

Mit PHP kann man unterschiedliche Informationen über eine Datei erheben.



Übungsdatei. Über eine Zählschleife werden Zufallszahlen zu einem String zusammengeführt. Der String wird als `zufall.txt` gespeichert (ca. 800 KiB).

```
<?php
    $inhalt = "Zufallszahlen: ";
    for($i = 0; $i < 100000; $i++) {
        $inhalt = $inhalt . rand(10000, 99999) . " - ";
    }
    file_put_contents('zufall.txt', $inhalt);
?>
```



file_exists() Prüft ob eine Datei oder ein Verzeichnis existiert und liefert TRUE oder FALSE

```
if(file_exists('zufall.txt')) {echo 'zufall.txt existiert';}
```



Weitere Funktionen zur "Existenzprüfung" (Liefert: TRUE oder FALSE)

is_file()	prüft ob es sich um eine reguläre Datei handelt
is_readable()	prüft die Existenz und die Lesbarkeit
is_executable()	prüft ob die Datei ausführbar ist.
is_writable()	prüft ob in die Datei geschrieben werden kann.



filesize() gibt die Dateigröße in Byte zurück. Das Ergebnis kann mit dem Faktor 1024 in Kilobyte, Megabyte usw. umgerechnet werden!

```
echo filesize('zufall.txt') . ' Byte';
```



mime_content_type() ermittelt den MIME-Typ einer Datei und liefert einen String z. B. image/gif oder text/plain.

```
echo mime_content_type('zufall.txt');
```



pathinfo() ermittelt Infos über einen Dateipfad und liefert ein Array.

```
$datei = pathinfo('zufall.txt');
echo 'Basis: ' . $datei['basename'] . '<br>';
echo 'Dateierweiterung: ' . $datei['extension'] . '<br>';
echo 'Dateiname: ' . $datei['filename'] . '<br>';
```



stat() ermittelt Infos über eine Datei und liefert ein Array.

```
$dateiinfos = stat('zufall.txt');
$letztterZugriff = $dateiinfos['atime'];
echo date("Y-m-d H:m", $letztterZugriff);
```