

MANEJO BÁSICO DE
ARCHIVOS CON PHP.
FUNCIONES FOPEN, FGETS,
FPUTS, FCLOSE Y FEOF.
EJEMPLOS

INTRODUCCIÓN AL MANEJO DE ARCHIVOS CON PHP.

Cuando necesitamos crear, guardar, leer o escribir archivos en la creación de sitios web, el manejo de éstos se convierte en una prioridad. Para ello, vamos a ver una serie de funciones y algunos ejemplos para familiarizarnos con el manejo de archivos a nivel básico en php.



FUNCIÓN FOPEN

La función fopen() sirve para abrir ficheros (archivos). Su sintaxis general:

```
<?php //Ejemplo
    $fp = fopen(fichero, modoDeApertura);
?>
```

Donde \$fp es el descriptor o identificador del fichero abierto que necesitaremos más tarde.

fichero puede ser un archivo de texto, un archivo con extensión .php, o incluso la URL de una página web (por ejemplo <http://www.paginaweb.com/>). “Abrir” una web con fopen() puede ser muy útil cuando queremos comprobar si la web existe, o si está activa.

Si **fichero** empieza con "http://", se abre una conexión hacia la web especificada. Siempre debe ponérsele una barra (/) al final.

Si **fichero** empieza con "ftp://", se abre una conexión al servidor especificado.

Si **fichero** no empieza con ninguna de las cosas anteriormente dichas, se abre una conexión "directa" con el archivo especificado. Si no existe el archivo o dirección especificadas, se devuelve un error.

Hay distintos modos de apertura de archivos, vamos a ver los más habituales.

A la hora de abrir un archivo, hay que saber de qué forma queremos abrirlo. Podemos abrirlo para escritura y lectura, pero de distintas formas:

Modo	Observaciones
r	Abre el archivo sólo para lectura. La lectura comienza al inicio del archivo.
r+	Abre el archivo para lectura y escritura. La lectura o escritura comienza al inicio del archivo.
w	Abre el archivo sólo para escritura. La escritura comienza al inicio del archivo, y vacía el archivo. Si el archivo no existe, intenta crearlo.
w+	Abre el archivo para escritura y lectura. La lectura o escritura comienza al inicio del archivo, y vacía el archivo. Si el archivo no existe, intenta crearlo.
a	Abre el archivo para sólo escritura. La escritura comenzará al final del archivo. Si el fichero no existe se intenta crear.
a+	Abre el archivo para lectura y escritura. La lectura o escritura comenzará al final del fichero. Si el fichero no existe se intenta crear.

Ejemplos

```
<?php //Ejemplo
    $fp = fopen("/apr2/fichero.txt", "r");
    $fp = fopen("/apr2/fichero2.txt", "w");
    $fp = fopen("http://www.maralboran.com/texto.txt", "a+");
    $fp = fopen("ftp://ftp.elmundo.es/fichero.txt", "w");
?>
```

Ahora bien, con esta simple instrucción sólo tendremos el archivo abierto para leer, escribir o leer y escribir. Pero ahora tendremos que hacer uso de las otras instrucciones para escribir o recuperar los datos que estimemos oportunos.

FUNCIÓN FCLOSE

La función fclose(identificadorDelFichero) cierra un archivo abierto.

Esta función debe ser utilizada después de abrir y manipular el archivo como veremos en los siguientes ejemplos.

FUNCIONES FGETS Y FEOF

La función fgets() recupera el contenido de una línea de un archivo. Su sintaxis general es:

```
<?php //Ejemplo
    fgets(descriptorDelFichero);
?>
```

Ejemplo

```
<?php //Ejemplo
    // Leemos la primera línea de fichero.txt
    // fichero.txt tienen que estar en la misma carpeta que el fichero php
    // fichero.txt es un archivo de texto normal creado con notepad, por ejemplo.
    $fp = fopen("fichero.txt", "r");
    $linea = fgets($fp);
    fclose($fp);
?>
```

Ahora bien, con esto sólo leeremos la primera línea del fichero de texto, si quisiéramos leer línea a línea hasta el final necesitaríamos el uso de un bucle while, por ejemplo.

```
<?php //Ejemplo
    // Iremos leyendo línea a línea del fichero.txt hasta llegar al fin (feof($fp))
    // fichero.txt tienen que estar en la misma carpeta que el fichero php
    // fichero.txt es un archivo de texto normal creado con notepad, por ejemplo.
    $fp = fopen("fichero.txt", "r");
    while(!feof($fp)) {    $linea = fgets($fp);
                           echo $linea . "<br />";           }
    fclose($fp);
?>
```

Esto nos mostrará el contenido del fichero de texto. Comprobarás que hemos incluido una nueva función denominada feof (que viene significando algo así como file end of file o “marca de final de archivo”). La sintaxis general para esta función es: feof (identificadorDelArchivo)

La función feof nos devuelve true cuando hemos llegado al final de archivo y false si no lo hemos alcanzado todavía. La condición while (!feof(\$fp)) podríamos haberla escrito también de la siguiente manera: while (feof(\$fp)==false) . Esta sintaxis es equivalente, puedes utilizar la que prefieras, pero debes acostumbrarte a entender ambas formas de escritura.

Finalmente, fíjate como una vez hemos terminado de operar con el fichero escribimos la instrucción fclose(identificadorDelFichero) para cerrar la conexión, cosa que debemos hacer siempre.

FUNCIÓN FPUTS

La función fputs() escribe una línea en un archivo. Su sintaxis general es:

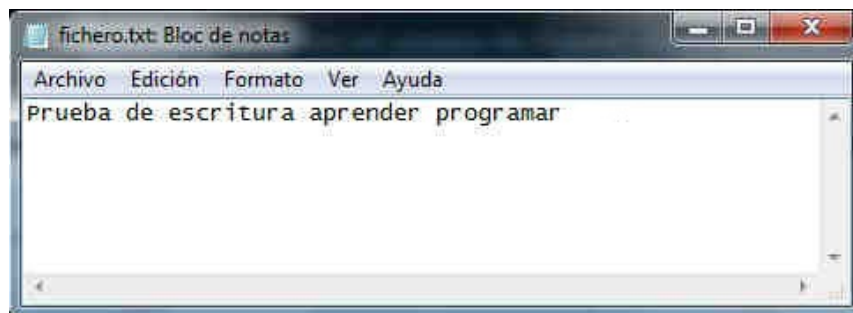
```
<?php //Ejemplo
      fputs(descriptorDelFichero, cadena);
?>
```

Ejemplo

```
<?php //Ejemplo
      // Escribimos una primera línea en fichero.txt
      // fichero.txt tienen que estar en la misma carpeta que el fichero php
      $fp = fopen("fichero.txt", "w");
      fputs($fp, "Prueba de escritura aprender programar");
      fclose($fp);
?>
```

Fíjate que al realizar la apertura del fichero en modo w, si el fichero no existe, será creado. Fíjate también como una vez hemos de terminado de operar con el fichero escribimos la instrucción fclose(identificadorDelFichero) para cerrar la conexión, cosa que debemos hacer siempre.

Si ahora abrimos el fichero con algún editor de textos como Notepad, veremos lo siguiente:



LECTURA COMPLETA DE
UN ARCHIVO DE TEXTO.
FUNCIÓN `file_get_contents`
y `nl2br`. EJERCICIOS
RESUELTOS Y EJEMPLOS.

LECTURA ARCHIVOS DE TEXTO CON PHP USANDO LA FUNCIÓN FILE_GET_CONTENTS.

Además de con la función fopen podemos realizar la lectura de ficheros de texto usando la función file_get_contents de PHP. La lectura de ficheros usando esta función es relativamente sencilla. Como podremos observar en los siguientes ejemplos y ejercicios, con dicha función podremos leer archivos de texto sin necesidad de ir recorriéndolos línea a línea para obtener su contenido.



FUNCIÓN FILE_GET_CONTENTS.

La sintaxis general a emplear es: file_get_contents(\$ruta_del_fichero);

Ejemplo.

```
<?php //Ejemplo
    $texto = file_get_contents("archivo.txt");
    echo $texto;
?>
```



Como podemos observar, el fichero es mostrado sin saltos de línea. Para convertir esos caracteres de salto de línea a etiquetas
 que den lugar a que se visualice el salto de línea en HTML y poder así, visualizarlos en un navegador, necesitamos usar la función nl2br.

Función nl2br

La función nl2br convierte los saltos de línea a etiquetas
 con lo que podremos ver esos saltos de línea en cualquier navegador. Para el ejemplo anterior podemos incluirla y así poder ver los saltos de línea correctamente. Escribe este código en un archivo como ejemplo1.php y visualiza el resultado.

```
<?php //Ejemplo
    $texto = file_get_contents("archivo.txt");
    $texto = nl2br($texto);
    echo $texto;
?>
```

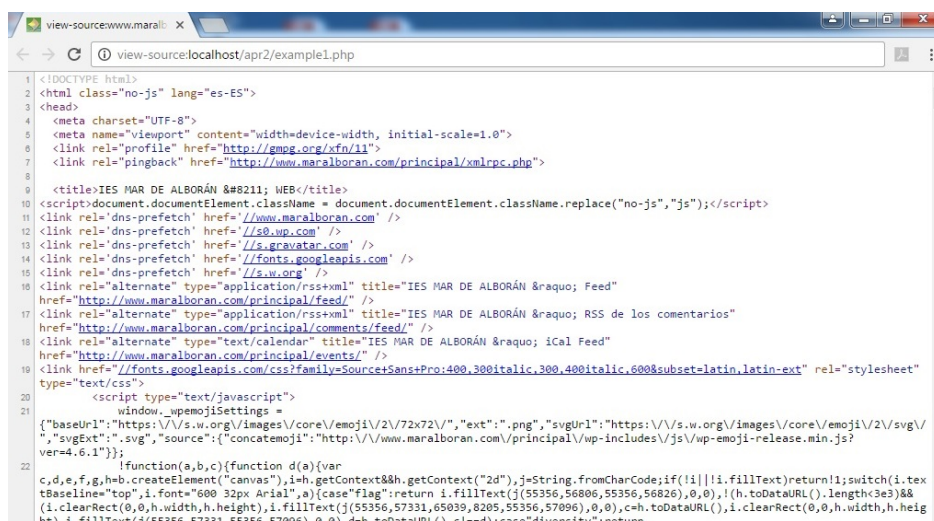


OTROS USOS DE FILE_GET_CONTENTS

Como dato adicional, la función `file_get_contents`, puede obtener el contenido de una página web en forma de código HTML, es decir, su código fuente.

Ejemplo.

```
<?php //Ejemplo
    $texto = file_get_contents("http://www.maralboran.com");
    echo $texto;
?>
```



Haciendo un uso más avanzado de esta función podrías hacer cosas como:

- Introducir un fragmento de código de otra página web dentro de tu propia página web. Para ello extraerías el contenido usando esta función y seleccionarías aquella parte que quieres mostrar dentro de tu página web.
- Hacer búsquedas de forma similar a como las hacen buscadores tipo Google o Yahoo. Una vez capturada la información de la página, tendrías que analizarla y almacenarla en bases de datos para darle el tratamiento que quisieras.
- Copiar el contenido de una web en otra web.
- Diversas aplicaciones adicionales.

EJERCICIO

Crea un fichero que lleve por nombre `miArchivoDePrueba.txt` e introduce en él el siguiente texto:

Una base de datos es un sistema informático a modo de almacén.

En este almacén se guardan grandes volúmenes de información.

Por ejemplo, imaginemos que somos una compañía telefónica y deseamos tener almacenados los datos personales y los números de teléfono de todos nuestros clientes, que posiblemente sean millones de personas.

A continuación crea un archivo denominado `ejemplo2.php` donde usando la función `file_get_content` recuperes en una variable el texto que se encuentra en ese fichero y lo muestres en el navegador.

LECTURA Y ESCRITURA
(fwrite) DE ARCHIVOS DE
TEXTO CON PHP. MODOS
DE ACCESO. EJERCICIOS
RESUELTOS Y EJEMPLOS.

LECTURA Y ESCRITURA DE ARCHIVOS DE TEXTO CON PHP.

En esta ocasión veremos unos ejemplos prácticos de lectura y escritura de ficheros de texto usando PHP. El primer lugar, debemos crear un archivo de texto (con cualquier editor, notepad por ejemplo, aunque también puede valer el bloc de notas de Windows) con algunas líneas de texto.



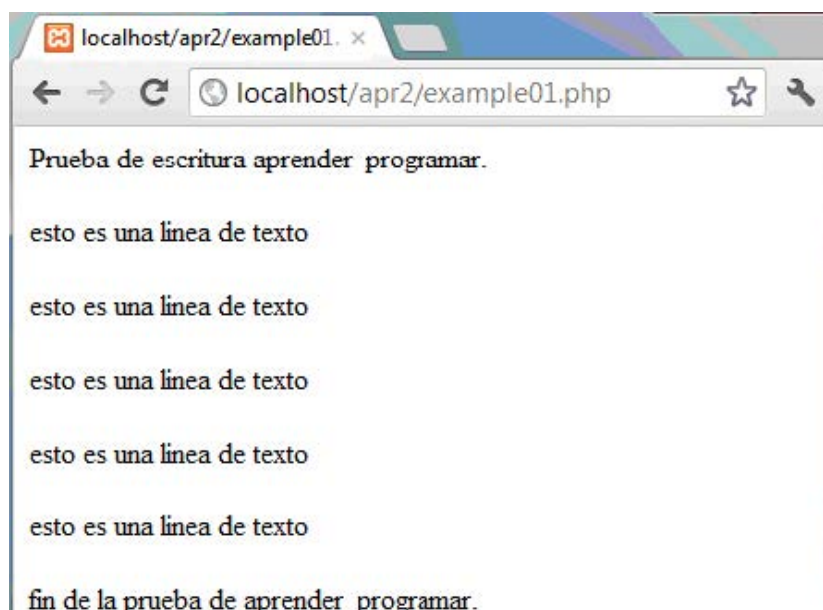
LECTURA DE FICHEROS

Para leer un archivo de texto línea a línea en PHP vamos a utilizar la función `fgets()`. Por ejemplo si queremos leer línea por línea el archivo 'archivo.txt' que se encuentra ubicado en la misma carpeta que el archivo 'leer.php', tenemos que escribir en el archivo PHP:

```
<?php //Ejemplo
    $file = fopen("archivo.txt", "r");
    while(!feof($file)) {
        echo fgets($file). "<br />";
    }
    fclose($file);
?>
```

Como podemos observar, lo primero que debemos hacer es abrir el fichero 'archivo.txt' y luego ir leyendo línea a línea hasta llegar al final. Recordemos que "r" es el modo de apertura que abre el archivo sólo para lectura. La lectura comienza al inicio del archivo. Después cerramos el fichero.

Con esto conseguiremos mostrar por pantalla el contenido del fichero 'archivo.txt' tal y como podemos ver en la siguiente imagen de ejemplo.



El contenido del fichero de texto "archivo.txt" se supone que era el siguiente:

Prueba de escritura de aprender programar.

esto es una linea de texto

esto es una linea de texto

esto es una linea de texto

esto es una linea de texto

esto es una linea de texto

fin de la prueba de aprender programar.

Lo que hemos hecho con nuestro pequeño programa es ir extrayendo líneas mientras no se detecta el final del fichero (feof). La condición `while(!feof($file))` también podría haberse escrito de esta otra manera: `while(feof($file)==false)`, con lo que obtendríamos el mismo resultado.

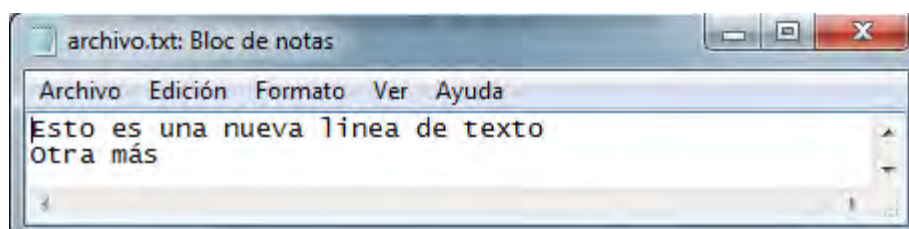
ESCRITURA DE FICHEROS

Para escribir un archivo de texto en PHP podemos utilizar la función `fwrite()`. Por ejemplo si queremos escribir línea por línea un archivo 'archivo.txt' que se encuentre ubicado en la misma carpeta que el archivo 'escribir.php', tenemos que escribir en el archivo PHP:

```
<?php //Ejemplo,  archivo escribir.php
    $file = fopen("archivo.txt", "w");
    fwrite($file, "Esto es una nueva linea de texto" . PHP_EOL);
    fwrite($file, "Otra más" . PHP_EOL);
    fclose($file);
?>
```

Nota: `PHP_EOL` (end of line) introduce un salto de línea en PHP. Mediante la concatenación con un punto forzamos el salto de línea después del texto introducido.

Una vez escribimos en nuestro navegador la ruta del fichero `escribir.php` se ejecuta el código asociado. Después de la ejecución, podemos ver el fichero creado con un editor de texto cualquiera y observaremos el siguiente contenido:



Tener en cuenta que el modo de apertura de archivo que hemos usado es w. Si recordamos el significado de este modo:

w: Abre el archivo sólo para escritura. La escritura comienza al inicio del archivo, y elimina el contenido previo del archivo. Si el archivo no existe, intenta crearlo.

AÑADIR CONTENIDO A UN FICHERO DE TEXTO

A veces no queremos reemplazar el contenido que exista en un fichero de texto, sino añadir un contenido adicional al final de lo que ya exista en el mismo. Vamos a ver cómo podemos hacerlo.

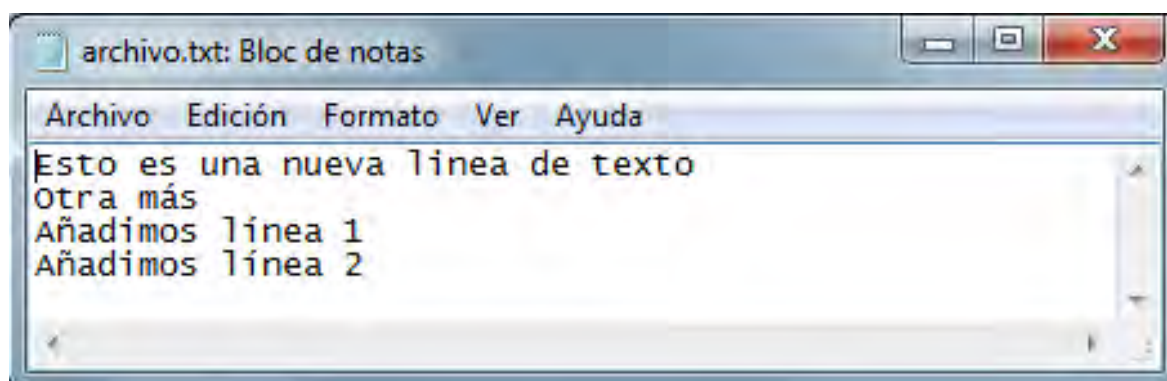
A modo de ejemplo añadiremos unas líneas de texto al final del fichero anteriormente escrito:

```
<?php //Ejemplo
    $file = fopen("archivo.txt", "a");
    fwrite($file, "Añadimos línea 1" . PHP_EOL);
    fwrite($file, "Añadimos línea 2" . PHP_EOL);
    fclose($file);
?>
```

Tener en cuenta que el modo de apertura de archivo que hemos usado es a. Si recordamos el significado de este modo:

a: Abre el archivo para sólo escritura. La escritura comenzará al final del archivo sin eliminar el contenido previo existente. Si el fichero no existe se intenta crear.

Ahora podemos ver el fichero creado con un editor de texto cualquiera y observaremos el siguiente contenido (no se ha reemplazado el contenido existente, sino que se ha ampliado):



Observamos que hemos añadido las dos nuevas líneas al final del archivo.

EJERCICIO

Crea el siguiente código PHP:

a) Una función (tipo procedimiento, no hay valor devuelto) denominada `escribirNumerosMod` que reciba dos parámetros: un array de valores enteros y una cadena de texto que puede ser "sobreescribir" ó "ampliar". La función debe proceder a: escribir cada uno de los números que forman el contenido del array en una línea de un archivo `datosEjercicio.txt` usando el modo de operación que se indique con el otro parámetro. Si el archivo no existe, debe crearlo.

Ejemplo: El array que se pasa es `$numeros = array(5, 9, 3, 22);` y la invocación que se utiliza es `escribirNumerosMod($numeros, "sobreescribir");` En este caso, se debe eliminar el contenido que existiera previamente en el archivo y escribir en él 4 líneas, cada una de las cuales contendrá los números 5, 9, 3 y 22.

b) Una función (tipo procedimiento, no hay valor devuelto) denominada `leerContenidoFichero` que reciba como parámetro la ruta del fichero y muestre por pantalla el contenido de cada una de las líneas del fichero.

c) Crea código php donde a través de la función `escribirNumerosMod` escribas en el fichero los números 2, 8, 14. Luego, mediante la función `leerContenidoFichero` muestra el contenido del fichero. Ahora con la función `escribirNumerosMod` amplía el contenido del fichero y añádele los números 33, 11 y 16. Muestra nuevamente el contenido del fichero por pantalla. Finalmente, escribe el fichero pasándole un array con los número 4, 99, 12 y parámetro <<sobreescribir>> para eliminar los datos que existieran previamente. Muestra el contenido del fichero por pantalla y un mensaje de despedida.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.