

Lösungen zu Übungsaufgaben

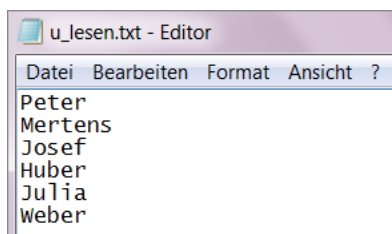
Quelle: Maurice (2015); Theis (2016)

Teil: Ein- und Ausgabe, Web-Formularanbindung

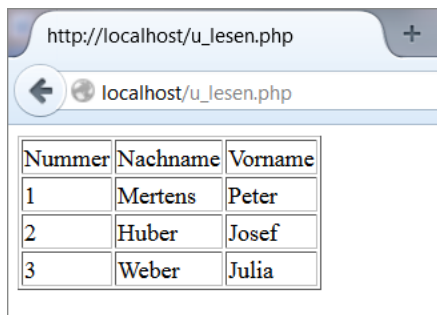
Kapitel: "9. Ein- und Ausgabe"

Aufgabe 9.1:

Schreiben Sie mithilfe eines Texteditors mehrere Namen in eine Datei (*u_lesen.txt*). Jeder Name soll zweizeilig geschrieben werden: in der ersten Zeile der Vorname und in der zweiten Zeile der Nachname (wie die folgende Abbildung zeigt).



Erstellen Sie ein PHP-Programm, das diese Datei öffnet und liest sowie die Namen (zusätzlich mit laufender Nummer) in einer HTML-Tabelle auf dem Bildschirm ausgibt (wie die folgende Abbildung zeigt).



Lösung:

```
<html>
<body>
<?php
    if(!file_exists("u_lesen.txt"))
        exit("Datei konnte nicht gefunden werden");

    $fp = @fopen("u_lesen.txt","r");
    if(!$fp)
        exit("Datei konnte nicht geöffnet werden");

    echo "<table border='1'>";
    echo "<tr> <td>Nummer</td> <td>Nachname</td> <td>Vorname</td> </tr>";
```

```

$nr = 0;

while (!feof($fp))
{
    $vn = fgets($fp, 100);
    $nn = fgets($fp, 100);
    $nr = $nr + 1;
    echo "<tr> <td>$nr</td> <td>$nn</td> <td>$vn</td> </tr>";
}

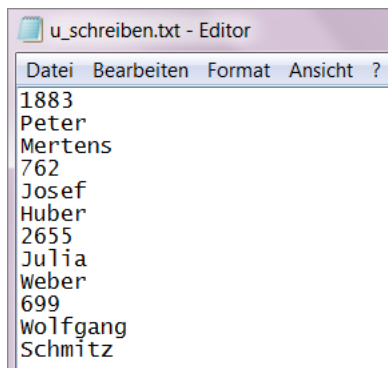
echo "</table>";
fclose($fp);
?>
</body>
</html>

```

Aufgabe 9.2:

In einer Datei (*u_schreiben.txt*) stehen mehrere Datensätze in der folgenden Form (siehe Abbildung):

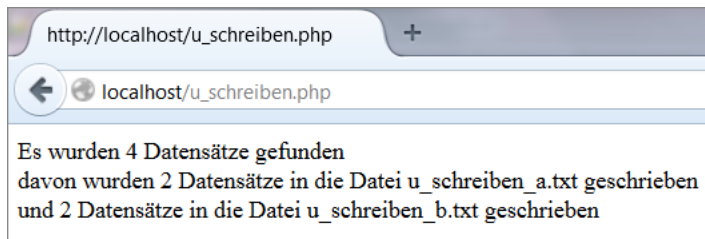
- Erste Zeile: laufende Nummer
- Zwei Zeile: Vorname
- Dritte Zeile: Nachname



Schreiben Sie ein PHP-Programm, das diese Datei öffnet und liest sowie die Datensätze in der gleichen Form in zwei verschiedenen Dateien ausgibt:

- Datensätze mit einer laufenden Nummer kleiner als 1.000 in der Datei *u_schreiben_a.txt*
- alle anderen Datensätze in der Datei *u_schreiben_b.txt*

Die Dateien sollen bei jedem Programmaufruf überschrieben werden. Auf dem Bildschirm soll eine Kontrollausgabe wie in folgender Abbildung erfolgen (hier mit den oben angegebenen Beispieldaten).



Hinweis: Die Funktion `fgets()` liest die Zeilen einschließlich des Zeilenumbruchs in eine Variable ein. Wird diese Variable in eine Datei ausgegeben, so wird auch der Zeilenumbruch ausgeführt. Er braucht nicht zusätzlich ausgegeben zu werden.

Lösung:

```
<html>
<body>
<?php
    if(!file_exists("u_schreiben.txt"))
        exit("Datei konnte nicht gefunden werden");

    $fpdata = @fopen("u_schreiben.txt","r");
    $fpa = @fopen("u_schreiben_a.txt","w");
    $fpb = @fopen("u_schreiben_b.txt","w");
    if(!($fpdata && $fpa && $fpb))
        exit("Eine der Dateien konnte nicht geöffnet werden");

    /* Zähler initialisieren */
    $gz = 0;
    $az = 0;
    $bz = 0;

    while (!feof($fpdata))
    {
        $gz = $gz + 1;
        $nr = fgets($fpdata, 100);
        $vn = fgets($fpdata, 100);
        $nn = fgets($fpdata, 100);

        /* in erste Ausgabedatei */
        if ($nr<1000)
        {
            $az = $az + 1;
            fputs ($fpa, "$nr$vn$nn");
        }

        /* in zweite Ausgabedatei */
        else
        {
```

```

        $bz = $bz + 1;
        fputs ($fpb, "$nr$vn$nn");
    }
}

/* alle Dateien schließen */
fclose($fpdata);
fclose($fpa);
fclose($fpb);

/* Kontrolle */
echo "Es wurden $gz Datensätze gefunden<br />";
echo "davon wurden $az Datensätze in die Datei u_schreiben_a.txt geschrieben<br />";
echo "und $bz Datensätze in die Datei u_schreiben_b.txt geschrieben";
?>
</body>
</html>

```

Datei: u_schreiben_a
=====

62
Josef
Huber
699
Wolfgang
Schmitz

Datei: u_schreiben_b
=====

1883
Peter
Mertens
2655
Julia
Weber

Kapitel: "10. Web-Formularanbindung"

Aufgabe 10.1:

Ergänzen Sie das folgende Beispiel durch zwei weitere Felder für eine E-Mail-Adresse und eine Telefonnummer. Die Eingaben dieser beiden Felder sollen ebenfalls ausgegeben werden.

```

<form action="verarbeitung.php" method="get">
Ihr Vorname: <br />
<input type="text" name="vorname" size="20" maxlength="30" />
<br />

```

```
Ihr Nachname: <br />
<input type="text" name="nachname" size="20" maxlength="30" />
<br />
<input type="submit" value="Abschicken" />
</form>
```

Lösung:

Zuerst muss das Formular um die beiden Felder ergänzt werden:

```
//formular.php
<form action="verarbeitung.php" method="get">
Ihr Vorname: <br />
<input type="text" name="vorname" size="20" maxlength="30" />
<br />
Ihr Nachname: <br />
<input type="text" name="nachname" size="20" maxlength="30" />
<br />
E-Mail: <br />
<input type="text" name="email" size="20" maxlength="30" />
<br />
Telefon: <br />
<input type="text" name="telefon" size="20" maxlength="30" />
<br />
<input type="submit" value="Abschicken" />
</form>
```

Im Verarbeitungsskript wird in zwei zusätzlichen Zeilen auf die Inhalte der neu ergänzten Formularfelder zugegriffen:

```
// verarbeitung.php
```

```
<?php
echo "Ihre Eingaben<br />\n";
echo "Vorname: " . htmlspecialchars($_GET["vorname"])
    . "<br />\n";
echo "Name: " . htmlspecialchars($_GET["nachname"])
    . "<br />\n";
echo "E-Mail: " . htmlspecialchars($_GET["email"])
    . "<br />\n";
echo "Telefon: " . htmlspecialchars($_GET["telefon"])
    . "<br />\n";
?>
```

Aufgabe 10.2:

Modifizieren Sie das Beispiel aus Aufgabe 10.1 so, dass die folgenden Bedingungen erfüllt werden:

- Das Formular und die Auswertung des Formulars sollen innerhalb desselben Skripts stehen.
- Die Daten sollen per POST gesendet werden
- Außerdem soll entweder die Ausgabe oder das Formular angezeigt werden.

Lösung:

Das um zwei Felder ergänzte Formular:

```
// fomular_oder_auswertung.php
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<title>Beispielformular</title>
</head>
<body>
<?php
    if (!isset($_POST["vorname"])) {
        ?>
        <form action="<?php echo
            htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="post">
            Ihr Vorname: <br />
            <input type="text" name="vorname" size="20" maxlength="30" />
            <br />
            Ihr Nachname: <br />
            <input type="text" name="nachname" size="20" maxlength="30" />
            <br />
            E-Mail: <br />
            <input type="text" name="email" size="20" maxlength="30" />
            <br />
            Telefon: <br />
            <input type="text" name="telefon" size="20" maxlength="30" />
            <br />
            <input type="submit" value="Abschicken" />
        </form>
        <?php
        } else {
            echo "Ihre Eingaben<br />\n"
                . "Vorname: " . htmlspecialchars($_POST["vorname"])
                . "<br />\n Name: " . htmlspecialchars($_POST["nachname"])
                . "<br />\n E-Mail: " . htmlspecialchars($_POST["email"])
                . "<br />\n Telefon: " . htmlspecialchars($_POST["telefon"])
                . "<br />\n";
        }
    }
```

```
?>
</body>
</html>
```

Aufgabe 10.3:

Das folgende Programm mit einer Funktion, die zu einem Nettobetrag den Bruttobetrag berechnet, soll so verändert werden, dass ein Benutzer den Nettobetrag über ein Formular eingeben kann (s. Abbildung). Außerdem kann er über eine Auswahlliste den Mehrwertsteuersatz wählen. Nach Absenden des Formulars findet die Umrechnung statt, und das Ergebnis wird ausgegeben.

```
// Programm

function brutto2($netto, $mwstSatz){
    return $netto * (100 + $mwstSatz) / 100;
}
$betrag = 23;
$bruttowert = brutto2($betrag, 7);
echo "$betrag ergibt $bruttowert inkl. MWSt<br />\n";

$bruttowert2 = brutto2($betrag, 19);
echo "$betrag ergibt $bruttowert2 inkl. MWSt<br />\n";
```

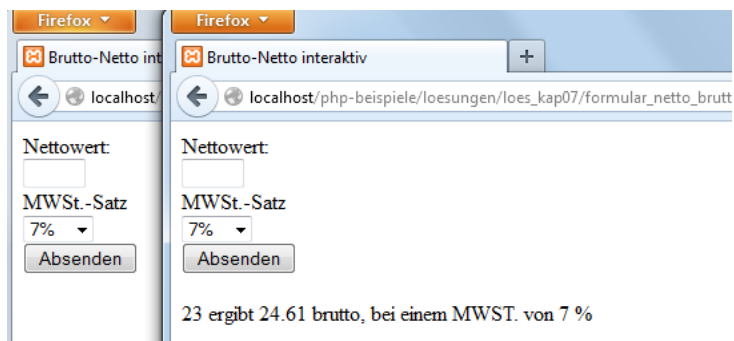


Abbildung: Brutto-Netto-Rechner interaktiv

Lösung:

Die einfachste Variante des Formulars zur Eingabe für den Betrag, zu dem der Bruttowert berechnet werden soll, sieht folgendermaßen aus:

```
// formular_netto_brutto.php
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
```

```

<title>Brutto-Netto interaktiv</title>
</head>
<body>
    <form method="get" action="<?php
        htmlspecialchars($_SERVER['PHP_SELF']); ?>">
        Nettowert:<br />
        <input type="text" name="wert" size="3" /><br />
        MWSt.-Satz<br />
        <select name="faktor">
            <option value="7">7%</option>
            <option value="19">19%</option>
        </select> <br />
        <input type="submit" value="Absenden" />
    </form>
    <?php
        function bruttoberechnen($wert, $faktor)
        {
            return $wert * (100 + $faktor) / 100;
        }
        if (!empty($_GET["wert"])) {
            $netto = $_GET["wert"];
            $faktor = $_GET["faktor"];
            $ergebnis = bruttoberechnen ($netto, $faktor);
            echo "<br />$netto ergibt $ergebnis brutto, bei einem MWST.
                von $faktor %";
        }
    ?>
</body>
</html>

```

Beim Formular verwenden wir für den Faktor für die Berechnung folgende option-Felder innerhalb des select-Elements:

```

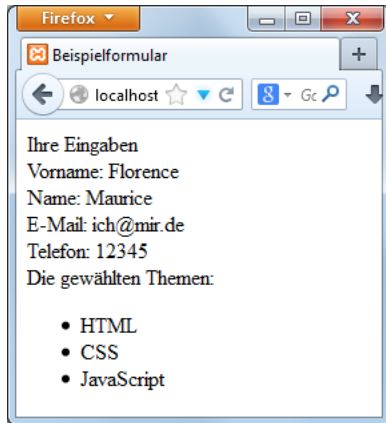
<option value="7">7%</option>
<option value="19">19%</option>

```

Dadurch wird 7% oder 19% angezeigt, jedoch der Wert 7 oder 19 übertragen (das, was bei value steht), und damit lässt sich rechnen.

Aufgabe 10.4:

Ergänzen Sie beim Formular aus Aufgabe 10.2 Checkboxes, beispielsweise zur Auswahl von interessanten Themen. Lassen Sie die gewählten Themen dann als ungeordnete Liste ausgeben!



Lösung:

Die für die Checkboxes ergänzten Teile sind im folgenden Beispiel hervorgehoben:

```
// formular_mit_checkbox.php
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<title>Beispielformular</title>
</head>
<body>
    <?php
        if (!isset($_GET["vorname"])) {
            ?>
            <form action="<?php echo
                htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="get">
                Ihr Vorname: <br />
                <input type="text" name="vorname" size="20" maxlength="30" />
                <br />
                Ihr Nachname: <br />
                <input type="text" name="nachname" size="20" maxlength="30" />
                <br />
                E-Mail: <br />
                <input type="text" name="email" size="20" maxlength="30" />
                <br />
                Telefon: <br />
                <input type="text" name="telefon" size="20" maxlength="30" />
                <br />
                Interessengebiete: <br />
                <input type="checkbox" name="thema[]" value="HTML" />HTML
                <input type="checkbox" name="thema[]" value="CSS" />CSS
                <input type="checkbox" name="thema[]" value="JavaScript"
                    />JavaScript
                <input type="checkbox" name="thema[]" value="PHP" />PHP<br />
                <input type="submit" value="Abschicken" />
            </form>
        }
    </?php>

```

```
</form>
<?php
    } else {
        echo "Ihre Eingaben<br />\n"
            . "Vorname: " . htmlspecialchars($_GET["vorname"])
            . "<br />\n Name: " . htmlspecialchars($_GET["nachname"])
            . "<br />\n E-Mail: " . htmlspecialchars($_GET["email"])
            . "<br />\n Telefon: " . htmlspecialchars($_GET["telefon"])
            . "<br />\n";
        if (isset($_GET["thema"]) && is_array($_GET["thema"])){
            echo "Die gewählten Themen: <br />\n ";
            echo "<ul>\n";
            foreach($_GET["thema"] as $th) {
                echo "<li>" . htmlspecialchars($th) . "</li>\n";
            }
            echo "</ul>\n";
        }
    }
    ?>
</body>
</html>
```