

PHP – Handle CSV File

Popular spreadsheet programs use the CSV file format (which stands for Comma Separated Values) to export worksheet data in plain text. Each line in the file represents one row of the worksheet, with values in each column separated by commas.

PHP's filesystem function library provides two functions – **fgetcsv()** and **fputcsv()** – respectively to read data from a CSV file into an array and put the array elements in a CSV file.

The fgetcsv() Function

The `fgetcsv()` function reads the line from the file pointer, and parses it into CSV fields.

```
fgetcsv(  
    resource $stream,  
    ?int $length = null,  
    string $separator = ",",  
    string $enclosure = "\"",  
    string $escape = "\\"  
): array|false
```

The **\$stream** parameter is a handle to the file resource, opened in **read mode**. The default separator symbol to parse the fields is comma, you can specify any other symbol if required.

The `fgetcsv()` function returns an indexed array containing the fields. If the function encounters any error, it returns **false**.

To demonstrate the use of `fgetcsv()` function, store the following text as "hello.txt" in the current working directory.

```
Name, Email, Post, Salary  
Ravishankar, ravi@gmail.com, Manager, 40000  
Kavita, kavita@hotmail.com, Assistant, 25000  
Nandkumar, nandu@example.com, Programmer, 30000
```

Example

The following PHP code reads the CSV data from this file, and returns an array. The fields in the array are then rendered in a HTML table –

```
<?php
$filename = 'hello.csv';
$data = [];

// open the file
$f = fopen($filename, 'r');

if ($f === false) {
    die('Cannot open the file ' . $filename);
}

// read each line in CSV file at a time
while (($row = fgetcsv($f)) !== false) {
    $data[] = $row;
}

// close the file
fclose($f);
echo "<table border=1>";
foreach ($data as $row) {
    echo "<tr>";
    foreach($row as $val) {
        echo "<td>$val</td>";
    }
    echo "</tr>";
}
echo "</table>";
?>
```

It will produce the following **output** –

Name	Email	Post	Salary
Ravishankar	ravi@gmail.com	Manager	40000
Kavita	kavita@hotmail.com	Assistant	25000
Nandkumar	nandu@example.com	Programmer	30000

The fputcsv() Function

The fputcsv() function puts an indexed array with its elements separated by commas, at the current file pointer position of a CSV file.

```
fputcsv(  
    resource $stream,  
    array $fields,  
    string $separator = ",",  
    string $enclosure = "\"",  
    string $escape = "\\",  
    string $eol = "\n"  
): int|false
```

The target file must be opened in write mode. The second mandatory parameter is an array consisting of comma separated fields. As in case of `fgetcsv()` function, the default separator is comma.

Example

In the following code, a two dimensional array of comma separated values is written into a CSV file.

```
<?php  
$data = [  
    ["Name", "Email", "Post", "Salary"],  
    ["Ravishankar", "ravi@gmail.com", "Manager", "40000"],  
    ["Kavita", "kavita@hotmail.com", "Assistant", "25000"],  
    ["Nandkumar", "nandu@example.com", "Programmer", "30000"],  
];  
$filename = 'employee.csv';  
  
// open csv file for writing  
$f = fopen($filename, 'w');  
  
if ($f === false) {  
    die('Error opening the file ' . $filename);  
}  
  
// write each row at a time to a file  
foreach ($data as $row) {  
    fputcsv($f, $row);  
}  
  
// close the file  
fclose($f);  
?>
```

The "employee.csv" file should be created in the current working directory, after the above program is executed.