

MANEJO DE FICHEROS

PHP

FICHEROS

- Como veremos más adelante las cookies y nos sirven para guardar datos del cliente, pero no los guardan de forma permanente.
- La forma más segura de guardar los datos sin que estos se pierdan es creando un archivo.
- Un fichero es un almacén de datos en dispositivos externos que hace que los datos sean persistentes al finalizar los programas

Proceso de trabajo con Ficheros

- Apertura del fichero
 - Obtención del puntero
- Procesamiento de fichero
 - obtener o grabar información
 - (leer/escribir)
- Cierre del fichero

Proceso de trabajo con Ficheros

- Las operaciones sobre ficheros suelen constar de tres fases:
 - Apertura del fichero
 - Se abre el fichero, indicando si se realizarán operaciones para leer, escribir o añadir al final del mismo.
 - Esta operación devuelve un descriptor de fichero que se usará en el resto de funciones.

Proceso de trabajo con Ficheros

- ¿Para qué sirve el puntero que devuelve fopen?
 - Define un canal a través del cual se accede al fichero.
 - Desde que el fichero está abierto se trabaja con el puntero.
 - Cuando se abre el fichero, el puntero se coloca al principio del fichero para esperar instrucciones.
- Procesamiento del fichero
 - Lectura-Escritura.
- Cierre del fichero

Proceso de trabajo con Ficheros

- Apertura `fopen()`:
 - Los archivos en PHP se abren con la función `fopen()`, que requiere dos parámetros:
 - El nombre del archivo que se quiere abrir y el modo en el que abrir el archivo.
 - La función devuelve un puntero en el archivo si es satisfactoria, o cero si no lo es, por eso se debe comprobar si ha fallado la operación.
 - Si no se ha podido abrir el fichero hacemos que finalice la ejecución del script devolviendo un mensaje de error.
 - En caso contrario el script continúa ejecutándose

Modo	Descripción
r	Apertura para lectura. Puntero al principio del archivo
r+	Apertura para lectura y escritura. Puntero al principio del archivo
w	Apertura para escritura. Puntero al principio del archivo y lo sobrescribe. Si no existe se intenta crear.
w+	Apertura para lectura y escritura. Puntero al principio del archivo y lo sobrescribe. Si no existe se intenta crear.
a	Apertura para escritura. Puntero al final del archivo. Si no existe se intenta crear.
a+	Apertura para lectura y escritura. Puntero al final del archivo. Si no existe se intenta crear.
x	Creación y apertura para sólo escritura. Puntero al principio del archivo. Si el archivo ya existe dará error E_WARNING. Si no existe se intenta crear.
x+	Creación y apertura para lectura y escritura. Mismo comportamiento que x.
c	Apertura para escritura. Si no existe se crea. Si existe no se sobrescribe ni da ningún error. Puntero al principio del archivo.
c+	Apertura para lectura y escritura. Mismo comportamiento que C.

APERTURA

- 'a+' lectura/escritura. Puntero al final. Añade no borra.
- 'x', 'x+' Igual que w, w+ pero evitando la creación si el fichero existe. Si el archivo ya existe dará error E_WARNING. Si no existe se intenta crear..
- 'c', 'c+' Igual que w , w+ pero sin truncar el fichero cuando ya existe. Si no existe se crea. Si existe no se sobre-escribe ni da ningún error. Puntero al principio del archivo.

APERTURA

- Para verificar que la operación `fopen()` ha tenido éxito:
`$fichero = fopen("datos.txt", 'r+');`--de lectura/escritura
`$fichero = fopen("datos.txt", 'w+');`--de lectura/escritura
 - Si no se ha podido abrir el fichero finaliza la ejecución del script devolviendo un mensaje de error:
 - **`if(!$fichero) die("ERROR: no se ha podido abrir el fichero de datos");`**
 - `die()`: Provoca la finalización de la ejecución del script mostrando el mensaje recibido como parámetro *f*.
 - En caso contrario el script continúa ejecutándose

CIERRE FICHEROS

- CIERRE:
 - Aunque php cierra todos los punteros de archivos al final de la ejecución del script, aunque se considera una buena práctica cerrar los archivos manualmente con `fclose()`.
 - `fclose(Puntero-de-apertura);`

procesamiento de Ficheros

Lectura

- `fgets($puntero, [bytes])`
 - Obtiene una línea del fichero
 - Se termina de leer cuando llega al final de línea, final del fichero o el último byte de datos.
 - `byte` indica cuantos bytes (caracteres) queremos leer del fichero (opcional)
- `fread($puntero, longitud)`
 - Es similar a `fgets()`, pero se lee todo el fichero(o hasta el carácter `longitud`), no va línea por línea como `fgets()`
- Para almacenar todo el contenido del fichero en un array(sin ser necesario abrir el fichero):
 - `file('nombre_fichero')`

- Ejemplo: leer el contenido de un fichero línea a línea:

```
$a = fopen('datos.txt', 'r');  
while(!feof($a)) {  
    echo fgets($a) . '<br>';  
}  
fclose($a);
```

- Ejemplo: leer el contenido de un fichero de una vez.

```
$a = file('datos.txt');  
foreach($a as $linea)  
    echo $linea . '<br>';
```

- Ejemplo: mostrar una página web de otro sitio web(las imágenes fallan si las URL son relativas)

```
$a =file('http://www.upm.es/index.html');  
foreach($a as $linea)  
    echo $linea;
```

Ejemplo de lectura con fread.

```
<?php
$gestor = fopen("misdatos.txt", "r");
$contenido = "";
echo $contenido .= fread($gestor,
8192);
fclose($gestor);
?>
```

- La variable \$contenido guardará el contenido que hayamos leído. Esta función requiere dos parámetros, el archivo abierto y la longitud que queremos leer de dicho archivo (en bytes)

Ejemplo de lectura con fread.

```
$file = "miarchivo.txt";  
$fp = fopen($file, "r");  
$contenido = fread($fp, filesize($file))
```

- La variable \$contenido guardará el contenido que obtengamos con la función fread(). Esta función requiere dos parámetros, el archivo abierto y la longitud que queremos leer de dicho archivo (en bytes). En este caso hemos empleado la función filesize() para obtener el tamaño del archivo y así devolver todo su contenido

Escritura de Ficheros

- `fwrite ($fichero, cadena, longitud)`
 - Escribe en `$fichero` los caracteres de `cadena`.
 - Si se añade el parámetro `longitud`, escribe hasta que termine la cadena o hasta que se alcancen los caracteres indicados en este parámetro, lo que antes ocurra.
 - Devuelve el número de caracteres escrito.
- `fputs` es una alias de `fwrite`

```
$a = fopen('datos.txt', 'w+');  
fwrite($a, "nueva linea\r\n");  
fputs($a, "otra linea\r\n");  
fclose($a);
```

FUNCIONES PARA PROCESAR FICHEROS

- **feof()** es utilizada con frecuencia en el manejo de archivos en PHP. Esta función comprueba si el puntero se encuentra al final del archivo. Se utiliza cuando se recorre un archivo línea por línea.
- **Seeking** sirve para mover el puntero de un archivo. Para mover el puntero se puede usar la función `fseek()`, y para encontrar la posición de un puntero dado `ftell()`.
- **rewind** —Establece el indicador de posición de archivo al principio del archivo.

FUNCIONES PARA PROCESAR FICHEROS

```
$file = "miarchivo.txt";
$texto = "Hola que tal 12345";
$fp = fopen($file, "w");
fwrite($fp, $texto);
fclose($fp);
$fp = fopen($file, "r");
// Si lo hemos abierto con r, siempre empieza desde el principio:
echo ftell($fp) . "<br>"; // Devuelve 0
// Colocamos el apuntador en la posición 10:
fseek($fp, 10);
// Mostramos la posición actual:
echo ftell($fp) . "<br>"; // Devuelve 10
// Se puede indicar una posición sin contenido:
fseek($fp, 1000);
echo ftell($fp) . "<br>"; // Devuelve 1000
// Para ir al final del archivo se emplea un tercer argumento en fseek:
fseek($fp, 0, SEEK_END);
echo ftell($fp) . "<br>"; // Devuelve 18
```

Ejemplo

- El siguiente ejemplo, vamos a abrir un fichero de solo lectura y vamos a recuperar la información posicionándonos de forma arbitraria en distintas posiciones.
- Primero nos posicionamos en el carácter número 19 y recupera mediante lectura los 18 caracteres siguientes del Texto2.txt, a continuación, se posiciona en el carácter número 15 y lee los siguientes 4 caracteres y por último, se posiciona al principio con la función **rewind()** y se leen los primeros 14 caracteres.

```
<Body>
```

```
<?Php
```

```
$nombre="Texto2.txt";
```

```
$f = fopen($nombre, r ) or die("Error al abrir el archivo: $nombre");
```

```
echo "<H3>";
```

```
fseek($f, 19);
```

```
echo fread($f, 18);
```

```
echo "<Br>";
```

```
fseek($f, 15);
```

```
echo fread($f, 4);
```

```
echo "<Br>";
```

```
rewind($f);
```

```
echo fread($f, 14);
```

```
fclose($f);
```

```
?>
```

```
</h3>
```

```
</Body>
```

```
</Htm>
```

FUNCIONES PARA PROCESAR FICHEROS

- Otras operaciones:
 - **Copiar un fichero:**
 - `copy(origen, destino)`
 - **Renombrar (o mover) un fichero:**
 - `rename(nombre_original, nombre_final)`
 - **Borrar un fichero:**
 - `unlink(fichero)`
 - **Truncar un fichero**
 - `ftruncate(descriptor, longitud)`
 - Trunca el archivo a la longitud en bytes dada.
 - **Tamaño de un fichero:**
 - `filesize(path)` → en bytes
 - **Última modificación del fichero**
 - `filemtime(path)` Devuelve marca de tiempo en que se modificó, formatearla con `date`

FUNCIONES PARA PROCESAR FICHEROS

- Otras operaciones:
 - **chgrp** → Cambia el grupo de un archivo
 - **chmod** → Cambia permisos de un archivo
 - **chown** → Cambia el propietario de un archivo
 - **is_executable** → Indica si el archivo es ejecutable
 - **is_file** → Indica si el archivo es un archivo regular
 - **is_link** → Indica si el archivo es un enlace simbólico
 - **is_readable** → Indica si es posible leer el archivo
 - **is_uploaded_file** → Indica si un archivo fue cargado a través de HTTP POST
 - **is_writable** → Indica si el nombre de archivo es escribible
 - **is_writeable** → Alias de **is_writable**
 - **filetype(path)** → Devuelve el tipo de un archivo.

EJEMPLO

Envíe sus datos para guardarlos en el fichero

Nombre:

Telefono:

Email:

Clave:

nombre : Jose

Telefono : 222222222

correo : jose@gmail.com

nombre fichero : miarchivo

Tu archivo se ha guardado con el nombre "miarchivo.txt"

```
<?php
if (isset($_REQUEST['enviar']))
{ $nombre=$_POST['nombre'];
    echo " nombre : $nombre<br>";
    $tel=$_POST['tel'];
    echo " Telefono : $tel<br>";
    $email=$_POST['email'];
    echo " correo : $email<br>";
    $clave=$_POST['clave'];
    echo " nombre fichero : $clave<br>";
//4-de escritura y lectura, si existe machaca y si no existe escribe y crea
    $archivo=fopen($clave.'.txt','w+');
fwrite($archivo, "Nombre: ".$nombre." Teléfono: ".$tel." E-mail: ".$email." Clave: ".$clave."<br>");
echo "Tu archivo se ha guardado con el nombre \"\$clave.txt\"<br>";
//cerrar archivo
    fclose($archivo);
}
else
{echo'<h1> Envie sus datos para guardarlos en el fichero</h1>';
?>
<body>
<form action="ejmfig1.php" method="post">
<p>Nombre: <input type="text" name="nombre" /></p>
<p>Telefono: <input type="text" name="tel" /></p>
<p>Email: <input type="text" name="email" /></p>
<p>Clave: <input type="text" name="clave" /></p>
<p><input type="submit" value="enviar" name='enviar' /></p>
</form>
<?php
;}
?>
</body>
```