

Manejo de Archivos

Dentro de las utilidades más importantes de un lenguaje de programación, está el manejo de de archivos, en programación Web y en el caso de PHP, de igual manera podemos tener muchas utilidades al poder manipular archivos, entre algunas de ellas tenemos:

- Importación de datos desde un archivo CSV.
- Exportación de datos a CSV o SQL.
- Manipular archivos XML
- Manipular archivos JavaScript
- Manipular Archivos CSS
- Manipular archivos HTML
- Mucho más....

Abrir y cerrar archivos

Para abrir un archivo, necesitamos de la función `fopen()` la cual recibe como parámetros, la ruta del archivo u el modo en el cual se abrirá el archivo, estos pueden ser:

- **'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 al fichero al final del fichero. Si el fichero no existe se intenta crear.
- **'a+'** Apertura para lectura y escritura; coloca el puntero al fichero al final del fichero. Si el fichero no existe se intenta crear.
- **'x'** Creación y apertura para sólo escritura; coloca el puntero al fichero al principio del fichero. Si el fichero ya existe, la llamada a `fopen()` fallará devolviendo `FALSE` y generando un error de nivel `E_WARNING`.
- **'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.

- **'C+'** Abrir el fichero para lectura y escritura; de otro modo tiene el mismo comportamiento que 'c'.

La función `fclose()`, nos sirve para cerrar el archivo.

Ejemplo:

```
$archivo = fopen('files/directorio.csv','r+');  
fclose($archivo);
```

Lectura de Archivos

Tenemos varias funciones para leer archivos.

- **`file_get_content()`**. Transfiere el contenido del archivo a una cadena. No es necesario previamente abrir el archivo. **Ejemplo:**

```
$cadena = file_get_contents('files/directorio.csv');  
echo $cadena;
```

- **`file()`**. Transfiere el archivo completo a un array. No es necesario previamente abrir el archivo. **Ejemplo:**

```
$arreglo = file('files/directorio.csv');  
print_r($arreglo);
```

- **fread()**. Lee el contenido de un archivo hasta el tamaño en bytes. Es más eficiente y recomendable usar esta función y para ello si se requiere abrir previamente el archivo. Ejemplo:

```
$archivo = fopen('files/directorio.csv','r');  
$tam = filesize('files/directorio.csv');  
$contenido = fread($archivo,$tam);  
echo $contenido;  
fclose($archivo);
```

Escritura de Archivos

Básicamente tenemos dos opciones para realizar escritura de archivos, recordar al abrir el archivo, este esté en modo escritura (w).

- **fwrite()**. Escribe el contenido de una cadena en un archivo, puede recibir también (opcional),k el número de bytes a escribir.

```
$archivo = fopen('files/directorio1.csv','w');  
  
fwrite($archivo,"esto es lo que escribo\n");  
fwrite($archivo,"esto es lo que escribo 2");  
  
fclose($archivo);
```

- **file_put_contents()**. Al igual que el fwrite, escribe una cadena en un archivo, sin embargo este tiene la particularidad de escribir desde el final del archivo con la bandera **FILE_APPEND**. No necesita que se abra el archivo.

```
file_put_contents('files/directorio1.csv',"Texto después del que ya tiene",FILE_APPEND | LOCK_EX);
```

Ejercicio Práctico

Una de las funcionalidades más importantes en el manejo de archivos, es poder migrar información de un lado a otro. Por ejemplo si tenemos una base de datos diferente a MySQL o un archivo de excel por ejemplo, desde estos podemos exportar los datos de una tabla u hoja de cálculo a un archivo de texto plano como es un csv (comma-separated values) y mediante la lectura del archivo con PHP podemos crear sentencias SQL para poblar nuestra base de datos.

Importar datos desde un csv y exportar a un archivo .sql.

Para importar datos desde un archivo de texto plano, podemos usar la función `file()`, y para generar el archivo .sql utilizaremos la función `file_put_contents()`.

```
<?php
```

```
$arreglo = file('files/directorio.csv');
```

```
foreach($arreglo as $fila){
```

```
    $campos = explode(',', $fila);
```

```
    $nombre = $campos[0];
```

```
    $edad = $campos[1];
```

```
    $profesion = $campos[2];
```

```
    $direccion = $campos[3];
```

```
    $fono = $campos[4];
```

```
    $cel = $campos[5];
```

```
    $email = $campos[6];
```

```
    $contenido .= "INSERT INTO contacto VALUES(NULL,
```

```
        '" . $nombre . "',
```

```
        '" . $edad . "',
```

```
        '" . $profesion . "',
```

```
        '" . $direccion . "',
```

```
        '" . $fono . "',
```

```
        '" . $cel . "',
```

```
        '" . trim($email) . "');\n";
```

```
}
```

```
if(file_put_contents('files/directorio.sql',$contenido,FILE_APPEND |  
LOCK_EX)){  
  
    $mensaje = "La importación de datos fue todo un éxito";  
  
    }else{  
  
        $mensaje = "La importación fallo";  
        die;  
    }  
echo $mensaje;  
  
?>
```

Tarea:

1.- Crear una tabla en la base de datos con los siguientes campos:

- id
- pais
- codigo

2.- Llenar la tabla con 5 registros desde el phpMyAdmin.

3.- Realizar la exportación de la tabla, desde el phpMyAdmin en formato csv.

4.- Abrir el archivo csv en Excel, borrar la columna correspondiente al ID, aumentar 5 registros más y exportar en formato csv.

6.- Con un script de php importar los datos del archivo csv y exportar a la base de datos correspondiente. Deben crear un formulario que permita subir el archivo correspondiente y una vez procesado, este debe ser borrado.