



DATEIEN FÜR DIE SPEICHERUNG VON DATEN

DATEN AUS EINER DATEI EINLESEN

```
<?php
$handle = fopen('beispiel.txt', 'r');

$result = array();
if($handle) {
    $i = 0;
    while(!feof($handle)) {
        $content = fgets($handle);
        $result[$i] = $content;
        $i++;
        print $content."<br>\n";
    }
    fclose($handle);
} else {
    print "Datei konnte nicht geöffnet werden<br>\n";
}
```

- <https://www.php.net/manual/de/function.fopen.php>
- <https://www.php.net/manual/de/function.fgets.php>
- <https://www.php.net/manual/de/functionfeof.php>
- <https://www.php.net/manual/de/function.fclose.php>

- ❖ Befehl `fopen()` öffnet eine Datei oder URL
- ❖ nach dem Dateinamen folgt der Modus => spezifiziert Zugriffstyp
 - `r` - nur zum Lesen geöffnet, platziert Dateizeiger am Dateianfang
 - `r+` - Lesen und Schreiben, platziert Dateizeiger am Dateianfang
 - `w` - nur zum Schreiben, platziert Dateizeiger am Dateiende.
 - `w+` - zum Schreiben und Lesen geöffnet, platziert Dateizeiger am Dateiende. Existiert Datei nicht, versucht, diese zu erzeugen
 - `a` - zum Erweitern der bestehenden Inhalte
- ❖ `fopen()` erzeugt als Rückgabewert einen Handle => bietet im Programm Zugriff auf Datei und muss in Variable gespeichert werden
 - Üblich `$handle` oder `$fh` (file handle) zu nennen
- ❖ Um jede Zeile aus File einzulesen => `while`-Schleife
 - `feof` - Prüft, ob ein Dateizeiger am Ende der Datei steht.
 - Solange er nicht am Ende ist hat er Wert `false`
- ❖ `fgets()` - Liest eine Zeile von der Position des Dateizeigers.
- ❖ `fclose()` - schließt den Zugriff auf die Datei wieder und sollte immer nach der letzten Verwendung des Handles eingefügt werden

DATEN IN EINER DATEI SPEICHERN

- ❖ Modus beim Öffnen für das Schreiben: w
- ❖ `fputs()` - Diese Funktion ist ein Alias für: `fwrite()`
`fwrite()` schreibt den Inhalt der Zeichenkette `string` in die Datei, auf welche der Dateizeiger `handle` zeigt.
- ❖ Mit `\n` Zeilenumbruch in der Datei erzeugen

```
<?php
$handle = fopen('beispiel.txt', 'w');
if($handle) {
    for($i = 0; $i < 10; $i++){
        fputs($handle, ($i+1) . "\n");
    }
    fclose($handle);
} else {
    print "Datei konnte nicht geöffnet werden.<br>\n";
}
```

- <https://www.php.net/manual/de/function.fwrite.php>

BEISPIEL

```
<form method="post" action="sameSiteAgain.php">
  Name: <input type="text" name="name"><br>
  E-Mail: <input type="text" name="e-mail"><br>
  <input type="submit" value="Senden">
</form>
<?php
$handle = fopen("beispiel.txt","w");
if ($handle) {
    if (!empty($_POST['name']) && !empty($_POST['e-mail'])) {
        if ($_POST['name'] != "") {
            $name = $_POST['name'];
            fputs($handle, $name."\n");
        }
        if ($_POST['e-mail'] != "") {
            $email = $_POST['e-mail'];
            fputs($handle, $email."\n");
        }
    }
    fclose($handle);
} else {
    print "Die Datei konnte nicht geöffnet werden.<br>\n";
}
?>
```

❖ Superglobale Variablen

Superglobals — Superglobals sind Built-in-Variablen, die immer in allen Gültigkeitsbereichen (s.g. Scopes) verfügbar sind

\$_GET
\$_POST
\$_REQUEST
\$_SERVER
...

- <https://www.php.net/manual/de/language.variables.superglobals.php>
- <https://www.php.net/manual/de/reserved.variables.request.php>
- <https://www.php.net/manual/de/reserved.variables.post.php>
- <https://www.php.net/manual/de/reserved.variables.get.php>

DIE DATEIRECHTE BEACHTEN

- ❖ Auf Linx-Webservern gibt es Zugriffsrechte auf Dateien
- ❖ Sollen verhindern, dass Unbefugte die Datei verwenden, löschen, ändern
- ❖ Rechte werden in Zahl gespeichert
 - 0 – an der Datei bestehen keine Rechte
 - 1 – Datei darf ausgeführt werden
 - 2 & 4 – Lese- bzw. Schreibrechte
 - 6 – Lese und Schreibrechte
 - 3 – Nutzer darf Datei ausführen und lesen
- ❖ Wenn alle Nutzer Lese- und Schreibrechte erhalten sollen: 0666

KLEINE ÜBUNG

- ❖ Schreibe ein Programm, das ein Formularfeld im Browser anzeigt. Dieses soll den Nutzer auffordern, eine beliebige Zahl einzugeben. Erstelle ein Textdokument mit verschiedenen Zahlen (jeweils in einer eigenen Zeile) und lese diese mit dem Programm ein. Multipliziere daraufhin die Zahlen mit dem Wert, den der Nutzer über den Browser eingegeben hat. Erstelle ein neues Textdokument und speichere die Ergebnisse darin ab.
Achtung: Zahlen aus Formularfeldern sollten mit intval() erst in einen Integer formatiert werden

```
<?php
if (!empty($_REQUEST['zahl'])) {
    print "<p>Ihre Eingabe wurde erfasst und für die
Berechnung des neuen Dokuments verwendet.</p>\n";
}
?>
```

```
<form method="post" action="aufgabe11-1.php">
    Geben Sie eine beliebige Zahl ein:
    <input type="text" name="zahl"><br>
    <input type="submit" value="Senden">
</form>
```

```
<?php
if (!empty($_REQUEST['zahl'])) {
    $handle = fopen("zahlen.txt","r");
    $handle2 = fopen("ergebnis.txt","w");
    $ergebnis = array();
    if ($handle) {
        $i = 0;
        while (!feof($handle)) {
            $inhalt = fgets($handle);
            $ergebnis[$i] = $inhalt;
            $i++;
        }
    }
}
```

```
if ($_REQUEST['zahl'] != "") {
    $faktor = $_REQUEST['zahl'];
    foreach ($ergebnis as $wert) {
        $wert = intval($wert)*$faktor;
        fputs($handle2, $wert."\n");
    }
    fclose($handle);
    fclose($handle2);
} else {
    print "Die Datei konnte nicht geöffnet werden.<br>\n";
}
?>
```



ENDE

QUELLE: PHP & MYSQL FÜR EINSTEIGER
ISBN: 978-3-96645-009-6