

UT4 Desarrollo de aplicaciones web con PHP

Ficheros y directorios

Ficheros

Librería Fisesystem

La función **fopen()** permite abrir archivos que se encuentren en nuestro servidor o una URL.

Pasamos 2 parámetros: nombre del archivo y el modo en el que se abrirá.

La función devuelve un **puntero** al fichero si es satisfactoria o **FALSE/cero** si no lo es.

Una vez termino, cierro el fichero con **fclose()**

```
$nombreArchivo = 'archivo.txt';
```

```
$archivo = fopen($nombreArchivo, 'r');
```

```
if ($archivo) {  
    echo "El archivo se abrió correctamente."  
    fclose($archivo); // No olvides cerrar  
} else {  
    echo "No se pudo abrir '$nombreArchivo'.  
}
```

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.

Ficheros

Lectura carácter

Se abre un fichero en modo lectura y se lee carácter a carácter usando **fgetc()**, que devuelve el siguiente carácter a partir del indicador de posición.

Para detectar el final del fichero se usa la función **feof()**, que devuelve TRUE cuando ya se ha llegado al final.

```
$fp = fopen("miarchivo.txt", "r");  
if ($fp === FALSE) {  
    echo "No existe el fichero o no se pudo leer<br>";  
}else{  
    while(!feof($fp) ){  
        $car=fgetc($fp);  
        echo $car;  
    }  
}  
fclose($fp);
```

Ficheros

Lectura con formato

Para leer un fichero que sigue un formato se puede usar **fscanf()**: lee una línea y le aplica un formato.

Una de las formas de usarla es pasando variables para que almacene en ellas los valores leídos.

En este caso devuelve el número de valores leídos correctamente.

```
$valores = fscanf($fichero, $formato, $var1, ...);
```

Supongamos que tenemos un fichero de texto que representa una matriz de 3 filas y 3 columnas.

```
23 234 611
```

```
1233 565 123
```

```
123 54 757
```

Indicamos que cada línea está formada por 3 números: "%d %d %d"

Si queremos leer dos veces un fichero **rewind()**, sitúa el indicador de posición de nuevo al principio del fichero.

```
<?php
```

```
// Ruta del archivo
```

```
$filePath = "matriz.txt";
```

```
// Asegurarse de que el archivo existe
```

```
if (file_exists(filename: $filePath)) {
```

```
    // Abrir el archivo para lectura
```

```
    $file = fopen(filename: $filePath, mode: "r");
```

```
    if ($file) {
```

```
        echo "Lectura fila por fila:";
```

```
        while (!feof(stream: $file)) {
```

```
            // Leer cada fila con fscanf
```

```
            fscanf(stream: $file, format: "%d %d %d", vars: &$num1, $num2, $num3);
```

```
            echo "<br> $num1, $num2, $num3 ";
```

```
        }
```

```
        fclose(stream: $file);
```

```
    } else {
```

```
        echo "No se pudo abrir el archivo.";
```

```
    }
```

```
} else {
```

```
    echo "Archivo no encontrado.";
```

```
}
```

Ficheros

Lectura con formato

Ejemplo muestra cómo recorrer un fichero con el formato anterior utilizando **fscanf()**.

Ficheros

Lectura completa

También son útiles **file_get_contents()**: devuelve una cadena con el contenido de un fichero.

file_put_contents(): escribe datos en un fichero.

Funcionan directamente a partir de la ruta del fichero, no hace falta llamar a fopen() y fclose().

```
<?php
$contenido=file_get_contents("fichero_ejemplo.txt");
echo "Contenido del fichero: $contenido<br>";
$res=file_put_contents("fichero_salida.txt", "Fichero creado con file put_contents");
if($res){
    echo "Fichero creado con éxito";
}else{
    echo "Error al crear el fichero";
}
```

Ficheros

Magic constants (<https://www.php.net/manual/es/language.constants.magic.php#constant.dir>)

Hay 9 **constantes mágicas** que cambian dependiendo de donde se usan (de ahí que se llamen mágicas). Todas estas constantes "mágicas" se resuelven en tiempo de compilación, a diferencia de las constantes regulares, que se resuelven en tiempo de ejecución. Algunas:

__DIR__ - ruta del directorio en el que se encuentra el archivo actual. Útil para incluir archivos y trabajar con rutas relativas

__LINE__ - línea actual del archivo en el que se utiliza

__FILE__ - ruta al archivo actual

```
// Supongamos "index.php" en la carpeta "proyecto".  
// Usamos __DIR__ para incluir "config.php" que está en la misma carpeta.  
require_once __DIR__ . '/config.php';  
// Podemos usar __DIR__ para incluir un archivo en una subcarpeta.  
require_once __DIR__ . '/biblioteca/funciones.php';  
// Imprimimos el directorio actual  
echo "Directorio actual: " . __DIR__;
```

Leer y escribir con fread y fwrite

fread(\$fichero, \$longitud): permite leer \$longitud bytes

fwrite(\$fichero, \$cadena): permite escribir \$cadena en un fichero

```
<?php
// Creamos, escribimos y cerramos un fichero

$file = "miarchivo.txt";
$texto = "Hola, esto está escrito en el fichero.";

$fp = fopen($file, "w");
// Si queremos añadir texto, debe abrirse a+
//$fp = fopen($file, "a+");

fwrite($fp, $texto);
fclose($fp);

?>
```

Lectura ->

(*******)**401crearFicheroTXT**: Escribe el resultado de la consulta de listar las personas/alumnos en un fichero listadoPersonas.txt

<- Escritura

```
<?php
// Abriendo un archivo y leyendo su contenido

$file = "miarchivo.txt";
$fp = fopen($file, "r");

// filesize() nos devuelve el tamaño del archivo
$contents = fread($fp, filesize($file));
echo $contents;

// Cerramos la conexión con el archivo
fclose($fp);

?>
```


Directorios

- Cambia el directorio actual al directorio *\$directory*.

```
bool chdir (string $directory)
```

- Obtiene el directorio actual.

```
string getcwd ( void )
```

- Devuelve un *array* con los archivos y directorios que se encuentran en *\$directory*:
SCANDIR_SORT_ASCENDING (0), SCANDIR_SORT_DESCENDING (1) o SCANDIR_SORT_NONE (sin orden):

```
scandir (string $directory, [0|1])
```

- Abre un **gestor de directorio** para ser usado en llamadas posteriores como *closedir()*, *readdir()*.

```
resource opendir (string $path [, resource $context ] )
```

- Lee una entrada desde un gestor de directorio abierto por *opendir()*.

```
string readdir ([resource $dir_handle])
```

- Cierra un gestor de directorio abierto.

```
void closedir ([resource $dir_handle ])
```

Ejemplos

```
<?php
// Directorio: /directorio/actual
echo getcwd() . "\n";
```

```
// Directorio: /directorio/actual/carpeta
chdir('carpeta');
echo getcwd() . "\n";
```

```
// Directorio: c:/xampp/htdocs
chdir('c:/xampp/htdocs');
echo getcwd() . "\n";
?>
```

```
<?php
//Array de archivos y directorios
chdir('c:/xampp/htdocs');
$directorio = "misPHP";
$archivos = scandir($directorio, 1);
print_r($archivos);
?>
```

```
<?php
// Contenido del directorio
chdir('c:/xampp/htdocs');
$directorio = "misPHP";
if ($gestor = opendir($directorio)) {
    echo "Gestor de directorio: $gestor\n<br />";
    echo "Entradas:\n<br />";
    // Iteramos sobre el directorio:
    while (false !== ($entrada = readdir($gestor))) {
        echo "$entrada\n <br />";
    }
    closedir($gestor);
}
?>
```

Directorios

Comprobar si es directorio: *is_dir()*

Y si lo que queremos es **crear un directorio**:

mkdir(ruta, modo)

modo: 0777 (todos los permisos)

En Windows estos permisos se ignoran:

mkdir(ruta);

OJO: asegúrate de que no existe el directorio antes de crearlo

Si queremos eliminar directorio: ***rmdir(ruta)***

Si queremos eliminar un fichero: ***unlink(ruta)***

(***)Crea sripts para probar estas funciones.

```
<?php
// Creamos un directorio
chdir('c:/xampp/htdocs/misPHP');
$directorio = "nuevo";
mkdir($directorio);
?>
```

pdf

Con PHP podemos **crear archivos pdf** utilizando la **librería FPDF**.

Descarga y pega el directorio fpdf dentro de htdocs (<http://www.fpdf.org/>)

Recuerda la ruta para referenciarla correctamente.

Creamos un pdf básico:

```
<?php
require('../fpdf/fpdf.php');

$pdf=new FPDF();
$pdf->AddPage();
$pdf->SetFont('Arial','B',16);
$pdf->Cell(40,10,'PDF con generado con PHP y FPDF');
//Nombre del fichero y opción de descarga
$pdf->Output('basico.pdf','D');
?>
```

- **AddPage**: crea una nueva página en el PDF.
- **SetFont**: define fuente, estilo y tamaño a utilizar.
- **Cell**: permite imprimir un texto (ancho, alto, texto...).
- **Output**: genera el archivo y le asigna un nombre. D: descarga.

(***)**402crearFicheroPDF**: mira esta documentación

<https://desarrolloweb.com/manuales/manual-fpdf.html>, y crea un script que genere un pdf con encabezado y pie que incluya una imagen, un enlace y texto de un archivo (listado de las personas o alumnos). Mejora: cada registro en una página.