

Dateien und Verzeichnisse

**Dateien und Verzeichnisse lesen,
schreiben und ergänzen**

Webapplikationen realisieren Programmierung in PHP

Lesen von einer Datei - 1

- Mit der Funktion *file(\$dateiname)* wird eine Datei ganz eingelesen und als Array zurückgegeben.
- Jedes Feld des Arrays entspricht dabei einer Zeile (Record) der Datei.
- Der Zeilenumbruch bleibt erhalten.
- Im Fehlerfall gibt die Funktion *false* zurück.
- Es gibt verschiedene Möglichkeiten eine Arrayvariable zu erstellen

Lesen von einer Datei - 2

Beispiel:

```
<?php echo "<?xml version='1.0' encoding='ISO-8859-1' ?>
<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Strict//EN,
    'http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd'>";
echo "<html xmlns='http://www.w3.org/1999/xhtml' lang='de, xml:lang='de'>";
echo "<head><title>Ausgabe einer Datei</title></head>";
echo "<body>";
echo "<h1>Ausgabe einer Datei</h1>";
echo "<pre>";

$lines = file("read.php");

foreach($lines as $line) {
    echo htmlentities($line);
    echo "</pre>";
    echo "</body>";
    echo "</html>"
}
?>
```

Hinweis: Die Funktion *htmlentities(\$text)* konvertiert alle nicht alpha-numerischen Zeichen in ihr HTML-Equivalent.

In eine Datei schreiben

- Das Schreiben in eine Datei ist um einiges aufwändiger und fehlerbehaftet.
- Das folgende Beispiel zeigt exemplarisch wie es geht.
- Für die Details sei auf www.php.net verwiesen.

```
$filename = "test.txt";
$text = "Der Text zum Anfügen\n";

if (!$handle = fopen($filename, "a")) {
    echo "datei $filename kann nicht zum schreiben geöffnet oder
    angelegt werden";
} elseif (!fwrite($handle, $text)){
    echo "fehler beim schreiben von $filename";
} elseif (!fclose($handle)){
    echo "fehler beim schliessen von $filename";
}
```

Records überschreiben

- In PHP existiert keine Funktion, die ein Überschreiben der Daten in einer Datei übernimmt.
- Solange die neuen Daten genau gleich lang wären, wie die zu ersetzenden Daten, wäre dies noch möglich.
- Sobald aber die Anzahl Bytes zwischen alt und neu variiert, müssten umfangreiche Verschiebeaktionen vorgenommen werden.

Records überschreiben – In kleineren Dateien

Solange die Datei nur wenige Kilobyte gross ist, können Sie folgendes Vorgehen wählen:

1. Einlesen der Datei in einen Array (Funktion: `file`)
2. Suchen der Zeile mit den Daten die geändert werden sollen.
3. Ändern der Daten im Array
4. Schreiben des ganzen Arrays in die ursprüngliche Datei (Funktion: `file_put_contents`)

Die `file_put_contents(...)` Funktion

- Ab Version 5 kennt PHP die `file_put_contents(...)`-Funktion welche das Schreiben in eine Datei massiv vereinfacht.
- Siehe php.net

Records überschreiben – In grösserem Dateien

Bei grossen Dateien ist es nicht möglich, diese komplett in den Speicher einzulesen. Daher müssen solche Dateien Stück für Stück gelesen werden.

1. Öffnen der Datei „alt“ im Read-Modus (fopen)
2. Öffnen der Datei „neu“ im Write-Modus
3. Solange Lesen der Datei „alt“ nicht End-of-File liefert (fread)
 1. Allfällige Änderungen an den Daten durchführen
 2. Schreiben der Daten in Datei „neu“ (fwrite)
4. Schliessen der Dateien „alt“ und „neu“ (fclose)
5. Löschen der Datei „alt“ (unlink)
6. Umbenennen der Datei „neu“ in „alt“ (rename)

Daten einfügen und löschen

- Wie Daten am Ende einer Datei eingefügt werden, haben Sie bereits beim Schreiben der Funktion `append` gelernt.
- Wie können Sie aber mittendrin Daten einfügen oder Daten aus einer Datei löschen?
- Am Einfachsten ist in beiden Fällen das oben beschriebene Vorgehen für das Ändern von grossen Dateien.
- Anstatt die Daten zu ändern, müssen Sie an der entsprechenden Stelle der Datei
 - die neuen Daten einfügen
 - die gelesenen Daten nicht schreiben

mittels Array:

Wenn Sie die Daten in einen Array eingelesen haben (nur bei kleineren Dateien) können Sie mit der Funktion *`array_splice`* (siehe php.net) einzelne Zeilen einfügen oder entfernen.

Verzeichnisse

Der folgende Codeauszug illustriert exemplarisch die Ausgabe der Dateinamen und Dateigrößen von jpg-Bildern in einem Verzeichnis images.

```
chdir("images");  
foreach (glob("*.jpg") as $filename) {  
    echo "$filename size " . filesize($filename) . "<br/>\n";  
}
```

Aufgaben / Übungen

- *Viel Spass !!!*
- *Abgabe: wie immer spätestens kommenden Donnerstag 12h auf den BSCW-Server hochladen.*

12_1_1: Ausgabe einer Datei

- Untersuchen Sie den vom obigen PHP-Beispiel ausgegebenen HTML-Code.
- Was würde passieren, wenn man die Zeilen direkt ausgibt ohne die Funktion *htmlspecialchars(...)* darauf anzuwenden?
- Schreiben Sie Ihre Vermutung auf und probieren Sie es anschließend aus.

12_1_2: Alternat. Ausgabe einer Datei

- Die Ausgabe der Datei hätte auch mit den beiden folgenden Alternativen erfolgen können. Probieren Sie diese Alternativen aus und nennen Sie je einen Grund die für, bzw. gegen diese Alternativen sprechen.

- Alternative 1

```
... <pre>
    echo htmlentities(implode("",file("readalt1.php")));
</pre> ...
```

- Alternative 2

```
... <pre>
    echo htmlentities(file_get_contents("readalt2.php"));
</pre> ...
```

12_1_3: fopen, fwrite und fclose

12_1_4: append(\$filename, \$text)

- 12_1_3: Verwenden Sie das obige Beispiel um bei jedem Aufruf des Scripts einen Zeitstempel in eine Datei *log.txt* zu schreiben.
- 12_1_4: Schreiben Sie eine Funktion *append(\$filename, \$text)* welche das Anfügen von Text an eine Datei kapselt.
- Die Funktion soll *true* zurückgeben, wenn das Schreiben geklappt hat.
- Andernfalls soll die entsprechende Fehlermeldung zurückgegeben werden.

12_1_5: write(\$filename, \$bak, \$text)

- Schreiben Sie eine Funktion *write(\$filename, \$bak, \$text)* welche einen Text in eine Datei schreibt.
- Allfällig vorhandene Daten sollen dabei überschrieben werden, nicht bevor jedoch von der alten Datei vorgängig eine Sicherung mit der angegebenen Endung *\$bak* angelegt wurde.
- Tipp: Verwende die Funktionen *rename(...)* und *unlink(...)*.

12_1_6: Page-Counter

12_1_7: Gästebuch

- 12_1_6: Erstellen und testen Sie unter Verwendung der eingebauten Funktion *file(...)* und der selbstgeschriebenen Funktion *write(...)* einen Page-counter.
- 12_1_7: Erstellen und testen Sie unter Verwendung der eingebauten Funktion ein Gästebuch.
 - *file(...)* und der selbstgeschriebenen Funktion
 - *write(...)*

12_1_8: Bildergalerie

12_1_9: Exif-Bilddaten

- 12_1_8: Erstellen und testen Sie unter Verwendung der eingebauten Funktion eine Bildgalerie die alle jpg-Bilder in einem Verzeichnis ausgibt.
 - `chdir(...)`
 - `glob(...)`
- 12_1_9: Verwenden Sie die eingebauten *Exif-Funktionen* (siehe php.net) um zu jedem Bild zusätzlich auszugeben, wann und mit welcher Kamera es erstellt wurde.