



PHP 02 LÖSUNGEN

KLEINE ÜBUNG

- T Schreibe ein Programm, das ermittelt, ob die Warenbestände für ein bestimmtes Produkt auf 0 gefallen sind und gegebenenfalls eine Meldung ausgibt, dass der Artikel nicht mehr verfügbar ist.

```
<?php
$bestand = 4;

if($bestand == 0) {
    print "Der gewünschte Artikel ist leider nicht mehr verfügbar.";
}
```

KLEINE ÜBUNG

T Erstelle zunächst ein Array, in dem sich ein Produkt mit dem zugehörigen Preis und die Anzahl der verfügbaren Artikel befindet.

Schreibe nun ein Programm, das:

Den Käufer darüber informiert, dass der Artikel nicht verfügbar ist, falls der Warenbestand auf 0 steht.

Ausgibt, dass das Produkt versandkostenfrei geliefert wird, falls mindestens ein Artikel vorrätig ist und falls der Preis bei mindestens 20 Euro liegt

Anzeigt, dass für die Lieferung 5 Euro Versandkosten anfallen, falls der Artikel verfügbar ist, der Preis jedoch unter 20 Euro liegt.

```
<?php

$produkt = array(
    'Produktname' => "Semmel",
    'Preis' => 15,
    'Anzahl' => 3
);

if($produkt['Anzahl'] == 0) {
    print "Das Produkt ".$produkt['Produktname']." ist leider nicht mehr verfügbar.";
} elseif($produkt['Preis'] >= 20) {
    print "Das Produkt ".$produkt['Produktname']." ist verfügbar und wird versandkostenfrei geliefert.";
} else {
    print "Das Produkt ".$produkt['Produktname']." ist verfügbar. Für die Lieferung fallen 5 € an";
}
```

KLEINE ÜBUNG

- T Erstelle drei Programme, die von 1 bis 10 zählen, verwende hierfür jeweils einmal eine while-, eine for- und eine do-while-Schleife

```
<?php

$i = 0;
while ($i < 10) {
    $i++;
    print $i."<br>\n";
}

for ($i = 0; $i < 10; $i++) {
    print ($i + 1) . "<br>\n";
}

$i = 0;
do {
    print ($i+1)."<br>\n";
    $i++;
} while ($i < 10);
```

- T Erstelle ein zusammengesetztes Array für das Sortiment eines Obst- und Gemüsehändlers mit Äpfel, Birnen, Tomaten und Zucchini. Die erste Ebene soll einen numerischen Index haben. Die zweite soll hingegen assoziativ sein (mit den Bezeichnungen Produkt, Preis und Sonderangebot). Bei der Angabe zum Produkt handelt es sich um eine Zeichenkette mit der entsprechenden Obst- oder Gemüsesorte, beim Preis um eine Zahl und die Angabe Sonderangebot soll eine boolesche Variable sein.
- Erstelle eine foreach-Schleife, die den Produktnamen und den Preis angibt. Sollte es sich dabei um ein Sonderangebot handeln, soll vor diesen Angaben „Achtung Sonderangebot!“ erscheinen.

```
<?php

$produkt[0]['Produktname'] = "Apfel";
$produkt[0]['Preis'] = 1.99;
$produkt[0]['Sonderangebot'] = false;
$produkt[1]['Produktname'] = "Birne";
$produkt[1]['Preis'] = 0.99;
$produkt[1]['Sonderangebot'] = true;
$produkt[2]['Produktname'] = "Tomate";
$produkt[2]['Preis'] = 2.99;
$produkt[2]['Sonderangebot'] = true;
$produkt[3]['Produktname'] = "Zucchini";
$produkt[3]['Preis'] = 1.99;
$produkt[3]['Sonderangebot'] = false;

foreach($produkt as $item) {
    if($item['Sonderangebot']) {
        print "Achtung Sonderangebot<br>\n";
    }
    print $item['Produktname']. "<br>\n";
    print $item['Preis']. "<br>\n";
}
```

- T Ergänze das vorherige Beispiel um die Ausgabe des entsprechenden Feldnamens. Das Sonderangebot kannst du weg lassen.

```
<?php

$produkt[0]['Produktname'] = "Apfel";
$produkt[0]['Preis'] = 1.99;

$produkt[1]['Produktname'] = "Birne";
$produkt[1]['Preis'] = 0.99;

$produkt[2]['Produktname'] = "Tomate";
$produkt[2]['Preis'] = 2.99;

$produkt[3]['Produktname'] = "Zucchini";
$produkt[3]['Preis'] = 1.99;

foreach($produkt as $item) {
    foreach ($item as $feldname => $inhalt) {
        print $feldname .": ". $inhalt."<br>\n";
    }
    print "<br>\n";
}
```

- T Erstelle eine Funktion, die die ersten zehn Potenzen einer Zahl berechnet. Dazu kommt der Operator ** zum Einsatz. Diesem wird die Basis vorangestellt und darauf folgt der Exponent. Das Hauptprogramm soll dabei den Ausgangswert übermitteln. Die Rückgabe erfolgt durch ein Array. Die Ausgabe des Arrays erfolgt im Hauptprogramm.

```
<?php

function potenz($wert) {
    $ergebnis = array();
    for ($i = 0; $i < 10; $i++) {
        $ergebnis[$i] = $wert**($i+1);
    }
    return $ergebnis;
}

$potenzen = potenz(2);
foreach ($potenzen as $ausgabe) {
    print $ausgabe . "<br>";
}
```

T Führe nun auch die Ausgabe des Arrays in einer eigenen Funktion durch. Lagere beide Funktionen in eine separate Datei aus und binde diese in das Hauptprogramm ein

T Hauptprogramm

```
<?php
include("function.php");
ausgabe(potenz(2));
```

T File function.php

```
<?php
function potenz($wert) {
    $ergebnis = array();
    for ($i = 0; $i < 10; $i++) {
        $ergebnis[$i] = $wert**($i+1);
    }
    return $ergebnis;
}

function ausgabe($potenzen) {
    foreach ($potenzen as $ausgabe) {
        print $ausgabe . "<br>";
    }
}
```


- T Suche in der Online-Dokumentation eine Funktion, um die Quadratwurzel einer Zahl zu bestimmen und eine weitere Funktion, mit der du eine Zufallsvariable generierst. Informiere dich über die Anwendung dieser beiden Funktionen. Erstelle nun ein Programm, dass die Wurzel eines zufälligen Werts berechnet und sowohl den Ausgangswert als auch das Ergebnis ausgibt.

```
<?php
$zufallszahl = rand();
$wurzel = sqrt($zufallszahl);
print "Die Wurzel aus " . $zufallszahl . " ist " . $wurzel . "<br>\n";
```



ENDE

QUELLE: PHP & MYSQL FÜR EINSTEIGER
ISBN: 978-3-96645-009-6