

Mögliche Filetypen sind:

- block: block special device
- char: character special device
- dir: directory
- fifo: FIFO (named pipe)
- file: regular file
- link: symbolic link
- unknown: unknown file type

Die Ziffernfolge 666 informiert über die Rechte des Besitzers, der Gruppe und der Anderen. Dabei bedeutet die erste Ziffer 6, dass der Besitzer der Datei lesen (4) und schreiben (2), aber nicht ausführen darf. Die zweite Ziffer 6 informiert über die – hier gleichen – Rechte der Gruppe und die letzte Ziffer über die aller anderen Benutzer.

Die möglichen Rechte der einzelnen Benutzer sind:

Besitzer			Gruppe			Andere		
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

r von read = lesen, w von write = schreiben,
x von execute = ausführen

Beispiel:

Für eine Datei wird als Schlüssel für die Zugriffsrechte die Zahl 754 geliefert. Die einzelnen Berechtigungen werden folgendermaßen ermittelt:

7 = Besitzer: lesen (4) + schreiben (2) + ausführen (1);

5 = Gruppe: lesen (4) + ausführen (1);

4 = Andere: lesen (4).

8.3.9 Arbeiten mit Formularen

Mit PHP-Skripten können Daten aus HTML-Formularen verarbeitet werden.

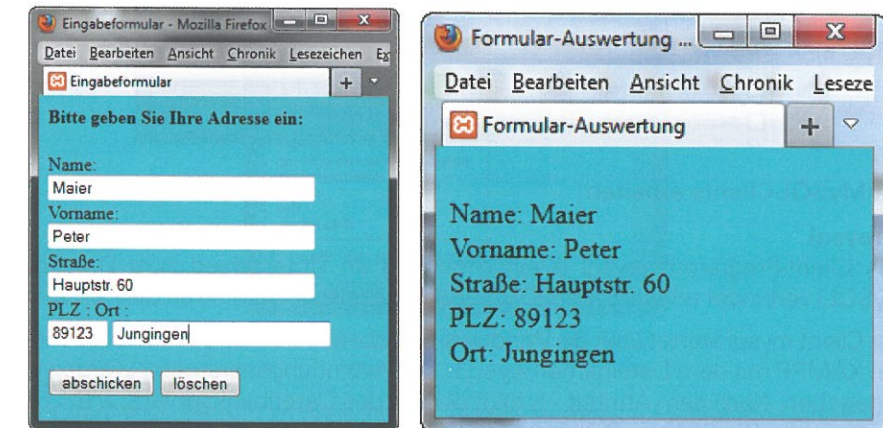
Beispiel:

In einem HTML-Formular wird die Adresse (Name, Vorname, Straße, PLZ, Ort) eingetragen. Über eine Befehlsschaltfläche sollen die Daten an ein PHP-Skript gesendet und dort angezeigt werden.

Das Listing des Eingabeformulars lautet z. B.:

```
<html>
<head><title>Eingabeformular</title></head>
<body text="#000000" bgcolor="#00E0FF">
  <h4>Bitte geben Sie Ihre Adresse ein:</h4>
  <form action="form_aus.php" method="POST">
    Name:<br><input type="Text" name="Name"
    size="35"><br>
    Vorname:<br><input type="Text" name="Vorname"
    size="35"><br>
    Straße:<br><input type="Text" name="Strasse"
    size="35">
    <br>
    PLZ : Ort :<br><input type="Text" name="PLZ"
    size="5" >
    <input type="Text" name="Ort"
    size="28"><br><br>
    <input type="Submit" name="submit" value="ab
```

```
schicken">
  <input type="reset" name="loeschen" value="löschen">
</form>
</body>
</html>
```



Zunächst erstellt man ein Formular im HTML-Code. Dort wird festgelegt, dass z. B. durch die Methode POST die Formulardaten an ein PHP-Skript, z. B. form_aus.php, gesendet werden. Dies geschieht durch die HTML-Anweisung

```
<form action="form_aus.php" method="post">.
```

Durch Betätigen der Schaltfläche abschicken werden die Daten an die Datei form_aus.php gesendet.

Das Listing der Datei form_aus.php lautet z. B.:

```
<html><head>
  <title>Formular-Auswertung</title></head>
<body text="#000000" bgcolor="#00E0FF">
  <?php
    $name      =$_POST["Name"];
    $vorname   =$_POST["Vorname"];
    $strasse   =$_POST["Strasse"];
    $plz       =$_POST["PLZ"];
    $ort       =$_POST["Ort"];
    echo "<br>Name:      ", $name;
    echo "<br>Vorname:   ", $vorname;
    echo "<br>Straße:     ", $strasse;
    echo "<br>PLZ:        ", $plz;
    echo "<br>Ort:         ", $ort;
  ?>
</body></html>
```

Die Standardvariable \$_POST[] überträgt die Daten im Datenformat eines Arrays. Die Elemente werden z. B. mit \$vorname=\$_POST["Vorname"] ausgelesen und einer lokalen Variablen zugewiesen. Mithilfe des Befehls echo... können sie angezeigt werden.

8.4 Das Datenbanksystem Maria DB

Hinweis:

Die Datenbanksysteme MariaDB und MySQL sind nahe zu gleichwertig zu verwenden.

MariaDB ist ein relationales Datenbanksystem. Es wird eine Client-Server-Architektur benutzt, wobei der Datenbankserver mysqld.exe als Hintergrundprozess auf einem Server läuft. Es können beliebig viele Clients auf den Server zugreifen. Das Multithreading-Konzept ermöglicht, verschiedene Anfragen an den Server abzuarbeiten.