

Programación III

PHP

Clase 4

Maximiliano Neiner

Temas a Tratar

- ❖ Subir archivos al servidor
- ❖ Variables de sesión
- ❖ Cookies

Temas a Tratar

- ❖ Subir archivos al servidor
- ❖ Variables de sesión
- ❖ Cookies

Subir archivos en PHP(1/3)

- Para poder subir archivos al servidor, es necesario crear un formulario en HTML que le permita a los usuarios seleccionar un archivo.

```
<!doctype html>
<html>
<head>
    <title>Subir Archivos</title>
</head>
<body>
    <form action="upload.php" method="post"
          enctype="multipart/form-data" >

        <input type="file" name="archivo" />
        <input type="submit" value="Subir" />
    </form>
</body>
</html>
```

Subir archivos en PHP(2/3)

- ❖ **Algunas reglas a seguir para el formulario HTML.**
 - El método del form debe ser POST.
 - El form necesita del atributo **enctype**. Dicho atributo especifica el contenido/tipo a usarse cuando se ‘submitea’ el form.
 - El input de tipo **FILE** permite mostrar una ventana modal para navegar en busca del archivo a ser subido.

- ❖ **Sin los requerimientos indicados la subida de archivos fallará.**

Subir archivos en PHP(3/3)

- Del lado del servidor, tenemos que manipular el archivo recibido en `$_FILES`.
- Utilizando funciones de PHP deberemos mover el archivo subido desde su ubicación temporal a la ubicación definitiva dentro del servidor.

```
<?php  
  
$destino = "uploads/" . $_FILES["archivo"]["name"] ;  
  
move_uploaded_file($_FILES["archivo"]["tmp_name"] , $destino) ;  
  
?>
```

\$_FILES

- ✖ Es una **super global** que existe a partir de PHP **4.1.0**.
- ✖ Es un array asociativo de elementos cargados al script actual a través del método POST.
 - **[name]** => nombre del archivo (con su extensión).
 - **[type]** => tipo del archivo (dato por el navegador).
 - **[tmp_name]** => carpeta temporal donde se guardará el archivo subido.
 - **[error]** => código de error (si es 0, no hubo errores).
 - **[size]** => tamaño del archivo, medido en bytes.

Demo

Atributos

- ❖ El nuevo atributo booleano ***multiple*** (de HTML5) permite que los usuarios seleccionen varios archivos para ser subidos al servidor.
- ❖ El atributo ***accept***, permite filtrar (en el cliente) los tipos de archivos que se permitirán subir al servidor.

```
<form action="upload.php" method="post"
      enctype="multipart/form-data" >

    <input type="file" name="archivos[]" multiple
           accept="image/png,.jpg,image/gif" />
    <input type="submit" value="Subir" />
</form>
```

Demo

Temas a Tratar

- ❖ Subir archivos al servidor
- ❖ Variables de sesión
- ❖ Cookies

Temas a Tratar

❖ Subir archivos al servidor

❖ Variables de sesión

- Generalidades
- Abrir una sesión
- Destruir una sesión

❖ Cookies

VARIABLES DE SESIÓN

- Las variables de sesión son una forma de guardar información de un usuario particular, que puede ser usada en distintas páginas del sitio Web.
- La información no es almacenada en el cliente.
- Dado que HTTP no mantiene estado entre páginas, la utilización de variables de sesión permite mantener información acerca de un solo usuario, y están disponibles para todas las páginas del sitio Web.
- Por defecto, las variables de sesión duran hasta que el usuario cierra el navegador.

Temas a Tratar

❖ Subir archivos al servidor

❖ Variables de sesión

- Generalidades
- Abrir una sesión
- Destruir una sesión

❖ Cookies

Abrir una Sesión (1/2)

- ❖ Una sesión se inicia con la función ***session_start()***.
- ❖ Dicha función debe estar declarada en cada script al que deseamos utilizar con variables de sesión.
- ❖ Las variables de sesión se establecen con la variable super global de PHP: ***\$_SESSION***.

```
<?php  
    session_start();  
  
    $_SESSION["CLAVE"] = "VALOR";  
?>
```

Abrir una Sesión (2/2)

- La mayoría de las sesiones establecen una clave de usuario en el cliente, que se parece a:
765487cf34ert8dede5a562e4f3a7e12.
- Cuando una sesión se abre en otra página, se examina el equipo para obtener una clave de usuario.
- Si hay una coincidencia, se accede a esa sesión, si no, se inicia una nueva sesión.

```
<?php  
    session_start();  
  
    echo $_SESSION["CLAVE"];  
?>
```

Temas a Tratar

❖ Subir archivos al servidor

❖ Variables de sesión

- Generalidades
- Abrir una sesión
- Destruir una sesión

❖ Cookies

Destruir una Sesión

- Para eliminar todas las variables de sesión globales y destruir la sesión, hay que usar `session_unset()` y `session_destroy()`

```
<?php
    //remueve todas las variables de sesión
    session_unset();

    //destruye la sesión
    session_destroy();

?>
```

Ejemplos

Temas a Tratar

- ❖ Subir archivos al servidor
- ❖ Variables de sesión
- ❖ Cookies

Temas a Tratar

- ❖ Subir archivos al servidor
- ❖ Variables de sesión
- ❖ Cookies
 - Generalidades
 - Establecer una cookie
 - Borrar una cookie

Cookies

- ✖ Una cookie se utiliza a menudo para identificar a un usuario.
- ✖ Una cookie es un pequeño archivo que el servidor guarda en el cliente.
- ✖ Cada vez que el mismo equipo solicita una página con un navegador, se enviará la cookie también.
- ✖ Con PHP, se puede tanto crear como recuperar valores de cookies.

Temas a Tratar

 Subir archivos al servidor

 Variables de sesión

 Cookies

- Generalidades
- Establecer una cookie
- Borrar una cookie

Establecer una cookie (1/2)

- **setcookie()** define una cookie para ser enviada junto con el resto de las cabeceras de HTTP.
- Al igual que otras cabeceras, las cookies deben ser enviadas antes de que el script genere ninguna salida (es una restricción del protocolo).
- Esto implica que las llamadas a esta función se coloquen antes de que se genere cualquier salida, incluyendo las etiquetas `<html>` y `<head>` al igual que cualquier espacio en blanco.
- Una vez que han sido enviadas las cookies, se puede acceder a ellas en la próxima carga de la página gracias a los arrays `$_COOKIE`.

Establecer una cookie (2/2)

```
bool setcookie(string name [, string value [, int expire = 0  
[, string path [, string domain [, bool secure = false  
[, bool httponly = false ]]]]] )
```

- ✖ A excepción del parámetro name, los parámetros son opcionales.
- ✖ Si existe algún tipo de salida anterior a la llamada de esta función, setcookie() fallará y retornará FALSE.
- ✖ Si setcookie() ejecuta satisfactoriamente, retornará TRUE.
- ✖ Esto no indica si es que el usuario ha aceptado la cookie o no.

Parámetros de setcookie (1/2)

- ✖ **name:** Indica el nombre de la cookie.
- ✖ **value:** Establece el valor de la cookie. Este valor se guarda en el cliente.
- ✖ **expire:** indica el tiempo en el que expira la cookie. Es una fecha Unix por tanto está expresada en números de segundos a partir de la presente época. (*1)
- ✖ **path:** Indica la ruta dentro del servidor en la que la cookie estará disponible. (*2)
- ✖ **domain:** El dominio para el cual la cookie está disponible. (*3)

Parámetros de setcookie (2/2)

- ❖ **secure:** Establece si la cookie sólo debiera transmitirse por una conexión segura HTTPS desde el cliente. Cuando se configura como TRUE, la cookie sólo se creará si es que existe una conexión segura. (*1)
- ❖ **httponly:** Cuando es TRUE la cookie será accesible sólo a través del protocolo HTTP. Esto significa que la cookie no será accesible por lenguajes de scripting, como JavaScript. (*2)

Ejemplo

```
$valor1 = 'valor_cookie_1';
$valor2 = 'valor_cookie_2';
$valor3 = 'valor_cookie_3';

setcookie("TestCookie1", $valor1);
setcookie("TestCookie2", $valor2, time()+3600);
setcookie("TestCookie3", $valor3, time()+3600, "/~cookie/", "test.com", 1);
```

Temas a Tratar

 Subir archivos al servidor

 Variables de sesión

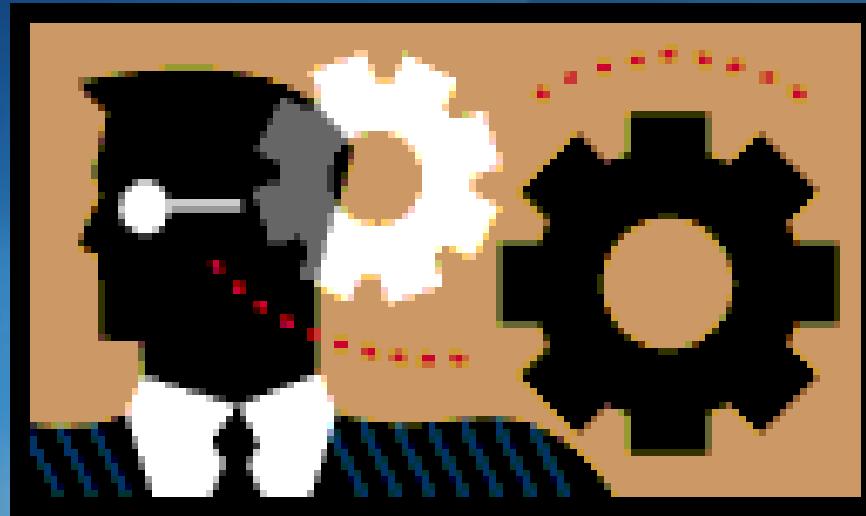
 Cookies

- Generalidades
- Establecer una cookie
- Borrar una cookie

Borrar una cookie

- Para borrar una cookie se debe asegurar que la fecha de expiración ya ha pasado, de este modo se detonará el mecanismo de eliminación del navegador.

```
setcookie("TestCookie2", " ", time()-3600);
setcookie("TestCookie3", " ", time()-3600, "/~cookie/", "test.com", 1);
```



Ejercitación