

Funktion	Beispiel	Bedeutung
rewind()	rewind(\$fp);	Setzt den Dateizeiger auf den Anfang einer Datei.
unlink()	unlink('filename');	Löscht eine Datei.
require()	require(\$zaehldatei);	Fügt eine Datei in das PHP-Skript ein.

**Beispiel:**

Um die Besucher einer Homepage zu zählen, wird ein Zähler programmiert, der den Zählerstand in einer Textdatei counter.txt speichert und am Bildschirm ausgibt.

```
<html>
<head>
  <title>PHP-Zähler</title>
</head>
<body bgcolor="#87ceff">
  <?php
    $zaehldatei="counter.txt";
    if(!file_exists($zaehldatei)) {
      $fp=fopen($zaehldatei,"w+");
      fwrite($fp,"0");
      fclose($fp);
    }
    $fp=fopen($zaehldatei,"r+");
    $zaehler=(int)fgets($fp,12);
    $zaehler++;
    rewind($fp);
    fwrite($fp,$zaehler);
    fclose($fp);

  ?>
  <div align="Center">
    <font size="+2">Sie sind der
      <? require($zaehldatei); ?>
      . Besucher!!</font></div>
</body>
</html>
```

In der Datei counter.txt wird nur der Zählerstand als Ganzzahl abgelegt. Der Name dieser Datei wird in einer Variablen festgelegt.

Wenn diese Datei (noch) nicht existiert, wird die if-Anweisung durchlaufen. Es wird diese Datei erzeugt und im Modus w+ geöffnet. Diese Betriebsart der Datei legt fest, dass sie zum Lesen und Schreiben geöffnet wird und – falls sie nicht existiert – zunächst erstellt wird. Mit der Funktion fwrite() wird die Zahl 0 in die Datei geschrieben. fclose() schließt die Datei wieder.

Die Funktion fopen() öffnet die nun sicher existierende Datei im Modus r+ zum Lesen und Schreiben. Mit fgets() wird die maximal 12 Zeichen lange Zahl als String ausgelesen und mit int() in eine Ganzzahl umgewandelt. Sie wird in der Variablen \$zaehler zwischengespeichert und mit \$zaehler++ um Eins erhöht. Die Funktion rewind() setzt den internen Zeiger auf den Anfang der Datei. Dorthin wird nun die um Eins erhöhte Zahl mit fwrite() geschrieben und die Datei wieder geschlossen.

Die Ausgabe erfolgt im HTML-Format, wobei der Zählerstand durch Einbinden der Datei mittels der Funktion require(\$zaehlerstand) angezeigt wird.



Dateien können mit der Funktion fopen() in verschiedenen Modi geöffnet werden. Die Betriebsart a öffnet die Datei und setzt den internen Dateizeiger auf das Ende der Datei. Dort sind jedoch nur Schreibvorgänge zugelassen. In der Betriebsart a+ ist diese Einschränkung nicht vorhanden, es kann auch gelesen werden. In beiden Betriebsarten wird die gewählte Datei erstellt, falls sie noch nicht existiert.

Hier eine Übersicht über die möglichen Betriebsarten für Dateioperationen:

Betriebsart	Erläuterung
"r"	Öffnet eine Datei nur zum Lesen.
"r+"	Öffnet eine Datei zum Lesen und zum Schreiben.
"w"	Öffnet eine Datei nur zum Schreiben.
"w+"	Öffnet eine Datei zum Lesen und Schreiben. Der Inhalt wird zunächst gelöscht. Falls die Datei nicht existiert, wird sie erstellt.
"a"	Öffnet die Datei nur zum Schreiben und setzt den Dateizeiger auf das Ende der Datei. Falls die Datei nicht existiert, wird sie erstellt.
"a+"	Öffnet die Datei zum Lesen und Schreiben und setzt den Dateizeiger auf das Ende der Datei.

**8.3.8 Zugriffsrechte auf Dateien**

Eine wichtige Information bei der Arbeit mit Dateien, speziell mit Datenbanken, sind die Zugriffsrechte. Sie informieren über die Berechtigungen verschiedener Benutzer bezüglich dieser Dateien.

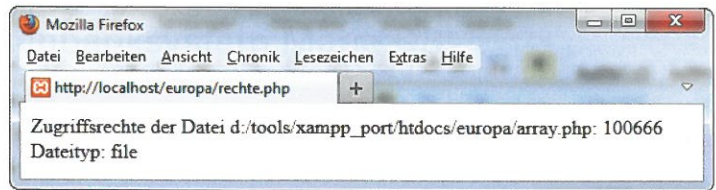
PHP bietet für die Verwaltung von Dateien und Zugriffsrechten geeignete Funktionen an:

Funktion	Beschreibung
filegroup();	Gibt die Gruppe des Dateibesitzers zurück.
fileowner();	Ermittelt den Besitzer der Datei.
fileperms();	Ermittelt die Zugriffsrechte der Datei.
filesize();	Ermittelt die Größe der Datei.
filetype();	Gibt den Dateityp zurück.
decoct();	Wandelt eine Dezimalzahl in eine Oktalzahl um.

**Beispiel:**

Mit einem PHP-Skript sollen die Zugriffsberechtigungen und der Dateityp, z. B. für die Datei D:\tools\xampp\_port\htdocs\europa\array.php, ermittelt werden.

```
<?php
  $datei = "d:/tools/xampp_port/htdocs/europa/array.php";
  $erg=fileperms($datei);
  echo "Zugriffsrechte der Datei $datei: ", decoct($erg),
  "<br>";
  echo "Dateityp: ", filetype($datei);
?>
```



Nach der Festlegung des Pfades der Datei in einer Variablen ermittelt die Funktion fileperms() die Zugriffsrechte auf diese Datei. Da das Ergebnis in dezimaler Schreibweise vorliegt, aber erst in oktaler Schreibweise richtig interpretiert werden kann, muss es mithilfe der Funktion decoct() umgewandelt werden. Die ersten drei Ziffern, hier 100, informieren über den Dateityp. Die Funktion filetype() gibt diesen Dateityp in einer Textvariablen als file aus.