

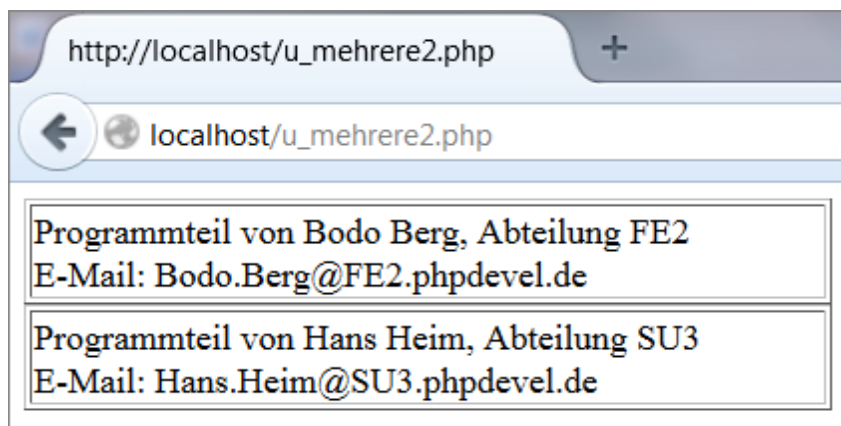
# Lösungen zu Übungsaufgaben

Quelle: Maurice (2015); Theis (2016)

## Kapitel: "7. Funktionen"

### Aufgabe 7.1:

Erstellen Sie eine Funktion *vermerk()*, die einen Entwicklervermerk erzeugt. Die Funktion soll von verschiedenen Entwicklern genutzt werden können. Vorname, Nachname und Abteilung werden als Parameter an die Funktion übergeben. Jedes Mal, wenn die Funktion aufgerufen wird, erscheint eine Ausgabezeile mit diesen Informationen und der E-Mail-Adresse. Die E-Mail-Adresse setzt sich wie folgt zusammen: *vorname.nachname@abteilung.phpdevel.de*. Testen Sie Ihre Funktion mit einem geeigneten Programm, in dem die Funktion mehrmals mit verschiedenen Informationen aufgerufen wird (Datei *u\_mehrere2.php*). Eine mögliche Ausgabe sehen Sie im Folgenden:

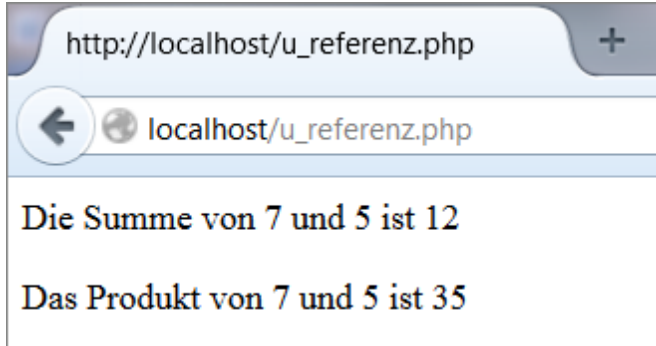


### Lösung:

```
<html>
<head>
<?php
    function vermerk($vor, $nach, $abt){
        echo "<table width='30%' border='1'>";
        echo "<tr><td>"
            . "Programmteil von $vor $nach, Abteilung $abt<br />"
            . "E-Mail: $vor.$nach@$abt.phpdevel.de"
            . "</td></tr>";
        echo "</table>";
    }
?>
</head>
<body>
<?php
    vermerk("Bodo", "Berg", "FE2");
    vermerk("Hans", "Heim", "SU3");
?>
</body>
</html>
```

**Aufgabe 7.2:**

Schreiben Sie ein PHP-Programm (Datei *u\_referenz.php*) mit einer Funktion *rechne()*. Dieser Funktion werden zwei Zahlen übergeben. Sie soll zwei Ergebnisse über die Parameterliste zurückliefern: zum einen die Summe der beiden übergebenen Zahlen und zum anderen das Produkt der beiden übergebenen Zahlen. Alle beteiligten Zahlen sollen im Hauptteil des Programms, also außerhalb der Funktion, ausgegeben werden. Verwenden Sie zur Übergabe die zweite Methode (Call-by-Reference). Nach einem Funktionsaufruf mit den Parametern 5 und 7 und der anschließenden Ausgabe erscheint eine Ausgabe wie im Folgenden.

**Lösung:**

```
<html>
<head>
<?php
    function rechne($a, $b, &$amp;su, &$amp;pr){
        $su = $a + $b;
        $pr = $a * $b;
    }
?>
</head>
<body>
<?php
    $x = 7;
    $y = 5;
    rechne($x, $y, $summe, $produkt);
    echo "<p>Die Summe von $x und $y ist $summe</p>";
    echo "<p>Das Produkt von $x und $y ist $produkt</p>";
?>
</body>
</html>
```

### Aufgabe 7.3:

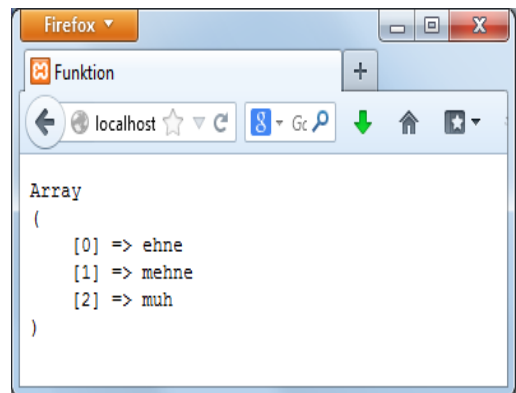
Arrays lassen sich gut mit `print_r()` ausgeben. Die hilfreichen Einrückungen sieht man allerdings nur im HTML-Quellcode der Seite oder wenn man um den Aufruf von `print_r()` `<pre>` und `</pre>` ergänzt:

```
echo "<pre>";
print_r($antworten);
echo "</pre>";
```

Schreiben Sie eine Funktion, die ein Array entgegennimmt und dieses Array mit `print_r()` und `<pre>` behandelt ausgibt. Testen Sie dann Ihre Funktion mit einem Array.

### Lösung:

```
function arrayinfo($arr) {
    echo "<pre>";
    print_r($arr);
    echo "</pre>";
}
$arr = array("ehne", "mehne", "muh");
arrayinfo($arr);
```



### Aufgabe 7.4:

Mit welchem regulären Ausdruck könnte man überprüfen, dass ein Dateiname ein JPEG-Bild ist? (Die übliche Endung für JPEG-Bilder kann sowohl `jpeg` als auch `jpg` sein!)

### Lösungshinweis:

Mit folgendem Code können Sie prüfen, ob etwas eine JPEG-Datei ist:

```
$datei = "blume.jpg";
if (preg_match("/\.(jpe?g$/", $datei)) {
    echo "passt";
}
```

Sehen wir uns das Muster noch einmal genau an:

`/\.(jpe?g$/`

`/` und `/` sind die Begrenzer.

- Am Anfang wird nach einem Punkt gesucht, der üblicherweise vor der Endung steht – denn uns interessieren ja nur Dateien mit der Endung `jpeg` oder `jpg`. Wenn `jpeg` an einer anderen Stelle steht, ist das für uns nicht relevant. Da der Punkt ein Sonderzeichen ist – er steht ja normalerweise für ein beliebiges Zeichen – müssen wir ihn über `\.` maskieren.
- Dann folgen die Zeichen `jpe`, wobei das `e` fakultativ ist – was durch das Fragezeichen (`?` steht für ein- oder keinmal.) gekennzeichnet wird.
- Nach dem `g` steht das `$`-Zeichen, um zu kennzeichnen, dass unser String mit der angegebenen Zeichenfolge enden muss.

**Aufgabe 7.5:**

Verwenden Sie `date()`, um die Meldung »Schönes Wochenende« am Samstag und Sonntag ausgeben zu lassen. An anderen Wochentagen soll hingegen »Gute Woche« erscheinen.

Lösungshinweis:

`date("w")` gibt den Wochentag als Zahl zwischen 0 (Sonntag) und 6 (Samstag) zurück. Das kann man in einer `if`-Anweisung benutzen.

```
date_default_timezone_set("Europe/Berlin");
$tag = date("w");
if ($tag == 0 || $tag == 6) {
    echo "Schönes Wochenende";
}
else {
    echo "Gute Woche";
}
```