

Correo electrónico

PHP dispone de una función que permite el envío de todo tipo de *mensajes de correo electrónico* desde una página web.

Son escasos los **hosting** que tienen activada esta función.

El motivo aducido, por muchos de ellos, para esta restricción no es otra que el riesgo de *abusos* mediante los mensajes conocidos como **spam**.

Como siempre, si tu interés es *publicar* y tienes necesidad de este servicio procura consultar sobre su disponibilidad en el *hosting* que elijas.

Sintaxis

La forma más simple de la función de correo es esta:

mail(dest, asunto, mensaje)

Donde **dest** es la dirección del destinatario, **asunto** es el texto que aparecerá como *Asunto* en el encabezado que recibirá el destinatario y **mensaje** el texto que aparecerá en el *cuerpo del mensaje*.

No te olvides de escribir *entre comillas* esos tres parámetros de la función.

Mensaje con cabeceras MIME

Una forma más completa es la siguiente:

mail(dest, asunto, mens, cabez)

Como puedes ver, en este caso añadimos un nuevo parámetro a la función (**cabez**) que debe estar *entre comillas* y que puede contener, separando con comas, lo siguiente:

From: Nombre <e-mail>

El texto que escribas en el parámetro **Nombre** (*¡cuidado, no lleva comillas!*) será el *nombre del remitente*, que aparecerá en el campo **De:** cuando el destinatario reciba el mensaje.

Donde dice **e-mail** es *obligado* escribir una dirección de correo que debe ir contenida *entre* < >. tal

Requisitos del sistema

La utilización de las funciones de correo electrónico requiere disponer de un servidor de correo electrónico instalado y activo y la modificación de la configuración inicial del fichero *php.ini*.

Si no tienes instalado este servidor puedes hacerlo ahora. En la página [Servidor de correo](#) tienes detallados ambos procesos (instalación y modificación de *php.ini*).

El mensaje más simple

Aquí tienes un ejemplo, el más simple, del uso de esta función:

```
<?
# insertamos la función mail (rojo) dentro de un condicional
# para tener una confirmación de envío y/o aviso de error
# es algo muy habitual para conocer el éxito de la ejecución
# de funciones
if(mail("juan@mispruebas.com", "Mi primer mensaje", "Este es el texto")){
    print "mensaje enviado";
}else{
    print "el mensaje no ha podido enviarse";
}
?>
```

[ejemplo94.php](#)

Un ejemplo más completo

En el ejemplo siguiente insertaremos algunos nuevos elementos comentados al margen. Observa con mucha atención la *estructura* del código. Fíjate que hemos insertado en **líneas diferentes** cada uno de los conceptos: From, Reply-To, etcétera y que **no hemos dejado ningún espacio** al comienzo de esas líneas.

No es por *capricho* ni por afán *estético*. Si insertáramos algún carácter delante de esas líneas se plantearían problemas en la estructura del mensaje y si no incluyéramos un *salto de línea* para cada uno de los conceptos también tendríamos ese mismo problema.

La sintaxis MIME es muy estricta en este sentido. ¡Tengamos mucho cuidado en esto!

Hay otra posibilidad sintáctica –como alternativa a los saltos de línea– ya conocida. Podríamos escribir todo en una sola línea sustituyendo los saltos de línea que ves aquí por **\n**, de forma que el script tuviera un aspecto similar al siguiente:

```
mail("juan@localhost", "Asunto", "Contenido", "\nReply-To: ... \nCc: ... \nBcc: ...")
```

donde, como ves, los **\n** sustituyen a los saltos de línea.

```
<?
mail("juan@mispruebas.com", "Varios destinatarios", "Cuerpo del mensaje",
"From: CursoPHP <juan@mispruebas.com>
Reply-To: andres@mispruebas.com
Cc: perico@mispruebas.com, andres@mispruebas.com
Bcc: andres@mispruebas.com, perico@mispruebas.com
X-Mailer: PHP/" . phpversion());
?>
```

[Usando código de ejemplo](#)

[Utilizando «separador \n»](#)

como puedes ver en el ejemplo.

Reply-To: correo

En **una nueva línea**, y *sin cerrar las comillas de la cadena iniciada en el encabezado anterior*, podemos indicar la **dirección de respuesta** del mensaje.

La dirección que escribamos donde dice **correo** (fíjate que **no** va entre comillas) será la dirección a la que se enviará la respuesta si el destinatario una vez recibido tu correo desea responder mediante la opción **Responder** de su programa de correo electrónico.

Cc: correo1,correo2,...

De igual forma que en el caso anterior *—en una nueva línea y sin cerrar comillas—* podemos incluir en **correo1**, **correo2**, etcétera, las direcciones de correo de las personas a las que deseamos *enviar copia* del mensaje.

No te olvides de **separar con comas** cada una de las direcciones que, como puedes ver en los ejemplos, **no** van entre **comillas**.

Bcc: correo1,correo2,...

Esta opción es idéntica en cuanto a su funcionamiento a la anterior, con la única diferencia que *esas direcciones no serán visibles por los destinatarios de los mensajes*.

X-Mailer:PHP/" . phpversion()

Es una *frivolidad* que se puede incluir en el **encabezado** del mensaje y que indica que versión de PHP con que ha sido creado.

¡Mucho cuidado!

Debemos insistir en sugerirte una **especial atención** a la sintaxis de este tipo de scripts.

Los contenidos generados por la función *mail* deben ser *interpretados* por el servidor de correo y —tanto en este caso como en los que veremos a continuación— los requisitos de formato de éstos servidores son muy estrictos.

De no adaptarse exactamente a sus especificaciones pueden producirse efectos extraños tales como: envíos incorrectos y/o apariencias indeseadas en los mensajes.

El mismo ejemplo, utilizando *variables*

Aquí tenemos un ejemplo donde los parámetros de envío proceden de variables PHP.

Recordemos que esas variables pueden transferirse —mediante un formulario— a un script que se encargue de su envío.

Como puedes observar, hemos puesto las direcciones de los destinatarios de las copias -visibles y ocultas- en sendos *arrays* y hemos añadido una función que: lee los array, los une en una cadena separándolos con comas y, por último, quita la *última coma añadida* utilizando la función **substr**.

```
<?
#variables destinatario, asunto, texto, etc.
$destino="andres@mispruebas.com";
$envia="Andrés Curso PHP";
$remite="andres@mispruebas.com";
$asunto="Mensaje experimental";
$texto="Esto es una prueba. No es spam";

#array de destinatarios de copias visibles

$c[0]="perico@mispruebas.com";
$c[1]="juan@mispruebas.com";

#crear la cadena con las direcciones
# y añadir las comas de separación

foreach($c as $pegar) {
    $cco .= $pegar;
    $cco .= ",";
};

#quitamos la coma del final de la cadena

$l=strlen($cco);

$cco=substr($cco,0,$l-1);

#array de destinatarios de copias OCULTAS

$b[0]="perico@mispruebas.com";
$b[1]="andres@mispruebas.com";

#crear la cadena con las direcciones
# y añadir las comas de separación

foreach($b as $pegar) {
    $bco .= $pegar;
    $bco .= ",";
};

#quitamos la coma del final de la cadena

$l=strlen($bco);

$cco=substr($bco,0,$l-1);

mail($destino, $asunto, $texto,
    "From: $envia <$remite>
    Reply-To: $remite
    Cc: $cco
    Bcc:$bco
    X-Mailer: PHP/" . phpversion());
?>
```

ejemplo96.php

Anterior

Índice

Siguiente