

MÓDULO 2ºDAW: Desarrollo Web en entorno Servidor.

Autor: Jorge López.



Tema 02: Gestión de Ficheros de texto en PHP.

01 Ficheros de texto

Apertura de ficheros en PHP

- `fopen()`

Cierre de un fichero

- `fclose()`

Ejemplo de creación de un fichero

- `fwrite()`

Lectura de un fichero

- `fread()`
- `rewind()`
- `feof()`
- `fgets()`
- `fgetc()`

Scripts de ejemplo:

- `fichero1.php`: creación de un fichero de texto
- `fichero2.php`: lectura de un fichero de texto

01 Ficheros de texto.

Apertura de ficheros en PHP:

Para abrir un fichero en PHP se utiliza la función **fopen()** que devuelve un recurso que apunta al fichero abierto. Los **3 parámetros** que se pasan a esta función son los siguientes:

```
@fopen('archivo.txt', 'a+b', true);
```

- **Nombre del fichero:** el nombre del fichero que se quiere abrir.
- **Modo de apertura:** especifica el tipo de acceso que se tendrá al fichero. Os incluyo **la tabla de la especificación oficial** para que veáis los diferentes modos de apertura.
- **Búsqueda:** El tercer parámetro es opcional, es un booleano que indica si debe buscar el archivo en la directiva **include_path** (además de buscar el fichero en la carpeta de trabajo buscará además en las rutas que haya especificadas en la cláusula **include_path**) en el archivo de configuración de PHP. (**no utilizaremos dicho parámetro**).

Modo	Descripción
r	Apertura para sólo lectura, <u>coloca el puntero al principio del archivo</u> .
r+	Apertura para lectura y escritura, <u>coloca el puntero al principio del archivo</u> .
w	Apertura para sólo escritura, <u>coloca el puntero al principio del archivo</u> y trunca el archivo a longitud cero, por lo que borrará todo su contenido. <u>Si el archivo no existe se intenta crear</u> .
w+	Apertura para lectura y escritura, <u>coloca el puntero al principio del archivo</u> y trunca el archivo a longitud cero, por lo que borrará todo su contenido. <u>Si el archivo no existe se intenta crear</u> .
a	Apertura para sólo escritura, <u>coloca el puntero al final del archivo</u> . <u>Si el archivo no existe se intenta crear</u> .
a+	Apertura para lectura y escritura, <u>coloca el puntero al final del archivo</u> (para <u>escritura</u>) para <u>lectura</u> lo coloca al principio. <u>Si el archivo no existe se intenta crear</u> .
x	Creación y apertura para sólo escritura, <u>coloca el puntero al principio del archivo</u> . Si el archivo ya existe, la llamada a fopen() fallará devolviendo FALSE y generando un error de nivel E_WARNING . Si el archivo no existe se intenta crear.
x+	Creación y apertura para lectura y escritura, <u>coloca el puntero al principio del archivo</u> . Si el archivo ya existe, la llamada a fopen() fallará devolviendo FALSE y generando un error de nivel E_WARNING . Si el archivo no existe se intenta crear.

Cierre de un fichero:

fclose():

Cerrar un fichero es tan fácil como abrirlo. Para ello utilizamos la función **fclose()** que toma como parámetro el puntero al fichero que queremos cerrar.

```
fclose($fichero);
```

Ejemplo de creación de un fichero:

Cuando abrimos un fichero también se puede especificar **b** de binario o **t** de texto en combinación con los diferentes modos de apertura. Solo en sistemas Windows se diferencia entre archivos binarios y de texto, así que si se quiere lograr la mayor portabilidad del código sería más eficiente abrir los archivos en modo binario.

fwrite() y fputs(): (ejemplo de fichero que no existe y se crea por primera vez):

PHP cuenta con una serie de métodos para escribir en un fichero abierto para escritura.

Podemos utilizar la función **fwrite()** o **fputs()**. Estas dos funciones funcionan de la misma manera, ya que **fputs()** es un alias de **fwrite()**, se les tiene que pasar 2 parámetros:

- el apuntador del fichero
- la cadena que quiere escribirse

El tercer parámetro es opcional e indica la longitud en bytes que se va a escribir.

La función devuelve el número de bytes escritos o **FALSE** si hubo algún error.

Ejemplo: una vez hayas ejecutado este script -> comprueba que en dicha carpeta se ha creado un fichero llamado **'fichero1.txt'** (comprueba su contenido):

Crea un fichero llamado prueba.php y prueba este código:

```
<?php
header('Content-Type: text/html; charset=UTF-8');

// abrimos el fichero en modo escritura
$fichero=@fopen('fichero1.txt', 'wb');
$cadena="Hola esto es un ejemplo de escritura en ficheros";
// los caracteres "\r\n" se utilizan para especificar fin de línea y línea nueva
$cadena=$cadena."\r\n";
$cadena=$cadena."Esto es otra frase";
$nescritos=fwrite($fichero, $cadena, strlen($cadena));
fclose($fichero);
echo "se han escrito en el fichero: ".$nescritos." caracteres";

?>
```

Nota: si hubiésemos puesto en el código lo siguiente, sólo se hubiesen escrito 20 caracteres

```
$nescritos=fwrite($fichero, $cadena, 20);
```

Lectura de un fichero:

PHP también cuenta con una serie de funciones para leer ficheros.

fread():

esta función toma 2 parámetros:

- el apuntador del fichero
- el número de bytes que queremos leer.

Esta función es poco manejable cuando queremos buscar en un fichero, así que PHP cuenta con funciones más precisas para leer líneas del fichero apoyándose en otras funciones relacionadas.

rewind():

esta función toma como parámetro el apuntador al fichero y sitúa el puntero del fichero al principio.

```
rewind($fichero);
```

Crea un fichero llamado **prueba.php** y prueba este código:

```
<?php
    header('Content-Type: text/html; charset=UTF-8');

    // abrimos el fichero en modo lectura
    $fichero = @fopen('fichero1.txt', 'rb', true );

    // leemos de golpe todo el fichero
    $contenido = fread($fichero, filesize('fichero1.txt'));
    echo "el contenido del fichero es: ";
    echo $contenido;
    echo "<br><br><br>";

    //situamos el puntero del fichero al principio
    rewind($fichero);

    // ahora leemos 15 caracteres
    $contenido = fread($fichero, 15);
    echo "el contenido de los 15 caracteres son: ";
    echo $contenido;

    fclose($fichero);
?>
```

feof():

esta función toma como parámetro el apuntador al fichero y devuelve TRUE si el puntero se encuentra al final del archivo.

```
feof($fichero);
```

fgets():

esta función se utiliza para leer línea a línea un fichero. Toma como parámetro:

- el apuntador del fichero
- opcionalmente una longitud (no lo utilizaremos este parámetro opcional)

y leerá una nueva línea o hasta que alcance la longitud establecida en el segundo parámetro opcional.

Utilizar esta función es la forma ideal de ir analizando el fichero línea a línea para encontrar la información que estamos buscando.

Crea un fichero llamado prueba.php y prueba este código:

```
<?php
header('Content-Type: text/html; charset=UTF-8');

// abrimos el fichero en modo lectura
$fichero = @fopen('fichero1.txt', 'rb', true );

if (!$fichero)
{
    echo 'No se puede abrir el fichero.';
}
else
{
    $num=1;
    // leemos líneas hasta que lleguemos al final del fichero
    while (!feof($fichero))
    {
        $linea = fgets ($fichero) ;
        echo 'Linea '.$num.': '.$linea.'<br>';
        $num++;
    }
}
?>
```

fgetc():

esta función es muy parecida a la anterior pero devuelve un solo carácter.

Crea un fichero llamado prueba.php y prueba este código:

(ejecuta el ejemplo anterior cambiando fgets por fgetc)

Las líneas vacías serán las que corresponden al carácter fin-de-línea.

Es un carácter de control que no se visualiza pero estar está.

readfile() o **file_get_contents()**: (no nos interesa ahora)

estas dos funciones sirven para leer **todo el fichero de una vez**.

La función **readfile()** toma como parámetro la ruta absoluta del fichero y muestra todo su contenido en pantalla sin la necesidad de haber abierto el fichero para lectura.

La función **file_get_contents()** funciona de manera similar pero no imprime el contenido en la salida estándar sino que lo guarda en una variable de tipo cadena.

Prueba las **2 funciones** con este ejemplo:

Crea un fichero llamado prueba.php y prueba este código:

```
<?php
header('Content-Type: text/html; charset=UTF-8');

// Prueba-1
readfile('fichero1.txt');

echo "<br><br>";

// Prueba-2
$archivo = file_get_contents('fichero1.txt', true);
echo $archivo;
?>
```

Scripts de ejemplo:

Ejecuta estos 2 fichero y mira el código:

- **fichero1.php**: **creación** de un fichero de texto
- **fichero2.php**: **lectura** de un fichero de texto