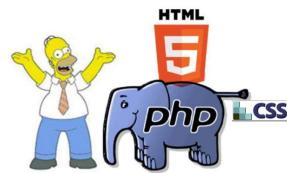


Skript

Grundlagen der Programmierung

Klasse 10 FI 5



Alan Kay

By the time I got to school, I had already read a couple hundred books.

I knew in the first grade that they were lying to me because I had already been exposed to other points of view.

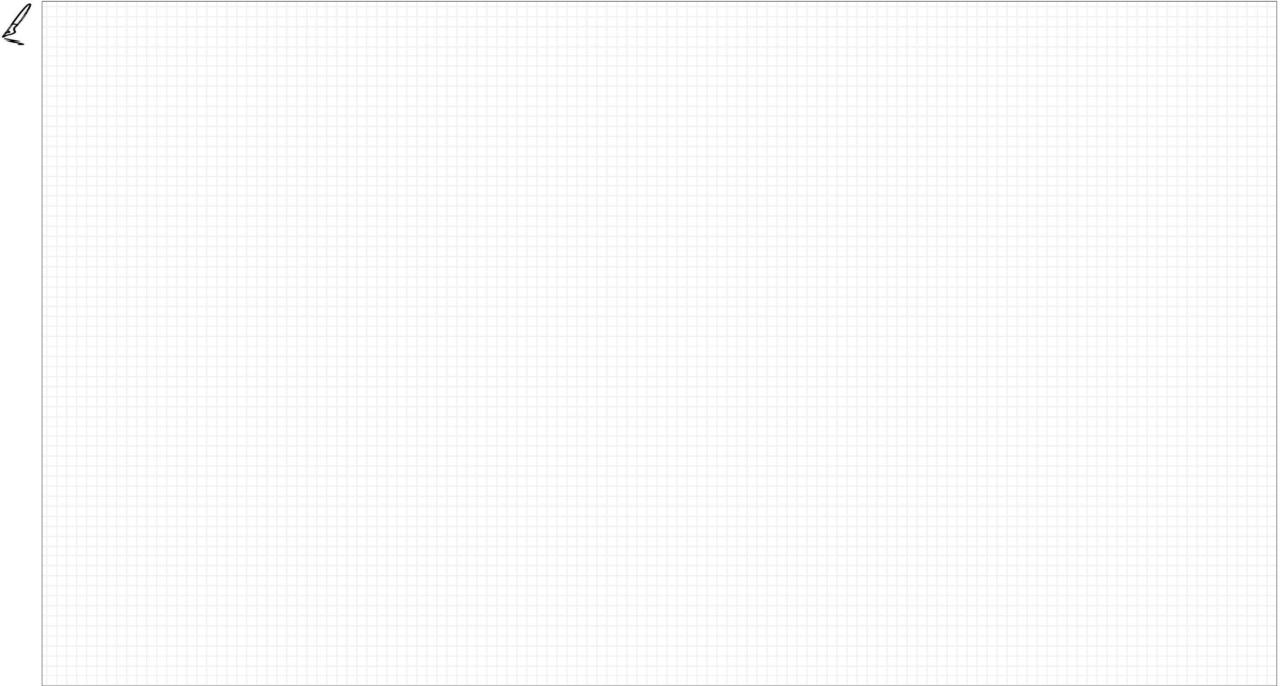
School is basically about one point of view — the one the teacher has or the textbooks have.

Inhaltsverzeichnis

| | | |
|-----|---|----|
| 1. | Einführung | 1 |
| 1.1 | Big Picture des Faches Anwendungsentwicklung und Programmierung | 1 |
| 1.2 | Entwicklung von Programmiersprachen | 2 |
| 1.3 | Übersetzungsarten | 3 |
| 1.4 | Programmiersprachenuniversum | 4 |
| 1.5 | Eigene Programmierkompetenz | 5 |
| 1.6 | Zusammenspiel von HTML, CSS, PHP, SQL und JavaScript | 5 |
| 1.7 | Hello World..... | 7 |
| 2. | Grundlagen PHP..... | 8 |
| 2.1 | Einführung in die Entwicklungsumgebung | 8 |
| 2.2 | Variablen und Rechenoperationen..... | 9 |
| 2.3 | Arrays..... | 17 |
| 2.4 | Mehrdimensionales Arrays..... | 23 |
| 2.5 | Kontrollstrukturen | 28 |
| 2.6 | Schleifen | 35 |
| 2.7 | Funktionen..... | 37 |
| 3. | Arbeiten mit Dateien | 45 |
| 4. | Objektorientierte Programmierung..... | 47 |
| 4.1 | Einführung in die Objektorientierung..... | 47 |
| 4.2 | Klassendiagramm..... | 58 |
| 4.3 | Vererbung..... | 66 |
| 4.4 | Abstrakte Klassen | 67 |
| 4.5 | Interface | 67 |
| 4.6 | Polymorphie | 69 |
| 5. | Allerlei..... | 69 |
| 6. | Entwurfsmuster..... | 70 |
| | Literaturverzeichnis..... | 79 |

1. Einführung

1.1 Big Picture des Faches Anwendungsentwicklung und Programmierung

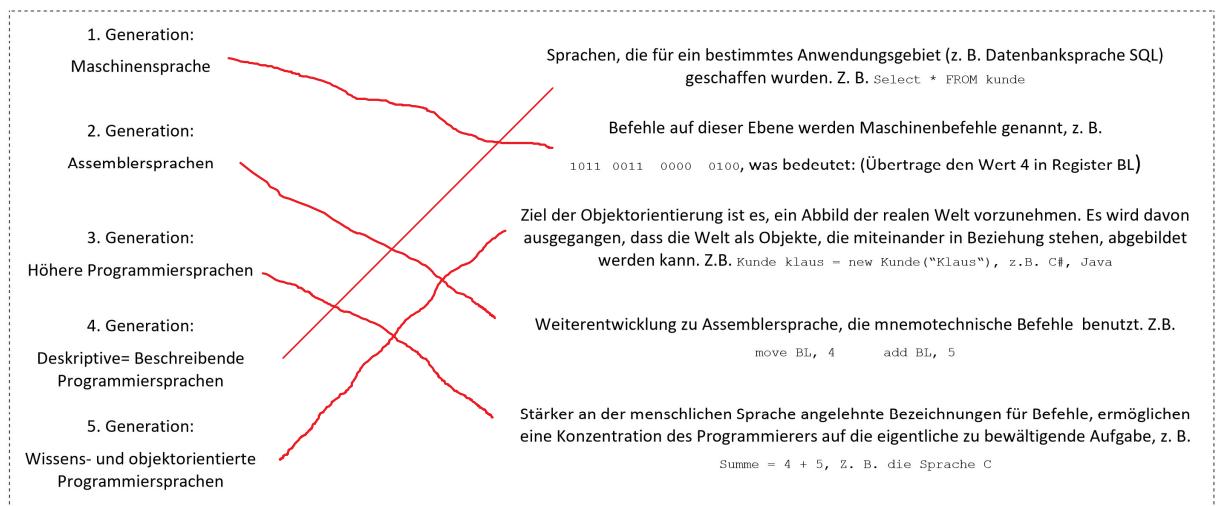


1

1.2 Entwicklung von Programmiersprachen

 Programmiersprachen haben sich im Verlauf der Zeit stetig weiterentwickelt. Allen Programmiersprachen ist jedoch gemein, dass diese letztlich in MaschinenSprache übersetzt werden müssen, so dass der Computer, genauer betrachtet das Rechenwerk in Kooperation mit dem Steuerwerk, die gewünschten Befehle verstehen und ausführen können.

Leider sind folgende Generationen von Programmiersprachen mit den jeweiligen Erklärungen durcheinandergeraten. Ihre Aufgabe ist es daher, die jeweilige Generation mit der jeweils passenden Erklärung durch eine Linie zu verbinden.



2

1.3 Übersetzungsarten

Arbeitsauftrag: Schauen Sie sich nachfolgendes Lernvideo an und beantworten Sie dabei folgende Fragen:

<https://youtu.be/F1GLYZ7fhvw>



Um welche Fragestellung handelt es sich hierbei überhaupt? Um was geht es?



Wie wird ein Programm überhaupt vom Computer ausgeführt

Beschriften Sie die beiden nachfolgenden Elemente und beschreiben Sie deren Eigenschaften.



Step-by-Step

Führt jede Anweisung hintereinander aus



Übersetzt den kompletten Code in die Maschinensprache

der dann vom Prozessor ausgeführt werden kann

+ keine Vorbereitungszeit, da jede Anweisung sofort ausgeführt werden kann

+ auf Fehler kann schneller reagiert werden
code-run-test

+

- Langsamer ??

-

+ schnellere Ausführung

-

- Längere Vorbereitungszeit
- Auf Fehler kann während der Ausführung nicht reagiert werden
- Code-compile-run-Zyklus
- prozessorabhängig

Finden Sie nun je Übersetzungsart zwei Programmiersprachen, die diese Übersetzungsart verwenden.



Php, python, javascript, perl...

c, c++, x++, delphi, go

Logischer Fehler
Compiler-Fehler

Finden Sie nun auch noch mindestens eine Programmiersprache, die einen zweifachen Übersetzungsprozess hat.



1.4 Programmiersprachenuniversum



Arbeitsauftrag 1: Erstellen Sie in Ihrer Gruppe auf einer PowerPoint Folien eine Übersicht über die von Ihnen gewählte Programmiersprache. Markieren Sie Ihre Wahl, so dass die nächste Gruppe erkennen kann, dass Sie diese Sprache bereits gewählt haben.

Die Folien sollen einem Plakat entsprechen und einerseits mögliche Einsatzszenarien darstellen sowie Vor- und Nachteile nennen. Gerne können Sie auch ein kleines Codefragment oder einen ausgedruckten Screenshot in Ihre Folien integrieren. Achten Sie hierbei darauf, die Daten grafisch ansprechend, gerne auch aufgelockert, darzustellen. Bereiten Sie sich zudem darauf vor, Ihre Folien Ihren Mitschülern später auch erklären zu können.

Drucken Sie Ihre zwei Folien anschließend aus und heften Sie es diese an der Wand/Tafel bei dieser Station mit einem Magnet an. Versuchen Sie, diese Aufgabe innerhalb von ca. 30 Minuten zu erledigen.

Sprachen/Techniken

- Java
- C#
- C++
- Java Script
- Typescript
- XML
- ColdFusion
- Ruby
- Python
- PHP
- SQL
- HTML
- CSS
- GO
- ABAP
- ...

 **Arbeitsauftrag 2:** Beantworten Sie nun schriftlich und in Stichpunkten auf dem an der Station ausliegenden Plakat die Frage, welche Sprache Ihrer Meinung nach die beste Programmiersprache ist. Sollte die von Ihnen präferierte Sprache schon auf dem Plakat genannt sein, ergänzen Sie die bereits dort stehende Aufzählung mit einem passenden Kommentar.



Arbeitsauftrag 3: Schauen Sie zudem nach, ob Sie Ihre gewählte Sprache im Tiboe Index wiederfinden. Natürlich recherchieren Sie dabei, was der Tiboe Index überhaupt ist und notieren Ihre Erkenntnis im nachfolgenden Feld.

<https://www.tiobe.com/tiobe-index/>



| | |
|---|----------------------|
|  | <input type="text"/> |
|---|----------------------|

1.5 Eigene Programmierkompetenz

Arbeitsauftrag: Schätzen Sie nun Ihre aktuelle eigenen Programmierkompetenz ein. Benutzen Sie hierzu folgende Kompetenzstufe (0=keine Kenntnisse, 1=wenig Kenntnisse, 2=Grundlagenkenntnisse, 3=Erweiterte Grundlagen, 4= Profikenntnisse) sind vorhanden:



| Programmiersprache | aktuell | 31.12 dieses Jahres | 31.12 nächsten Jahres | 31.12 übernächsten Jahres |
|--------------------|---------|---------------------|-----------------------|---------------------------|
| C | | | | |
| C++ | | | | |
| C# | | | | |
| Java | | | | |
| PHP | | | | |
| Java Script | | | | |
| CSS | | | | |
| HTML | | | | |
| Ruby | | | | |
| ASP.NET | | | | |
| SQL | | | | |
| | | | | |

Welche Ziele im Programmierumfeld haben Sie noch?



Wie wollen Sie diese Ziele erreichen?



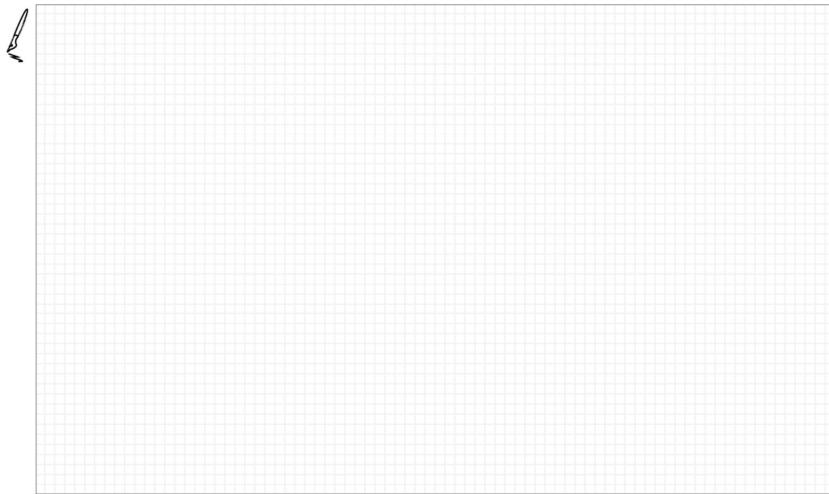
- Arbeitsauftrag:** Zeigen Sie einen Mitschüler oder Ihren Lehrer Ihre Planung und bitten Sie diesen um Feedback zu Ihren Plänen.

- Feedback erhalten**

Arbeitsauftrag: Nutzen Sie die bereit gestellten Puzzleteile, um das klassische, etwas vereinfachte

Zusammenspiel von HTML, CSS, PHP, SQL und JavaScript

darzustellen. Übertragen Sie anschließend das fertige Puzzlebild auf folgende Skizze



1. Benutzer gibt Ziel-URL im Browser ein.
2. Browser fordert unter Nutzung des HTTP-Protokolls(GET-Befehl) eine Webseite bei einem Webserver an.
3. Webserver nimmt die Anfrage über seinen Port 80 wahr und führt mittels des PHP Interpreters das angeforderte PHP-Skript aus.
4. PHP-Skript fordert vom Datenbankserver mittels SQL-Befehle Daten aus der Datenbank an.
5. Datenbankserver gibt die angeforderten Daten an Webserver zurück.
6. PHP-Skript verarbeitet die empfangenen Daten vom Webserver.
7. Webserver sendet die vorbereitete Webseite mittels des HTTP-Protokolls(Response).
8. Browser visualisiert die Webseite, indem dieser HTML, CSS und JavaScript ausführt.
9. Benutzer bekommt das Resultat angezeigt und kann wiederum reagieren.

Arbeitsauftrag: Schauen Sie sich nachfolgendes Lernvideo an und beantworten Sie dabei stichpunktartig folgende Fragen auf der Rückseite dieses Blattes.

<https://academind.com/learn/web-dev/frontend-vs-backend-vs-fullstack-development/>



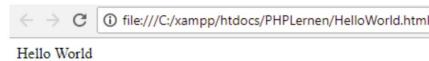
1. Mit welchen Techniken beschäftigt sich ein Frontendentwickler, mit welchen beschäftigt sich ein Backendentwickler?
2. Welcher Bereich pflegt den Kontakt zur Datenbank?
3. Welcher Bereich ist für die Validierung der Nutzerangaben zuständig?
4. Was bedeutet der zuletzt genannte Begriff „Jack of all Trades“?



1.7 Hello World

Arbeitsauftrag: Statisch Hello World sagen

Öffnen Sie den Editor Visual Studio Code¹ und erzeugen Sie eine neue Datei, in die Sie schlicht und einfach „Hello World“ schreiben. Speichern Sie diese Datei im folgenden Verzeichnis
C:\xampp\htdocs\[IHRNAME]\PHPLernen ab. Gehen Sie nun in das Verzeichnis und öffnen Sie die Datei mit dem Browser. Die Ausgabe könnte wie folgt aussehen:



Wie finden Sie diese statische Variante von Hello World? Markieren Sie den Ausdruck von Homer, der Ihre Gefühle am besten zum Ausdruck bringt.



O
 Informationstext:

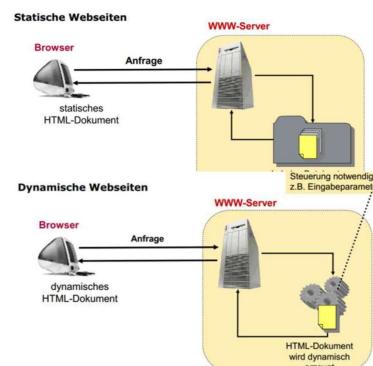
Während zu Beginn der Evolution des Webs statische Webseiten, die vorab auf Webservern abgelegten worden waren, Informationen statisch bereitstellten, passten dynamisch erzeugte Webseiten sich später individuell an den Informationsbedarf des Anfragenden an.²



O



O



Gängige Praxis ist mittlerweile die dynamische Erzeugung von HTML Code, der in Abhängigkeit der Nutzeranfrage erzeugt wird. Neben einem Editor bräuchten wir in diesem Fall auch einen Webserver, der die Sprache auf dem Server übersetzt, der die Dynamisierung herstellt. Im schulischen Umfeld nutzen wir zu diesem Zweck XAMPP³. Starten Sie einmal XAMPP und dort den PHP-Server(Apache). Öffnen Sie anschließend den Browser und geben Sie <http://localhost> ein.

Notieren Sie auf der Rückseite dieses Blattes stichpunktartig, was Sie sehen.

¹ "Visual Studio Code ist ein Quelltext-Editor von Microsoft. Er dient vorrangig der Entwicklung von Webanwendungen und ermöglicht die Programmierung mit Programmier- und Auszeichnungssprachen wie Batch, C#, C++, Clojure, CoffeeScript, CSS, Dart, Dockerfile, F#, Go, Groovy, Handlebars.js, HTML, Ini, Jade, Java, JavaScript, JSON, Less, Lua, Makefile, Markdown, Objective-C, Perl, PowerShell, PHP, Python, R, Razor, Ruby, Rust, Sass, SQL, Swift, TypeScript, Visual Basic und XML, wobei durch Plugins fast jede Sprache unterstützt werden kann.", Vgl.: https://de.wikipedia.org/wiki/Visual_Studio_Code

² Quelle: https://openhpi.de/courses/7/wiki/willkommen-zu-woche-1?module_item_id=677

³ „XAMPP ist ein Programmepaket von freier Software – ursprünglich vorwiegend im Umfeld des LAMP-Systems. XAMPP ermöglicht das Installieren und Konfigurieren des Webservers Apache mit der Datenbank MariaDB bzw. SQLite und den Skriptsprachen Perl und PHP (mit PEAR). Das X steht hierbei für die verschiedenen Betriebssysteme, auf denen es eingesetzt werden kann.“ Vgl. <https://de.wikipedia.org/wiki/XAMPP>

2. Grundlagen PHP

Hinweis zur verwendeten Literatur: Die nachfolgend niedergeschriebene Theorie sowie die daran angelehnten Übungsaufgaben basieren auf unterschiedlichen Werken dritter Autoren, die je nach Themengebiet im unterschiedlichen Umfang als Quelle dieses Skriptes dienten. Welche Bücher konkret verwendet wurden, können Sie dem im Anhang befindlichen Literaturverzeichnis entnehmen.

Weithin könnten folgende Internetquellen für das schnelle digitale Recherchieren hilfreich sein:



| | |
|-----------------------|---|
| Dokumentation von PHP | http://de2.php.net/ |
| Forum inkl. Tutorials | http://www.selfphp.de/ |
| Tutorialseite | https://www.w3schools.com/pHP/ |

2.1 Einführung in die Entwicklungsumgebung

Mit welchem Editor Sie im schulischen Umfeld arbeiten wollen, bleibt letztlich Ihren persönlichen Geschmack überlassen. Nachfolgend erhalten Sie einen Link mit der Einführung in Visual Studio Code. Sollten Sie noch nicht vertraut mit diesem Editor sein, gleichsam jedoch Visual Studio Code nutzen wollen, sollten Sie sich das verlinkte Kurzvideo anschauen.



| | |
|----------------------------------|---|
| Einführung in Visual Studio Code | https://code.visualstudio.com/docs/introvideos/basics |
|----------------------------------|---|

Im weiteren Programmierkurs werden Sie sicherlich auch das Bedürfnis verspüren, eine Debugfunktion nutzen zu können. Hierzu sei zu gegebener Zeit folgender Videolink empfohlen.



| | |
|---|---|
| Debugging mit Visual Studio Code und XAMPP einrichten | https://youtu.be/lZHNADo-WNA |
|---|---|



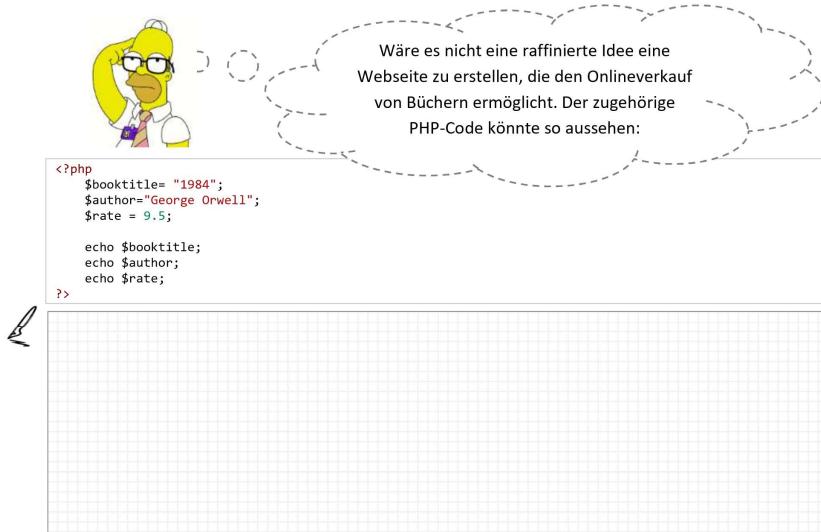
Übung

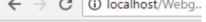
Erstellen Sie ein PHP-Skript, das beim Aufruf über den Browser mittels des Befehls `echo` eine der folgenden Figuren zeichnet.

```
*****
 *   *
 *   *
 *   *
 *****
 ***
```

2.2 Variablen und Rechenoperationen

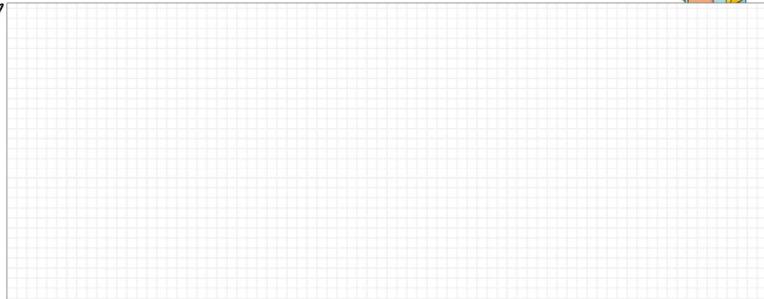
Wir reisen in das Jahr 1994 zurück. Im Rahmen des sich sukzessive entwickelnden Word Wide Webs hat ein junger, vielversprechender Hoffnungsträger folgende scheinbar geniale Idee:



| | |
|--|---|
| <p>Gerne zeigen wir Ihnen die zur Verfügung stehenden Bücher. Um Sie persönlich ansprechen zu können, bitte ich Sie jedoch zuerst Ihren Namen anzugeben.</p> <p><input type="text"/> Vorname <input type="text"/> Nachname <input type="button" value="Senden"/> <input type="button" value="Zurücksetzen"/></p> | <pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title></title> </head> <body> <p>Gerne zeigen wir Ihnen die zur Verfügung stehenden Bücher. Um Sie persönlich ansprechen zu können, bitte ich Sie jedoch zuerst Ihren Namen anzugeben.</p> <form action = "VariablenStoreMitParam.php" method = "post"> <p><input name = "firstname" /> Vorname</p> <p><input name = "lastname" /> Nachname</p> <p><input type = "submit" /></p> </form> </body> </html></pre> |
| <p> Guten Tag, Christoph Zobel Buch des Tages: 1984</p> | <pre><?php echo "Guten Tag, " . \$_POST["firstname"] . " " . \$_POST["lastname"] . " "; \$booktitle= "1984"; \$author="George Orwell"; \$finished = true; \$rate = 9.5; echo "Buch des Tages: "; echo \$booktitle; ?></pre> |



Was ist denn dann eigentlich `$_POST` für eine komische Variable, häää?



XSS(Cross-site scripting)-Gefahr: Immer dann, wenn Daten vom Client zum Server übertragen werden, sollten diese hinsichtlich ihrer Existenz sowie ihrer möglichen Manipulation hin überprüft werden. Aus didaktischen Gründen verzichten wir *an dieser Stelle* auf einen entsprechenden Exkurs.

Weitere Informationen zu den vielfältigen Gestaltungsmöglichkeiten von Formularen htmlseitig entnehmen Sie bitte nachfolgenden Link.



Möglichkeiten von Formularen

https://wiki.selfhtml.org/wiki/HTML/Tutorials/Formulare_erstellen_und_gestalten



Informationstext zu Variablen: Bitte lesen Sie sich nun nachfolgenden Informationstext durch und markieren Sie sich die Ihrer Meinung nach wichtigen Stellen.

PHP-Variablen beginnen mit einem \$-Zeichen, gefolgt von einem Variablenamen. Ein gültiger Variablenname beginnt mit einem Buchstaben oder einem Unterstrich, gefolgt von einer Kombination aus Buchstaben, Ziffern und oder Unterstrichen. Variablennamen sind *case-sensitive*, d.h. Groß- und Kleinschreibung machen einen Unterschied. Alles nach dem // ist ein Kommentar und wird ignoriert.



Aufgabe: Versehen Sie alle zulässigen Namensgebungen mit einem Haken.

```
<?php  
o $ some value = 'abc';  
o $number = 12.3;  
o $some$aigna% = '$^%';  
o $go_2_home = "ok";  
o $go_2_Home = 'no';  
o $isThisCamelCase = true;
```

Am Ende der Liste sehen Sie die CamelCase-Notation, die Mischung von Groß- und Kleinbuchstaben, was eine bevorzugte Schreibweise vieler Programmierer ist.⁴

⁴ Welche Richtlinie zur Definition von Variablen, Funktionen oder auch Klassen Sie nutzen, bleibt im schulischen Umfeld letztlich Ihnen überlassen. Gerne können Sie auch die Programmierrichtlinien Ihrer Firma anwenden. Wichtig ist nur, dass Sie einheitlich und ggf. in Absprache im Team Ihre Benennungen durchführen. Sollten Sie die

Datentypen

PHP ist schwach typisierend. Das heißt, dass der Datentyp einer Variablen automatisch von PHP zugewiesen wird. Hierzu „untersucht“ PHP den Wert, welcher der Variablen zugewiesen wird und vergibt entsprechend automatisch den vermeintlich passenden Datentyp.

PHP hat 8 primitive Datentypen, aber zurzeit konzentrieren wir uns auf die vier wichtigsten:

- Boolean: Sie besitzen entweder true oder false als Wert.
- Integer: Dies sind numerische Werte ohne Dezimalstellen, z.B. 2 oder 5
- Float: Fließkommazahlen mit einer Dezimalstelle, z.B. 2.3
- Strings: aneinandergesetzte Zeichen, die entweder von einem „ oder einem , umschlossen sind, z.B. „dies“ oder „das“.

Auch wenn PHP diese Datentypen besitzt, erlaubt es dennoch die Zuweisung verschiedener Typen an die gleiche Variable. Betrachten Sie den folgenden Code:

```
<?php  
$number = 123;  
var_dump($number);  
$number = 'abc';  
var_dump($number);  
Das Ergebnis im Browser wird wie folgt aussehen:  
int(123)  
string 'abc'
```

Zunächst weist man der Variablen \$number den Wert 123 zu. Die Variable besitzt deshalb den Datentyp Integer, was var_dump auch anzeigt. Danach weisen wir der gleichen Variablen einen String zu; PHP beklagt sich nicht deswegen. Man nennt dieses Verhalten type juggling (Typ-Jonglieren).



Aufgabe: Geben Sie nachfolgenden Code in der Konsole ein und finden Sie mit Ihrem Banknachbarn zusammen eine Erklärung für das Ergebnis:

```
<?php  
$a = "1";  
$b = 2;  
  
var_dump($a + $b);  
var_dump($a . $b);
```

Sie wissen bereits, dass der +-Operator die Summe zweier numerischen Werte zurückgibt. Der .-Operator fügt zwei Strings zusammen. Wenn PHP einen String mit einer Zahl addieren soll, versucht PHP diesen String in eine Zahl zu konvertieren. In unserem Falle ist dies möglich. In der letzten Zeile fügen wir zwei String zusammen. Die Variable \$b hält einen Integer-Wert; deshalb wird PHP diesen zunächst in einen String („2“) umwandelt, und dann mit dem anderen String verbunden. Das Ergebnis ist dann „12“.

Programmierrichtlinien Ihrer Firma noch nicht kennen, empfehle ich Ihnen beim nächsten Aufenthalt in der Firma danach zufragen.

Operatoren

Damit Variablen miteinander agieren können, benötigen Sie sogenannte Operatoren. Nachfolgend eine kleine Auswahl.

Arithmetische Operatoren

Arithmetische Operatoren sind leicht verständlich. Addition, Subtraktion, Multiplikation und Division (+, -, *, and /) funktionieren wie ihr Name es sagt. Modulo (%) gibt den Restwert einer Division zurück; der Exponent nimmt den ersten Operanden hoch den 2. Operanden. Die Negation versetzt den Operanden in das Negative; dies ist auch der einzige Operator, der nur einen Operanden besitzt.

```
<?php
$a = 10;
$b = 3;
var_dump($a + $b); // 13
var_dump($a - $b); // 7
var_dump($a * $b); // 30
var_dump($a / $b); // 3.333333...
var_dump($a % $b); // 1
var_dump($a ** $b); // 1000
var_dump(-$a); // -10
```

Zuweisungsoperatoren

Dieser wurden bereits vorher genutzt; er weist das Ergebnis eines Ausdrucks einer Variablen zu.

```
<?php
$a = 3 + 4 + 5 - 2;
var_dump($a); // 10
```

Es gibt eine Reihe von verkürzten Zuweisungsoperatoren. Man erstellt sie durch die Kombination eines arithmetischen mit einem Zuweisungsoperator. Einige Beispiele:

```
<?php
$a = 13;
$a += 14; // same as $a = $a + 14;
var_dump($a);
$a -= 2; // same as $a = $a - 2;
var_dump($a);
$a *= 4; // same as $a = $a * 4;
var_dump($a);
```

Vergleichsoperatoren

Vergleichsoperatoren werden am häufigsten genutzt. Sie nehmen zwei Operanden und vergleichen diese. Das Ergebnis des Vergleichs wird üblicherweise als boolean zurückgegeben. Ein Beispiel:

```
<?php
var_dump(2 < 3); // true
var_dump(3 < 3); // false
var_dump(3 <= 3); // true
var_dump(4 <= 3); // false
var_dump(2 > 3); // false
var_dump(3 >= 3); // true
var_dump(3 > 3); // false
```

Logische Operatoren

Diese vergleichen logische Operationen, also Aussagen wie True oder False miteinander; der Rückgabewert ist ein boolescher Wert. Die meistverwendeten Operatoren sind “(not), \$\$ (and) und || (oder)

```
<?php
var_dump(true && true); // true
var_dump(true && false); // false
var_dump(true || false); // true
var_dump(false || false); // false
var_dump(!false); // true
```

Increment- / Decrement- Operatoren

Dies sind abkürzende Schreibweisen wie += bzw. -=; sie werden nur mit Variablen verwendet. Sie benötigen besondere Beachtung, weil sie sich nicht immer intuitiv verhalten:

++: Links von der Variable wird der Wert der Variable um eins erhöht und dann als Ergebnis weiterverarbeitet. Rechts von der Variable eingesetzt, wird zunächst der Wert der Variable zurückgegeben und dann um eins erhöht.

-: Wie bei ++, nur wird der Wert der Variable jeweils um eins vermindert.

Let's see an example:

```
<?php
$a = 3;
$b = $a++; // $b is 3, $a is 4
var_dump($a, $b);
$b = ++$a; // $a and $b are 5
var_dump($a, $b);
var_dump($g);
```

Strings

Strings sind alphanumerische Zeichenketten, d.h. sie enthalten entweder Ziffern, Buchstaben bzw. Sonderzeichen. Der Programmierer hat relativ häufig mit Strings zu tun; PHP stellt dafür eine ganze Menge an Funktionen zur Verfügung, die man unter <http://php.net/manual/en/ref.strings.php> betrachten kann. Wir werden nur die am häufigsten verwendeten Funktionen ansprechen.

```
<?php
$text = ' How can a clam cram in a clean cream can? ';
echo strlen($text); // 45
$text = trim($text); echo $text; // How can a clam cram in a clean cream can?
echo strtoupper($text); // HOW CAN A CLAM CRAM IN A CLEAN CREAM CAN?
echo strtolower($text); // how can a clam cram in a clean cream can?
$text = str_replace('can', 'could', $text);
echo $text; // How could a clam cram in a clean cream could?
echo substr($text, 2, 6); // w coul
var_dump(strpos($text, 'can'));// false
var_dump(strpos($text, 'could'));// 4
```

Im obigen Beispiel wird ein String mit verschiedenen Funktionen bearbeitet:

- **strlen:** Diese Funktion gibt die Anzahl der Zeichen zurück, die ein String enthält.
- **trim:** Diese Funktion nimmt alle Leerzeichen links und rechts vom String weg und gibt ihn anschließend zurück
- **strtoupper/strtolower:** Diese Funktionen wandeln alle Zeichen eines String in Groß- oder Kleinbuchstaben um
- **str_replace:** Diese Funktion ersetzt alle Vorkommen eines Teilstrings im betrachteten String durch einen anderen String (Suchen/Ersetzen)
- **substr:** Diese Funktion holt aus einem String eine Zeichenkette heraus, die durch die übergebenen Positionen definiert sind.
- **strpos:** Diese Funktion zeigt die Position der ersten Fundstelle einer zu suchenden Zeichenkette innerhalb eines Strings.

Zusätzlich gibt es einen String-Operator (.), der zwei Strings zusammenfügt.

Strings können weiterhin durch einzelne oder doppelte Anführungszeichen umschlossen werden. Der Unterschied besteht darin, dass mit einfachen Anführungszeichen ein String so dargestellt wird, wie er ist. Mit doppelten Anführungszeichen hingegen werden Escape-Zeichen sowie Variablen innerhalb des Strings unterschiedlich behandelt.

Escape-Zeichen: Dies sind besondere Zeichen, die nicht einfach dargestellt werden können. Beispiele dafür sind newline (Neue Zeile) oder Tabulatoren. Um sie darzustellen, werden Escape-Zeichen verwendet. Die sind das sog. Backslash-Zeichen, gefolgt von einem weiteren Zeichen. Beispielsweise ist \n das Zeichen für NeueZeile, \t bedeutet einen Tabulator.



Fragen an Dr. Simpson (jun.)

F: Wie kann ich erkennen, welchen Datentyp PHP einer Variablen zugewiesen hat?

A: Will man den Typ einer Variable erkennen, erlaubt der Befehl `echo gettype($name);` die Ausgabe des Datentyps.

F: Im Text stand, das PHP schwach typisierend ist. Gibt es folglich auch stark typisierende Sprachen?

A: Ja natürlich. So zum Beispiel C# oder Java. Dort muss der Datentyp einer Variablen bei der Deklaration explizit mit angegeben werden. Beispielcode in C#: `int nummer= 10;` Weist man dann aus Versehen einen nicht zum Datentyp passenden Wert zu, erzeugt dies eine Fehlermeldung.

F: Ist es auch möglich, die Typzuweisung an eine Variable zu erzwingen?

A: Will man den Typ einer Variable erkennen, erlaubt der Befehl `echo gettype($name);` die Ausgabe des Datentyps.

F: Was ist denn der Unterschied bei einem Vergleich zwischen == und ===?

A: Es gibt auch mehrere Operatoren, die auf Gleichheit prüfen. Der == - Operator überprüft die Gleichheit nach einer evtl. notwendigen Typumwandlung. Der === - Operator (Identität) überprüft ohne Typumwandlung.

```
$a = 3;
$b = '3';
var_dump($a == $b); // true
var_dump($a === $b); // false
```

Übungen

1. Erstellen Sie eine Anwendung, bei der eine von Nutzer in einen Formular eingegebene Ganzzahl mit sich selbst multipliziert und anschließend am Bildschirm ausgibt.

2. In einem Programm müssen die Inhalte der Variablen \$v1 und \$v2 getauscht werden. Ordnen Sie die folgenden Anweisungen in die richtige Reihenfolge, in dem Sie 1,2,3 passend vor der Anweisung notieren.

| | |
|--|-------------|
| | \$v2=\$v1 |
| | \$v1=\$temp |
| | \$temp=\$v2 |

3. Simulieren Sie nun die Anmeldemaske einer Kartenreservierung für ein Kino. Neben einem Begrüßungstext soll der Nutzer aufgefordert werden, seinen Vornamen, Nachnamen sowie die Anzahl der gewünschten Karten nacheinander einzugeben. Alle diese Informationen werden auf der Serverseite mit PHP anschließend übersichtlich noch einmal ausgegeben werden. Geben Sie zusätzlich den Gesamtpreis der Reservierung aus, wenn eine Karte 8 € kostet.

4. Erstellen Sie mit folgenden Angaben ein PHP-Skript.

| Variable | Bezeichnung | Variable | Preis (netto) |
|-----------------------|---------------|-----------------|---------------|
| \$bezeichnung_tisch | Schreibtisch | \$preis_tisch | 1999.00 € |
| \$bezeichnung_stuhl | Bürostuhl | \$preis_stuhl | 589.00 € |
| \$bezeichnung_lampe | Lampe | \$preis_lampe | 29.00 € |
| \$bezeichnung_pctisch | Computertisch | \$preis_pctisch | 999.00 € |

- a. Berechnen Sie den Gesamtpreis (\$netto_gesamt) der eingekauften Artikel, indem Sie die zuvor von Ihnen angelegten Variablenwerte addieren lassen.
- b. Berechnen Sie für den gerade berechneten Gesamtpreis den Bruttopreis (\$brutto_gesamt) mithilfe einer Konstanten namens MWST. Recherchieren Sie hierzu, wie man eine Konstante in PHP erzeugt. Der Mehrwertsteuersatz, der zur Berechnung verwendet wird, beträgt 19 %. Die verwendete Zeichenkette für die Währung Euro stellen Sie bitte ebenfalls über eine Konstante (EURO) bereit.
- c. Berechnen Sie zusätzlich die Bruttopreise aller Artikel.
- d. Lassen Sie alle errechneten Werte in verständlicher Form mit Beschriftungen anzeigen.

Netto-Gesamtpreis der eingekauften Artikel: 3616 Euro.
 Brutto-Gesamtpreis der eingekauften Artikel: 4303.04 Euro.

Brutto-Preis Schreibtisch: 2378.81 Euro.
 Brutto-Preis Bürostuhl: 700.91 Euro.
 Brutto-Preis Schreibtischlampe: 34.51 Euro.
 Brutto-Preis Computertisch: 1188.81 Euro.

5. Mit der Widmark-Formel können wir die Blutalkoholkonzentration abschätzen. Sie wird mit folgender Formel berechnet:

$$c = \frac{A}{m * r}$$

mit

$$A = V * e * \rho$$

wobei

- * c : Alkoholkonzentration im Blut in [g/kg]
- * A : Aufgenommene Masse des Alkohols in [g]
- * r : Verteilungsfaktor im Körper – Männer: $r \approx 0,7$ – Frauen: $r \approx 0,6$ – Kinder: $r \approx 0,8$
- * m : Masse der Person in [kg]
- * V : Volumen des Getränks in [ml]
- * e : Alkoholvolumenanteil in [%] (z.B. Bier $\approx 0,05$)
- * ρ : Dichte von Alkohol [g/ml] → $\rho \approx 0,8$ g/ml

Aufgabenstellung: Schreiben Sie ein Programm, das die Blutalkoholkonzentration mit der Widmark-Formel berechnet. Die zur Berechnung notwendigen Daten sollen dabei vom Nutzer im Rahmen einer Formulareingabe ermittelt werden können. Das Ergebnis soll mit einen von Ihnen frei wählbaren situativen Hinweis versehen werden.

2.3



Oh neeein! Nur *ein Buch* zum Verkauf anzubieten erscheint mir nun doch nicht besonders raffiniert zu sein. Doch wie soll ich nun bloß ein weiteres Buch auf der Serverseite mit PHP anbieten können?

Ihre Aufgabe: Klären Sie mit Ihrem Banknachbarn, was unseren Freund Homer Kopfzerbrechen bereitet und halten Sie Ihre Erkenntnisse im nachfolgenden Feld fest.



Ich bin so klug, ich bin so klug! K-L-U-K!

```
<?php  
echo "Guten Tag, " . $_POST["firstname"] . " " . $_POST["lastname"] . "</br>";  
  
$books = ['1984', 'Romeo and Juliet'];  
  
foreach($books as $book)  
{  
    echo $book . " </br>";  
}
```



Informationstext zu Arrays: Bitte lesen Sie sich nun nachfolgenden Informationstext durch und markieren Sie sich die Ihrer Meinung nach wichtigen Stellen.

Wenn Sie schon Erfahrung mit Programmiersprachen haben, werden Sie zwei Datenstrukturen kennen, die sehr viel gemeinsam haben und besonders nützlich sind: *Listen* und *Maps*. Eine Liste ist eine geordnete Ansammlung von Elementen, während eine Map eine Sammlung von *Schlüssel-Wert-Paaren* darstellt. Sehen wir uns ein Beispiel an:

```
List: ["Harry", "Ron", "Hermione"]
```

```
Map: { "name": "James Potter", "status": "dead" }
```

In PHP haben wir Arrays, die beide vorher angesprochenen Datenstrukturen implementieren.

Arrays anlegen

Man kann auf verschiedene Arten Arrays erzeugen. Man kann ein leeres Array anlegen oder bereits am Anfang mit Daten füllen.

```
<?php
$empty1 = [];
$empty2 = array();

$names1 = ['Harry', 'Ron', 'Hermione'];
$names2 = array('Harry', 'Ron', 'Hermione');

$status1 = [ 'name' => 'James Potter', 'status' => 'dead' ];
$status2 = array('name' => 'James Potter', 'status' => 'dead' );
```

Im diesem Beispiel erzeugen wir sowohl eine Liste(\$names) als auch eine Map(\$status).

Später werden wir sehen, dass Listen wie Maps behandelt werden. Intern ist der \$names1 eine Map, ihre Schlüssel sind sortierte Ziffern! Eine andere Darstellung für \$names1 zeigt den Zusammenhang.

```
$names1 = [ 0 => 'Harry', 1 => 'Ron', 2 => 'Hermione' ];
```

Die Schlüssel eines Arrays können beliebige alphanumerische Werte sein, wie beispielsweise Buchstaben oder Zahlen. Die Werte eines Arrays können beliebig sein: Strings, Ziffern, Boolesche Werte, andere Arrays und so weiter. Sehen sie folgendes Beispiel:

```
<?php
$books = [ '1984' => [ 'author' => 'George Orwell', 'finished' => true, 'rate' => 9.5 ],
           'Romeo and Juliet' => [ 'author' => 'William Shakespeare', 'finished' => false ] ];

var_dump($books);
```

Dieser Array ist eine Liste, die zwei array-basierte Maps enthält. Jede Map enthält verschiedene Werte wie String, Doubles und Booleans.

Füllen von Arrays

Arrays sind nicht unveränderlich sondern können auch nach ihrer Erzeugung geändert werden. Sie können das Array dabei wie eine Liste oder wie eine Map ansprechen. Bei einer Map sprechen Sie den Schlüssel an bzw. formulieren Sie einen neuen Schlüssel. Bei einer Liste fügen Sie einfach ein weiteres Element hinzu.

```
<?php
$names = ['Harry', 'Ron', 'Hermione'];
$status = [ 'name' => 'James Potter', 'status' => 'dead' ];

$names[] = 'Neville';

$status['age'] = 32;

print_r($names, $status);
```

Wenn man Elemente aus einem Array herausnehmen will, kann man die unset-Funktion verwenden.

```
<?php
$status = [ 'name' => 'James Potter', 'status' => 'dead' ];
unset($status['status']);

print_r ($status);
```

Ergebnis: Der neue \$status-Array enthält nur noch den Schlüssel name.

Ausgabe eines Arrays im Rahmen einer Schleife

```
<?php
$status = [ 'name' => 'James Potter', 'status' => 'dead' ];

foreach($status as $key => $value)
{
    echo $key . ":" . $value;
}
```

Zugriff auf Array-Elemente

Der Zugriff auf die Inhalte eines Arrays ist leicht, da nur der Schlüssel angegeben werden muss. Sie müssen deshalb verstehen, wie Listen funktionieren. Intern werden Listen wie Maps behandelt, die einen sortierten numerischen Schlüssel haben. Dieser beginnt immer bei 0 und endet damit bei n-1.

Sie können jeden Schlüssel einem vorhandenen Array hinzufügen, auch wenn vorher nur numerische Schlüssel angegeben wurden. Wenn man einen Wert hinzufügen will, wird ihn PHP hinter den letzten numerischen Schlüssel anhängen. Die Ausgabe eines bestimmten Teils eines Arrays wird durch die Angabe seinen Schlüssel ermöglicht:

```
<?php
$names = ['Harry', 'Ron', 'Hermione'];
print_r($names[1]);
//prints 'Ron'
```

Der Zugriff auf einen nicht-existenten Schlüssel wird einen NULL-Wert zurückgeben.

The empty and isset functions

Es gibt zwei nützliche Funktionen, um Informationen über den Inhalt eines Arrays zu erhalten. Die empty-Funktion gibt true zurück, wenn ein Array keinerlei Inhalt hat. Die isset-Funktion gibt an, ob die angefragte Position im Array vorhanden ist oder nicht.

Suchen in einem Array

Eine der am häufigsten benutzten Funktionen im Zusammenhang mit Arrays ist wohl `in_array()`. Die Funktion hat zwei Parameter, nämlich den gesuchten Wert und den Array, in dem gesucht werden soll. Die Funktion gibt True zurück, wenn der Wert gefunden wird, ansonsten False. Noch nützlicher ist manchmal die Funktion `array_search()`. Sie gibt die Position der Fundstelle zurück oder False.

Sortieren von Arrays

Ein Array kann nach verschiedenen Kriterien sortiert werden. Die Default-Sortierung (Reihenfolge der Eingabe), kann durch die Sortierung nach Schlüsseln oder den Werten ersetzt werden. Beim Sortieren kann noch darauf geachtet werden, ob die Schlüssel erhalten werden sollen oder nicht. Eine komplette Übersicht erhalten Sie unter [*http://php.net/manual/en/array.sorting.php*](http://php.net/manual/en/array.sorting.php).

Weitere Funktionen bei Arrays (verkürzt):

| | |
|---|--|
| <code>array_flip(\$feld)</code> | Im Array werden Indizes mit Werten vertauscht. |
| <code>array_key_exists(wert, \$feld)</code> | Prüfung, ob ein Schlüssel in einem Array vorhanden ist. |
| <code>array_keys(\$feld)</code> | Liefert die Indizes des angegebenen Arrays zurück. |
| <code>array_merge(\$feld1, \$feld2...)</code> | Fügt die Elemente mehrerer Arrays zu einem Array zusammen. |
| <code>array_push(\$feld, werte)</code> | Das Array wird um den oder die angegebenen Werte am Ende des Arrays erweitert. |
| <code>array_search(wert, \$feld)</code> | Das Array wird nach dem angegebenen Wert durchsucht. |
| <code>array_sum(\$feld)</code> | Addiert die Werte des Arrays und liefert das Ergebnis zurück. |
| <code>array_unique(\$feld)</code> | Es wird ein neues Array erstellt, aus dem doppelte Werte des angegebenen Arrays gelöscht wurden. |
| <code>array_values(\$feld)</code> | Alle Werte des Arrays werden zurückgeliefert. |
| <code>count(\$feld)</code> | Gibt die Anzahl der Elemente des Arrays zurück. |
| <code>sort(\$feld)</code> <code>rsort(\$feld)</code> | Sortiert die Werte des angegebenen Arrays aufsteigend bzw. absteigend. |
| <code>ksort(\$feld)</code> <code>krsort(\$feld)</code> | Sortiert das angegebene Array nach Schlüssel aufsteigend bzw. absteigend. |

Alle Arrayfunktionen können Sie der PHP Dokumentation je nach Bedarf entnehmen.

Fragen an Dr. Simpson (jun.)

F: Ist es auch möglich, in PHP typsichere Arrays zu erzeugen?

A: Da PHP an sich eine schwach typisierende Sprache ist, ist eine unmittelbare Typsicherheit, d. h. Sicherstellung, dass immer nur der angegebene passende Typ in ein Array eingefügt wird, nicht möglich. Es gibt jedoch Work-arounds um dies zu realisieren. Lese bei Interesse hierzu einfach im Internet nach.

F: Woran kann ich eigentlich erkennen, ob ich ein indiziertes oder assoziatives Array nutzen soll?

A: Die Antwort ergibt sich aus dem realen Kontext, den Sie abbilden wollen. Liegen bereits eindeutige Schlüssel(Keys) in der realen Welt vor, z. B. ein KFZ-Kennzeichen für ein Auto, dann bietet sich ein assoziatives Array an.

Übungen:

1. Best Friends
 - a. Erstellen Sie ein Array, das Ihre drei besten Klassenfreunde speichert und diese anschließend wieder ausgibt.
 - b. Ein Freund ist in Ungnade gefallen, löschen Sie diesen aus dem Array. Geben Sie den Inhalt des Arrays anschließend wieder aus.
 - c. Sie haben eine neue Freundin kennengelernt, sie heißt Uschi. Speichern Sie auch diese in Ihrem Array. Geben Sie den Inhalt das Array anschließend wieder aus.
2. Sie wollen herausfinden, wie viele Kilometer Sie im Durchschnitt mit einer Tankfüllung fahren können. Hierzu haben Sie sich folgende Werte notiert: (1020, 923, 780, 890).
 - a. Berechnen Sie die durchschnittliche Kilometerreichweite für Ihr Fahrzeug.
 - b. Ermitteln Sie den durchschnittlichen Kraftstoffverbrauch je 100 km unter der Annahme, dass Sie einen 70 Liter Tank haben.
3. Erstellen Sie ein Array mit den Zahlen von 0 bis 9. Erstellen Sie anschließend ein zweites Array und weisen Sie diesen den Inhalt des ersten Arrays in umgekehrter Reihenfolge zu. Folgendes Beispiel soll dies simulieren. $[0,1,2,3,4,5,6,7,8,9] \rightarrow [9,8,7,6,5,4,3,2,1,0]$
4. Erstellen Sie ein Programm, das die Ziehung der Lottozahlen simuliert. Füllen Sie zunächst ein Array mit sechs Zahlen, die zufällig über folgenden Befehl erzeugt werden: `rand(5, 15);`
Der Spieler kann anschließend sechs Zahlen zwischen 1 und 49 eingeben, die in einem weiteren Array gespeichert werden. Das Programm zeigt anschließend beide Zahlenreihen untereinander an. Der Abgleich der richtig erratenen Zahlen wird zunächst dem Nutzer überlassen.
Erweiterung für Schnelle: Erstellen Sie eine Methode, die die Überprüfung der Zahlen selbstständig ausführt.
5. Sie sollen ein Programm entwickeln, das Buchstabe in Morse-Code-Zeichen übersetzt. Diese können Sie folgender Liste entnehmen.





```
//Oder alternativ:  
$person[] = array("Johanna", "schwedisch", "19 Jahre");
```

Ausgabe des Inhalts eines mehrdimensionalen indizierten Arrays

Will man nun die Daten von Maria ausgegeben bekommen, würde dies mit folgenden Befehl funktionieren:

```
echo $person[1][0] . " ist " . $person[1][2] . " alt und spricht " . $person[1][1] . ".<br>"
```

Die erste eckige Klammer beinhaltet den Index des äußeren Arrays, die zweite eckige Klammer den Index des inneren Arrays.

Sollen hingegen alle Daten eines mehrdimensionalen indizierten Arrays ausgegeben werden, kann die mit einer doppelten for-Schleife geschehen:

```
<?php  
for($i=0; $i < count($person[0]); $i++)  
{  
    for($y=0; $y < count($person[$i]); $y++)  
    {  
        echo $person[$i][$y];  
        echo "<br>";  
    }  
}
```

Assoziatives mehrdimensionales Array anlegen

```
<?php  
$land = array(  
    "Spanien" => array("Hauptstadt" => "Madrid", "Sprache" => "Spanisch", "Waehrung" => "Euro", "Flaeche" => "504545 qkm"),  
    "England" => array("Hauptstadt" => "London", "Sprache" => "Englisch", "Waehrung" => "Pfund", "Flaeche" => "130335 qkm"),  
    "Portugal" => array("Hauptstadt" => "Lissabon", "Sprache" => "Portugiesisch", "Waehrung" => "Euro", "Flaeche" => "92345 qkm")  
>;
```

Während **Spanien** der Schlüssel des ersten äußeren Arrays ist, ist **Hauptstadt** der erste Schlüssel des inneren Arrays.

Aufgabe: Leiten Sie die **Syntax** für das Erstellen eines indizierten mehrdimensionalen Arrays aus obigen Beispiel ab und notieren Sie diese im nachfolgenden Feld.

Erweitern von mehrdimensionalen assoziativen Arrays

```
<?php  
$land["Ungarn"]["Hauptstadt"] = "Budapest"; $land["Ungarn"]["Sprache"] = "Ungarisch";  
$land["Ungarn"]["Waehrung"] = "Forint"; $land["Ungarn"]["Flaeche"] = "98036 qkm";
```

```
//oder alternativ:
$land["Ungarn"] = array("Hauptstadt" => "Budapest", "Sprache" => "Ungarisch", "Währung" =>
"Forint", "Fläche" => "93036 qkm")
```

Ausgabe des Inhalts eines mehrdimensionalen assoziativen Arrays

```
<?php
//Anlegen
$lands = array(
    "Spanien" => array("Hauptstadt" => "Madrid", "Sprache" => "Spanisch", "Währung" => "Euro", "Fläche" => "504442 qkm"),
    "England" => array("Hauptstadt" => "London", "Sprache" => "Englisch", "Währung" => "Pfund", "Fläche" => "130035 qkm",
    "Portugal" => array("Hauptstadt" => "Lissabon", "Sprache" => "Portugiesisch", "Währung" => "Euro", "Fläche" => "92245 qkm")
);
//Ausgabe
foreach ($lands as $key =>$land)
{
    foreach ($land as $key => $value) {
        echo $key . " : " . $value;
        echo "<br>";
    }
}
```



Bitte erstellen Sie nun für mich einen Lösungsvorschlag für das Erstellen des mehrdimensionalen Arrays meines Onlinebücherhandels. Das nachstehende Bild zeigt, wie ich mir die Ausgabe nach der Eingabe des Namens auf der Startseite vorstelle:

① Ausgangsformular zur Namenseingabe

Guten Tag!

Gern zeigen wir Ihnen die zur Verfügung stehenden Bücher. Um Sie persönlich ansprechen zu können, bitte ich Sie jedoch zuerst Ihren Namen anzugeben.

| | |
|---------------------------------------|--------------|
| <input type="text"/> | Vorname |
| <input type="text"/> | Nachname |
| <input type="button" value="Senden"/> | Zurücksetzen |

② Anzeige aller vorhandenen Bücher

NEU!

Guten Tag, Christoph Zobel!
Folgende Bücher stehen zur Auswahl:
author:George Orwell
finished:yes
rate:9.5€
author:William Shakespeare
finished:no
rate:9.5€

Personal
Kunden
anspräde!

Fragen an Dr. Simpson (jun.)

F: Woran erkenne ich, wann die Nutzung eines eindimensionalen oder mehrdimensionalen Arrays sinnvoll ist?

A: Die Antwort ergibt sich auch hier aus dem realen Kontext, den Sie abbilden wollen. Liegt zum Beispiel eine Struktur im Sinne eines Weihnachtskalenders vor, lassen sich die zwei Dimensionen Tag und Inhalt bei einem Türchen erkennen. Tipp: Die Orientierung am realen Bild, das man abbilden will, erscheint eigentlich immer sinnvoll.

F: Es gibt doch bei der Nutzung von mehrdimensionalen Arrays auch die praktische list-Funktion? Warum wurde diese nicht beim Auslesen des mehrdimensionalen assoziativen Arrays verwendet?

A: Wie in jeder Programmiersprache gibt es auch bei PHP verschiedene Wege zum Ziel. Die Nutzung der zweifach verschachtelten foreach-Schleife ist hier nur exemplarisch zu verstehen, auch das Auslesen über andere Schleiftypen ist denkbar. Bei der list-Funktion ist jedoch zu beachten, dass diese nur bei mehrdimensionalen Arrays verwendet werden können, deren inneres Array indiziert ist.

Übungen:

- Erstellen Sie ein mehrdimensionales indiziertes Array mit folgenden Inhalten und geben Sie diese anschließend aus.⁶

| Nr. | Titel | Originaltitel | Dt. Erstausstrahlung |
|-----|---------------------------|-----------------------------------|----------------------|
| 1 | Es weihnachtet schwer | Simpsons Roasting on an Open Fire | 6. Dez. 1991 |
| 2 | Bart wird ein Genie | Bart the Genius | 20. Sept. 1991 |
| 3 | Der Versager | Homer's Odyssey | 11. Okt. 1991 |
| 4 | Eine ganz normale Familie | There's No Disgrace Like Home | 13. Sep. 1991 |

- Sie erhalten Sie ehrenvolle Aufgabe, eine Tabelle zu erstellen, die die Teams mit den meisten Siegen bei der Fussballweltmeisterschaft der Herren auflistet. Diese letztlich in HTML generierte Tabelle wird mit einen zuvor erstellten passenden mehrdimensionalen Array erstellt.

Erweiterung für Schnelle: Es sollen nun auch die Jahre aufgezählt werden, in denen die Teams Weltmeister wurden.



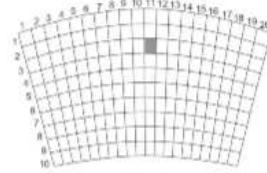
⁶ Quelle: <http://de.simpsons.wikia.com/wiki/Episodenguide>

3. Erstellen Sie für einen Kinosaal ein Reservierungssystem(stark vereinfacht), das es ermöglicht, freie Plätze je nach Kundenwunsch zu reservieren. Den Kinosaal können Sie sich wie folgt vorstellen:

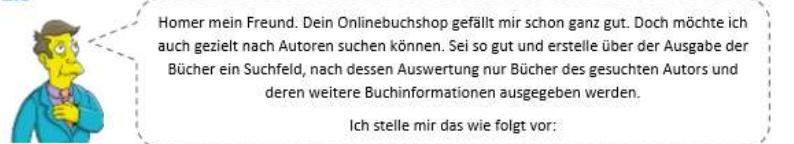
a) Erstellen Sie zu diesem Zwecke zuerst ein mehrdimensionales Array und belegen Sie das Array an allen Stellen mit dem Wert `true`. Bieten Sie dem Nutzer sodann an, durch die Eingabe der Werte Reihe sowie Platznummer einen Sitzplatz zu reservieren, sofern dieser noch frei ist. Dies kann der Nutzer so oft wiederholen, bis dieser den Vorgang durch die Eingabe von `x` abbricht.

b) Erweitern Sie Ihr Programm nun dahingehend, dass die Gesamtzahl der noch verfügbaren Plätze vor jeder Nutzerabfrage angezeigt wird.

c) Es soll nun ermöglicht werden, dass der Nutzer eine Reihe eingeben kann und eine Anzahl von zusammenhängenden Sitzplätzen, die er in dieser Reihe haben möchte. Das Programm soll sodann klären, ob es in der gewünschten Reihe, die gewünschte Anzahl zusammenhängender Plätze gibt. Die entsprechenden Platznummern werden an den Aufrufer zurückgegeben.



2.5



① Ausgangsformular zur Nameneingabe

Gerne zeigen wir Ihnen die zur Verfügung stehenden Bücher. Um Sie persönlich ansprechen zu können, bitte ich Sie jedoch zuerst Ihren Namen anzugeben.

Vorname
Nachname
Senden Zurücksetzen

② Suchleiste mit Suchfeld

Gutes Tag, Christoph Zobel

William Shakespeare Gesuchter Autor: NEU!

Folgende Bücher stehen zur Auswahl:
author:Georges Orwell
finished yes
rate: 8.5
author: William Shakespeare
finished no
rate: 9.5

③ Ergebnisanzeige der Suchfeldauswertung

Title: Romeo and Juliet
author: William Shakespeare
finished no
rate: 9.56

↳ Sofern kein Treffer, entsprechende Meldung anzeigen

„Versuchen ist der erste Schritt zum Versagen.
Infofern viel Erfolg!“



Informationstext zu Kontrollstrukturen: Bitte lesen Sie sich nun nachfolgenden Informationstext durch und markieren Sie sich die Ihrer Meinung nach wichtigen Stellen.

If/Else-Kontrollstruktur

Kontrollstrukturen sind ein zentraler Bestandteil fast jeder Programmiersprache. Erst diese ermöglichen einen flexiblen, unterschiedlichen Fluss des Programmablaufes in Abhängigkeit der Auswertung von Variablen. Nachfolgendes Beispiel zeigt den einfachen Einsatz einer if/else-Struktur.

```
<html>
<body>
<p>Bitte tragen Sie das Zugangspasswort ein</p>
<form action = "ifelse_zugang.php" method =
"post">
    <p><input name = "pw" /></p>
    <p><input type = "submit" />
    <input type = "reset" /></p>
</form>
</body>
</html>
<?php
    if ($_POST["pw"] == "bingo")
    {
        echo "Zugang gestattet";
    }
    else
    {
        echo "Zugang verweigert";
    }
?>
```

Das erste Skript zeigt das zuerst aufgerufene HTML-Dokument an, dass die in der Abbildung dargestellte Eingabemaske erzeugt. Das zweite Dokument ist das PHP-Skript, das in obiger Action-Methode aufgerufen wird. Dort werden dann die überlieferten Variablen mit ausgewertet.

Natürlich kennt PHP auch logische Operatoren wie das Logische Oder (Zeichen `||`), das Logische Und (Zeichen `&&`) sowie den Ungleich (Zeichen `!=`). Weitere Operatoren sowie dessen Auswertung können Sie den genannten Internetquellen entnehmen. Anzumerken ist an dieser Stelle, dass selbstverständlich auch verschachtelte `if/else` Strukturen sowie `switch/case`-Anweisungen möglich sind.

Switch-Kontrollstruktur

Eine andere Kontrollstruktur ähnlich zu `if..else` ist `switch ... case`. Diese Struktur wertet nur einen Ausdruck aus und führt einen Block aus in Abhängigkeit seines Wertes, wie nachfolgendes Beispiel zeigt:

```
<?php
    switch ($title) {
        case 'Harry Potter': echo "Nice story, a bit too long."; break;
        case 'Lord of the Rings': echo "A classic!"; break;
        default: echo "Dunno that one."; break;
    }
```

Der `switch`-Teil erhält einen Ausdruck, in unserem Fall die Variable `$title` und definiert dann eine Reihe von möglichen Fällen (Werte) für diese Variable. Wenn ein möglicher Wert in der Prüfung übereinstimmt, wird der dazugehörige Block aufgerufen.

Sobald eine `break`-Anweisung gefunden wird, verlässt PHP den `switch`-Block. Wenn kein Fall gefunden wird, führt PHP den `default`-Fall aus.

```
<?php
    $title = 'Twilight';
    switch ($title)
    {
        case 'Harry Potter': echo "Nice story, a bit too long.";
        case 'Twilight': echo 'Uh...';
        case 'Lord of the Rings': echo "A classic!";
        default: echo "Dunno that one.";
    }
```

Wenn Sie diesen Code ausführen, wird man sehen dass PHP Uh...A classic! Dunno that one ausgibt. PHP beginnt mit dem 2ten Case und führt die weiteren Cases aus, weil es kein break findet. Dies ist normalerweise nicht gewünscht.



Erstellen Sie nun Ihren Lösungsvorschlag für mich für den ehrenwerten Wunsch von Rektor Skinner.



Fragen an Dr. Simpson (jun.)

F: Muss jede Switch-Kontrollstruktur auch einen Default-Fall angeben?

A: Nein. Die Angabe eines Defaultfalles ist optional und je nach Situation sinnvoll oder auch nicht.

F: Benötige ich nach jeder Anweisung den Befehl break?

A: Es ist auch wichtig zu wissen, dass die break-Anweisungen notwendig sind, wenn man den switch-case-Block verwenden will. Ohne break wird PHP die weiteren Cases ausführen, auch wenn dies nicht gewünscht ist.



Didli dudli mein Freund. Haben wir da nicht ein wichtiges Thema im Rahmen des Informationstextes vergessen? Macht ja nichts, gerne erzähle ich dir mehr über das äußerst spannende Thema

Struktogramme¹

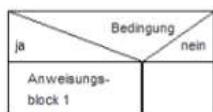
Zur Darstellung von Sachverhalten der strukturierten Programmierung wurden vor vielen Jahren verschiedene Notationsformen entwickelt. Die wichtigsten sind der Programmablaufplan und das Struktogramm.

Das Nassi-Shneiderman-Diagramm, auch Struktogramme genannt, ist ein Diagrammtyp zur Darstellung von Programmentwürfen im Rahmen der Methode der strukturierten Programmierung.

Die meisten der nachfolgenden Strukturböcke können ineinander geschachtelt werden. Das aus den unterschiedlichen Strukturböcken zusammengesetzte Struktogramm ist im Ganzen rechteckig, also genauso breit wie sein breitester Strukturblock.

Linearer Ablauf (Sequenz)

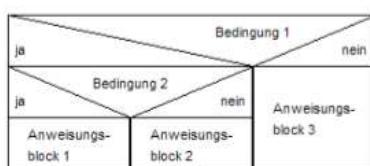
| | |
|-------------|--|
| Anweisung 1 | Jede Anweisung wird in einen rechteckigen Strukturblock geschrieben. |
| Anweisung 2 | Die Strukturböcke werden nacheinander von oben nach unten durchlaufen. |
| Anweisung n | Leere Strukturböcke sind nur in Verzweigungen zulässig. |

Verzweigung (Alternativen)**Einfache Auswahl**

Alternativ: bedingte Verarbeitung, Selektion, einfache Selektion (if)
 Nur wenn die Bedingung zutreffend (wahr) ist, wird der Anweisungsblock 1 durchlaufen. Ein Anweisungsblock kann aus einer oder mehreren Anweisungen bestehen. Trifft die Bedingung nicht zu (falsch), wird der Durchlauf ohne eine weitere Anweisung fortgeführt (Austritt unten).

Zweifache Auswahl

Alternativ: alternative Verarbeitung, alternative Verzweigung (if then else)
 Wenn die Bedingung zutreffend (wahr) ist, wird der Anweisungsblock 1 durchlaufen. Trifft die Bedingung nicht zu (falsch), wird der Anweisungsblock 2 durchlaufen. Ein Anweisungsblock kann aus einer oder mehreren Anweisungen bestehen. Austritt unten nach Abarbeitung des jeweiligen Anweisungsblocks.

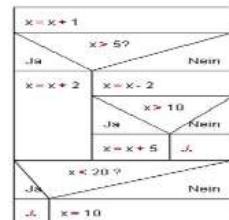
Verschachtelte Auswahl

Es folgt eine weitere Bedingung. Die Verschachtelung ist ebenso im Nein-Fall (noch) möglich.

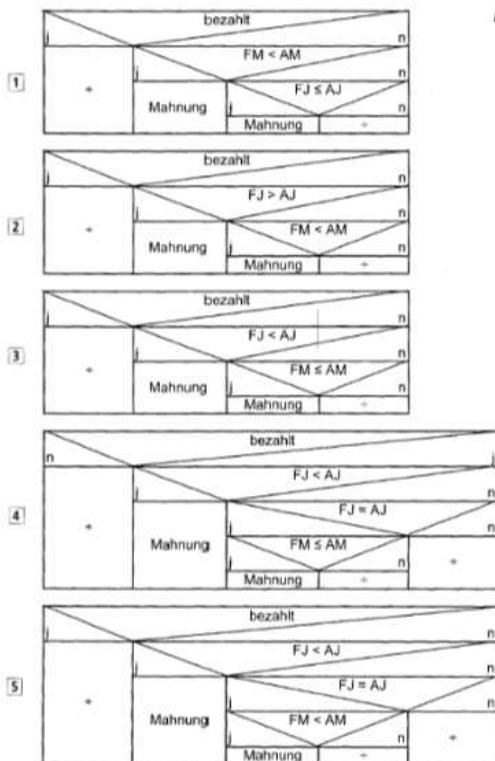
Übungen zu Struktogrammen:

1. Welchen Wert hat die Variable x nach Durchlauf des Programms, wenn:

- a) X zu Beginn den Wert 1 hat
- b) X zu Beginn den Wert 17 hat



2. Für ein Mahnprogramm soll eine Verarbeitungslogik entworfen werden. Die Mahnung wird generell erst im auf den Fälligkeitsmonat folgenden Monat aufgelöst, d.h. der genaue Fälligkeitstag wird bei der Verarbeitung nicht berücksichtigt. Kontrollieren Sie, welches Struktogramm eine richtige Lösung des Problems enthält.



4. Im Rahmen der nachfolgenden Aufgabe sollen Sie einen so genannten Schreibtischtest durchführen. Es stellt sich dabei zwangsläufig die Frage, was ein Schreibtischtest überhaupt ist. Fragen wir hierzu einen einschlägig bekannten Experten, erhalten wir folgende Antwort:

„Der Schreibtischtest ist ein Verfahren, das im Bereich der Softwareentwicklung verwendet wird, um Algorithmen oder Routinen auf Richtigkeit zu prüfen. Der Schreibtischtest wird nicht mit Hilfe eines Rechners durchgeführt, sondern vielmehr im Kopf des Entwicklers“, das zumindest habe ich auf Wikipedia gelesen.

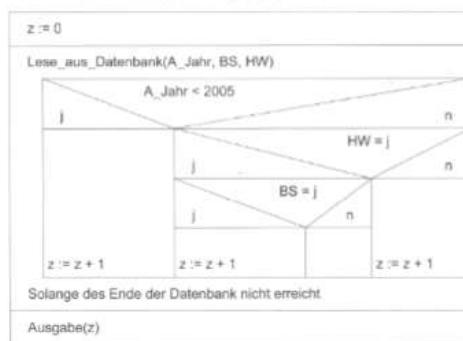


Im neuen Netzwerk der Schwarzenbach KG sollen alle Arbeitsplatz-PCs einheitlich mit einem neuen Betriebssystem betrieben werden und es sollen auch noch alte Arbeitsplatz-PCs weiterverwendet werden.

Zur Ermittlung der benötigten Anzahl Lizenzen soll ein entsprechendes Programm erstellt werden. Ein Mitarbeiter der IT-Abteilung hat bereits folgendes Struktogramm erstellt.

Hinweis

- Das Programm greift auf die Hardware-Verwaltungs-Datenbank zu..
- HW: Hardware-Ausstattung des PCs erfüllt Anforderung (j oder n)
- A_Jahr: Jahr, in dem PC angeschafft wurde
- BS: Installiertes Betriebssystem neu (j oder n)



Testen Sie die im Struktogramm dargestellte Logik mit einem Schreibtischtest anhand folgender Daten:

| A_Jahr | BS | HW |
|--------|----|----|
| 2004 | n | j |
| 2004 | n | n |
| 2006 | n | n |
| 2007 | n | j |
| 2008 | j | j |
| 2009 | j | j |
| 2009 | j | j |

Wie viele Lizenzen müssen laut Ihrem Schreibtischtest beschafft werden?

1. Erstellen Sie nun ein Programm, das mittels eines Eingabeformulars einen Nutzer dabei hilft, dessen Kosten für einen Tankvorgang zu berechnen. Der Nutzer soll hierzu die Anzahl der getankten Liter eingeben sowie die getankte Sorte. Ein zweites Skript berechnet daraufhin basierend auf im Skript gespeicherten Preisen die Gesamtkosten, die anschließend dargestellt werden. Sollte die getankte Menge größer oder gleich 100 Liter sein, wird nochmals 5% Rabatt vom Preis abgezogen.

2. Erstellen Sie ein Programm, das die erreichten IHK Punkte, die in einem Formular eingegeben werden, in Noten umrechnet und ausgibt. Folgende Punkteverteilung sieht die IHK vor:
Sollten ein unpassender Wert eingegeben werden, wir der Nutzer darauf aufmerksam gemacht.

| | |
|---------------------|--------------------|
| sehr gut | ausreichend |
| 100 - 92 Punkte | 66 - 50 Punkte |
| gut | mangelhaft |
| 91 - 81 Punkte | 49 - 30 Punkte |
| befriedigend | ungenügend |
| 80 - 67 Punkte | 29 - 0 |

2.6 Schleifen



Ich muss mich dringend über weitere Kontrollstrukturen und hierbei insbesondere über Schleifen informieren, eine Schleife, die foreach-Schleife, hatten wir ja schon einmal benutzt. Nur leider habe ich gerade überhaupt keine Zeit hierfür, da ich gerade beruflich auf Hawaii unterwegs bin. Daher ergeht folgender Arbeitsauftrag an Sie:

Bereiten Sie eine kurze Schulung vor, in der Ihre Mitschüler die Ihnen zugeteilte Wiederholungsstruktur kennenlernen.

Überlegen Sie sich hierbei zuerst, wie Sie diese Thematik Ihren Mitschülern näherbringen können. Nach einer kurzen Einführung, die sowohl die Syntax als auch das Struktogramm dieser Struktur beinhalten muss, sollten Ihre Mitschüler auch eine kleine von Ihnen erstellte Aufgabe lösen können. Bitte bereiten Sie zu diesem Zwecke Ihren Mitschülern auch ein Handout vor.

Bearbeitungszeit: 70 Minuten

Präsentations- und Übungszeit: 30 Minuten

Übungen zu Schleifen:

1. Erstellen Sie unter Nutzung einer Wiederholungsstruktur ein Programm, das die Zahlen 1 bis 1000 ausgibt.

2. Erstellen Sie unter Nutzung einer Wiederholungsstruktur ein Programm, das die Zahlen 1000 bis 0 in 100er Schritten ausgibt.

3. Schreiben Sie ein Programm, dass eine vom Nutzer eingegebene Zahl wiederholt halbiert bis diese kleiner als 0,001 ist.

4. Erzeugen Sie die Ausgabe des kleinen Einmaleins mit Hilfe einer geschachtelten For-Schleife in einer Tabelle, die wie folgt aussieht:

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|-----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
| 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60 |
| 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 | 70 |
| 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80 |
| 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 | 90 |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |

Tipp: Schleifen lassen sich auch in sich verschachteln...

5. Erstellen Sie ein Programm, das den Nutzer fragt, wie viele PIN-Codes er erzeugt haben möchte. Erzeugen Sie daraufhin die gewünschte Anzahl von Codes und geben Sie diese an den Nutzer zurück. Gehen Sie davon aus, dass ein Code grundsätzlich vierstellig ist und nur aus Zahlen bestehen sollte.

Mögliche Erweiterung: Stellen Sie sicher, dass nur einmalige und keine mehrfach vorkommenden Codes zurückgegeben werden. Weiterhin können Sie die Anzahl der Codestellen dynamisch vom Nutzer vorgeben lassen. *Tipp:* Nutzen Sie zum Erzeugen der Zufallszahl folgende Funktion:

<http://php.net/manual/de/function.random-int.php>

6. Erstellen Sie ein Programm (inklusive Struktogramm vorab), das den Verlauf des Wertverlustes von Investitionsgütern darstellt. Der Nutzer soll dabei den Anschaffungswert und den die Abschreibungsdauer eingeben. Basierend auf diesen Werten werden dann der Anfangswert, der Abschreibungsbetrag sowie der Restwert am Jahresende ausgegeben. Die Ausgabe könnte wie folgt aussehen

| Nutzungsjahr | Anfangswert | Abschreibung | Restwert |
|--------------|-------------|--------------|----------|
| 1 | x.ooo,xx | x.ooo,xx | x.ooo,xx |
| n | x.ooo,xx | x.ooo,xx | x.ooo,xx |

Zur Info:

- Abschreibungsbetrag = Anschaffungswert/Abschreibungsdauer (für alle Jahre gleich hoch)
- Der Restwert am Jahresanfang = Anschaffungswert am Jahresanfang - Abschreibungsbetrag des Jahres



Didli dudli. Du hast im Übrigen gerade die drei Kontrollstrukturen der Strukturierten Programmierung kennengelernt. Hierbei handelt es sich um ein programmiersprachenübergreifendes *Programmierparadigma*, das verlangt, sich auf der untersten Ebene des Codes auf lediglich drei Kontrollstrukturen zu beschränken: **Sequenz, Verzweigung, Schleifen.**

Die bekannteste Konsequenz dieses Prinzips ist die Vermeidung von Sprunganweisungen wie *GOTO*. Durch die konsequente Implementierung von Teilprogrammen und von Kontrollstrukturen vermeiden strukturierte Programme Codewiederholungen, was Vorteile bei der Suche von Programmfehlern mit sich bringt, die Programme kürzer und übersichtlicher macht und daher die Wartung von Software vereinfacht. Die strukturierte Programmierung ist heute in fast allen Bereichen eine Selbstverständlichkeit.

2.7



Mein lieber Homer. Ich komme aus der Zukunft um dich auf einen besorgniserregenden Qualitätsmangel in deinem Code aufmerksam zu machen. Du musst doch stets folgenden Grundsatz beachten:

Don't Repeat Yourself (DRY-Prinzip)

```

//?php
//Anlegen

$booksNovel = [ [ 'title' => 'To Kill A Mockingbird', 'author' => 'Harper Lee', 'available' => true, 'pages' => 336, 'isbn' => 9780061120084 ], [ 'title' => '1984', 'author' => 'George Orwell', 'available' => true, 'pages' => 267, 'isbn' => 9780547249643 ], [ 'title' => 'One Hundred Years Of Solitude', 'author' => 'Gabriel Garcia Marquez', 'available' => false, 'pages' => 457, 'isbn' => 9785267006323 ] ];

$booksNonFiction = [ [ 'title' => 'PHP 7 Programming Cookbook', 'author' => 'Doug Bierer', 'available' => false, 'pages' => 610, 'isbn' => 1785883445 ] ];

//Ausgabe
foreach ($booksNovel as $book)
{
    foreach ($book as $key => $value) {
        echo $key . " :" . $value;

        if (!$book['available'])
        {
            echo "Not available";
        }

        echo "</br>";
    }
}

//Ausgabe
foreach ($booksNonFiction as $book)
{
    foreach ($book as $key => $value) {
        echo $key . " :" . $value;

        if (!$book['available'])
        {
            echo "Not available";
        }

        echo "</br>";
    }
}

```



Ihre Aufgabe: Klären Sie mit Ihrem Banknachbarn, welche Codequalitätsmängel in obigen Code festzustellen ist. Was sind die Gefahren, die daraus erwachsen? Halten Sie Ihre Erkenntnisse im nachfolgenden Feld fest.

In unserer obigen Onlinebuchhandelsseite haben wir eine Funktionalität genutzt, die ausgelagert werden könnte. Die Funktion erwartet ein Array, welches ein Buch darstellen soll und erzeugt eine HTML-Ausgabe für das Buch. Der Code ist der gleiche wie zuvor, nur jetzt in einer Funktion gekapselt.

```

//functions.php
<?php
    function printableTitle(array $book): string {
        $result = '<i>' . $book['title'] . '</i> - ' . $book['author'];
        if (! $book['available']) {
            $result .= '<b>Not available</b>';
            echo "<br>";
        }
        return $result;
    }
}

<?php
require_once 'functions.php';
//Anlegen

$booksNovel = [ [ 'title' => 'To Kill A Mockingbird', 'author' => 'Harper Lee', 'available' => true, 'pages' => 336, 'isbn' => 9780061120084 ], [ 'title' => '1984', 'author' => 'George Orwell', 'available' => true, 'pages' => 267, 'isbn' => 9780547249643 ], [ 'title' => 'One Hundred Years Of Solitude', 'author' => 'Gabriel Garcia Marquez', 'available' => false, 'pages' => 457, 'isbn' => 9785267006323 ] ];

$booksNonFiction = [ [ 'title' => 'PHP 7 Programming Cookbook', 'author' => 'Doug Bexler', 'available' => false, 'pages' => 610, 'isbn' => 1785883445 ] ];

//Ausgabe
foreach ($booksNovel as $book)
{
    foreach ($book as $key => $value) {
        echo printableTitle($book);
    }
}

//Ausgabe
foreach ($booksNonFiction as $book)
{
    foreach ($book as $key => $value) {
        echo printableTitle($book);
    }
}

```

Wichtig ist, dass obige Datei nun die Datei *functions.php* einbinden muss um dann die darin definierte Funktion nutzen zu können. Die Schleifen sehen nun deutlich aufgeräumter aus. Doch stellt sich hier die Frage, ob eine weitere Optimierung denkbar wäre?

Ihre Aufgabe: Klären Sie mit Ihrem Banknachbarn, welche Optimierungsmöglichkeit in obigen Code noch vorhanden ist? Halten Sie Ihre Erkenntnisse wiederum im nachfolgenden Feld fest.





Informationstext zu Funktionen: Bitte lesen Sie sich nun nachfolgenden Informationstext durch und markieren Sie sich die Ihrer Meinung nach wichtigen Stellen.

Eine Funktion ist ein wiederverwendbarer Code-Block, der eigenen Funktionsnamen trägt. Sie kennen bereits einige vordefinierte Funktionen wie `empty()`, `in_array()` oder `var_dump()`, die von PHP bereitgestellt werden. Es ist aber auch ohne größeren Aufwand möglich, eigene Funktionen zu erstellen, um Funktionalitäten zu kapseln um diese mehrfach verwendbar zu machen.

Funktionen deklarieren

Eine Funktion wird definiert durch

- das Schlüsselwort `function`
- den Funktionsnamen
- die optionalen Argumente
- den Anweisungsblock
- den Datentyp eines möglichen Rückgabewertes

Ein einfaches Beispiel:

```
<?php
function addNumbers($a, $b) {
    $sum = $a + $b;
    return $sum;
}

$result = addNumbers(2, 3);
```

Der Funktionsname lautet `addNumbers`; er nimmt zwei Parameter entgegen: `$a` und `$b`. Innerhalb des Funktionsblocks wird eine neue Variable `$sum` definiert, die die Summe der beiden Argumente berechnet. Anschließend gibt die Funktion das Ergebnis mit Hilfe des `return`-Statements zurück. Um die Funktion nutzen zu können, muss man diese mit dem Funktionsnamen aufrufen und zwei Argumente übergeben. Diese werden durch Komma getrennt.

Funktionsargumente

Eine Funktion braucht häufig Informationen, die man ihr zur Verfügung stellen muss. Diese Argumente erhalten einen lokalen Namen. Die Informationen werden in der jeweiligen Reihenfolge an die Argumente übergeben.

Funktionen können auch optionale Parameter haben, d.h. man muss ihnen keinen Wert übergeben, sondern stellt einen Default-Wert zur Verfügung. Siehe dazu folgendes Beispiel:

```
<?php
function addNumbers($a, $b, $printResult = false) {
    $sum = $a + $b;
    if ($printResult) {
        echo 'The result is ' . $sum;
    }
    return $sum;
}

$sum1 = addNumbers(1, 2);
$sum1 = addNumbers(3, 4, false);
$sum1 = addNumbers(5, 6, true); // it will print the result
```

Diese Funktion hat zwei notwendige und ein optionales Argument. Wenn keine Wert für das dritte Argument übergeben wird (`$printResult`), dann wird in diesem Falle der Wert `false` verwendet.

Die Argumente, die der Funktion übergeben werden, sind Kopien der Originalwerte. Wenn sie innerhalb der Funktion verändert werden, hat das keine Auswirkungen auf die Werte der Originalvariablen. Man nennt dies Copy By Value.

```
<?php
function modify($a) {
    $a = 3;
}
$a = 2;
modify($a);
var_dump($a); // prints 2
```

Return-Statement

Man kann zwar mehrere RETURN-Statements innerhalb einer Funktion schreiben, aber PHP wird nach dem Ausführen des ersten Statements die Funktion verlassen. Mehrere RETURN-Statements sind dann sinnvoll, z.B. wenn sie verschiedene Bedingungen (if) innerhalb einer Funktion haben.

```
<?php
function loginMessage() {
    if (isset($_COOKIE['username'])) {
        return "You are " . $_COOKIE['username'];
    } else {
        return "You are not authenticated.";
    }
}
```

Wenn die Funktion nichts zurückgeben soll, kann man auf die RETURN-Anweisung innerhalb einer Funktion verzichten. Die Funktion endet dann einfach mit dem Erreichen der letzten Anweisung innerhalb des Funktionsblocks.

Gültigkeitsbereich (Scope) von Variablen

Ein weiterer wichtiger Punkt ist der Gültigkeitsbereich (Scope) von Variablen. Da die Variable `$sum` innerhalb des Funktionsblockes definiert wurde, ist sie auch nur innerhalb dieses Blockes verwendbar. Hätte man die Variable `$sum` außerhalb der Funktion definiert, wäre sie innerhalb der Funktion nicht sichtbar gewesen, es sei denn, man hätte sie als Parameter übergeben. Die Übergabe erfolgt dabei als Copy By Value.

```
1  <?php
2
3      $sum = 0;
4      $summe2 = 20;
5      echo $sum;
6      echo $summe2;
7
8      function addiere($a , $b, $temp)
9      {
10          echo $sum;
11          $summe2 = $a + $b;
12          echo "<p>";
13          echo $sum;
14      }
15
16      addiere(4, 5, $summe2);
17      echo $summe2;
20
```

| Notice: Undefined variable: sum in C:\xampp\htdocs\scratch.php on line 10 | | | | |
|--|--------|--------|-----------|-------------------|
| Call Stack | | | | |
| # | Time | Memory | Function | Location |
| 1 | 0.4065 | 357728 | {main}() | ...scratch.php:0 |
| 2 | 0.4065 | 357728 | addiere() | ...scratch.php:16 |

| Notice: Undefined variable: sum in C:\xampp\htdocs\scratch.php on line 13 | | | | |
|--|--------|--------|-----------|-------------------|
| Call Stack | | | | |
| # | Time | Memory | Function | Location |
| 1 | 0.4065 | 357728 | {main}() | ...scratch.php:0 |
| 2 | 0.4065 | 357728 | addiere() | ...scratch.php:16 |



Fragen an Dr. Simpson (jun.)

F: Unterstützt PHP eigentlich überladene Methoden?
A: Nein - PHP unterstützt keine überladenen Methoden.

Überladen (overloading) bedeutet, dass man für zwei Funktionen identische Funktionsnamen verwendet, die aber verschiedene Argumente besitzen. Da man in PHP Argument definieren kann, ohne den Datentyp angeben zu müssen, wüsste PHP deshalb nicht, welche Funktion aufgerufen werden soll.

F: Ist es auch möglich einer Funktion ein Argument als Referenz zu übergeben?

A: Ja. Man signalisiert dieses Verhalten durch ein Voranstellen eines Ampersands & vor dem Variablenamen.

```
<?php
function modify(&$a)
{ $a = 3; }

Dies ist allerdings grundsätzlich nicht zu empfehlen.
Zwar gibt es in PHP Funktionen, die Argumente bei
Referenz übernehmen, so zum Beispiel bei der
Übergabe von Arrays. Dieses Verhalten jedoch
künstlich mit obigen Ansatz zu realisieren, sollte
vermieden werden, da sonst der nachfolgende
Programmierer an dieser Stelle eventuell überrascht
werden könnte.
```

F: Gibt es nicht neben der oben gezeigten Einbindung von Funktionen über **require_once** 'functions.php'; noch weitere Möglichkeiten, externe Skripte und deren Funktionen zur Nutzung im aktuellen Skript einzubinden?

A: Ja, natürlich. Folgende Einbindungsmöglichkeiten stehen zur Verfügung:

include: Dies wird versuchen, die angegebene Datei bei jedem Aufruf einzubinden, auch wenn dies vorher schon geschehen war. Falls die Datei nicht gefunden wird, wird PHP eine Warnung ausgeben, aber das Programm weiter fortsetzen.

require: Dies macht das Selbe wie include; wirft aber einen Fehler, falls die Datei nicht gefunden wurde.

include_once: Macht das gleiche wie include, wird die Datei aber nur beim ersten Mal einbinden. Nachfolgende Aufrufe werden ignoriert.

require_once: Mach das gleiche wie require; nachfolgende Aufrufe werden aber ignoriert.

Übungen zu Funktionen:

1. Erstellen Sie eine Funktion, die zwei Zahlen übergeben bekommt und die Summe dieser Zahlen zurückgibt. Die Ausgabe dieser Summe soll nach erfolgter Rückgabe aus der Funktion ausgegeben werden.
2. Programmieren Sie eine Funktion, die die Fakultät von einer übergebenen natürlichen Zahl errechnet und an den Aufrufer zurückgibt. Beispiel $3! = 1 \cdot 2 \cdot 3 = 6$
3. Für ein Unternehmen sollen wir ein Programm entwickeln, welches für erfasste LKW automatisch die entsprechende LKW-Mautgebühr berechnet.
Folgende Mautgebühren wurden uns dafür vorgegeben:

| Schadstoffklasse | A | B | C | D | E | F |
|------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| bis 3 Achsen | 12,50 ct/km | 14,60 ct/km | 15,70 ct/km | 18,80 ct/km | 19,80 ct/km | 20,80 ct/km |
| ab 4 Achsen | 13,10 ct/km | 15,20 ct/km | 16,30 ct/km | 19,40 ct/km | 20,40 ct/km | 21,40 ct/km |

Aufgabenstellung Schreiben Sie eine Funktion, die nach der oben beschriebenen Regelung die LKW-Maut für eine Fahrt berechnet. Die Mautgebühr ergibt sich unter anderem aus dem Kilometerpreis mal der Anzahl der gefahrenen Kilometer.

4. Gesamtwiederholung (angelehnt an GA I Winter 2014)

- a) Erstellen Sie ein zweidimensionales Array mit folgenden Daten

| Einkommen | Miete |
|-----------|----------|
| 4.200,00 | 1.200,00 |
| 900,00 | 340,00 |
| 1.800,00 | 600,00 |
| 3.600,00 | 1.100,00 |
| 2.700,00 | 800,00 |
| 5.900,00 | 1.300,00 |

- b) Erstellen Sie nun eine Methode mit dem Namen *prozente*, die auf Basis des erstellten Arrays die vorhandenen Einkommen und Mieten einer Einkommensgruppe zuordnet und dort das Einkommen je Gruppe sowie die Miete je Gruppe aufaddiert. Die gewünschte Anzahl der Einkommensgruppen und deren Staffelung in EUR werden der Methode als Parameter *anzahlGruppen* übergeben. Das folgende Beispiel zeigt die Zuordnung von Einkommen zu Einkommensgruppen für den Fall, dass die Methode folgende Parameter übergeben bekommt: *anzahlGruppen=5* und *staffelung=1000*.

| Einkommen (in EUR) | Rechnung zur Ermittlung der Einkommensgruppe | Zuordnung zu Einkommensgruppe | Bereich (in EUR) |
|--------------------|--|-------------------------------|------------------|
| 900,00 | $900 / 1.000 = 0,9$ | 0 | < 1.000 |
| 1.800,00 | $1.800 / 1.000 = 1,8$ | 1 | 1.000 – 1.999 |
| 2.700,00 | $2.700 / 1.000 = 2,7$ | 2 | 2.000 – 2.999 |
| 3.600,00 | $3.600 / 1.000 = 3,6$ | 3 | 3.000 – 3.999 |
| 4.200,00 | $4.200 / 1.000 = 4,2$ | 4 | ≥ 4.000 |
| 5.900,00 | $5.900 / 1.000 = 5,9$ | 4 | ≥ 4.000 |

Wie Sie erkennen können, wird zum Beispiel der erste Datensatz, ein Einkommen in Höhe von 900,00 Euro der ersten Einkommensgruppe zugeordnet, indem das Einkommen durch die Staffelung geteilt wird. Das gerade überprüfte Einkommen sowie die dazu passende Miete muss in der entsprechenden Einkommensgruppe Gruppe gespeichert werden.

Die Methode *prozente* soll nach der Zuordnung aller Einkommen zu Einkommensgruppen den durchschnittlichen Prozentsatz je Einkommensgruppe, der für Miete aufgewendet wird, ermitteln. Die Daten dieser Berechnung sollen in einem eindimensionalen Array gespeichert werden und an den Aufrufer der Methode zurückgegeben werden. Dieses könnte wie folgt aussehen (die linke Spalte entspricht dem Index des Arrays und zugleich der Einkommensgruppe).

| | |
|---|----|
| 0 | 38 |
| 1 | 33 |
| 2 | 30 |
| 3 | 31 |
| 4 | 25 |

c) Erstellen Sie nun eine weitere Methode *sortProzente*, die als Übergabeparameter das gerade erhaltene eindimensionale Array aus der Methode *prozente* erhält und daraus ein zweidimensionales Array mit aufsteigend sortierten prozentualen Anteilen erstellt. Folgendes Beispiel zeigt den möglichen Rückgabewert der Methode:

Beispiel für Ausgangs-Array *prozente*

| Einkommensgruppe (entspricht dem Index des Arrays) | Prozentualer Anteil |
|--|----------------------------|
| 0 | 38 |
| 1 | 33 |
| 2 | 30 |
| 3 | 31 |
| 4 | 25 |

Beispiel für Ergebnis-Array *sortProzente*

| Einkommensgruppe (entspricht dem Index des Arrays) | Prozentualer Anteil |
|--|----------------------------|
| 4 | 25 |
| 2 | 30 |
| 3 | 31 |
| 1 | 33 |
| 0 | 38 |

Verwenden Sie dazu folgende Idee zum Sortieren:

- Erstellen eines zweidimensionalen Arrays *sortProzente*:
Anzahl Zeilen: Länge des Arrays *prozente*
Anzahl Spalten: 2
- Initialisieren der Spalte 0 mit den Werten 0, 1, 2, 3, ..., Länge *prozente* - 1
(Einkommensgruppen)
- Initialisieren der Spalte 1 mit den Werten aus dem Array *prozente* (prozentualer Anteil der Miete je Einkommensgruppe)
- *Durchgang 1*: für alle j von 0 bis Anzahl Zeilen von *sortProzente* - 2
Wenn der *prozentuale Anteil* (Wert) der Zeile j größer ist als der *prozentuale Anteil* (Wert) der Zeile (j+1), dann Daten der Zeile j mit den Daten der Zeile (j+1) vertauschen.
- Wiederholen von *Durchgang 1* bis die Daten gemäß Aufgabenstellung sortiert sind.

3. Arbeiten mit Dateien

Homer, mein guter Freund. Sicherlich hast du Interesse diese wertvolle Diskette zu erwerben. Sie enthält eine Sammlung von völlig legal erworbenen Büchern, die du für mich als kleiner Freundschaftsdienst über deinen Onlinebuchshop verkaufen kannst. Welche Bücher vorhanden sind, kannst du der beiliegenden Datei entnehmen, Partner.

Was bin ich nur für ein Glückspilz. Ich bin zwar kein religiöser Mensch, aber wenn es dich wirklich gibt, DANKE ich dir Superman!



Wie man Dateien importiert, erscheint mir doch ein Leichtes zu sein. Nutze daher den von mir erstellten Informationstext und passe anschließend meine Startseite so an, dass diese unter dem Suchfeld alle Bücher auflistet mit den entsprechenden Daten, die sich in der Datei im JSON-Format befinden.

Ach ja, die Datei liegt bei dir auf dem Klassenaustauschlaufwerk!



Informationstext zum Arbeiten mit Dateien: Bitte lesen Sie sich nun nachfolgenden Informationstext durch und markieren Sie sich die Ihrer Meinung nach wichtigen Stellen

Dateien lesen

In unserem Code definieren wir eine Liste von Büchern, die wir bisher innerhalb des Codes halten. Dies ist unpraktisch bei einer größeren Anzahl von Büchern, womit wir uns Gedanken machen müssen, wie wir diese Daten auslagern.

Auch wenn man natürlich in Datenbanken denken könnte, bietet das Dateisystem eine Vielzahl von Formaten, in denen man diese Dateien speichern könnte, wie beispielsweise CSV oder XML-Dateien. Heutzutage ist eines der weitverbreitetsten Formate zum Speichern von Daten in Webanwendungen das JSON-Format. PHP erlaubt es Arrays in das JSON-Format und umgekehrt zu speichern und benötigt dafür nur zwei Funktionen: `json_encode` und `json_decode`.

Folgenden Informationen befinden sich in der Datei `books.json`:

```
[
{
  "title": "To Kill A Mockingbird", "author": "Harper Lee",
  "available": true, "pages": 286, "isbn": 9780061120084
},
{
  "title": "1984", "author": "George Orwell", "available": true, "pages": 267,
  "isbn": 9780547249643
},
{
  "title": "One Hundred Years Of Solitude",
  "author": "Gabriel Garcia Marquez", "available": false, "pages": 457,
  "isbn": 9785267006323
}
]
```

Der Quellcode-Schnipsel ist eine JSON-Darstellungen unseres Arrays aus PHP. Wir lesen diese Datei nun mit Hilfe der Funktion `file_get_contents` ein und wandeln sie in einen PHP-Array um. Ersetzen Sie den Array mit diesen beiden Zeilen Code.

```
$booksJson = file_get_contents('books.json');
$books = json_decode($booksJson, true);
```

Mit der ersten Funktion laden wir den Inhalt der gesamten Datei in eine Variable als JSON-String. Die zweite Funktion wandelt dann diesen String in einen Array um. Der zweite Parameter der Funktion steht für einen Array, ansonsten würde PHP Objekte erzeugen, was erst im weiteren Verlauf eine Rolle spielen wird.

```
$booksJson = file_get_contents('/home/user/bookstore/books.json');
$booksJson = file_get_contents(__DIR__ . '/books.json');
```

Wenn man Dateien lädt, muss man deren absoluten oder relativen Pfad kennen. Zum Aufbau des Pfadnamens ist die Konstante `__DIR__` hilfreich, die immer den absoluten Pfadnamen des PHP-Files liefert.

Writing files

Sollten Änderungen zur Laufzeit erfolgt sein, sollten diese natürlich wieder in die persistente Quelle – in unseren Fall der JSON-Dateien – überführt werden. Dies kann mit folgenden Befehlen erfolgen:

```
$booksJson = json_encode($books);
file_put_contents(__DIR__ . '/books.json', $booksJson);
```

`JSON_encode` macht das Gegenteil von `json_decode`; es wandelt einen Array in einen JSON-String um. Die `file_put_contents`-Methode schreibt dann in die als Parameter übergebene Datei.



Aufgabe: Ordnen Sie folgende Begrifflichkeiten aus den bisherigen Kapiteln Ihrer passenden Beschreibung zu.

| | |
|--------------------------|---|
| <code>\$POST</code> | Takes a JSON encoded string and converts it into a PHP variable. |
| <code>JSON_encode</code> | If called from within a function, the statement immediately ends execution of the current function, and returns its argument as the value of the function call. |
| <code>__DIR__</code> | The directory of the file. If used inside an include, the directory of the included file is returned. |
| <code>Return</code> | Returns the JSON representation of a value |
| <code>JSON_decode</code> | The statement is similar to a series of IF statements on the same expression. In many occasions, you may want to compare the same variable (or expression) with many different values, and execute a different piece of code depending on which value it equals to. |
| <code>switch</code> | An associative array of variables passed to the current script via the HTTP POST method |

4. Objektorientierte Programmierung

4.1 Einführung in die Objektorientierung



Lieber Freund Homer. Du begebst erneut einen schwerwiegenden Fehler. Dein Code ist nicht objektorientiert aufgebaut. Schau dir doch die Welt an und erkenne, was Alan Kay (Erfinder der ersten OOP-Sprache Smalltalk, geboren in Springfield ;-)) zur Objektorientierung sagte:

1. Alles ist ein Objekt
2. Objekte kommunizieren durch das Senden und Empfangen von Nachrichten (welche aus Objekten bestehen),
3. Objekte haben ihren eigenen Speicher (strukturiert als Objekte),
4. Jedes Objekt ist die Instanz einer Klasse (welche ein Objekt sein muss),
5. Die Klasse beinhaltet das Verhalten aller ihrer Instanzen (in der Form von Objekten in einer Programmliste),
6. Um eine Programmliste auszuführen, wird die Ausführungskontrolle dem ersten Objekt gegeben und das Verbleibende als dessen Nachricht behandelt"



Didli dudli, das ist doch Quatsch. Objektorientierte Programmierung führt zu unnötig viel Code, der in PHP, da PHP auch nicht objektorientiert entwickelt werden kann, völlig unnötig ist. Also ich kann nun wirklich keinen Nutzen aus dem objektorientierten Ansatz erkennen.



Flender, du bist ein dummer Mensch. Dein strukturierter Ansatz ist mindestens ebenso rückständig wie deine völlig belödeten Frisur. Die Strukturierte Programmierung hat ein paar Nachteile. Ein Nachteil ist zum Beispiel, dass zwischen einer Daten- und Funktionsschicht Unterschiede sind. Eine ganzheitliche systemische Betrachtung bleibt dabei außen vor.

Vier Aspekte machen im Gegensatz dazu die Objektorientierung überlegen. Homer wird dir diese sicherlich gerne erklären.



Natürlich Herr Professor aus der Zukunft. Ich vervollständige nur noch schnell meine Notizen zu den vier Prinzipien.



Ihre Aufgabe: Lesen Sie nachfolgenden Informationstext durch und vervollständigen Sie anschließend mit passenden Erklärungen Homers Notizen.

Klasse =

Objekt=

Datenkapselung=

Vererbung=

(später zu ergänzen)

Polymorphie=

(später zu ergänzen)

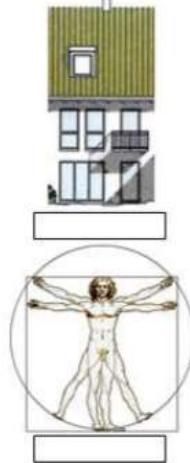
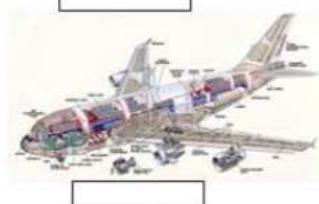


Informationstext zur OOP:

Heutzutage ist OOP eines der häufigsten Programmierparadigmen⁷ und PHP unterstützt es hervorragend. Objekte kann man sich als Real-Life-Elemente vorstellen. Jedes Objekt hat Attribute und damit Werte, welches es von anderen Objekten des gleichen Typs (Klasse) unterscheidet. Es kann auch über seine Methoden ein bestimmtes Verhalten ausführen. Ein Klasse definiert, wie ein Objekt aussieht und was es machen kann. Es ist so etwas wie eine Blaupause für ein Objekt.

⁷ Weitere Programmierparadigmen sind XXXXXX. + Leseempfehlung

 **Aufgabe:** Verbinden Sie mit einer Linie die konkreten Objekte mit ihrer Klassenvorlage. Notieren Sie jeweils unter dem Bild, ob es sich um ein Objekt oder um eine Klasse handelt.



Das objektorientierte Paradigma begreift daher die Welt als Objekte, die Daten halten, Methoden ausführen können und dabei mit anderen Objekten in einer Interaktion stehen.

Aufgabe: Überlegen Sie sich, welche Klassen es in Ihrer beliebten Berufsschule gibt.



Aufgabe: Nehmen wir unser Homers Bookstore-Beispiel und denken darüber nach, welche echten Klassen es enthält. Halten Sie Ihrer Erkenntnisse hier fest: