Assoziative Arrays

In assoziativen Arrays wird als *key* statt der Index-Nummer ein ganz eigenes Schema erzeugt. Dieses darf man sich (fast) frei ausdenken. Man kann bei diesem „Ersatz-Schlüssel“ einfache Strings aber auch Zahlen benutzen, um es sich bei der Abfrage zu erleichtern. Der *key* wird dann in Gänsefüßchen notiert.

Beispiel:

```
   $SSVReutlingen["torwart"] = "Gühring";
   $SSVReutlingen["mittelfeld1"] = "Rill";
   ...
```

Damit werden Arrays noch wertvoller, da jedes Element eines Arrays nun zwei miteinander verknüpfte Informationen enthält, im Beispiel oben den Namen des Spielers (wie bisher) aber zusätzlich noch die Position.

Ein weiteres Beispiel, wie ein assoziatives Array definiert werden kann:

```
$wochentage = array(
"so" => "Sonntag",
"mo" => "Montag",
"di" => "Dienstag",
"mi" => "Mittwoch",
"do" => "Donnerstag",
"fr" => "Freitag",
"sa" => "Samstag");

echo $wochentage["mo"];
```

Die foreach Schleife

Für Arrays gibt es den speziellen Schleifentyp foreach. Während die Verwendung bei "normalen" Arrays durch eine for-Schleife ersetzt werden kann, ist es bei assoziativen Arrays nicht mehr möglich, mit einer Zählschleife alle Elemente auszugeben[1)].

Mit diesem Schleifentyp werden stets alle einzelnen Arrayelemente eines Arrays durchlaufen. Eine foreach-Schleife beginnt mit dem Schlüsselwort foreach. Dann folgt nach der öffnenden Klammer das Array bzw. die Variable, die das Array enthält, welches durchlaufen werden soll. Danach folgt das Schlüsselwort as und eine neue Variable. In dieser Variable wird für jeden Schleifendurchlauf der neue Wert des nächsten Arrayelements gespeichert. Nach der folgenden schließenden Klammer beginnt der Schleifenrumpf.

```
  foreach ($hauptstadt as $stadt) {
    echo $stadt."<br />";
  }
  // gibt nacheinander die Werte aus dem Array aus
```

Mit der foreach Schleife kann man auch gleichzeitig auf Schlüssel und Werte zugreifen:

```
   foreach ($hauptstadt as $key => $value) {
    print '$hauptstadt['.$key.']=' . $value;
    print " ----> Schlüssel: <tt>" . $key . "</tt>  Wert: <tt>" . $value .
"</tt><br />";
  }
```

(A1)

Schreibe eine hauptstadt.php-Datei, in der du ein Array `$hauptstadt` mit 10 verschiedenen Hauptstädten Europas deklarierst. Verwende als *key* die im Web gebräuchliche Länderkennung (evtl. recherchieren). Gib anschließend die Länder mit ihren Hauptstädten aus.

Hinweis:

Bei der Ausgabe von Arrays in Verbindung mit einem Text kann man verschiedene Varianten wählen. Eine davon lautet:

```
   echo "<p>Die Hauptstadt von Frankreich ist $hauptstadt['fr'] <br /></p>";
```

Beachte die unterschiedlichen Anführungszeichen!

(A2)

Gegeben sei folgendes Array:

```
$bildschirmtyp = array('flachbildschirm' => "billig und gut",
'roehrenbildschirm' => "sehr billig, aber schwer", 'minibildschirm' =>
"billig und klein", 'plasmabildschirm' => "teuer, gut und schwer");
```

Erzeuge unter Verwendung des Arrays folgende Ausgabe:

```
Flachbildschirme sind billig und gut, Minibildschirme billig und klein.
Plasmabildschirme sind teuer, gut und schwer.
```

Fügen zu dem Array noch den Astronautenbildschirm hinzu, der riesig und teuer ist. Erzeugen die entsprechende Ausgabe.

(A3)

Gegeben sei folgender Code, der ein assoziatives Array definiert:

```
$typ['bungalow'] = "flach";
$typ['pyramide'] = "spitz";
$typ['hütte'] = "einfach";
$typ['hochhaus'] = "hoch";
$typ['wolkenkratzer'] = "sehr hoch";
```

Gib eine Liste der folgenden Form aus:

```
Haustyp: bungalow, Dach: flach
```

Zum knobeln: Wie kann man erreichen, dass der erste Buchstabe des Haustyps immer groß geschrieben wird?

Wie könnte man das Array alternativ definieren? Schreibe den Code um und teste!

(A4) Geheimschrift

Der Heimlichkeit wegen benutzen wir einen geheimen Code für Wochentage. Dabei ersetzen wir jeweils den Wochentag durch ein anderes Substantiv, nämlich

```
Montag/Baumhaus
Dienstag/Ventilator
Mittwoch/Unterricht
Donnerstag/Behördenwegweiser
Freitag/Fahrradschloss
Samstag/Kettenschaltung
Sonntag/Montag.
```

Außerdem ersetzen wir einige Schulfächer:

```
Wirtschaftsinformatik/Freude
Deutsch/Brezeln
Mathematik/Kaugummi
```

(1) Erstelle ein assoziatives Array namens `$geheimcode`, in dem der zu verschlüsselnde

Wochentag/Schulfach als Schlüssel fungiert und der Geheimcode als jeweiliger Wert ($geheimcode['Montag'] soll also den Wert Baumhaus besitzen).

(2) Schreibe ein Programm, um folgende Mitteilung zu entschlüsseln:

```
Letzten Behördenwegweiser hatten wir Kaugummi, das war anstrengend. Zum Glück blieb dann nur noch der Fahrradschloss. Kettenschaltung und Montag konnte ich mich nicht richtig entspannen, denn Baumhaus erwartet uns schon wieder Freude. Naja ... besser als Brezeln.
```

Ersetze dabei einfach die codierten Wörter durch die entsprechende Arrayvariable.

Tipp: Du kannst einen Text in ein Array aus den einzelnen Wörtern umwandeln:

```
$worte = explode(" ", $text);
```

Nun kannst du mit einer `foreach` Schleife für jedes Wort überprüfen, ob es eine entprechende Codierung gibt: Die Funktion `array_key_exists($key, $array)` überprüft, ob der Schlüssel `$key` im `$array` existiert. So kannst du, je nach Ergebnis dieses Tests entweder einfach das Wort oder das decodierte Wort ausgeben.

(3) Schreibe wie oben ein Programm, das folgenden Text in Ihre Geheimschrift übersetzt:

```
Am Freitag haben wir Mathematik, am Dienstag ebenfalls. Mein Lieblingsfach ist Deutsch, das wir leider nur am Mittwoch haben. Am meisten freue ich mich immer darauf, wenn ich am Sonntag Wirtschaftsinformatik lernen darf.
```

[1)]
Warum nicht?

From:
https://info-bw.de/ -

Permanent link:
https://info-bw.de/faecher:informatik:oberstufe:php:assoziative_arrays:start

Last update: **28.04.2021 15:40**