phplens.com/lens/dl/adodb-time.zip>.

El archivo comprimido contiene un fichero **-adodb-time.inc.php-** con funciones PHP de que se comportan de forma idéntica a las nativas de PHP y que, además, permiten utilizar fechas anteriores a 1970 y valores negativos del tiempo Unix.

Para utilizar estas funciones bastaría con poner en el script: *include("adodb_time.inc.php");* y sustituir las funciones de fecha de acuerdo con lo que se indica en esta tabla:

Función PHP	Función Adodb-time
getdate()	adodb_getdate()

<code>date()</code>	<code>adodb_date()</code>
<code>gmdate()</code>	<code>adodb_gmdate()</code>
<code>mktime()</code>	<code>adodb_mktime()</code>

En [este enlace](#) puedes comprobar los resultados de la aplicación de estas funciones que, como verás, son idénticos a los que hemos visto en el tema [Funciones de fecha](#) con la salvedad de que en este caso se admiten **fechas anteriores a 1970** y **tiempos Unix negativos**.

Anterior



Índice



Siguiente

