cpifp Los Enlaces

DESARROLLO WEB EN ENTORNO SERVIDOR

2º DESARROLLO DE APLICACIONES WEB

TEMA 10. RESUMEN COOKIES Y SESIONES

cpifp Los Enlaces

T10. RESUMEN COOKIES Y SESIONES

1. SESIONES

Una sesión es la navegación que hace un determinado usuario por un sitio web. La sesión se refiere a todas las peticiones que realiza el usuario vinculadas al mismo sitio web.

Durante toda la sesión el usuario puede haber establecido preferencias y elegido opciones que interese recordar. Eso es lo que se denomina el **estado de una sesión**.

¿Cómo almacenamos el estado de una sesión?

cpifp Los Enlaces

T10. RESUMEN COOKIES Y SESIONES

2. COOKIES

Una de las opciones que manejan los navegadores es el almacenamiento mediante cookies.

Las cookies son archivos de texto que almacenan información sobre el usuario y que se guardan en el propio ordenador del usuario. Contienen esencialmente nombres de variables y valores (de tipo string) para dichas variables.

El mayor inconveniente con el uso de cookies es que el cliente de un sitio web puede prohibir o borrar las cookies almacenadas.

T10. RESUMEN COOKIES Y SESIONES

2. COOKIES

La forma de crear una cookie desde PHP es con la función

```
setcookie (nombre[,valor][,expiración][,ruta]
          [,dominio][,segura][,solohttp])
```

Parámetros:

- **nombre**: se crea el elemento $_COOKIE[“nombre”].
- **valor**: si no se indica este parámetro se borra la cookie.
- **expiración**: por defecto es 0, se puede configurar con time( ) + 3600 o el tiempo en segundos que queremos que dure.

T10. RESUMEN COOKIES Y SESIONES

2. COOKIES

La forma de crear una cookie desde PHP es con la función

```
setcookie (nombre[,valor][,expiración][,ruta]
          [,dominio][,segura][,solohttp])
```

Parámetros:

- **dominio**: por defecto el dominio actual.
- **segura**: por defecto *false*, si indicamos *true* solo se almacenará si la comunicación es mediante protocolo seguro.
- **solohttp**: https://diego.com.es/cookies-http

cpifp Los Enlaces

T10. RESUMEN COOKIES Y SESIONES

2. COOKIES

Ejemplo de almacenamiento de datos en cookies:

```
<h1>Ejemplo de uso de cookie</h1>
  Introduzca su nombre:
  <form action="ejemplo1_2crearcookie.php"  method="GET">
    <input type="text" name="nombre"><br>
    <input type="submit" value="Enviar">
  </form>
```

cpifp Los Enlaces

T10. RESUMEN COOKIES Y SESIONES

2. COOKIES

Ejemplo de almacenamiento de datos en cookies:

```
<?php setcookie("ejemplocookie", $_GET['nombre'], time()+3600,"/",""); ?>
//Parámetros rellenados: nombre, valor, expiración, ruta y dominio
<!DOCTYPE html>
...
    <h1>Ejemplo de uso de cookie</h1>
    <p>Se ha establecido una cookie de nombre  <b>ejemplocookie</b> con el valor: <b><?php print $_GET['nombre']; ?></b>, que será válida durante 1 hora.</p>
```

cpifp Los Enlaces

T10. RESUMEN COOKIES Y SESIONES

2. COOKIES

Ejemplo de almacenamiento de datos en cookies:

```
<h1>Ejemplo de uso de cookie</h1>
<?php
   //Así obtenemos el valor de la cookie y lo mostramos.*/
   echo "Se ha recibido la cookie de nombre <b>ejemplocookie</b> con valor:
   <b>".$_COOKIE['ejemplocookie']."</b>";
?>
```

T10. RESUMEN COOKIES Y SESIONES

2. COOKIES

Se puede sobreescribir el valor de una cookie con la misma función `setcookie()`. Simplemente utilizaremos el mismo nombre de la cookie almacenada y en el segundo parámetro le daremos su nuevo valor.

Si utilizamos el valor `false` o lo dejamos vacío estaremos borrando la cookie:

```
setcookie("nombre");
```

cpifp Los Enlaces

T10. RESUMEN COOKIES Y SESIONES

2. COOKIES

Ejemplo de modificación de cookies:

```php
<?php
  if(isset($_COOKIE['contador'])) {
    setcookie('contador', $_COOKIE['contador'] + 1, time() + 365*24*60*60);
    // Se incrementa la cookie contador una unidad. Caduca en un año
  } else {
    // Se crea y envía la cookie al ordenador cliente. Caduca en un año
    setcookie('contador', 1, time() + 365*24*60*60);
  }?>
```

cpifp Los Enlaces

T10. RESUMEN COOKIES Y SESIONES

2. COOKIES

Ejemplo de modificación de cookies:

```
<body>
<?php
  echo "<p style='text-align:right;font-weight:bold;'>Número de visitas:" ;
    if (!isset($_COOKIE['contador']))
      echo 1;
    else echo $_COOKIE['contador'];
  echo "</p>";
</body>
```

T10. RESUMEN COOKIES Y SESIONES

2. COOKIES

En la cookies no se pueden almacenar datos binarios, sólo strings. Para convertir datos binarios a texto, PHP dispone de la función `serialize()`:

```
setcookie("notas",serialize(array(9,7,6,5)));
```

Para recuperar dichos datos, que ahora se encuentran en formato *string*, se usa la función `unserialize()`:

```
$array=unserialize($_COOKIE["notas"]);
```

T10. RESUMEN COOKIES Y SESIONES

3. VARIABLES DE SESIÓN

Ventajas respecto al uso de cookies:

- Pueden funcionar con las cookies desactivadas pasando el identificador de sesión por GET.
- Almacenan más datos que una cookie, incluidos binarios.
- Los datos de sesión se almacenan en el servidor, donde las medidas de seguridad son mayores que en el cliente normalmente.

T10. RESUMEN COOKIES Y SESIONES

3. VARIABLES DE SESIÓN

Desventajas respecto al uso de cookies:

- Vulnerables ante *session hijacking* (secuestro de sesión).
- El identificador de sesión se almacena mediante cookies, si éstas están deshabilitadas en el lado del cliente hay que implementar un método para comunicar dicho identificador entre páginas.

cpifp Los Enlaces

T10. RESUMEN COOKIES Y SESIONES

3. VARIABLES DE SESIÓN

Para crear una nueva sesión o mantener abierta una sesión ya creada se usa la función `session_start()`.

A partir de entonces, su usan las variables de sesión mediante el array $_SESSION.

Para eliminar una variable de sesión se utiliza la función `unset()` y para finalizar la sesión `session_destroy()`.

cpifp Los Enlaces

T10. RESUMEN COOKIES Y SESIONES

3. VARIABLES DE SESIÓN

Ejemplo de almacenamiento de datos en cookies:

```
<?php
  session_start();  // Se crea la sesión al acceder al sitio
?>
...
<body>
<?php $_SESSION['var']='Hola mundo';  // Se crea una variable de sesión
      echo $_SESSION['var']; ?>
</body>
```

T10. RESUMEN COOKIES Y SESIONES

3. VARIABLES DE SESIÓN

Ejemplo de almacenamiento de datos en cookies:

```
<?php
  session_start();  // Se continúa con la sesión creada en la página
anterior
?>
...
<body>
<?php echo $_SESSION['var'];
      unset ($_SESSION['var']); ?>  // Se elimina la variable de sesión
</body>
```

cpifp Los Enlaces

T10. RESUMEN COOKIES Y SESIONES

3. VARIABLES DE SESIÓN

Ejemplo de almacenamiento de datos en cookies:

```
<?php
  session_start();  // Se continúa con la sesión creada
?>
...
<body>
<?php unset ($_SESSION);      // Se elimina el array para liberar memoria
      session_destroy(); ?>  // Se finaliza la sesión
</body>
```

T10. RESUMEN COOKIES Y SESIONES

3. VARIABLES DE SESIÓN

Si se elimina el array $_SESSION en algún momento para borrar todas las variables de sesión a la vez pero se quiere continuar con la sesión, se debe crear de nuevo con `$_SESSION = array()`.

El tiempo de espera de la sesión de PHP depende de la configuración del servidor o de las directivas relevantes *session.gc_maxlifetime* en el archivo *php.ini*. Por lo general, el tiempo de espera predeterminado de la sesión de PHP es de 24 minutos (1440 segundos) y no es necesario usar la función `session_destroy()`.

T10. RESUMEN COOKIES Y SESIONES

3. VARIABLES DE SESIÓN

Existen dos formas de pasar el identificador de sesión entre peticiones:

- Mediante una cookie: `setcookie("PHPSESSID", "dh047snn89mtibpck3tth5rk83")`
- Mediante GET: `localhost/ejemplo3?PHPSESSID=dh047snn89mtibpck3tth5rk83`

Se puede configurar en el servidor el método a utilizar desde el archivo *php.ini*:

➔ `session.use_cookies = 1 (ó 0)`

➔ `session.use_trans_id = 0 (ó 1)`

cpifp Los Enlaces

T10. RESUMEN COOKIES Y SESIONES

3. VARIABLES DE SESIÓN

Además, PHP cuenta con las funciones `session_name` y `session_id` para obtener o modificar los datos de sesión:

```
<?php
  session_start();  // Se crea la sesión
?>
...
<body>
<?php
   echo session_id(),"<br>";    // Pedimos que escriba el identificador
único
   echo session_name(),"<br>";  // Y el nombre de la sesión
```

T10. RESUMEN COOKIES Y SESIONES

3. VARIABLES DE SESIÓN

Para hacer uso del método GET, debemos pasar este nombre y su ID a través de la URL:

```
<?php
  session_start();  // Se crea la sesión
?>
...
<body>
  <a href="ejemplo3_mostrarsesion2.php?
    <?php echo session_name()."=".session_id()?>
  ">Volver a llamar esta página</a>
```