

FICHEROS

En php existen funciones propias del lenguaje que nos permiten acceder al fichero, tanto para escribir en él como para leer, aquí se explicaran algunas de estas

Paso 1: abrir un fichero

Para esto utilizaremos la sentencia `@fopen`, que nos devolverá el puntero hacia este fichero, hay que tener en cuenta que es buena idea, agregarlo a una variable para operar con él más tarde, en esta sentencia recurriremos dos parámetros, la ruta y el tipo de acceso

```
$fichero = @fopen("index.txt" , 'r');  
if (!$fichero) {  
    die('Problema con el fichero');  
}  
while(!feof($fichero)){
```

En caso de que no exista devolverá false.

Como podemos ver hay diferentes parámetros para los ficheros, que se muestran en la siguiente tabla

Una lista de los modos posibles de <code>fopen()</code> usando <code>mode</code>	
mode	Descripción
'r'	Apertura para sólo lectura; coloca el puntero al fichero al principio del fichero.
'r+'	Apertura para lectura y escritura; coloca el puntero al fichero al principio del fichero.
'w'	Apertura para sólo escritura; coloca el puntero al fichero al principio del fichero y trunca el fichero a longitud cero. Si el fichero no existe se intenta crear.
'w+'	Apertura para lectura y escritura; coloca el puntero al fichero al principio del fichero y trunca el fichero a longitud cero. Si el fichero no existe se intenta crear.
'a'	Apertura para sólo escritura; coloca el puntero del fichero al final del mismo. Si el fichero no existe, se intenta crear. En este modo, fseek() solamente afecta a la posición de lectura; las lecturas siempre son pospuestas.
'a+'	Apertura para lectura y escritura; coloca el puntero del fichero al final del mismo. Si el fichero no existe, se intenta crear. En este modo, fseek() no tiene efecto, las escrituras siempre son pospuestas.
'x'	Creación y apertura para sólo escritura; coloca el puntero del fichero al principio del mismo. Si el fichero ya existe, la llamada a fopen() fallará devolviendo false y generando un error de nivel E_WARNING . Si el fichero no existe se intenta crear. Esto es equivalente a especificar las banderas <code>O_EXCL O_CREAT</code> para la llamada al sistema de <code>open(2)</code> subyacente.
'x+'	Creación y apertura para lectura y escritura; de otro modo tiene el mismo comportamiento que 'x'.
'c'	Abrir el fichero para sólo escritura. Si el fichero no existe, se crea. Si existe no es truncado (a diferencia de 'w'), ni la llamada a esta función falla (como en el caso con 'x'). El puntero al fichero se posiciona en el principio del fichero. Esto puede ser útil si se desea obtener un bloqueo asistido (véase flock()) antes de intentar modificar el fichero, ya que al usar 'w' se podría truncar el fichero antes de haber obtenido el bloqueo (si se desea truncar el fichero, se puede usar ftruncate() después de solicitar el bloqueo).
'c+'	Abrir el fichero para lectura y escritura; de otro modo tiene el mismo comportamiento que 'c'.
'e'	Establecer la bandera 'close-on-exec' en el descriptor de fichero abierto. Disponible solamente en PHP compilado en sistemas que se ajustan a POSIX.1-2008.

Con estos podremos realizar diferentes funciones
para cerrar usaremos `fclose` junto con el identificado

Manual ficheros PHP
Diego Casero Martín

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