**Lektion 14: Filesystem-Functions (Dateisystem-Funktionen)**

Mit PHP können Sie auf das Dateisystem des Servers zugreifen. Dadurch können Sie Ordner und Text-Dateien durch PHP-Skripte manipulieren.

Sie können PHP z.B. dazu benutzen, eine Textdatei zu lesen oder zu schreiben. Oder Sie können alle Dateien eines angegebenen Ordners auflisten lassen. Es gibt viele Möglichkeiten und PHP kann Ihnen eine Menge langwierige Arbeit ersparen.

Hier werden wir uns anschauen, wie Sie PHP dazu benutzen können, mit Ordnern und Dateien zu arbeiten. Das Ziel ist, Ihnen einen schnellen Überblick zu geben. In der nächsten Lektion werden wir uns die verschiedenen Möglichkeiten dann genauer anschauen. Wir werden nicht alle Möglichkeiten behandeln. Um eine komplette Übersicht zu erhalten, möchten wir wiederholt auf die Dokumentation verweisen.

[[Dokumentation](http://www.php.net/manual/en/function.filemtime.php)filemtime](http://www.php.net/manual/en/function.filemtime.php)

Gibt den Zeitpunkt wieder, zu dem der Inhalt einer Datei das letzte Mal geändert wurde (als UNIX timestamp - siehe Lektion 4).

[[Dokumentation](http://www.php.net/manual/en/function.fileatime.php)fileatime](http://www.php.net/manual/en/function.fileatime.php)

Gibt den Zeitpunkt wieder, an dem auf eine Datei zuletzt zugegriffen, bzw. diese geöffnet wurde (als UNIX timestamp - siehe Lektion 4).

[[Dokumentation](http://www.php.net/manual/en/function.filesize.php)filesize](http://www.php.net/manual/en/function.filesize.php)

Gibt die Größe einer Datei in Bytes aus.

Lassen Sie uns versuchen, diese drei Eigenschaften bei der Datei herauszufinden, die Sie gerade ansehen: "/tutorials/php/lesson14.php"

<html>

<head>

<title>Dateisystem</title>

</head>

<body>

<?php

// Eigenschaften herausfinden und ausgeben

echo "<h1>Datei: lesson14.php</h1>";

echo "<p>Wurde zuletzt editiert: " . date("r", filemtime("lesson14.php"));

echo "<p>Wurde zuletzt geöffnet: " . date("r", fileatime("lesson14.php"));

echo "<p>Größe: " . filesize("lesson14.php") . " bytes";

?>

</body>

</html>

 - [Beispiel anzeigen](http://de.html.net/tutorials/php/lesson14_ex1.php)

**Ordner**

PHP erlaubt Ihnen auch, mit den Ordnern auf dem Server zu arbeiten. Wir werden wieder nicht alle verschiedenen Möglichkeiten durchgehen, sondern nur ein Beispiel zeigen. Wieder sei uns der Hinweis auf die Dokumentation gestattet.

[[Dokumentation](http://php.net/manual/en/function.opendir.php)opendir](http://php.net/manual/en/function.opendir.php)

Öffnet einen vorgegebenen Ordner.

[[Dokumentation](http://php.net/manual/en/function.readdir.php)readdir](http://php.net/manual/en/function.readdir.php)

Gibt den Dateinamen der nächsten Datei in einem geöffneten Ordner aus. (vergl. [[Dokumentation](http://php.net/manual/en/function.opendir.php)opendir](http://php.net/manual/en/function.opendir.php))

[[Dokumentation](http://php.net/manual/en/function.closedir.php)closedir](http://php.net/manual/en/function.closedir.php)

Schließt einen vorgegebenen Ordner.

Das folgende Beispiel listet den Inhalt des Ordners "tutorials/php/" auf.

<html>

<head>

<title>Dateisystem</title>

</head>

<body>

<?php

// Öffnet den Ordner

$folder = opendir("../../tutorials/php/");

// Schleife über alle Dateien im Ordner

while (($entry = readdir($folder)) != "") {

echo $entry . "<br />";

}

// Schließt den Ordner

$folder = closedir($folder);

?>

</body>

</html>

 - [Beispiel anzeigen](http://de.html.net/tutorials/php/lesson14_ex2.php)

Im Beispiel wird zuerst das Verzeichnis "../../tutorials/php/" geöffnet. Dann werden mit einer Schleife die Namen der jeweils nächsten Datei geschrieben, solange es weitere Dateien gibt. Zum Schluss wird der Ordner geschlossen.

In der nächsten Lektion werden wir uns anschauen, wie man aus einer Textdatei liest, bzw. in eine Textdatei schreibt.