

Las tablas de valores

En las tablas que tenemos aquí a la derecha estamos viendo – clasificadas por tipos– algunas *variables predefinidas* de PHP con sus *valores actuales*.

Esta información (variables y valores) está siendo extraída de la configuración de tu servidor y de este documento.

Es muy probable que esté llamando tu atención el hecho de que **dos nombres** de variable distintos compartan el mismo **valor**.

Intentaremos ver el por qué de esa duplicidad.

Las variables de las columnas de la izquierda comienzan todas por **\$**, mientras que las de la derecha lo hacen por **\$HTTP** y eso es algo **muy importante**.

En el primer caso se alude a las variables **superglobales** que hemos comentado en la página anterior, mientras que en el otro las variables *no tienen ese carácter*.

Razones de la duplicidad

La *duplicidad* de variables se justifica por lo siguiente: Las variables **superglobales** se introdujeron en PHP a partir de la versión 4.1.0 y no existían con anterioridad.

Con muy buen criterio, los *desarrolladores* PHP optaron por mantener las variables predefinidas de las versiones anteriores para evitar que los programadores que migraran sus aplicaciones a estas nuevas versiones de PHP se vieran obligados a *reescribir* el *código* de sus scripts.

Es por esta razón, por la que se mantienen dos **variables** distintas para recoger el mismo valor.

Observa la sintaxis de los nombres de las variables y comprobarás la similitud que existe entre ambas. Los corchetes y sus contenidos son exactamente iguales en ambos casos.

Tal es el caso de estas dos, que, por cierto, devuelven en número de la IP a través de la que está accediendo el usuario a nuestro servidor:

\$_SERVER['REMOTE_ADDR']
y
\$HTTP_SERVER_VARS['REMOTE_ADDR']

cuya similitud resulta evidente.

Aunque los **valores** de ambas variables van a ser siempre **idénticos**, no ocurre lo mismo con su **ámbito**.

Las primeras son de ámbito **superglobal** y no necesitan ser declaradas expresamente con globales cuando se trata de utilizar sus valores dentro de cualquier función.

En el segundo caso va a ser **imprescindible** que sean *declaradas* como **globales** antes

Variables de servidor

\$_SERVER['COMSPEC']	\$HTTP_SERVER_VARS['COMSPEC']
C:\WINDOWS\COMMAND.COM	
\$_SERVER['DOCUMENT_ROOT']	\$HTTP_SERVER_VARS['DOCUMENT_ROOT']
c:/apache/htdocs	
\$_SERVER['HTTP_ACCEPT']	\$HTTP_SERVER_VARS['HTTP_ACCEPT']
image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, application/x-icq, */*	
\$_SERVER['HTTP_ACCEPT_ENCODING']	\$HTTP_SERVER_VARS['HTTP_ACCEPT_ENCODING']
gzip, deflate	
\$_SERVER['HTTP_ACCEPT_LANGUAGE']	\$HTTP_SERVER_VARS['HTTP_ACCEPT_LANGUAGE']
es	
\$_SERVER['HTTP_CONNECTION']	\$HTTP_SERVER_VARS['HTTP_CONNECTION']
Keep-Alive	
\$_SERVER['HTTP_HOST']	\$HTTP_SERVER_VARS['HTTP_HOST']
localhost	
\$_SERVER['HTTP_USER_AGENT']	\$HTTP_SERVER_VARS['HTTP_USER_AGENT']
Mozilla/4.0 (compatible; MSIE 6.0; Windows 98; bri1)	
\$_SERVER['PATH']	\$HTTP_SERVER_VARS['PATH']
C:\WINDOWS;C:\WINDOWS\COMMAND;C:\ARCHIV~1\ARCHIV~1\GTK\2.0\BIN;C:\APACHE;C:\APACHE\PHP4;C:\APACHE\PHP5	
\$_SERVER['PHPRC']	\$HTTP_SERVER_VARS['PHPRC']
c:/Apache/php5/	
\$_SERVER['REMOTE_ADDR']	\$HTTP_SERVER_VARS['REMOTE_ADDR']
127.0.0.1	
\$_SERVER['REMOTE_PORT']	\$HTTP_SERVER_VARS['REMOTE_PORT']
1670	
\$_SERVER['SCRIPT_FILENAME']	\$HTTP_SERVER_VARS['SCRIPT_FILENAME']
c:/apache/htdocs/pdfinal/php16.php	
\$_SERVER['SERVER_ADDR']	\$HTTP_SERVER_VARS['SERVER_ADDR']
127.0.0.1	
\$_SERVER['SERVER_ADMIN']	\$HTTP_SERVER_VARS['SERVER_ADMIN']
pruebas@localhost	
\$_SERVER['SERVER_NAME']	\$HTTP_SERVER_VARS['SERVER_NAME']
localhost	
\$_SERVER['SERVER_PORT']	\$HTTP_SERVER_VARS['SERVER_PORT']
80	
\$_SERVER['SERVER_SIGNATURE']	\$HTTP_SERVER_VARS['SERVER_SIGNATURE']
Apache/1.3.33 Server at localhost Port 80	
\$_SERVER['SERVER_SOFTWARE']	\$HTTP_SERVER_VARS['SERVER_SOFTWARE']
Apache/1.3.33 (Win32) PHP/4.4.0	
\$_SERVER['WINDIR']	\$HTTP_SERVER_VARS['WINDIR']
C:\WINDOWS	
\$_SERVER['GATEWAY_INTERFACE']	\$HTTP_SERVER_VARS['GATEWAY_INTERFACE']
CGI/1.1	
\$_SERVER['SERVER_PROTOCOL']	\$HTTP_SERVER_VARS['SERVER_PROTOCOL']
HTTP/1.1	
\$_SERVER['REQUEST_METHOD']	\$HTTP_SERVER_VARS['REQUEST_METHOD']
GET	
\$_SERVER['QUERY_STRING']	\$HTTP_SERVER_VARS['QUERY_STRING']
\$_SERVER['REQUEST_URI']	\$HTTP_SERVER_VARS['REQUEST_URI']
/pdfinal/php16.php	
\$_SERVER['SCRIPT_NAME']	\$HTTP_SERVER_VARS['SCRIPT_NAME']
/pdfinal/php16.php	
\$_SERVER['PATH_TRANSLATED']	\$HTTP_SERVER_VARS['PATH_TRANSLATED']
c:/apache/htdocs/pdfinal/php16.php	
\$_SERVER['PHP_SELF']	\$HTTP_SERVER_VARS['PHP_SELF']
/pdfinal/php16.php	
\$_SERVER['argv']	\$HTTP_SERVER_VARS['argv']
\$_SERVER['argc']	\$HTTP_SERVER_VARS['argc']

de que sus valores puedan ser utilizados dentro de una función.

¿Nombres algo raros?

Probablemente te extrañará – justificadamente– la longitud y la estructura un tanto rara de estos nombres de variables. Cuando tratemos el tema de los *arrays asociativos* veremos que esa es la sintaxis habitual de ese tipo de variables.

Las nombres de las variables de cada uno de los tipos, sólo se diferencian en lo contenido entre los **corchetes** porque se trata de *distintos elementos* del mismo **array asociativo** y –tal como veremos– esa es la sintaxis típica de los *array*.

¿Para qué sirven?

No vamos a agobiarte con una enumeración de variables y contenidos, pero a poco que observes las tablas de valores te darás cuenta de que es muy abundante y muy importante la información que recogen.

Si analizas las *variables de servidor* te darás cuenta de que aparece un montón de información relativa a su configuración: nombre, rutas, nombres de páginas, IP del servidor, etcétera.

Con los demás tipos ocurre algo similar.

Los distintos tipos

Veamos los diferentes tipos de variables predefinidas que existen en PHP. Por ahora, no te preocupes demasiado sobre la forma de utilizarlas. Las incluimos aquí como una simple enumeración y con una breve descripción de su utilidad.

En temas posteriores haremos referencia a ellas. Por el momento nos bastará con conocer su existencia.

Estamos viendo los valores de las variables de *de entorno (ENV)* y las de *servidor (SERVER)*, pero, además de ellas, existen algunas otras cuyos nombres y utilidades vamos a comentarte.

Variables de sesión

Las identificaremos por los nombres **\$_SESSION** o por **\$HTTP_VARS**.

Este tipo de variables las utilizaremos cuando hagamos mención al uso de **sesiones**.

La utilización de sesiones –ya abundaremos en ello– es una forma de recoger, de *forma temporal* en un documento del mismo carácter, información específica generada a través de los accesos de cada uno de los usuarios.

Por ejemplo, cuando accedes al *Aula Virtual* de este curso y escribes tu clave y contraseña se crea un documento temporal en el servidor del CNICE con un número único y exclusivo para ese acceso – identificador de sesión– que te permite acceder a diferentes apartados de ese espacio sin necesidad de que reescribas, en cada una de las páginas, esos mismos valores.

--

Variables de entorno

\$_ENV['TMP']	\$HTTP_ENV_VARS['TMP']
C:\\WINDOWS\\TEMP	
\$_ENV['TEMP']	\$HTTP_ENV_VARS['TEMP']
C:\\WINDOWS\\TEMP	
\$_ENV['PROMPT']	\$HTTP_ENV_VARS['PROMPT']
\$p\$g	
\$_ENV['winbootdir']	\$HTTP_ENV_VARS['winbootdir']
C:\\WINDOWS	
\$_ENV['COMSPEC']	\$HTTP_ENV_VARS['COMSPEC']
C:\\WINDOWS\\COMMAND.COM	
\$_ENV['PATH']	\$HTTP_ENV_VARS['PATH']
C:\\WINDOWS;C:\\WINDOWS\\COMMAND;C:\\ARCHIV~1\\ARCHIV~1\\GTK\\2.0\\BIN;C:\\APACHE;C:\\APACHE\\PHP4;C:\\APACHE\\PHP5	
\$_ENV['windir']	\$HTTP_ENV_VARS['windir']
C:\\WINDOWS	
\$_ENV['CMDLINE']	\$HTTP_ENV_VARS['CMDLINE']
START C:\\Apache\\Apache.exe -w -f \"C:\\Apache\\conf\\httpd_modulo4.conf\" -d \"C:\\Apache\\\"	

Variables GLOBALES

\$GLOBALS['TMP']
C:\\WINDOWS\\TEMP
\$GLOBALS['TEMP']
C:\\WINDOWS\\TEMP
\$GLOBALS['PROMPT']
\$p\$g
\$GLOBALS['winbootdir']
C:\\WINDOWS
\$GLOBALS['COMSPEC']
C:\\WINDOWS\\COMMAND.COM
\$GLOBALS['PATH']
C:\\WINDOWS;C:\\WINDOWS\\COMMAND;C:\\ARCHIV~1\\ARCHIV~1\\GTK\\2.0\\BIN;C:\\APACHE;C:\\APACHE\\PHP4;C:\\APACHE\\PHP5
\$GLOBALS['windir']
C:\\WINDOWS
\$GLOBALS['CMDLINE']
START C:\\Apache\\Apache.exe -w -f \"C:\\Apache\\conf\\httpd_modulo4.conf\" -d \"C:\\Apache\\\"
\$GLOBALS['DOCUMENT_ROOT']
c:/apache/htdocs
\$GLOBALS['HTTP_ACCEPT']
image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, application/x-icq, */*
\$GLOBALS['HTTP_ACCEPT_ENCODING']
gzip, deflate
\$GLOBALS['HTTP_ACCEPT_LANGUAGE']
es
\$GLOBALS['HTTP_CONNECTION']
Keep-Alive
\$GLOBALS['HTTP_HOST']
localhost
\$GLOBALS['HTTP_USER_AGENT']
Mozilla/4.0 (compatible; MSIE 6.0; Windows 98; bri1)
\$GLOBALS['PHPRC']
c:/Apache/php5/
\$GLOBALS['REMOTE_ADDR']
127.0.0.1
\$GLOBALS['REMOTE_PORT']
1670
\$GLOBALS['SCRIPT_FILENAME']
c:/apache/htdocs/pdffinal/php16.php
\$GLOBALS['SERVER_ADDR']
127.0.0.1
\$GLOBALS['SERVER_ADMIN']

El carácter efímero de las sesiones seguramente lo has comprobado más de una vez cuando al *actualizar* una página te ha aparecido un mensaje diciendo que *la sesión ha caducado*.

Variables del método POST

Las identificaremos por los nombres **\$HTTP_POST_VARS** o por **\$_POST**.

Este tipo de variables –que utilizaremos con frecuencia– recogen la información que se envía desde el **cliente** para ser utilizada por el **servidor**.

Recuerda el carácter *dinámico* de PHP y que ese dinamismo (interacción *cliente* – *servidor*) requiere que el servidor *guarde* los datos remitidos por el cliente.

Variables del método GET

Las identificaremos por los nombres **\$HTTP_GET_VARS** o por **\$_GET**

Son *muy similares* a las anteriores. La existencia de los dos tipos se justifica porque también existen dos tipos de **métodos** (maneras) de enviar datos desde el *cliente* hasta el *servidor*.

Cuando el método de envío es el llamado GET los datos se recogen en variables de este tipo, y, por el contrario, si ese método envío fuera POST se recogerían en aquellas.

Variables de transferencia de ficheros

Las identificaremos por el nombre **\$HTTP_FILES_VARS** o por **\$_FILES**.

Para el caso de *transferencia* de ficheros desde el *cliente* al *servidor* –«*subir ficheros*»– es necesario un *procedimiento* distinto de los anteriores.

Será en este caso cuando se utilicen variables de este tipo.

El tipo GLOBALS

A diferencia de las anteriores, las variables de este tipo disponen de una sintaxis única –**\$GLOBALS**– sin que quepa ninguna otra opción.

Su finalidad es recoger en una *tabla* los nombres de todas las variables establecidas como globales –en cada momento– así como sus valores. Si observas la tabla que tienes aquí a la derecha, quizá te sorprenda leer *nombre* o *página*. ¿De donde han salido esos valores?. Bueno... en esta página utilizamos *scripts* PHP y esos son los nombres de unas variables que hemos incluido en esos *script*.

Conocida la existencia de los diferentes tipos de *variables predefinidas* y vista esta tabla –a modo de ejemplo de su utilidad– no será preciso que profundicemos más en el asunto.

Lo trataremos en el momento en el que tengamos que hacer uso de cada una de ellas.

pruebas@localhost
\$GLOBALS['SERVER_NAME']
localhost
\$GLOBALS['SERVER_PORT']
80
\$GLOBALS['SERVER_SIGNATURE']
Apache/1.3.33 Server at localhost Port 80
\$GLOBALS['SERVER_SOFTWARE']
Apache/1.3.33 (Win32) PHP/4.4.0
\$GLOBALS['WINDIR']
C:\WINDOWS
\$GLOBALS['GATEWAY_INTERFACE']
CGI/1.1
\$GLOBALS['SERVER_PROTOCOL']
HTTP/1.1
\$GLOBALS['REQUEST_METHOD']
GET
\$GLOBALS['QUERY_STRING']
\$GLOBALS['REQUEST_URI']
/pdffinal/php16.php
\$GLOBALS['SCRIPT_NAME']
/pdffinal/php16.php
\$GLOBALS['PATH_TRANSLATED']
c:/apache/htdocs/pdffinal/php16.php
\$GLOBALS['PHP_SELF']
/pdffinal/php16.php
\$GLOBALS['argc']
0
\$GLOBALS['opcion']
I
\$GLOBALS['anterior']
php15.php
\$GLOBALS['siguiente']
php17.php
\$GLOBALS['nombre']
nombre
\$GLOBALS['clave']
siguiente

[Anterior](#) [Indice](#) [Siguiente](#)

