

****Guía de Manejo de Archivos CSV en PHP****

¿Qué es un archivo CSV?

CSV significa "Comma-Separated Values" (valores separados por comas). Es un formato de archivo que guarda datos en texto plano, donde cada línea representa una fila y cada valor en la línea está separado por una coma (`,`). Un archivo `usuarios.csv` podría verse así:

```
...
usuario1,password1,5
usuario2,password2,2
usuario3,password3,1
...
```

Cada línea representa un usuario y contiene tres datos: nombre de usuario, contraseña y un contador.

Funciones de PHP para Manejo de Archivos CSV

PHP tiene varias funciones que simplifican la lectura y escritura de archivos CSV.

`fopen()`

`fopen()` abre un archivo para lectura o escritura. Devuelve un "handle" (o identificador) que representa el archivo abierto. Este handle se usa con otras funciones para leer o escribir en el archivo.

```
```php
$handle = fopen("usuarios.csv", "r"); // Abre el archivo para leer ("r")
...
```

#### **\*\*Modos de apertura:\*\***

- `"r"`: Solo lectura.
- `"w"`: Escritura. Si el archivo existe, lo vacía; si no, lo crea.
- `"a"`: Escritura al final del archivo (no lo vacía).
- `"r+"`: Lectura y escritura sin vaciar el archivo.

#### **##### `fgetcsv()`**

`fgetcsv()` lee una línea del archivo CSV y la convierte en un array, donde cada elemento es un valor de la línea (separado por comas u otro delimitador).

```
```php
while (($data = fgetcsv($handle, 1000, ",")) !== false) {
    print_r($data); // Imprime el array de valores de una línea
}
...
```

- **Parámetros:**

- `\$handle`: El archivo abierto.
- `1000`: Longitud máxima de la línea.

- ``,``: Delimitador de valores (por defecto es una coma, pero puede ser cualquier otro carácter).

`fputcsv()`

`fputcsv()` escribe un array en el archivo CSV. Convierte cada elemento del array en una cadena y los separa con el delimitador especificado.

```
```php
```

```
$data = ["usuario4", "password4", "1"];
```

```
fputcsv($handle, $data); // Escribe en el archivo como: usuario4,password4,1
```

```
```
```

- ****Parámetros****:

- `$handle`: El archivo abierto.

- `$data`: El array a escribir.

- ``,``: Delimitador de valores (opcional).

`fclose()`

`fclose()` cierra un archivo previamente abierto con `fopen()`. Es importante cerrar el archivo cuando terminamos de trabajar con él para liberar recursos.

```
```php
```

```
fclose($handle);
```

```
```
```

Ejemplo Completo: Leer y Escribir en un Archivo CSV

Aquí tienes un ejemplo de cómo usar estas funciones para leer y luego escribir en un archivo CSV.

Leer el archivo `usuarios.csv` y guardar los datos en un array

```
```php
```

```
$archivo = "usuarios.csv";
```

```
$usuarios = [];
```

```
if (($handle = fopen($archivo, "r")) !== false) {
```

```
 while (($data = fgetcsv($handle, 1000, ",")) !== false) {
```

```
 $usuarios[] = $data; // Cada línea se guarda en el array $usuarios
```

```
 }
```

```
 fclose($handle);
```

```
}
```

```
print_r($usuarios); // Muestra todos los usuarios en el archivo
```

```
```
```

Modificar el archivo `usuarios.csv`

Supongamos que queremos aumentar en 1 el contador de logins del usuario `usuario1`.

```
```php
foreach ($usuarios as &$usuario) {
 if ($usuario[0] == "usuario1") {
 $usuario[2] += 1; // Incrementa el contador
 break;
 }
}

// Escribir el array actualizado en el archivo CSV
if (($handle = fopen($archivo, "w")) !== false) {
 foreach ($usuarios as $usuario) {
 fputcsv($handle, $usuario); // Guarda cada usuario de nuevo en el archivo
 }
 fclose($handle);
}
```
```

Resumen de Flujo de Trabajo en un Archivo CSV

1. **Abrir el archivo** con `fopen()`.
2. **Leer línea por línea** usando `fgetcsv()` (cuando se necesita leer).
3. **Modificar los datos en un array PHP**.
4. **Escribir el array completo en el archivo CSV** usando `fputcsv()`, si hay cambios.
5. **Cerrar el archivo** con `fclose()`.

Ejemplo Completo Usando Todas las Funciones

```
```php
<?php
$archivo = "usuarios.csv";

// Leer usuarios del archivo
$usuarios = [];
if (($handle = fopen($archivo, "r")) !== false) {
 while (($data = fgetcsv($handle, 1000, ",")) !== false) {
 $usuarios[] = $data;
 }
 fclose($handle);
}

// Supongamos que queremos incrementar el contador de "usuario1"
foreach ($usuarios as &$usuario) {
 if ($usuario[0] == "usuario1" && $usuario[1] == "password1") {
 $usuario[2] += 1; // Incrementar el contador de logins
 break;
 }
}
```

```

}

// Guardar usuarios modificados en el archivo
if (($handle = fopen($archivo, "w")) !== false) {
 foreach ($usuarios as $usuario) {
 fputcsv($handle, $usuario);
 }
 fclose($handle);
}

echo "Archivo CSV actualizado con éxito.";
?>
'''

```

Este ejemplo ilustra cómo:

- Leer todos los usuarios desde un archivo CSV.
- Modificar datos específicos (aquí, el contador de logins).
- Sobrescribir el archivo con los datos actualizados.