Variables de sesión

La verdadera utilidad de trabajar con *sesiones* estriba en la posibilidad de **propagar** junto con ellas los *valores* de las *variables de sesión*.

Se trata de *ir añadiendo y propagando* variables con sus valores, y de la posibilidad de utilizarlas de la misma forma que se utilizarían las *variables externas* enviadas a través de un *formulario*.

Igual que ocurría en caso de los formularios, también las variables de sesión pueden ser tratadas de forma distinta según como estén configurado PHP (**php.ini**) y cual sea la versión de PHP que utilicemos.

Register globals

Las variables de sesión tienen, como ocurría en otros casos, una sintaxis común (que no depende de la configuración de la directiva register globals) y una específica –añadida a la anterior– para el caso en que register globals esté activado.

A diferencia de los casos que hemos visto anteriormente, cuando se trata de sesiones no existe la similitud sintáctica que existía en aquellos casos.

Esa es la razón por la que no vamos a incluirla. Su utilidad práctica es nula y podría crearnos cierta confusión.

El hecho de que PHP mantenga activas las funciones de esa opción obedece únicamente a razones de tipo práctico. La decisión de eliminarlas de las nuevas versiones podría ocasionar serios perjuicios a programadores que tienen desarrolladas sus aplicaciones con la antigua sintaxis y que se verían obligados a modificar el código fuente de todos sus scripts anteriores.

Manejo de variables

Las funciones más importantes para el manejo de variables de sesión son las siguientes:

$_SESSION[*'var'*]

es una de las formas de definir una variable de sesión.

El índice *var* debe contener entre

Con cualquier opción de *register_globals*

Aquí tenemos un ejemplo en el que utilizamos las funciones PHP específicas para el tratamiento de sesiones.

En el caso de que la versión de PHP no admitiera **superglobales** habría que sustituir $_SESSION por $HTTP_SESSION_VARS y tener presente el carácter no global de esta última variable.

```
<?
# iniciamos la sesión
session_start();
# visualizamos el identificador de sesión
echo "Este es el identificador de sesion: ",session_id(),"<br>";
# registramos una variable de sesión asignandole un nombre
$_SESSION['variable1'];
#asignamos un valor a esa variable de sesión
$_SESSION['variable1']="Filiberto Gómez";
# registramos una nueva variable de sesión
#asignandole directamente un valor
$_SESSION['variable2']="Otro filiberto, este Pérez";
#comprobamos la existencia de la variables de sesión
echo "Mi variable1 esta registrada: ",
isset($_SESSION['variable1']),"<br>";
#leemos el contenido de esa variable
  print "Su valor es: ".$_SESSION['variable1']."<br>";
#comprobamos la existencia de la otra variable y la visualizamos
echo "Mi variable2 esta registrada :",
                    isset($_SESSION['variable2']),"<br>";
print $_SESSION['variable2']."<br>";
#destruimos la variable1
unset($_SESSION['variable1']);
echo "La variable1 ha sido destruida:",
                      isset($_SESSION['variable1']),"<br>";
print $_SESSION['variable1']."<br>";
#destruimos todas las variables restantes
unset($_SESSION);
#comprobamos que han sido destruidas

echo "La variable1 ya estaba vacia:",
                      isset($_SESSION['variable1']),"<br>";
print $_SESSION['variable1']."<br>";

echo "También ha sido destruida la variable2: ",
                                $_SESSION['variable2'],"<br>";
print $_SESSION['variable2']."<br>";

?>
```

ejemplo128.php

Propagación de sesiones

Los tres scripts siguientes son un ejemplo del uso de sesiones para la propagación de sesiones.

Funcionan bajo cualquier forma de register_globals y también en el caso en que las cookies estuvieran desactivadas en el navegador del cliente

```
<?
/* recuerda que entre <? y la primera línea no puede haber
   líneas en blanco ni tampoco puede haberla encima de <?
   aunque como en este caso, si admite líneas de comentario
   pero no líneas en blanco */
```

comillas– el nombre que pretendamos asignarle a esa *variable de sesión*.

Si la variable ya existiera le reasignaría un valor nulo.

$HTTP_SESSION_VARS['*v*']

Es complementaria de la anterior en el caso de que PHP acepte variables *superglabales*.
En en caso de que no fueran aceptadas sería la opción alternativa a aquella.

Ambas se comportan igual que cualquier otra variable. Tanto si existieran previamente como si no hubieran sido creadas anteriormente le asignaría el *valor* indicado.

unset($_SESSION);

La función **unset** destruye las variables contenidas en el paréntesis. En este caso, al contener el array $_SESSION destruiría todas las variables contenidas en él.

unset($_SESSION['*var*']);

Es similar a la anterior. En este caso solo sería destruida la variable de sesión indicada en *var*.

isset($_SESSION['*var*']);

La función **isset** –no es específica del tratamiento de sesiones– devuelve un valor *booleano* (UNO ó NUL) según que exista o no exista la variable contenida en el paréntesis. De hecho, se comporta con las variables de sesión de forma idéntica a como lo haría con cualquier otro tipo de variable.

Propagación de las variables

Las variables $_SESSION['*var*'] creadas en cualquier página, se propagan a todas las demás páginas a las que se propague la sesión, sin que para ello sea necesaria ninguna actuación específica.

Bastará con propagar la sesión siguiendo los procedimientos descritos en la página anterior para que las variables sean transferidas automáticamente.

Esa es la verdadera utilidad de este tipo de variables.

Recuerda que para transferir otros tipos de variables teníamos que recurrir a formularios o a incluir todas las parejas *nombre=valor* en la dirección de la página que habría de recibirlas. No cabe duda que este método añade un gran factor de comodidad y utilidad a la transferencia de información entre páginas del mismo espacio de servidor.

Aunque resulta obvio, quizá no está

```
# deactivamos la opcion de que las páginas puedan guardarse
# en la cache del navegador del cliente
session_cache_limiter('nocache,private');
# le asignamos un nombre a la sesión
# aunque lo habitual sería dejar el nombre por defecto
# que le asigna la configuración de php.ini
session_name('pruebas');
# iniciamos la sesion
session_start();
# creamos variables de sesion y les asignamos valores
$_SESSION['valor1']=25;
$_SESSION['valor2']="Ambrosio de Morales";
$_SESSION['variable3']="Una prueba más";
/* cerramos el script e insertamos un enlace a otra página
   y propagamos la sesión incluyendo en la llamada
   el nombre de la session y su identificador
   En esta página no se visualizaría nada. Solo el enlace */
?>
<A Href="ejemplo130.php?<?echo session_name()."=".session_id()?>">
                  Propagar la sesion</A>
```

ejemplo129.php

```
<?
/* pese a que la sesion viene de la página anterior
   tenemos que poner nuevamente session_cache_limiter
   ya que esta instruccion no se conserva
   solo es válida para la página en la que esta definida
   También tenemos que poner en session_name el mismo
   nombre de la página anterior, de no hacerlo
   PHP entendería que se trata de iniciar una sesion distinta
   Por ultimo también debemos iniciar la sesión
   es obligatorio iniciarla */
session_cache_limiter('nocache,private');
session_name('pruebas');
session_start();
/* comprobaremos que la sesion se ha propagados
   visualizando el array asociativo $_SESSION
   que contiene todas la variables de Sesion */

foreach($_SESSION as $indice=>$valor){
        print("Variable: ".$indice." Valor: ".$valor."<br>");
}
# modificamos los valores de las variables de sesion
# de igual forma que si fueran variables de cualquier otro tipo

$_SESSION['valor1']+=87;
$_SESSION['valor2'] .=" bonito nombre";
# destruimos la tercera variable
unset($_SESSION['variable3']);

# propagamos la sesion a la página siguiente
# con identico proceso al del script anterior
?>
<A Href="ejemplo131.php?<?echo session_name()."=".session_id()?>">
                  Propagar la sesion</A>
```

```
<?
# identicos comentarios a los anteriores
session_cache_limiter('nocache,private');
session_name('pruebas');
session_start();
# este bucle nos confirmará que se han propagado
# los nuevos valores y que la tercera variable ha sido destruida
foreach($_SESSION as $indice=>$valor){
        print("Variable: ".$indice." Valor: ".$valor."<br>");
}
?>
```

de más comentar que las variables de sesión no se transmiten más que entre páginas alojadas en el mismo espacio de servidor.