



# PHP

## Lesen und Schreiben von Dateien



- Dies macht häufig Sinn, wenn sich eine komplexe Datenbank-Anbindung nicht lohnt.
- In unserem Beispiel verwenden wir eine Textdatei als Protokoll für alle Anmeldeversuche in unserem System.
- Dabei wird zunächst versucht, eine Datei mit dem Namen protokoll.txt aus demselben Verzeichnis, in dem das Skript ausgeführt wird, auszulesen.
- Diese Datei ist beim ersten Aufruf noch nicht vorhanden, aus diesem Grund muss ein passender Modus gewählt werden:
  - Modus „a“: Nur zum Schreiben geöffnet; platziere Dateizeiger auf Dateiende. Existiert die Datei nicht, versuche, diese zu erzeugen.
  - Der Modus ist immer abhängig vom Anwendungsfall.



- Wenn die Datei existiert, wird in der Variablen `$datei` eine Referenz auf die offene Datei abgelegt.
- Diese Referenz nennt man auch Ressource oder Handle.
- Über diese Referenz kann nun in die Datei geschrieben werden, für uns ist relevant:
  - Uhrzeit
  - Username
  - IP-Adresse (Datenschutzvorgaben beachten!)
- Nachdem `fwrite` abgeschlossen ist, kann die Bearbeitung mit `fclose` beendet werden.



- Wir erweitern unsere auswertung.php-Datei mit den folgenden Zeilen, um jeden Anmeldeversuch in die Datei protokoll.txt zu schreiben:

```
<?php  
  
// Protokolliere Anmeldeversuch  
$datei = fopen("protokoll.txt", "a");  
fwrite($datei, date("H:i:s") . " : " . $username . " mit " . $_SERVER['REMOTE_ADDR'] . PHP_EOL);  
fclose($datei);  
  
?>
```

```
19:08:02: Musterfrau mit 46.5.XXX.XXX  
19:08:08: Dopatka mit 46.5.XXX.XXX  
19:08:12: Test mit 46.5.XXX.XXX
```



Funktion	Bedeutung
\$fh=fopen(\$var,\$op)	<p>öffnet die in \$var angegebene Datei oder URL und gibt eine Referenz auf das geöffnete Objekt zurück; mögliche Werte für die Operation \$op sind:</p> <p>“a“: öffnen zum Schreiben; Referenz zeigt auf das Ende der Datei; eine nicht-existierende Datei wird angelegt</p> <p>“a+“: öffnen zum Lesen und Schreiben; Referenz zeigt auf das Ende der Datei; eine nicht-existierende Datei wird angelegt</p> <p>“r“: öffnen zum Lesen; Referenz zeigt auf den Anfang der Datei</p> <p>“r+“: öffnen zum Lesen und Schreiben; Referenz zeigt auf den Anfang der Datei</p> <p>“w“: öffnen zum Schreiben; Referenz zeigt auf den Anfang der Datei; existierende Datei wird auf 0Byte gesetzt; eine nicht-existierende Datei wird angelegt</p> <p>“w+“: öffnen zum Lesen und Schreiben; Referenz zeigt auf den Anfang der Datei; eine nicht-existierende Datei wird angelegt</p>



Funktion	Bedeutung
\$erg=fopen(\$fh)	schließt eine zuvor mit fopen geöffnete Datei; bei Erfolg wird TRUE, sonst FALSE zurückgeliefert
\$erg=fgets(\$fh,\$var)	liest Daten aus der Dateireferenz \$fh ein; entweder bis Zeilenumbruch, Datei-Ende oder bis zur Anzahl an Zeichen, die in \$var angegeben wurde
\$erg=fgetcsv(\$fh,\$var,\$trenner)	liest eine Zeile aus der geöffneten CSV-Datei (Comma Separated Values) \$fh aus; der Parameter \$var beinhaltet die Anzahl der zu lesenden Zeichen und muss größer sein als die längste Zeile in der Datei, da sonst das Ende der Zeile nicht gefunden wird; in \$trenner wird das Trennzeichen der CSV-Datei angegeben; die Rückgabe ist ein Datenfeld
fwrite(\$fh,\$var)	schreibt die als Zeichenkette in \$var übergebenen Daten in die Datei \$fh



Funktion	Bedeutung
\$erg=is_file(\$var)	wenn die Datei existiert und es eine reguläre Datei ist, gibt is_file TRUE, sonst FALSE zurück; in \$var wird der Pfad und der Name der Datei als Zeichenkette übergeben
\$erg=file_exists(\$var)	überprüft, ob eine in \$var übergebene Pfad- und Dateiangabe existiert und gibt TRUE zurück, wenn das der Fall ist und ansonsten FALSE
\$erg=filectime(\$var)	gibt das Datum und die Uhrzeit der letzten Änderung einer Datei in \$var als UNIX-Zeitstempel zurück
\$erg=filesize(\$var)	gibt die Grösse der Datei, die ggf. zusammen mit Pfadangabe in \$var angegeben wird, zurück; bei einem Zugriffsfehler wird FALSE zurück gegeben



Funktion	Bedeutung
\$erg=flock(\$fh,\$op)	schützt eine Datei \$fh vor Operationen, die in \$op übergeben werden; \$op kann sein: LOCK_SH: Verriegelung für Lesezugriff LOCK_EX: exklusive Verriegelung für Schreibzugriffe LOCK_UN: gibt eine Verriegelung wieder frei LOCK_NB: verhindert, dass die Funktion während der Verriegelung blockiert; diese Konstante kann zusätzlich zu den anderen Konstanten angegeben werden
\$erg=unlink(\$var)	löscht die in \$var übergebene Datei und gibt FALSE zurück, wenn die angegebene Datei nicht gelöscht werden konnte