

Dynamische Webprogrammierung

Skript

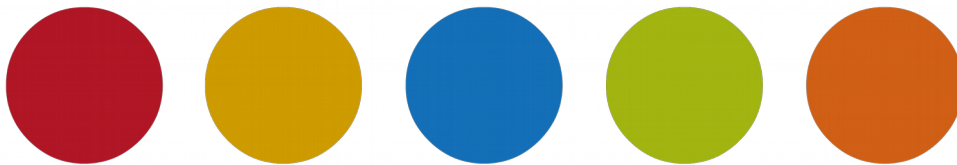
Materialsammlung

Schulung:	Informatik und Wirtschaftsinformatik
-----------	--------------------------------------

Unterrichtsbegleitendes E-Learning:

[→ E-Learning](#)

Stand: 19. Apr 2020



© Christine Janischek



<https://edublog.emotionalspirit.de/>

Inhaltsverzeichnis

1 Allgemeines.....	4
1.1 Arbeits- und Informationsmaterial: Einführung in PHP.....	6
1.2 Leittext: Einführung in PHP.....	7
2 Grundgerüst eines dynamischen Web-Layouts in PHP.....	11
2.1 Informationsblatt: Grundgerüst eines dynamisches Web-Layout.....	11
2.2 Leittext: Ein dynamisches Web-Layout in PHP.....	12
3 Formulare auswerten: Bmirechner.....	18
3.1 Informationsblatt: Formulare.....	18
3.2 Aufgabe: Bmirechner.....	20
3.3 Leittext: Bmirechner.....	21
3.3.1 Leittext: Bmirechner – Formular erzeugen.....	21
3.3.2 Leittext: Bmirechner – Formular in das dynamische Layout einbetten.....	25
3.3.3 Leittext: Bmirechner – Berechnung und Anzeige des Ergebnisses.....	27
4 Übung - Formulare auswerten: Notenrechner.....	31
5 Kontrollstrukturen.....	32
5.1 Informationsblatt: Fallunterscheidungen.....	32
5.2 Aufgaben: Einführende Übungen zu Fallunterscheidungen.....	34
5.3 Aufgabe: Taschenrechner.....	36
5.4 Aufgabe: Rabattrechner.....	37
5.5 Aufgabe: Milchautomat.....	39
5.6 Aufgabe: Erweiterter Bmirechner.....	40
5.7 Aufgabe: Parkautomat.....	42
6 Wiederholstrukturen.....	43
6.1 Informationsblatt: Kontrollstrukturen → Wiederholstrukturen.....	43
6.2 Informationsblatt: Datencontainer.....	45
6.3 Arbeitsblatt: Umsatzrechner.....	46
6.4 Arbeitsblatt: Darlehnsrechner - Kontrollstrukturen → Wiederholstrukturen.....	48
7 PHP und Datenbanken.....	51
7.1 Einführung in Sessions.....	51
7.1.1 Informationsblatt.....	51
7.1.2 Arbeitsblatt.....	52
7.2 Einführung in die Verschlüsselung.....	55
7.2.1 Informationsblatt.....	55
7.2.2 Arbeitsblatt.....	56
7.3 Einführung Datenbankzugriff.....	60
7.3.1 Informationsblatt.....	60

7.3.2 Arbeitsblatt.....	64
7.4 Einführung Datenbankoperationen.....	65
7.4.1 Informationsblatt.....	65
7.4.2 Arbeitsblatt.....	67

1 Allgemeines



Das Skript schildert den Umgang mit der Programmiersprache PHP anhand von konkreten Beispielen die sich auch in den Unterricht im Fachbereich Wirtschaftsinformatik respektive im Fachbereich Informatik einbetten lassen. Alle Oberstufenlehrpläne der Sekundarstufe 2 (Jahrgang 2 in Klasse 13 Berufliches Gymnasium, Berufskolleg Jahrgang 1, Jahrgang 2 und im Berufskolleg Wirtschaftsinformatik Jahrgang 1) enthalten eine Einheit „Webprogrammierung“.

Aktuelle Versionen des Skriptes selbst und die im Skript behandelten Quellcodes können Sie online herunterladen und testen:

[Skript & Sources:](#)

[Aktuelle Unterrichtssoftware und Materialien](#)

[Webseiten mit PHP \(objektorientiert\) programmieren](#)

[Skript & Sources:](#)

[Webprogrammierung im Unterricht](#)

[E-Learning OOP](#)



Für alle Inhalte gilt natürlich das Urheberrecht. Ich selber achte auch darauf. Um Details zur Creative-Commons-Lizenz für die von mir selbst verfassten Texte, Quellcodes und Grafiken zu erhalten, klicken Sie links auf das CC-BY-NC-SA-Logo. Für Ergänzungs- und/oder Verbesserungsvorschläge schreiben Sie mir bitte eine E-Mail: cjanischek@gmx.de

Alle Quellangaben wurden nach bestem Gewissen genannt und aufgeführt. Permanent begleitende Literatur waren:

[PHP01]

Achour, Mehdi , Betz, Friedhelm, et al, "PHP-Handbuch", 2001-2015,
<https://secure.php.net/manual/de/index.php>, zuletzt geprüft am 23.11.15

[PHP02]

w3schools, "PHP 5 Tutorial", 1999-2015, <http://www.w3schools.com/php/>, zuletzt geprüft am 23.11.15

[WEB01]



w3schools, "HTML(5) Tutorial", 1999-2015, <http://www.w3schools.com/html/default.asp>, zuletzt geprüft am 23.11.15

[HRE01]

Landesinstitut für Schulentwicklung Baden-Württemberg, "Handreichung zur Lehrplaneinheit 'Objektorientierte Systementwicklung'", 1999-2015, http://www.ls-bw.de/Handreichungen/reihe_beruflich/spezifisch/bg/Informatik/HR_JG1_OO%20vorlaeufig.pdf , zuletzt geprüft am 23.11.15

[HRE02]

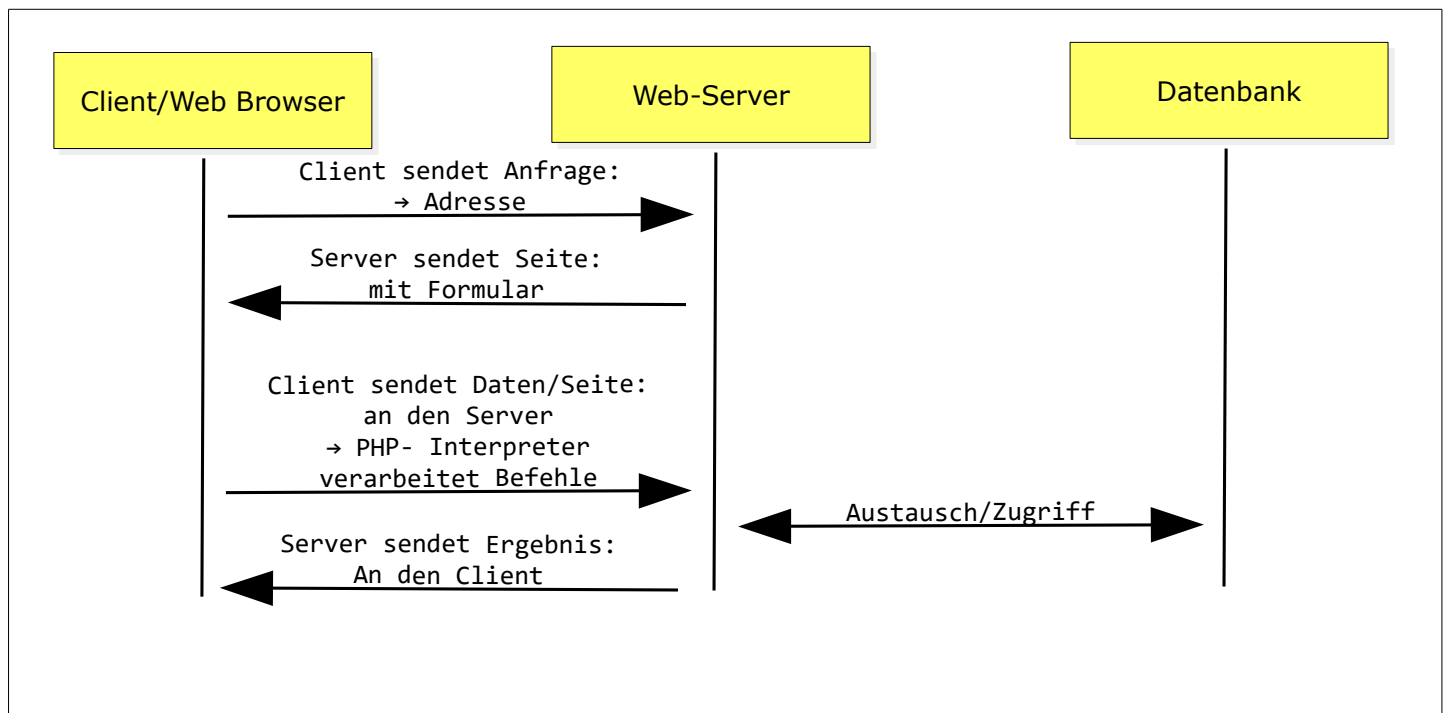
Landesinstitut für Schulentwicklung Baden-Württemberg, "Handreichung zur Lehrplaneinheit 'Design und Realisierung von Internetseiten'", 1999-2015, http://www.ls-bw.de/Handreichungen/reihe_beruflich/spezifisch/bg/Informatik/hr_inf_j2_bg.pdf , zuletzt geprüft am 23.11.15

[PBR01]

Vortrag ZAG Jahrestagung Informatik, "Ich-Kann-Liste", 2015, Pia Brunner

1.1 Arbeits- und Informationsmaterial: Einführung in PHP

Thema:	Einführung in PHP Autor: Christine Janischek Client-Server-Prinzip
--------	--

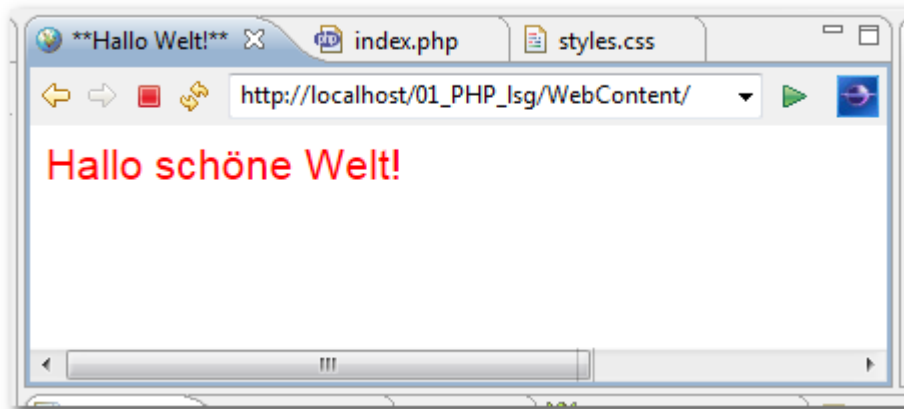


Grundkonzept und Technik¹

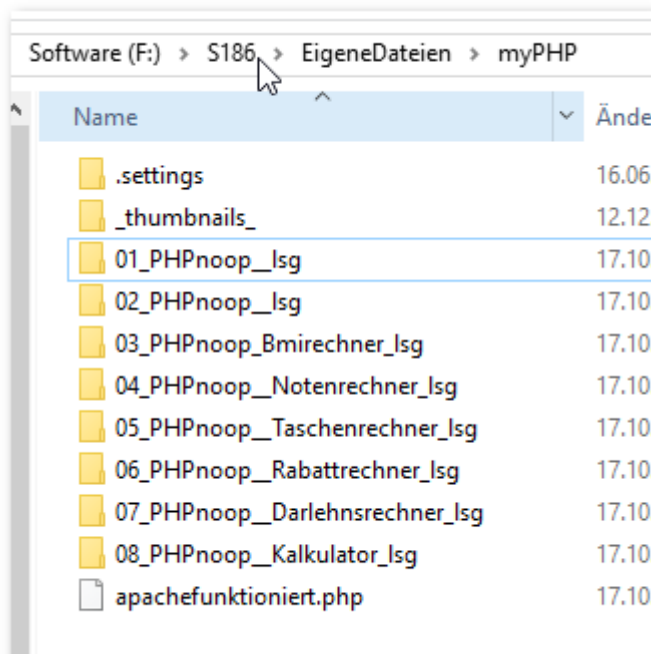
- x Webseiten werden durch den Zusatz von speziellen Tags programmierbar
- x Seitenbestandteile werden erst beim Aufruf der Seite durch den Client, quasi in Echtzeit zusammengebaut. Die Seite wird also dynamisch erzeugt!
- x Der Benutzer kann auf der Benutzeroberfläche interagieren (Eingaben tätigen und Schaltflächen bedienen). Die Anwendung verarbeitet die Daten, erzeugt und präsentiert ein Ergebnis. Durch PHP wird die HTML-Anwendung also erst richtig interaktiv und in gewisser Weise intelligent.
- x absolute Browser-Unabhängigkeit, da die Anwendungen auf dem Server ablaufen. Der Benutzer kann also keine Einschränkungen durch spezielle Einstellungen im Browser verursachen (z.B. deaktivieren von Javascript).
- x einfache Datenbankzugriffe auf eine MySQL-Datenbank, über die Schnittstelle ODBC auf Access-Datenbanken und sqlite-Datenbank (Mobile-Apps)
- x Grundlage z.B. für die meisten Content-Management-Systeme (CMS).

¹ Auszug in Anlehnung an das Skript von Johannes Kaiser

1.2 Leittext: Einführung in PHP



View: Ergebnis der Übung (index.php)



Arbeitsverzeichnis ausfindig machen.

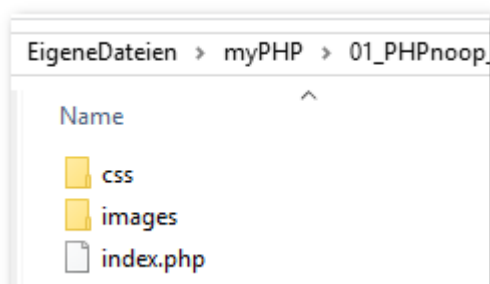
Öffnen Sie dazu Ihren USB-Stick und wählen Sie die Verzeichnisse → Informatikstick → EigeneDateien → myPHP.

Dieses Verzeichnis nutzen Sie künftig als Workspace für Ihre PHP-Projekte in Eclipse.

Hinweis:

Merken Sie sich unbedingt den Ort dieses Verzeichnisses:

\Informatikstick\EigeneDateien\myPHP

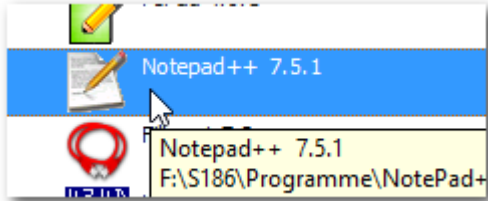
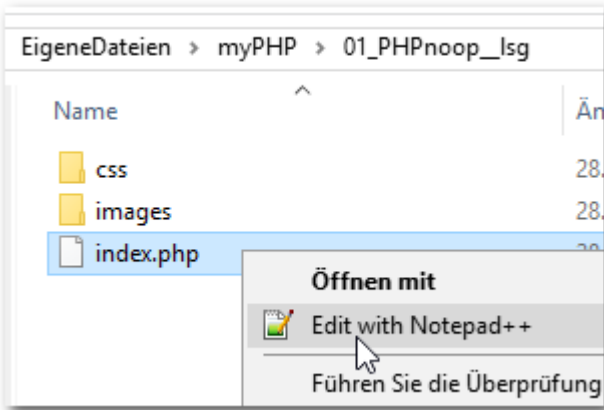
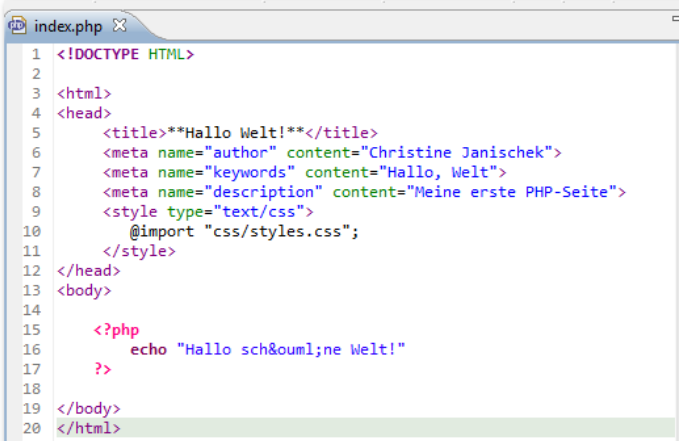


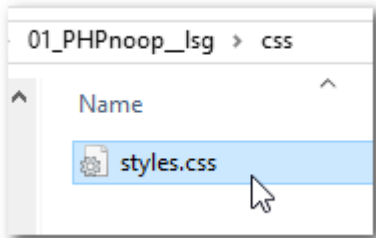
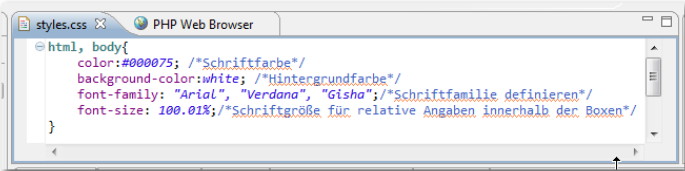
Projektverzeichnis und Struktur anlegen.

Legen Sie die nebenstehende Projektstruktur in das www/ Verzeichnis ihres Webserver oder nutzen Sie alternativ das Verzeichnis „myPHP“ in den „EigeneDateien“ Ihres Informatiksticks.

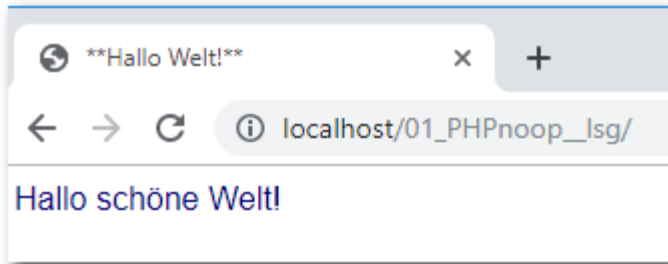
Hinweis:

Der local installierte Webserver → localhost

	(i.d.R. 127.0.0.1) muss auf dieses Verzeichnis zeigen, damit Sie die PHP-Anwendung im Browser testen können!
	<i>Notepad++ starten.</i> Starten Sie alternativ notepad++ über das Start-Menü Ihrer Digitalen Tasche (Informatik-stick).
	<i>index.php in einer geeigneten Entwicklungs-umgebung öffnen.</i> Notepad++ ist eine geeignete Entwicklungs-umgebung. In der Praxis findet Eclipse vielfach Anwendung. Öffnen Sie die Datei.
 Index.php	<i>Einfache Ausgaben in PHP erzeugen.</i> Ergänzen Sie nun den angezeigten Quellcode und Speichern Sie die Veränderung (STRG + S): <pre style="background-color: yellow; padding: 10px;"><?php echo "Hallo sch&ouml;ne Welt!" ?></pre> Eingabehilfe: <pre style="background-color: yellow; padding: 10px;"><!DOCTYPE HTML> <html> <head> <title>**Hallo Welt!**</title> <meta name="author" content="Ihr Name"> <meta name="keywords" content="Hallo, Welt"></pre>

	<pre> <meta name="description" content="Meine erste PHP-Seite"> <style type="text/css"> @import "css/styles.css"; </style> </head> <body> <!-- hier fehlt Quellcode ---> </body> </html> </pre>
	<i>Stildatei erzeugen.</i> Erzeugen Sie in der Entwicklungsumgebung eine Neue „noch leere“ Datei im Format „css“ und speichern Sie diese Datei unter dem Namen „styles.css“ im Verzeichnis „css“ ab.
	<i>Einfache Formatierung im CSS festlegen.</i> Öffnen Sie die Datei mit einem Doppelklick und übertragen Sie die folgenden Stilvorgaben: Eingabehilfe: <pre> html, body{ /*Schriftfarbe*/ color:#000075; /*Hintergrundfarbe*/ background-color:white; /*Schriftfamilie definieren*/ font-family: "Arial", "Verdana", "Gisha"; /*Schriftgröße für relative Angaben innerhalb der Boxen*/ font-size: 100.01%; } </pre>

im Web Browser



View: index.php

Testen Sie die Anwendung.

Öffnen Sie dazu den Internetbrowser und geben Sie den Pfad zur gerade erstellten Datei an. Geben Sie dazu den in der Grafik angezeigten Pfad ein.

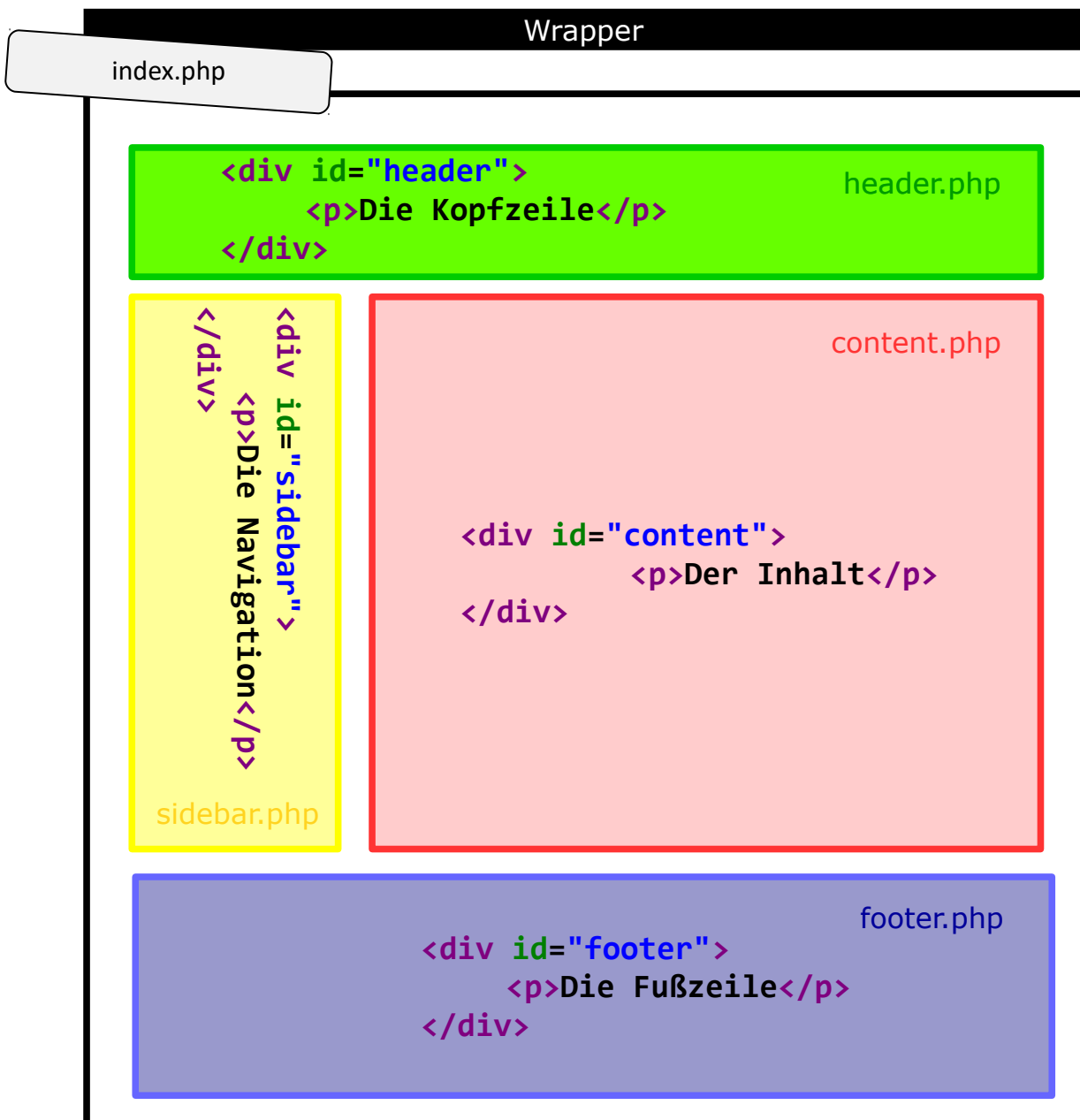
http://localhost/01_PHPnoop/

Herzlichen Glückwunsch Sie haben ihre erste PHP-Seite erstellt.

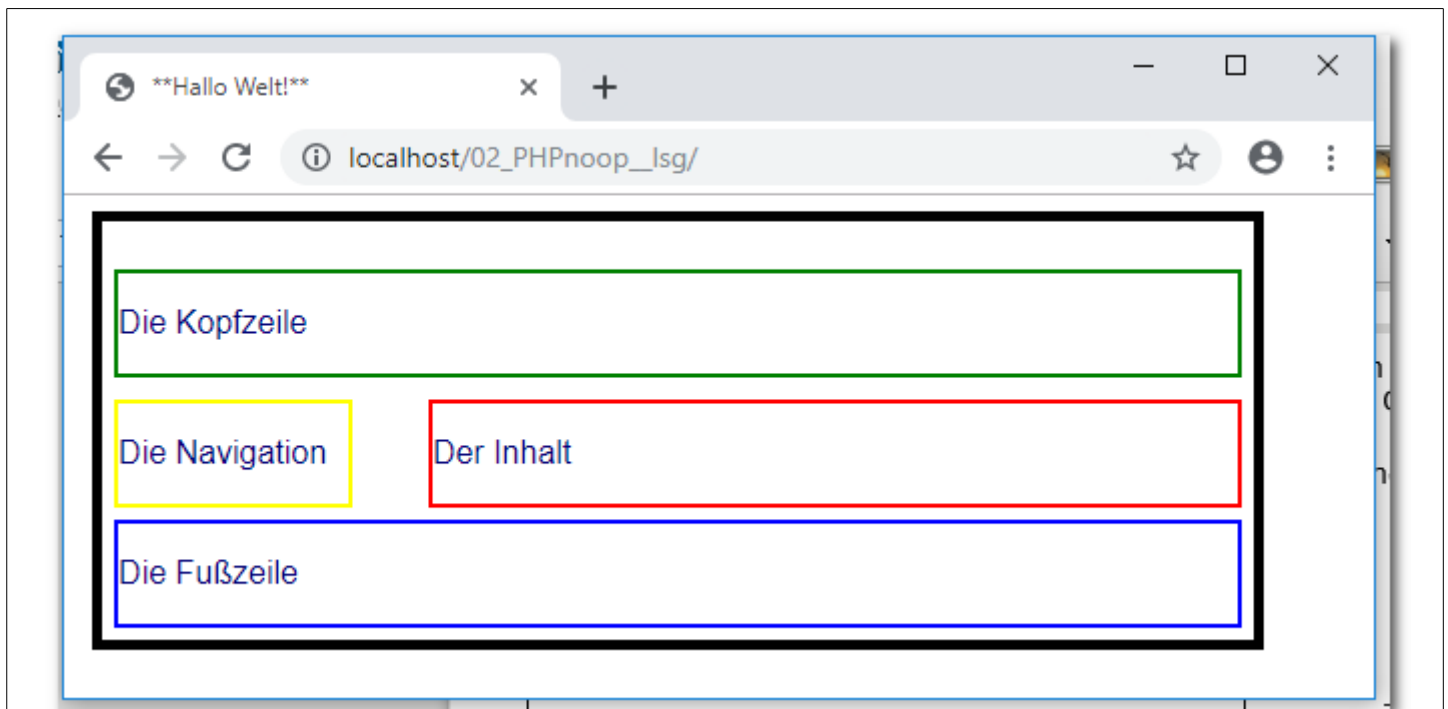
2 Grundgerüst eines dynamischen Web-Layouts in PHP

2.1 Informationsblatt: Grundgerüst eines dynamisches Web-Layout

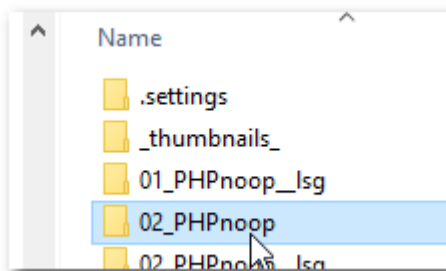
Thema:	Grundgerüst eines dynamischen Web-Layouts in PHP
	Autor: Christine Janischek



2.2 Leittext: Ein dynamisches Web-Layout in PHP



View: Ergebnis der Übung (index.php)

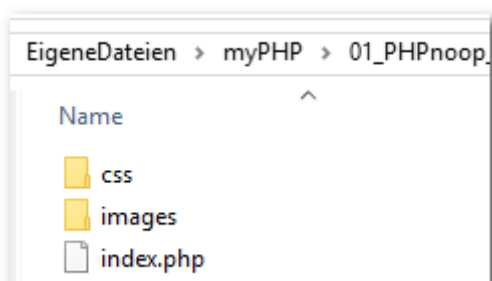


Projektverzeichnis anlegen.

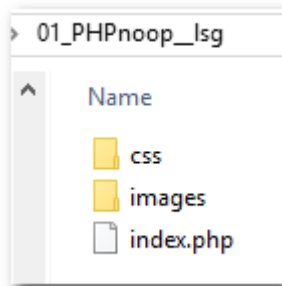
Legen Sie die nebenstehende Projektverzeichnis in das `www/` Verzeichnis ihres Webserver oder nutzen Sie alternativ das Verzeichnis „my-PHP“ in den „EigenenDateien“ Ihres Informatiksticks.

Hinweis:

Der local installierte Webserver → localhost (i.d.R. 127.0.0.1) muss auf dieses Verzeichnis zeigen, damit Sie die PHP-Anwendung im Browser testen können!



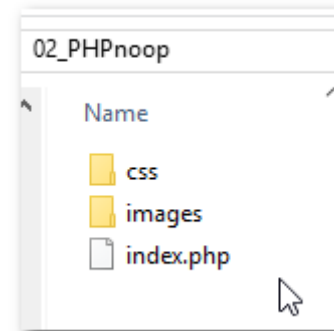
Letztes Projekt:



Prinzip der Wiederverwendung.

Öffnen und kopieren Sie die Ergebnisse des letzten Projektes des aktuellen Projektes.

Aktuelles Projekt:



Das Verzeichnis „images“ bleibt vorerst leer. Zu einem späteren Zeitpunkt das Verzeichnis nutzen.

Fortsetzung:

```
#header{
/*Rahmen 2 Pixel breit durchgehend grün*/
    border:2px solid green;
}
```

```
#content{
/*Rahmen 2 Pixel breit durchgehend grün*/
    border:2px solid red;

/*Ausrichtung des Elements rechts*/
    float: right;

/*Breite der Box: relativ 70%*/
    width: 70%;

/*Höhe der Box: automatische Ausrichtung*/
    height:auto;
}
```

```
#sidebar{
/*Rahmen 2 Pixel breit durchgehend gelb*/
    border:2px solid yellow;

/*Ausrichtung des Elements links*/
    float: left;

/*Breite der Box: relativ 20%*/
    width: 20%;
}
```

Ergänzen Sie die Stildatei um das Box-Modell.

Wir legen für alle Seitenbestandteile eine Div-Box an und legen eine eigene Rahmenfarbe fest.

Öffnen Sie die Stildatei → styles.css und ergänzen Sie den angezeigten Quellcode für das Box-Modell und speichern Sie die Änderungen (STRG + S).

Hinweis:

Die wrapper-Box wird später den Container bilden der die ganze Seite im Internetbrowser zusammenhält und ausrichtet. Weitere Details folgen zu einem späteren Zeitpunkt.

Anfang:

```
div{
/*Aussenabstand: relativ 1% von allen Seiten*/
    margin: 1%;
}
```

```
#wrapper{
/*Block Element*/
    display:block;
}
```

```
/*Höhe der Box: automatische Ausrichtung*/
height:auto;
}
```

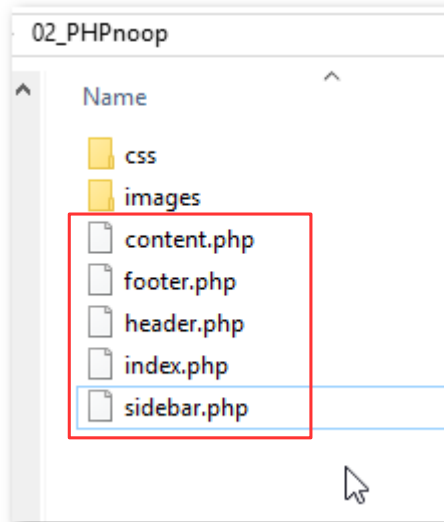
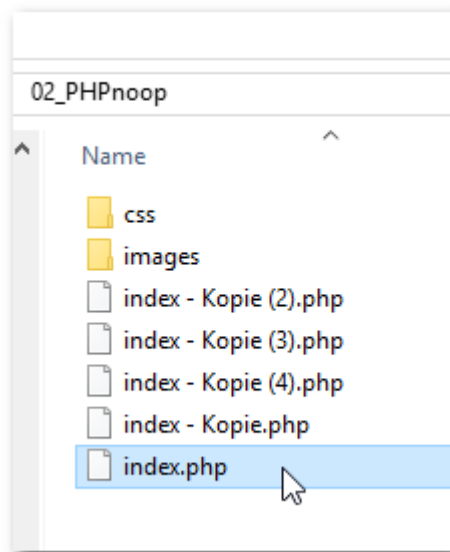
```
#footer{
/*Rahmen 2 Pixel breit durchgehend blau*/
border:2px solid blue;

/*Beendet das Umfließen von Elementen*/
clear:both;
}
```

```
/*Rahmen 5 Pixel breit durchgehend schwarz*/
border:5px solid black;
```

```
/*Breite der Box: relativ 90%*/
width: 90%;
}
```

Ergänzen Sie auch den nebenstehenden Quellcode.



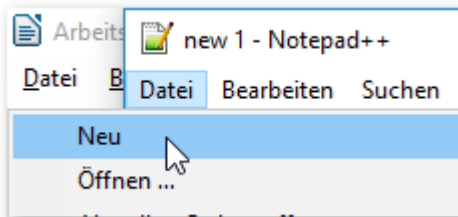
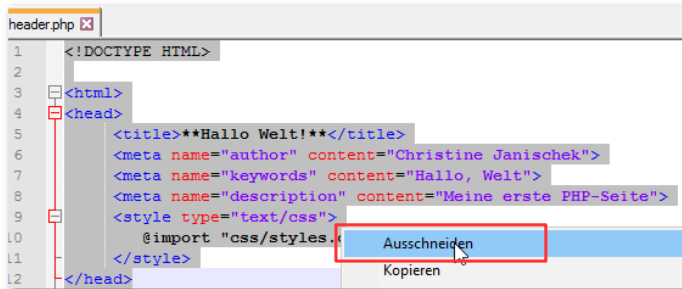
Prinzip der Zerlegung.

Um das Box-Modell zu nutzen benötigen wir weitere PHP-Dateien für

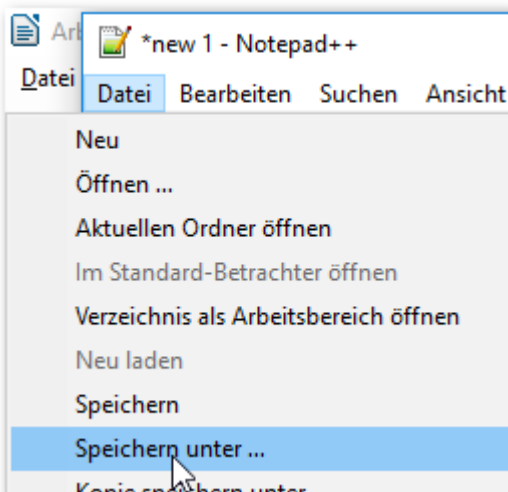
- die Kopfzeile (header),
- den Inhalt (content),
- die Navigation (sidebar) und
- die Fußzeile (footer).

Jeder Seitenbestandteil wird also in eine extra PHP-Datei ausgelagert.

Kopieren Sie dazu die Datei → index.php (→ STRG + C → STRG + V) vier mal und benennen Sie die Dateien wie in der nebenstehenden Grafik angezeigt um.



Menü-Leiste → Datei → Neu



Menü-Leiste → Datei → Speichern unter

Kopf zerlegen.

Wir zerlegen und differenzieren dazu

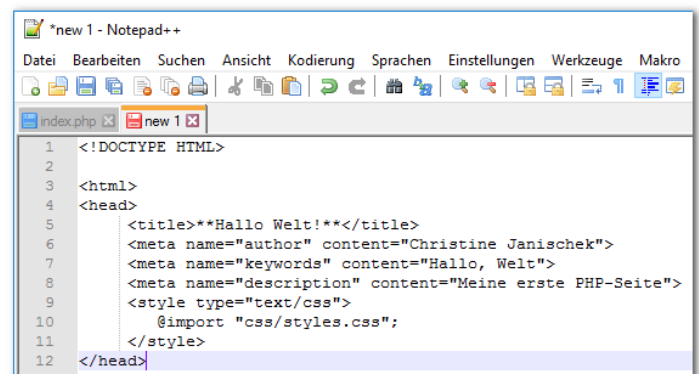
→ head

→ header

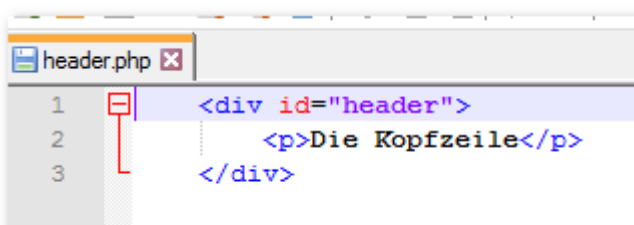
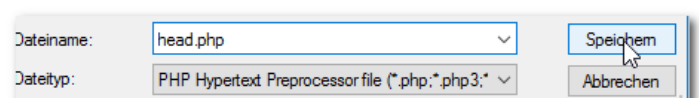
Öffnen Sie dazu zuerst die Datei

→ header.php im PHP Editor.

Schneiden Sie den oberen Teil (DOCTYPE und head) aus → STRG +X und fügen Sie den Quellcode in eine neue PHP-Datei ein STRG +V.



Nennen Sie die Datei → head.php und speichern Sie den Inhalt als php-Datei ab.



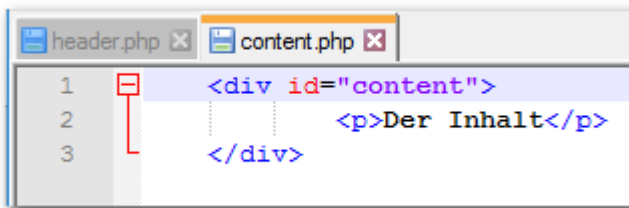
Inhalt der Kopfzeile (header) definieren.

Wechseln Sie dazu erneut in die Datei → header.php.

Verändern Sie den Quellcode wie folgt:

```
<div id="header">
    <p>Die Kopfzeile</p>
</div>
```

Speichern Sie die Veränderungen
→ STRG +S



```

1 <div id="content">
2     <p>Der Inhalt</p>
3 </div>

```

Den Inhalt (content) definieren.

Öffnen die Sie Datei → content.php.

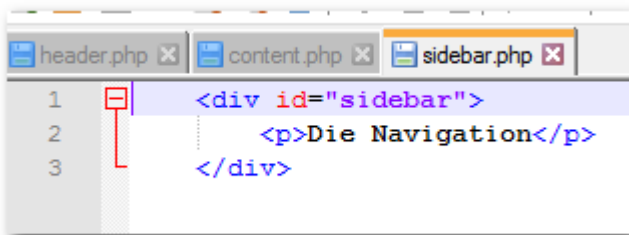
Ändern Sie die Datei → content.php, wie folgt:

```

<div id="content">
    <p>Der Inhalt</p>
</div>

```

Speichern Sie die Veränderungen
→ STRG + S



```

1 <div id="sidebar">
2     <p>Die Navigation</p>
3 </div>

```

Inhalte der Navigation (sidebar) definieren.

Ändern Sie auf die gleiche Weise den Quellcode für die → sidebar.php und footer.php.

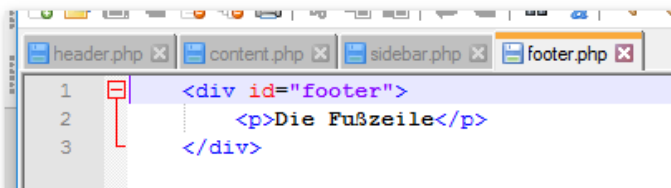
Ändern Sie die Datei → sidebar.php, wie folgt:

```

<div id="sidebar">
    <p>Die Navigation</p>
</div>

```

Speichern Sie die Veränderungen
→ STRG + S



```

1 <div id="footer">
2     <p>Die Fußzeile</p>
3 </div>

```

Inhalte der Fußzeile (footer) definieren.

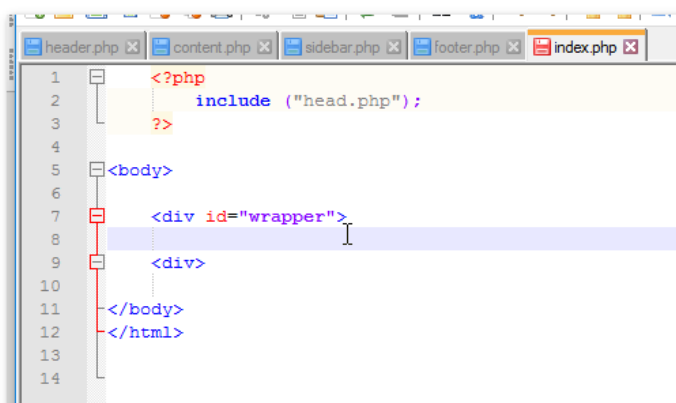
Ändern Sie die Datei → footer.php, wie folgt:

```

<div id="footer">
    <p>Die Fußzeile</p>
</div>

```

Speichern Sie die Veränderungen
→ STRG + S



```

1 <?php
2     include ("head.php");
3 ?>
4
5 <body>
6
7     <div id="wrapper">
8
9     <div>
10
11 </div>
12 </body>
13 </html>
14

```

index.php

Der PHP-include-Befehl:

Zusammenführung aller Seitenbestandteile

In der Datei → index.php werden alle Seiten:

```

→ head.php
→ header.php
→ sidebar.php
→ content.php
→ footer.php

```

integriert (zusammengeführt). Die wrapper-Box umfasst davon alle sichtbaren Seiten innerhalb des body-Tags.

Übernehmen Sie dazu den nebenstehenden Quellcode in die Datei → index.php und ergänzen Sie nachträglich die fehlenden


```
<?php
    include ("dateiname.php");
?>
```

Wenn der include-Befehl innerhalb einer Funktion aufgerufen wird, verhält sich der gesamte Code aus der aufgerufenen Datei, wie wenn er in der Funktion stünde. Folglich hat er den selben Variablen-Gültigkeitsbereich wie diese Funktion.

Bestandteile:

```
<?php
    include ("head.php");
?>

<body>

    <div id="wrapper">
        ...
    </div>

</body>
</html>
```

Nutzen Sie die Angaben im [Informationsblatt](#).

Im Web Browser



View: index.php

Testen Sie die Anwendung.

Öffnen Sie dazu den Internetbrowser und geben Sie den Pfad zur gerade erstellten Datei an. Geben Sie dazu den in der Grafik angezeigten Pfad ein.

http://localhost/02_PHPnoop/

Herzlichen Glückwunsch Sie haben ihr erstes dynamisches Layout erstellt.

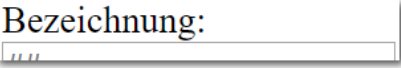
3 Formulare auswerten: Bmirechner

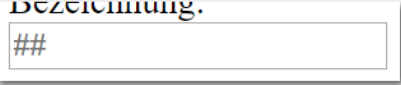
3.1 Informationsblatt: Formulare

Die eigentliche Fähigkeit von einer programmlogischen Sprache wie PHP liegt u.a. darin Eingaben und Entscheidung des Benutzers auf der Benutzeroberfläche auszuwerten und im Ergebnis zu berücksichtigen. Fundamental ist dazu die Verwendung von Formularen und deren Komponenten.

Die folgenden Anwendungen arbeiten jeweils mit einer Formulardatei und einer Datei die sowie die Programmlogik als auch die Konfiguration der gewünschten Ausgabe enthält.

Formulare und vielfach verwendete Komponenten:

<pre><!-- Grundgerüst für ein Formular --> <form name="formularname" method="post" action="auswertungsdatei.php"> <fieldset> <legend>Titel des Formulars</legend> <!-- hier folgen die Formularkomponenten --> </fieldset> </form></pre>	<i>Grundgerüst für ein Formular in HTML.</i> Mit Fieldset und Legende 
<pre><!-- Ein Bezeichnungsfeld --> <label for="Name Bezugskomponente">Bezeichnung:</label>
</pre>	<i>Bezeichnungsfeld</i> 
<pre><!-- Texteingabefeld --> <input type="text" name="tfName" id="tfName" placeholder="##" required="required" autofocus="autofocus" />

</pre>	<i>Texteingabefeld</i> 
<pre><!-- Radiobutton --> <input type="radio" name="rbOperator" value="wert" /> Operatorname
</pre>	<i>Radiobutton</i> 

```
<!-- Listenfeld -->
<label for="sName">Name der Liste:</label> <br />
  <select name="sName" size="1">
    <option value="1">Listenelement1</option>
    <option value="2">Listenelement2</option>
    <option value="3">Listenelement3</option>
    <option value="4">Listenelement4</option>
  </select> <br /><br /><br />
```

Listenfeld

Name der Liste:

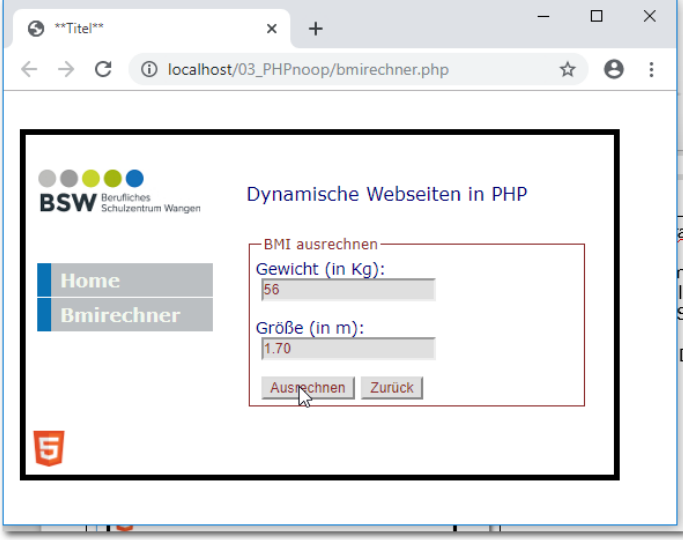
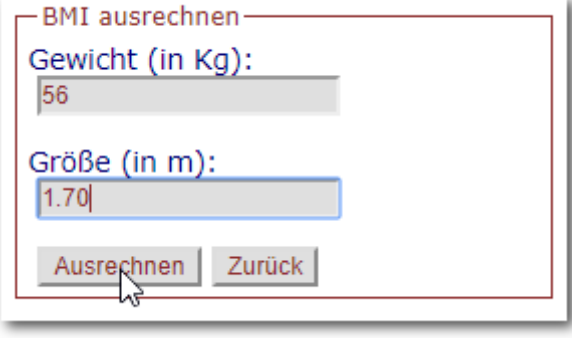
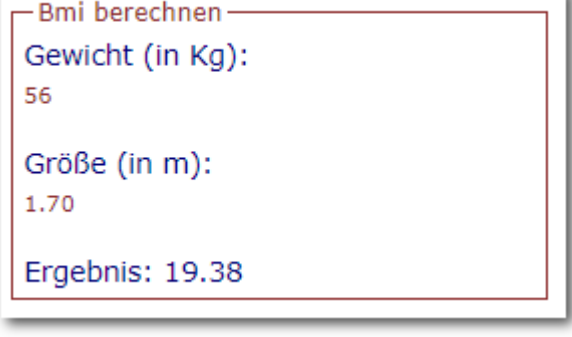
- Listenelement1
- Listenelement1
- Listenelement2
- Listenelement3
- Listenelement4

```
<!-- Schaltflächen für Ausrechnen und Zurück -->
<input type="submit" value="Ausrechnen" name="bmiAusrechnen"/>
<?php echo"<input type='button' value='Zurück' onClick='history.back()' />" ?>
```

Ausrechnen Zurück

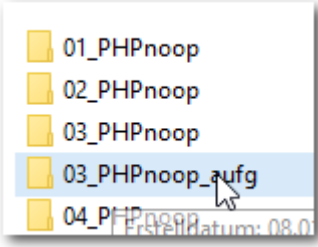
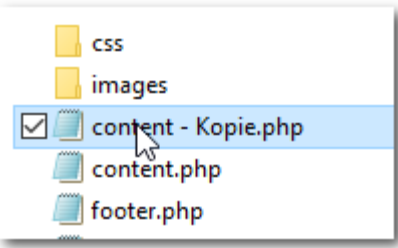
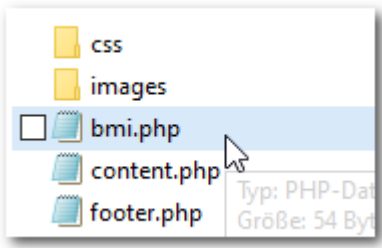
3.2 Aufgabe: Bmirechner

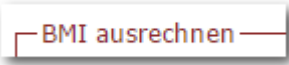
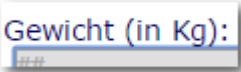
Realisieren Sie den Bmirechner. Erstellen Sie dazu die folgenden Dateien.

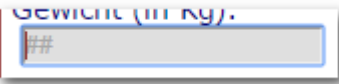
 Ergebnis der Übung: → bmirechner.php (View)	<i>Ziel: Ansicht Framework der Seite.</i> Oben die Kopfzeile mit dem Logo und einer Überschrift. Auf der linken Seite die Navigation. Auf der rechten Seite der Inhalt. Der Bmirechner → bmirechner.php
 Eingabe-Formular: → bmi.php (View)	<i>Ziel: Ansicht des Formulars.</i> Das Formular für den Bmirechner → bmi.php
 Steuerungs- und Ausgabe-Datei: → bmi1.php (Controller)	<i>Ziel: Anzeige der Berechnung.</i> Die gewünschte Auswertung der Eingaben und die Anzeige des Ergebnisses. → bmi1.php

3.3 Leittext: Bmirechner

3.3.1 Leittext: Bmirechner – Formular erzeugen

	<i>Projekt verwenden.</i> Verwenden Sie für die folgende Aufgabe das Projekt „03_PHPnoop_aufg“ als Ausgangsprojekt. Kopieren Sie das Projekt, fügen Sie das Projekt in Ihr www-Verzeichnis: „EigeneDateien/myPHP/“ in der Digitalen Tasche und benennen Sie es um in „03_PHPnoop“.
	<i>Formulardatei erzeugen.</i> Kopieren Sie dazu die Datei content.php mit STRG + C und fügen Sie die Datei STRG + V ein. Benennen Sie die Datei um.
	<i>Dateiname festlegen.</i> Geben Sie den Dateiname. Öffnen Sie die Datei in der Entwicklungsumgebung (z.B. Notepad++).
<pre><form name="bmirechnerformular" method="post" action="bmi1.php"> ... </form></pre>	<i>Quellcode für das Formular einbetten.</i> Fügen Sie dazu die Form-Box in die Content-Box ein. Definieren Sie die Eigenschaften für den Namen, Methode und Aktion, wie angegeben. Name Bezeichner für das Formular.

	Method In der Regel wird die Methode → post verwendet. • Post: unterdrückt die Angabe der Formparameter in der URL, die Länge der URL ist nicht begrenzt und der Nutzer kann auf die Auswertung nicht verlinken. • Get: zeigt die Formparameter in der URL an, die Länge der URL ist auf 3000 Zeichen begrenzt und der Nutzer kann auf die Auswertung verlinken. Action Enthält den Dateinamen der ereignissteuernden PHP-Datei → Controller. Die Datei wird aufgerufen, wenn der Nutzer die Schaltfläche vom Typ → submit im Formular anklickt.
<pre><fieldset> ... </fieldset></pre>	<i>Fieldset definieren.</i> Fügen Sie in die Form-Box die Fieldset-Box ein. Mit HTML5 ist es sinnvoll Formulare in sinnvolle Blöcke zusammenzufassen. Sie dienen der Gruppierung zusammengehörender Formularelemente.
 View <pre><legend>BMI ausrechnen</legend></pre>	<i>Legende definieren.</i> Fügen Sie in die Fieldset-Box die Legend-Box ein. Definieren Sie für die Legende den Titel.
 View <pre><label for="tfGewicht"> Gewicht (in Kg): </label>
</pre>	<i>Label-Komponente definieren.</i> Fügen Sie unterhalb der Legende und innerhalb der Fieldset-Box die Label-Komponente ein. Die Label-Komponente dient als Bezeichner für ein Eingabefeld, z.B. einem Textfeld. Für die Eigenschaft → for wird der Komponentennamen des Eingabefeldes angegeben.
<pre><input type="text" name="tfGewicht" id="tfGewicht" placeholder="##" required="required" autofocus="autofocus" />

</pre>  View Eigenschaften eines Textfeldes:	<i>Textfeld-Komponente definieren.</i> Fügen Sie unterhalb der Label-Komponente und innerhalb der Fieldset-Box die Textfeld-Komponente ein. Ein Textfeld ist eine Input-Komponente (Eingabefeld) vom Typ → text. Weitere Beispiele für Eingabe-Komponenten:

type

die Eigenschaft legt fest, auf welche Daten (Art der Erfassung und Datentyp) erfasst werden sollen.

Name

Bezeichner für die Komponente.

id

Wird gesetzt um das Feld, falls notwendig im Stylesheet individuell formatieren zu können.

placeholder

Enthält einen Eingabehinweis, um den Benutzer bei der korrekten Eingabe der Daten zu unterstützen.

required

Ist ein Flag (boolescher Wert) das gesetzt wird, wenn die Dateneingabe durch den Benutzer zwingend erforderlich ist. Neu in HTML5.

autofocus

Ist ein Flag (boolescher Wert) sorgt dafür, dass die Komponente beim laden der Seite fokussiert wird → der Cursor blinkt. Neu in HTML5.

<https://wiki.selfhtml.org/wiki/HTML/Formulare>

Es gibt weitere Eingabefelder:

Programming Rules			
Bezeichnung	Typ (type)	Präfix	Beispiel
Texteingabefeld	text	tf	tfGewicht
Passwortfeld	password	tf	tfPasswort
Radiobutton	radio	rb	rbGeschlecht
Checkbox	checkbox	cb	cbSalami
Button	submit	bt	btAusrechnen
Button	button	bt	btZurueck
Listenfeld	select	s	sSorte

Bezeichnung	Typ (type)	Präfix	Beispiel
Texteingabefeld	text	tf	tfGewicht
Passwortfeld	password	tf	tfPasswort
Radiobutton	radio	rb	rbGeschlecht
Checkbox	checkbox	cb	cbSalami
Button	submit	bt	btAusrechnen
Button	button	bt	btZurueck
Listenfeld	select	s	sSorte

Komponenten für die Eingabe der Größe einbetten.

Fügen Sie wie zuvor für das Gewicht die notwendigen Komponenten für die Größe ein.

name

→ tfGroesse

id

→ tfGroesse

placeholder

→ #.##

required

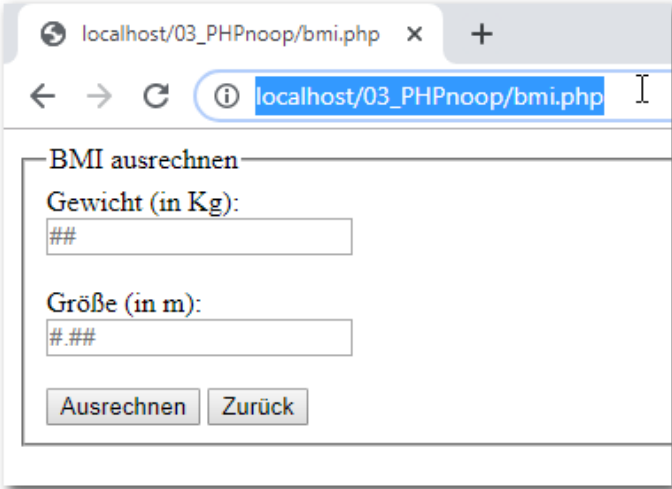
→ required

Schaltflächen einbetten.

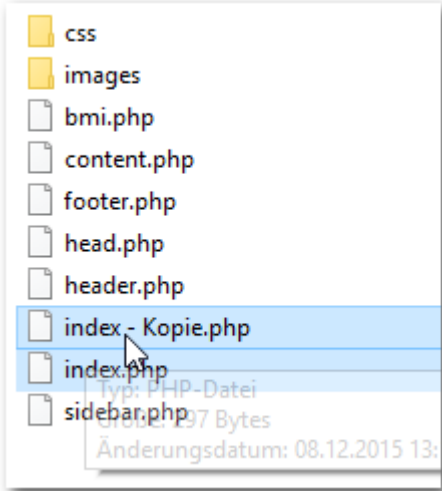
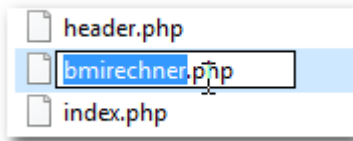
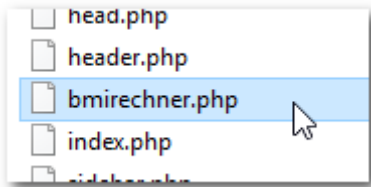
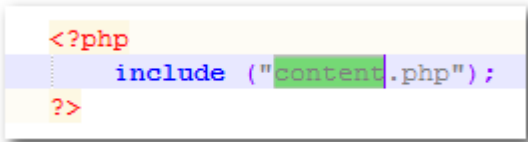
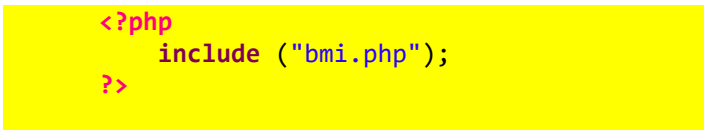
```
<input type="submit"
      value="Ausrechnen" name='bmiAusrechnen' />
```

```
<?php echo"<input type='button'
      value='Zurück'
      onClick='history.back()' />" ?>
```

Fügen Sie unterhalb der Textfeld-Komponente und innerhalb der Fieldset-Box die Button-Komponenten für die Schaltfläche → Ausrechnen und → Zurück ein.

	Mit dem Funktionsaufruf <pre><?php echo "... " ?></pre> wird die Ausgabe (→ echo) der Schaltfläche → zurück erzeugt. Auf diese Weise können wir via PHP das Ereignis über die Eigenschaft → onClick einbetten. Methodenaufruf → history.back(): Ist eine Javascript-Funktion. Die Methode ladet die vorhergehende URL aus der Historie des aktuellen Browserfensters.
	Testen Sie das Formular. Öffnen Sie dazu den Internetbrowser und geben Sie den Pfad zur gerade erstellten Datei an. Geben Sie dazu den Pfad ein. <pre>http://localhost/03_PHPnoop/bmi.php</pre> Herzlichen Glückwunsch Sie haben Ihr Formular erstellt. Betten Sie nun das Formular in Ihr dynamisches Layout ein. Folgen Sie dazu dem Leittext.

3.3.2 Leittext: Bmirechner – Formular in das dynamische Layout einbetten

	<i>Formulardatei in das bestehende Box-Model der Seite einbetten.</i> Kopieren Sie dazu die Datei → index.php mit STRG + C und fügen Sie die Datei STRG + V ein. Benennen Sie die kopierte Datei um.
	<i>Dateiname festlegen.</i> Öffnen Sie die Datei mit einem Doppelklick auf den Dateinamen: 
 	<i>Formular referenzieren.</i> Ersetzen Sie in der Datei → bmirechner.php die Referenz durch die Referenz auf die Formular-datei → bmi.php



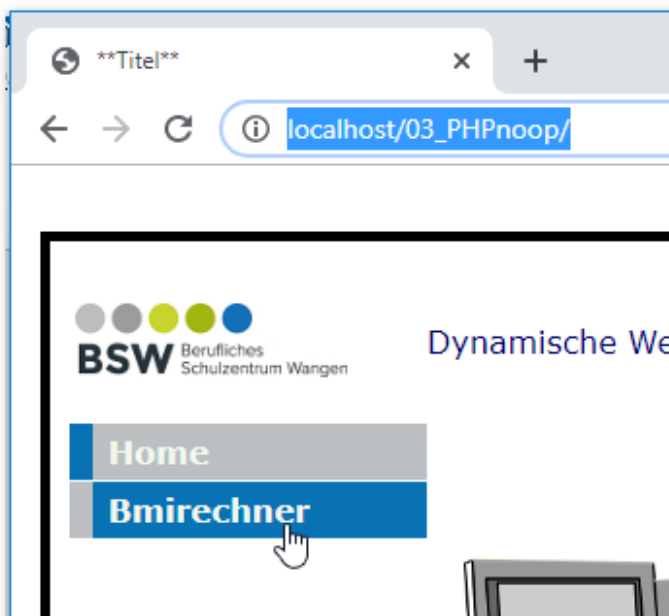
View: sidebar.php

```
<ul>
  <li><a href="index.php">Home</a></li>
  <li><a href="bmirechner.php"
    target="_parent">Bmirechner</a></li>
</ul>
```

Navigation erweitern.

Öffnen Sie die Datei → sidebar.php. Erweitern Sie die ungeordnete Liste innerhalb der Sidebar-Box um die Referenz (das Listenelement) auf die bmirechner.php-Datei.

Im Web Browser



View: index.php

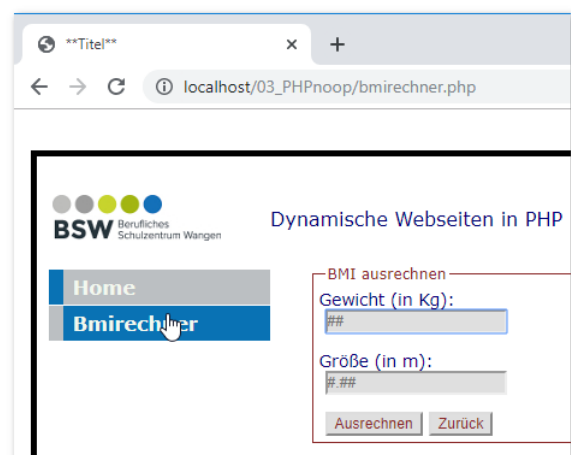
Testen Sie die Anwendung.

Öffnen Sie dazu den Internetbrowser und geben Sie den Pfad zur gerade erstellten Datei an. Geben Sie dazu den in der Grafik angezeigten Pfad ein.

http://localhost/03_PHPnoop/

Klicken Sie in der Navigationsleiste auf die Option Bmirechner.

Herzlichen Glückwunsch Sie haben den Rechner erfolgreich in die Seite eingebettet.



Der Rechner rechnet noch nicht! Folgen Sie dazu weiter dem Leittext

3.3.3 Leittext: Bmirechner – Berechnung und Anzeige des Ergebnisses

Funktionalitäten einbetten.

In den nächsten Schritten kümmern wir uns um die Funktionalität → Ausrechnen. Das Ereignis, also die Auswertung der Eingaben durch den Benutzer sollen dann erfolgen, wenn die Schaltfläche → Ausrechnen angeklickt wird.

Diese Auswertung kann eine Seitenbeschreibungssprache HTML nicht leisten. Wir nutzen deshalb die Programmiersprache PHP dafür.

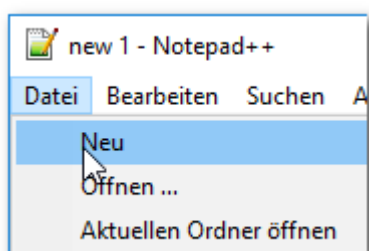


Die Controllerdatei fehlt noch!

Im Moment rufen Wir mit dem Klick auf die Schaltfläche → Ausrechnen eine Datei → bmi1.php auf (siehe Form-Action in der Datei → bmi.php):

```
<form name="bmirechnerformular" method="post" action="bmi1.php">
  <fieldset>
    <legend>BMI ausrechnen</legend>
```

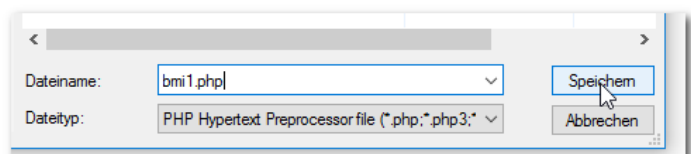
Diese Datei existiert aber noch nicht, deshalb erscheint auch die nebenstehende Fehlermeldung.



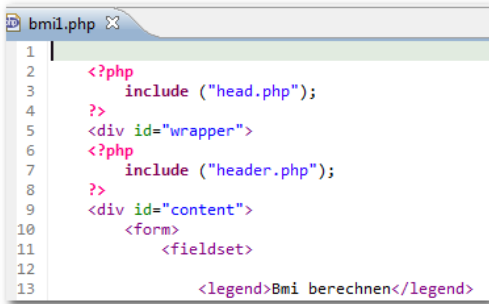
Controllerdatei einbetten.

Wir werden diese Datei nun erstellen und das EVA-Prinzip (**E**ingaben lesen, **v**erarbeiten und **a**usgeben) mit Hilfe von PHP anwenden.

Erzeugen Sie dazu die Datei → bmi1.php.



Öffnen Sie diese Datei mit einem Klick auf den Dateinamen.



Eingabehilfe:

```

<?php
    include ("head.php");
?>
<body>
<div id="wrapper">
    <?php
        include ("header.php");
    ?>
    <div id="content">
        <form>
            <fieldset>

                <legend>Bmi berechnen</legend>

```

Grundgerüst unseres Layouts einbetten.

Ersetzen Sie dazu den angegebenen Quellcode durch den nebenstehend angezeigten Quellcode.

Integrieren Sie dazu die Dateien:

→ head.php
→ header.php

und die Boxen:

→ wrapper
→ content
→ form
→ fieldset
→ legend

Hinweis:

Sie müssen die Content-, Form- und Fieldset-Box im Anschluss an die folgenden Implementierungen korrekt schließen!

```

<?php
//Eingaben lesen:
$pGewicht = $_POST['tfGewicht'];
$pGroesse = $_POST['tfGroesse'];

```

PHP-Tag

```

<?php
...
?>

```

EINGABE: Eingaben lesen

```

$pGewicht = $_POST['tfGewicht'];
$pGroesse = $_POST['tfGroesse'];

```

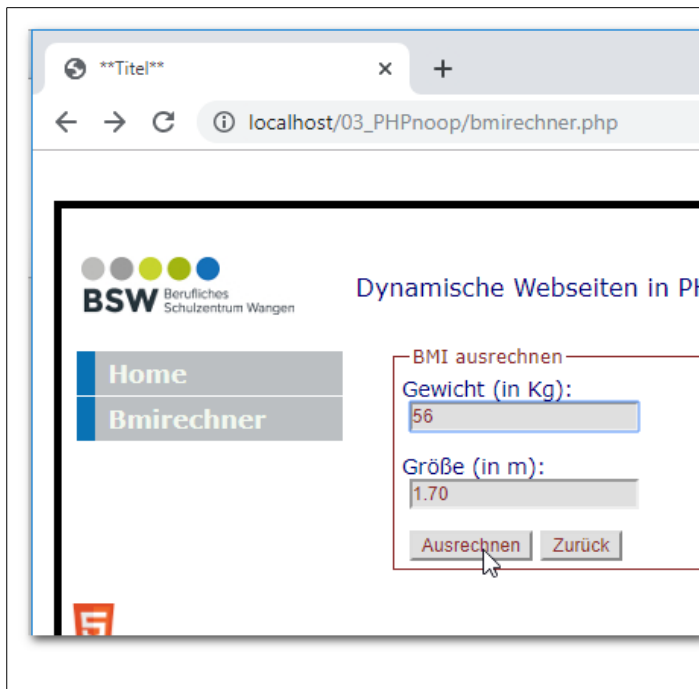
Eingabe: Eingaben Lesen.

Wir lesen nun im Ersten Schritt die Eingabewerte aus den Texteingabefeldern → tfGewicht, und → tfGroesse und übernehmen diese Werte in die PHP-Parameterattribute (Variablen) → pGewicht und → pGroesse.

Öffnen Sie dazu unterhalb der Legenden-Box eine neues PHP-Tag und übernehmen Sie den nebenstehenden Quellcode.

Die eigentliche Verarbeitung der eingegebenen Daten werden wir objektorientiert realisieren. Damit ergibt sich die Notwendigkeit eine Klasse, deren Eigenschaften und Verhaltensweisen zu definieren. Folgen Sie dazu weiter dem Leittext.

<pre>//Verhaltensweisen: Berechnende Funktion/Methode function berechne_bmi(\$pGewicht,\$pGroesse) { #####EINGABE:##### \$mGewicht = \$pGewicht; \$mGroesse = \$pGroesse; #####VERARBEITUNG:##### \$mBmi = \$mGewicht/(\$mGroesse * \$mGroesse); return \$mBmi; }</pre>	<i>Funktion/Methode definieren</i> Direkt im Anschluss an das Lesen der Eingabewerte wird die Funktion/Methode für die Berechnung deklariert.
<pre>/*Methodenaufruf: Die Berechnung*/ \$ausgabe = berechne_bmi(\$pGewicht,\$pGroesse); </pre>	<i>Funktions-/Methodenaufruf</i> Direkt im Anschluss an die Definition der Funktion/Methode verwenden wird diese auch. Dazu fangen wir den Rückgabewert (das Ergebnis) in einer neuen Variablen → \$ausgabe auf.
<pre>#####AUSGABE##### echo "Gewicht (in Kg): <h5>".\$pGewicht."</h5>"; echo "Größe (in m): <h5>".\$pGroesse."</h5>"; echo "Ergebnis: ".round(\$ausgabe,2); ?></pre>	<i>Gewünschte Ausgabe konfigurieren</i> Mit dem → echo-Befehl werden im Anschluss an den Funktions-/Methodenaufruf die Darstellung und Ausgabe der Eingabewerte und des berechneten Wertes veranlasst. Der berechnete Wert (→ \$ausgabe) wird auf zwei Kommastellen gerundet.
<pre></fieldset><!--Fieldset-Box schließen--> </form><!--Form-Box schließen--> </div><!--Content-Box schließen--> <?php include ("sidebar.php"); ?> <?php include ("footer.php"); ?> </div><!--Wrapper-Box schließen--> </body> </html></pre>	<i>Boxen schließen und Dateien einbetten.</i> Schließen Sie alle noch offenen Boxen in der richtigen Reihenfolge und inkludieren Sie die Navigation und die Fußzeile. Eingabehilfe: <pre></fieldset><!--Fieldset-Box schließen--> </form><!--Form-Box schließen--> </div><!--Content-Box schließen--> <?php include ("sidebar.php"); ?> <?php include ("footer.php"); ?> </div><!--Wrapper-Box schließen--> </body> </html></pre>



Testen

http://localhost/03_PHPnoop/bmirechner.php

Eingabewerte eingeben und auf → Ausrechnen klicken.

Gewünschte Ausgabe:



4 Übung - Formulare auswerten: Notenrechner

BSW Berufliches Schulzentrum Wangen

Dynamische Webseiten in PHP

Home
Bmirechner
Notenrechner

Notenrechner

Note in Mathematik:
2

Note in Deutsch:
5

Note in Englisch:
3

Note in BWL:
2

Ausrechnen Zurück

notenrechner.php

Notenrechner

Note in Mathematik:
2

Note in Deutsch:
5

Note in Englisch:
3

Note in BWL:
2

Ausrechnen Zurück

note.php

Notendurchschnitt ausrechnen

Mathe:
2

Deutsch:
5

Englisch:
3

Bwl:
2

Ergebnis: 3

note1.php

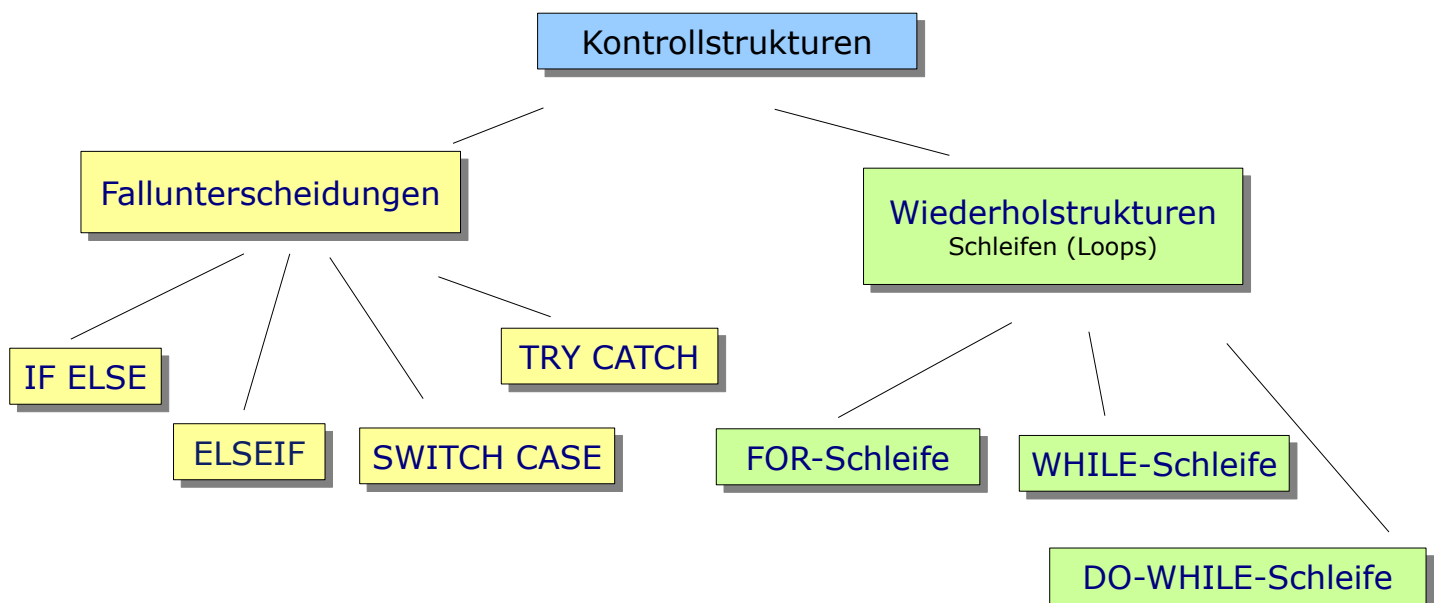
Arbeitsauftrag:

1. Erzeugen Sie ein neues Projekt um den → Notenrechner umzusetzen. Nutzen Sie das Informationsmaterial und die Lösung aus dem letzten Kapitel (→ kopieren und modifizieren), um die Aufgabe zu lösen.
2. Testen und Dokumentieren Sie alle Ergebnisse.

5 Kontrollstrukturen

5.1 Informationsblatt: Fallunterscheidungen

Thema: 	Kontrollstrukturen: Fallunterscheidungen Autor: Christine Janischek Informationsblatt: Überblick zu Fallunterscheidungen
---	--



Hinweis: Fallunterscheidungen sind Bestandteil von Verhaltensweisen (Methoden)		
Fallunterscheidung	Grundgerüst	Besonderheit
IF ELSE	<pre> if(Bedingung){ //Anweisung(en) }else{ //weitere Anweisung(en) } </pre>	Die Bedingung muss zum Wahrheitswert „true“ oder „false“ evaluieren können. Handelt 'nicht-geschachtelt' nur genau zwei Fälle ab. Große Werte zuerst prüfen, um logische Fehler zu vermeiden!
ELSEIF	<pre> if(Bedingung1){ //Anweisung(en) }elseif(Bedingung2){ </pre>	Kann beliebig viele Fälle behandeln. Große Werte zuerst prüfen, um

	<pre> //weitere Anweisung(en) }elseif(Bedingung3){ //weitere Anweisung(en) }else{ //weitere Anweisung(en) } </pre>	logische Fehler zu vermeiden! Die ELSEIF prüft nur solange bis eine Bedingung zu → true evaluiert, weitere wahren Bedingungen werden ignoriert.
SWITCH CASE	<pre> switch(Ausdruck){ case ausdruck_1: //Anweisung(en) break; case ausdruck_2: //Anweisung(en) break; case ausdruck_3: //Anweisung(en) break; default: //Anweisung(en) } </pre>	Kann Fälle in Abhängigkeit des Ausdrucks abhandeln. Der Ausdruck enthält einen Wert, die Wertigkeit kann variieren und enthält zumeist eines primitiven Datentyps (int, double, char,...). Mit dem → break wird nach erfolgreicher Prüfung die Kontrollstruktur verlassen. Wenn eine Methode mit Rückgabewert implementiert wird erzeugt der Aufruf → return denselben Effekt, wie mit dem → break. Der Einsatz des break-Befehls ist damit hinfällig.
TRY CATCH	<pre> try { //Anweisung(en) } catch (Exception \$e) { //Anweisung(en) } </pre>	Ausnahmebehandlung (Exception-Handling)

5.2 Aufgaben: Einführende Übungen zu Fallunterscheidungen

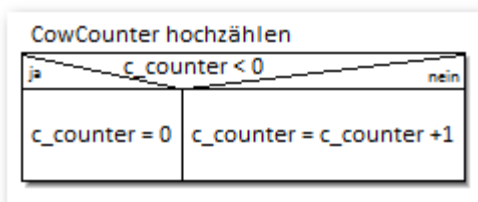
Information:

Ein Programmierer benötigt in der Praxis vielfach die Möglichkeit alternative Lösungswege zu entwickeln. Einige Strukturen sog. Algorithmen und die Kontrollstrukturen an sich, sind dazu das passende Handwerkszeug.

Aufgabe:

Üben Sie mit Hilfe der folgenden Aufgaben Ihr algorithmisches Denken. Setzen Sie dazu die folgenden Struktogramme in Quellcode (Programmcode) um. Nutzen Sie dazu nach Bedarf die Kontrollstrukturen aus der Übersicht. **Entwickeln Sie Funktionen/Methoden mit Parametern.**

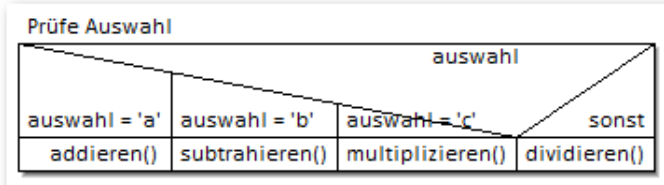
Programmieren Sie händisch mittels Papier und Bleistift!



Kühe zählen.

Wir möchten eine Anwendung entwickeln die Kühe zählen kann.

Eine Funktion/Methode soll in der Lage sein den Zähler um eine Einheit zu erhöhen Schreiben Sie die Anweisungen in eine Funktion/Methode → **cowCounter_hochzaehlen**.



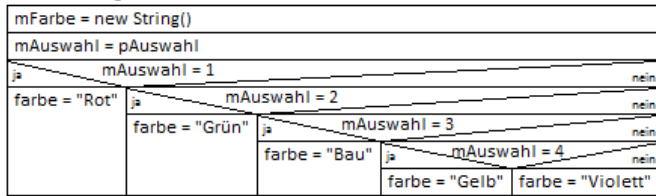
Auswahl prüfen.

Die Rechenoperation soll für einen Taschenrechner individuell wählbar sein.

Schreiben Sie die Anweisungen in eine Funktion/Methode → **berechne**.

Implementieren Sie auch die verwendeten Funktionen/Methoden für die Addition, Subtraktion, Multiplikation und Division zweier Zahlen.

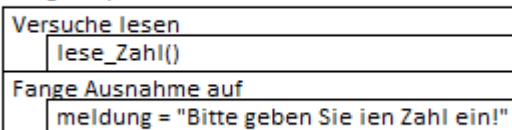
Farbe festlegen

*Farbe festlegen.*

Die Farbe soll für Grafiken individuell bestimmbar sein.

Schreiben Sie die Anweisungen in eine Funktion/Methode → **farbe_festlegen**.

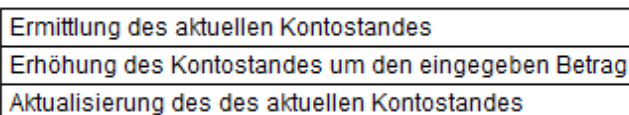
Eingabe prüfen

*Eingabe prüfen.*

Die Eingabe soll geprüft werden.

Schreiben Sie die Anweisungen in eine Funktion/Methode → **check_Eingabe**.

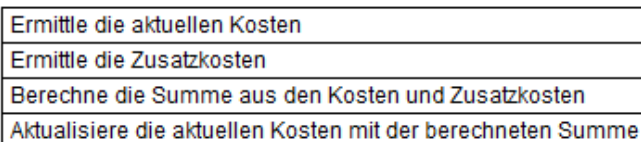
Einzahlen

*Einzahlen.*

Die Einzahlung auf einem Konto soll ermöglicht werden.

Schreiben Sie die Anweisungen in eine Funktion/Methode → **einzahlen**.

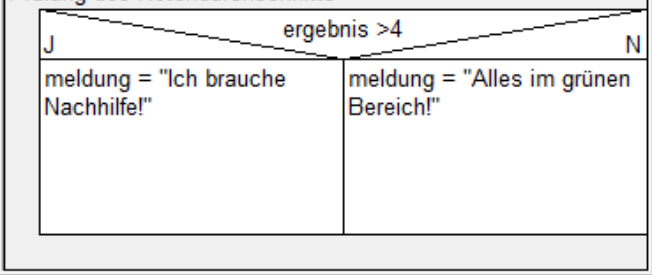
Kosten berechnen

*Kosten berechnen.*

Die Kosten für ein Haus ist mit Zusatzkosten verbunden.

Schreiben Sie die Anweisungen in eine Funktion/Methode → **berechne_Kosten**.

Prüfung des Notendurchschnitts

*Notenrechner erweitern*

Erweitern Sie den bestehenden Notenrechner um die nebenstehende Prüfung.

5.3 Aufgabe: Taschenrechner



BSW Berufliches
Schulzentrum Wangen

Dynamische Webseiten in PHP

Home

Bmirechner

Notenrechner

Taschenrechner

Zahlen eingeben

Zahl 1:

Zahl 2:

Operation auswählen

☒ + (addieren)

☐ * (multiplizieren)

☐ ^ (potenzieren)

☐ - (subtrahieren)

☐ / (dividieren)

taschenrechner.php

Zahlen eingeben

Zahl 1:

Zahl 2:

Operation auswählen

☒ + (addieren)

☐ * (multiplizieren)

☐ ^ (potenzieren)

☐ - (subtrahieren)

☐ / (dividieren)

Ergebnis berechnen

Zahl1:

12

Zahl2:

4

Operator:

+

Ergebnis: 16

trechner.php

trechner1.php

Arbeitsauftrag:

1. Erzeugen Sie ein neues Projekt um den → Taschenrechner umzusetzen. Nutzen Sie das Informationsmaterial und die Lösung aus den letzten Kapiteln (→ kopieren und modifizieren), um die Aufgabe zu lösen.
2. Testen und Dokumentieren Sie alle Ergebnisse.

5.4 Aufgabe: Rabattrechner



Dynamische Webseiten in PHP (noop)

- Home
- Bmirechner
- Notenrechner
- Taschenrechner
- Rabattrechner**

Der Inhalt

Rabatt- und Zahlungsbetrag berechnen

Betrag:

Menge:



rabattrechner.php

Rabatt- und Zahlungsbetrag berechnen

Betrag:

Menge:

rabatt.php

Ergebnis berechnen

Betrag:
150 €

Menge:
70

Rabattsatz:
15

Rabattbetrag:
22.5 €

Ergebnis: 127.5 €

rabatt1.php

Ermittlung des Rabattsatzes:

Ermittle Rabattsatz				
>=150	>=100	>=50	>=20	menge?
rabattsatz = 12	rabattsatz = 10	rabattsatz = 8	rabattsatz = 6	sonst rabattsatz = 0

Berechnungen

Des Rabattbetrags:

 $\text{rabattbetrag} = \text{betrag} / 100 * \text{rabattsatz};$

Des Zahlungsbetrags:

 $\text{zahlungsbetrag} = \text{betrag} - \text{rabattbetrag};$ *Arbeitsauftrag:*

1. Erzeugen Sie ein neues Projekt um den → Rabattrechner umzusetzen. Nutzen Sie das Informationsmaterial und die Lösung aus den letzten Kapiteln (→ kopieren und modifizieren), um die Aufgabe zu lösen.
2. Testen und Dokumentieren Sie alle Ergebnisse.

5.5 Aufgabe: Milchautomat



BSW Berufliches
Schulzentrum Wangen

Dynamische Webseiten in PHP (noop)

- Home
- Bmirechner
- Notenrechner
- Taschenrechner
- Rabattrechner
- Milchautomat**

Milchautomat

Menge:

Milchtyp wählen:

Vollmilch ▼

☐ Große Flasche (1.5 Liter)

☐ Normale Flasche (1.0 Liter)

☐ Mittlere Flasche (0.7 Liter)

☒ Kleine Flasche (0.5 Liter)

Milchautomat

Menge:

Milchtyp wählen:

Vollmilch ▼

☐ Große Flasche (1.5 Liter)

☐ Normale Flasche (1.0 Liter)

☐ Mittlere Flasche (0.7 Liter)

☒ Kleine Flasche (0.5 Liter)

Ergebnis berechnen

Menge (Anzahl der Flaschen):

3 Flaschen

Milchtyp:

Vollmilch

Flaschengröße:

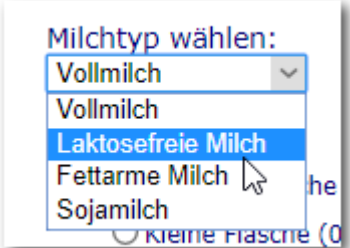
0.5 Liter Flaschen

Flaschenpreis:

0.65 €

Ergebnis: 1.95 €

Drop-Down-Menü:



Formeln:

Flaschenpreis = Milchpreis * Flaschengröße

Ergebnis = Menge * Flaschenpreis

Milchtyp	Milchpreis
Vollmilch	1.30
Laktosefreie Milch	1.45
Fettarme Milch	1.10
Sojamilch	1.5

Arbeitsauftrag:

1. Erzeugen Sie ein neues Projekt um den → Milchautomat umzusetzen. Nutzen Sie das Informationsmaterial und die Lösung aus dem letzten Kapitel (→ kopieren und modifizieren), um die Aufgabe zu lösen.
2. Testen und Dokumentieren Sie alle Ergebnisse.

5.6 Aufgabe: Erweiterter Bmirechner

bmirechner.php

bmi.php

bmi1.php

Tabellarische Darstellung:

Quelle: <http://www.bmi-rechner.net/bmi-tabelle.htm>

	BMI männlich	BMI weiblich
Untergewicht	unter 20	unter 19
Normalgewicht	20-25	19-24
Übergewicht	26-30	25-30
Adipositas	31-40	31-40
starke Adipositas	größer 40	größer 40

→ Interpretation des berechneten BMIs in Abhängigkeit des Geschlechts.

Tabellarische Darstellung:

Quelle: <http://www.bmi-rechner.net/bmi-tabelle.htm>

Der optimale Body Mass Index nach Alter:

Alter	optimaler BMI
19-24	19-24
25-34	20-25
35-44	21-26
45-54	22-27
55-64	23-28
älter als 65	24-29

→ Bestimmung des optimalen BMIs (Minimum und Maximum) anhand des Alters

Arbeitsauftrag:

1. Erzeugen Sie ein neues Projekt um den → Notenrechner umzusetzen. Nutzen Sie das Informationsmaterial und die Lösung aus dem letzten Kapitel (→ kopieren und modifizieren), um die Aufgabe zu lösen.
2. Testen und Dokumentieren Sie alle Ergebnisse.
3. Erstellen Sie die fehlenden Struktogramme für die Interpretation und die Bestimmung des optimalen BMIs.

5.7 Aufgabe: Parkautomat

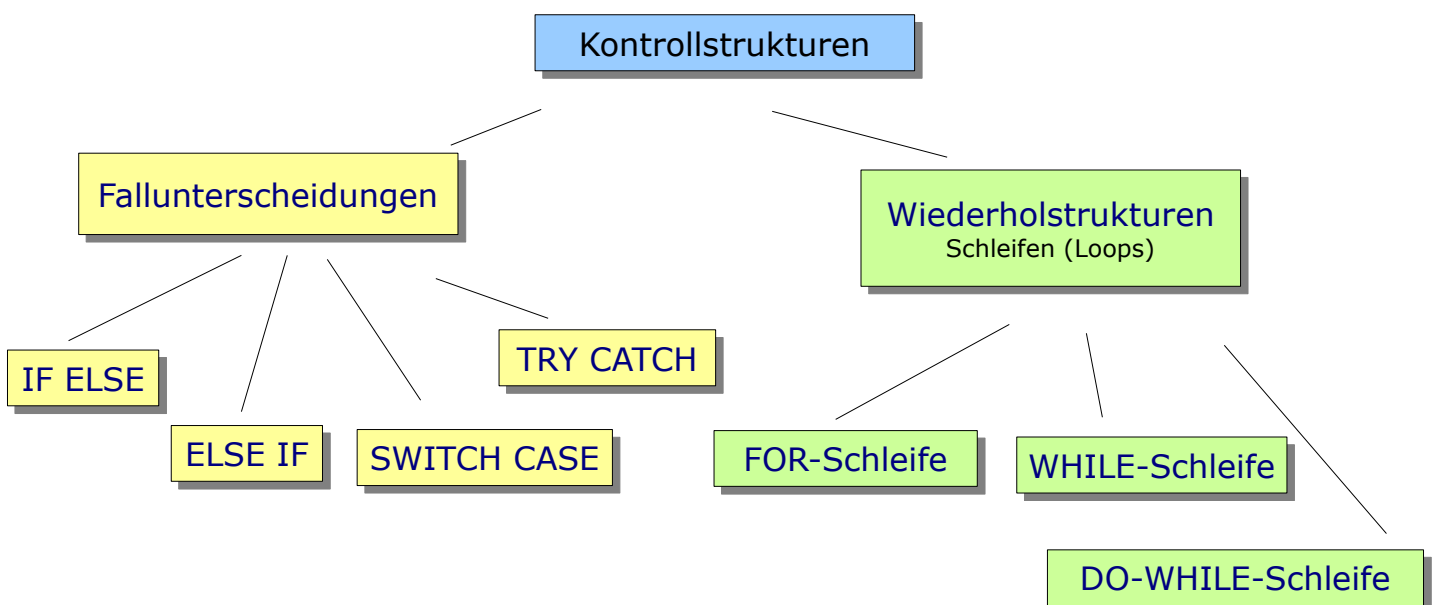
Arbeitsauftrag:

1. Entwerfen Sie selbständig die Software für einen Parkautomaten. Erzeugen Sie dazu ein neues Projekt um den → Parkautomat umzusetzen und zu testen. Nutzen Sie das Informationsmaterial und die Lösung aus dem letzten Kapitel (→ kopieren und modifizieren), um die Aufgabe zu lösen.
2. Testen und Dokumentieren Sie alle Ergebnisse.
3. Erstellen Sie die fehlenden Struktogramme.

6 Wiederholstrukturen

6.1 Informationsblatt: Kontrollstrukturen → Wiederholstrukturen

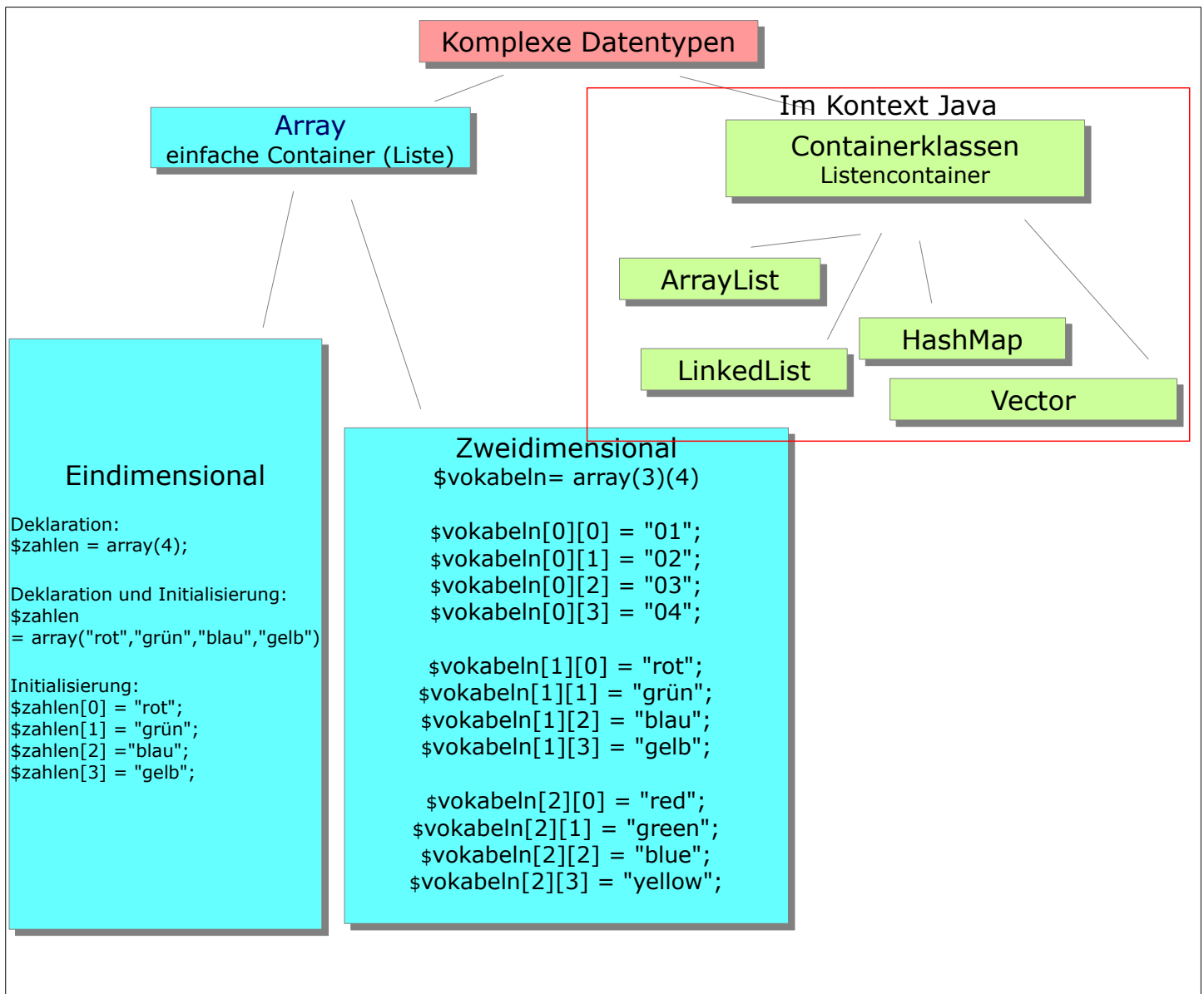
Thema: 	Kontrollstrukturen: Wiederholstrukturen Autor: Christine Janischek Informationsblatt: Überblick zu Wiederholstrukturen
---	---



Hinweis: Wiederholstrukturen sind Bestandteil von Verhaltensweisen (Methoden)		
Wiederholstrukturen	Grundgerüst	Besonderheit
FOR-SCHLEIFE	<pre> for (ausdruck1; ausdruck2; ausdruck3){ //Anweisung(en) } Beispiel: for(\$i = 10;\$i > 0;\$i--) { echo \$i; } </pre>	Ist die komplexeste Schleife. Ausdruck1 wird vor der Schleife ausgeführt und enthält in der Regel die Deklaration und Initialisierung der Schleifenzählervariablen. Am Anfang eines jeden Schleifendurchlaufs wird die Anweisung in Ausdruck2 ausgeführt. Wenn diese → True ist wird die Schleife fortgesetzt

		und die darunterliegenden Anweisungen werden ausgeführt. Anderenfalls (→ False) wird der Vorgang abgebrochen. Am Ende eines jeden Schleifendurchlaufs wird die Anweisung in Ausdruck3 ausgeführt.
WHILE-SCHLEIFE	<pre>while (bedingung){ //Anweisung(en) //Abbruchbedingung }</pre> Beispiel: <pre>\$i = 10; while(\$i > 0) { echo \$i; \$i--; }</pre>	Ist die einfachste Schleife. Die Anweisungen innerhalb der Schleife werden wiederholt ausgeführt, solange die Bedingung zutrifft, also zu → True evaluiert. Eine Abbruchbedingung z.B. ein Zähler stellt in der Regel sicher, dass die Schleife nicht endlos ausgeführt wird, die Bedingung zu gegebener Zeit zu → False evaluiert und der Vorgang abgebrochen wird.
DO-WHILE-SCHLEIFE	<pre>do{ //Anweisung(en) //Abbruchbedingung }while(bedingung);</pre> Beispiel: <pre>\$i = 10; do { echo \$i; \$i--; } while (\$i > 0);</pre>	Ist der While-Schleife sehr ähnlich. Der Unterschied liegt daran, dass die Bedingung erst am Ende des Schleifendurchlaufs geprüft wird.

6.2 Informationsblatt: Datencontainer



6.3 Arbeitsblatt: Umsatzrechner

Thema: 	Kontrollstrukturen: Wiederholstrukturen Autor: Christine Janischek Arbeitsblatt: Umsatzrechner (Filialumsätze)
--	--

Filialdaten eingeben

Filialnummer:

Filialname:

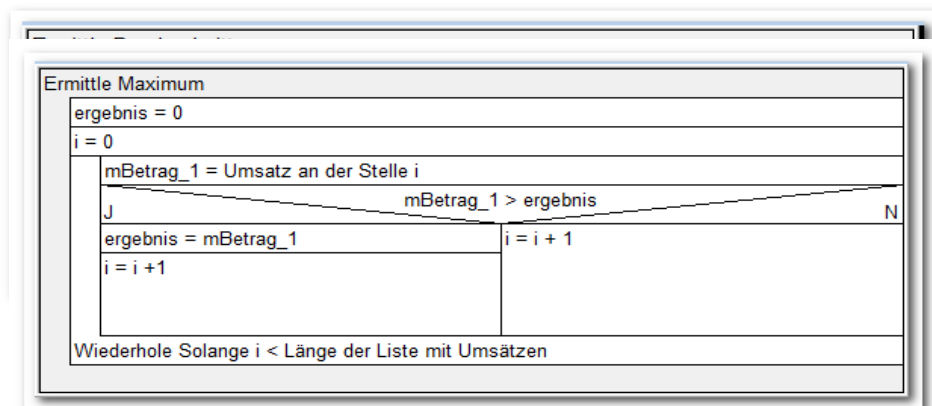
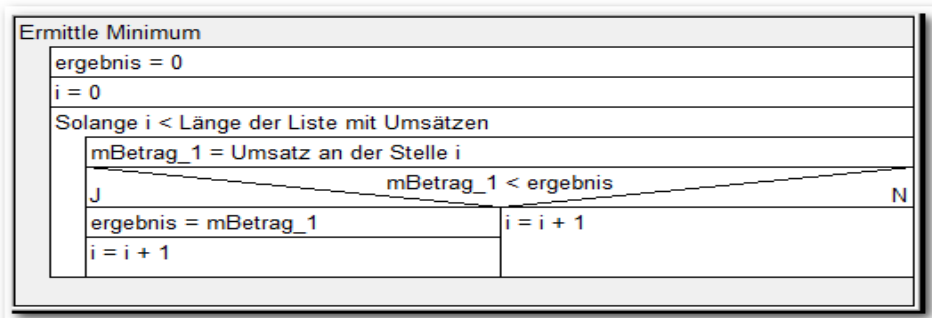
Zurück

Operationen

Für die Prüfung welche Schaltfläche angeklickt wurde.

```
//Prüfung: Wurde ein Button gedrückt?  
function pruefe_Button($umsaetze){  
  if(isset($_POST['btMinimum'])){  
    $ergebnis = ermittele_Minimum($umsaetze);  
  }elseif(isset($_POST['btMaximum'])){  
    $ergebnis = ermittele_Maximum($umsaetze);  
  }elseif(isset($_POST['btDurchschnitt'])){  
    $ergebnis = ermittele_Durchschnitt($umsaetze);  
  }else{  
    $ergebnis = 0;  
  }  
  return $ergebnis;  
}
```

Ihr Unternehmen hat mittlerweile einige Filialen. Für jede Filiale wird der Umsatz der letzten zwölf Monate in einer einfachen Liste erfasst.



Arbeitsauftrag:

1. Initialisieren Sie die Liste Umsatz der Filiale „Wangen“ in der Jahnstrasse 15 mit den folgenden Werten:
1000,1500,1100, 1200, 1150, 950,500,8800, 16000, 433, 8000, 5000
2. Welche Arten von Listentypen (Container) kennen Sie?
3. Nennen Sie fünf Kontrollstrukturen.
4. Erstellen Sie anhand der vorgegebenen Struktogramme den Quellcode für die Verhaltensweisen: ermittle Minimum, ermittle Maximum und ermittle Durchschnitt den Programmcode. Testen Sie den Quellcode!
5. Es hat sich ein Denkfehler eingeschlichen! Identifizieren Sie den Fehler und finden Sie einen Lösungsweg.
6. Welche Art Kontrollstrukturen eignen sich für die Behandlung der geschilderten Probleme?

6.4 Arbeitsblatt: Darlehnsrechner - Kontrollstrukturen → Wiederholstrukturen



BSW Berufliches
Schulzentrum Wangen

Dynamische Webseiten in PHP (noop)

- Home
- Bmirechner
- Notenrechner
- Taschenrechner
- Rabattrechner
- Milchautomat
- Umsatzrechner
- Darlehnsrechner**

Annuitätendarlehen berechnen

Finanzierungsbedarf:
#####.##

Zinssatz (nominal):
##.##

Tilgungssatz:
##.##

Zeitraum (in Jahren):
##

Ausrechnen
Zurück



→ darlehnsrechner.php (View)

Annuitätendarlehen berechnen

Finanzierungsbedarf:
8000

Zinssatz (nominal):
6.5

Tilgungssatz:
15

Zeitraum (in Jahren):
5

Ausrechnen
Zurück

Eingabe-Formular: → darlehen.php (View)

Fakten zum Finanzplan

Finanzierungsbedarf:
8000 €

Zinssatz:
6.5 %

Tilgungssatz:
15 %

Zeitraum:
5 Jahre

Annuität: 1720 €

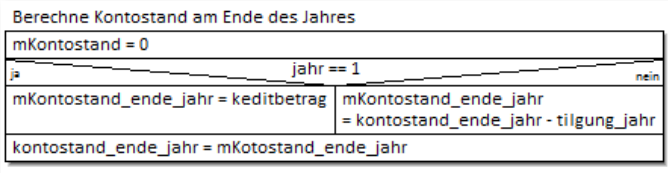
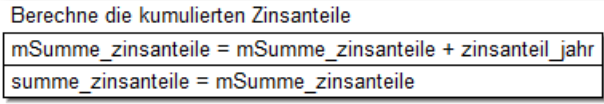
Finanzierungsplan						
Jahr	Kreditbetrag	Zinsanteil	Tilgung	Monatliche Belastung	Jährliche Belastung	Ende des Jahres
1	8000 €	520 €	0 €	143.33 €	1720 €	8000 €
2	8000 €	520 €	1200 €	143.33 €	1720 €	6800 €
3	6800 €	442 €	1278 €	143.33 €	1720 €	5522 €
4	5522 €	358.93 €	1361.07 €	143.33 €	1720 €	4160.93 €
5	4160.93 €	270.46 €	1449.54 €	143.33 €	1720 €	2711.39 €

Zinsen insgesamt: 2111.39 €

Ausgabe-Datei: → darlehen1.php (Controller)

```
/*Tabelle*/
th,td{
    border: 1px solid grey;/*Rahmen keiner*/
    font-size: 75%;/*relative Schriftgröße*/
}
```


	td{ text-align: right; } Stylesheet-Angaben für die Formatierung der Tabelle → style.css
Berechne Annuität mAnnuitaet = finanzierungsbedarf * (zinssatz + tilgungssatz)/100 ergebnis = mAnnuitaet	Der Annuität. Im vorliegenden Fall handelt es sich um ein Annuitäten-darlehen. Mittels der folgenden Berechnung kann die Annuität er-mittelt werden. Formel: ergebnis = finanzierungsbedarf * (zinssatz + tilgungssatz)/100
Ermittle Kreditbetrag j* jahr == 1 nein mKreditbetrag = finanzierungsbedarf mKreditbetrag = kontostand_ende_jahr kreditbetrag = mKreditbetrag kreditbetrag = mKreditbetrag	Der Kreditbetrag. Für das Erste Jahr soll für den Kreditbetrag der eingege-bene Finanzierungsbedarf übernommen werden. Für alle anderen Jahre soll für den Kreditbetrag der Kontostand am Ende des Jahres übernommen werden.
Berechne den Zinsanteil mZinsanteil = kreditbetrag * zinssatz /100 zinsanteil_jahr = mZinsanteil	Der Zinsanteil. zinsanteil = kreditbetrag * zinssatz /100
Ermittle Belastung im Jahr: Ermittle die Belastung im Jahr mBelastung_jahr = ergebnis belastung_jahr = mBelastung_jahr	Die Belastung im Jahr. Entspricht beim Annuitätendarlehen der Annuität.
Ermittle Tilgung im Jahr Berechne Tilgung im Jahr j* jahr == 1 nein mTilgung_jahr = 0 mTilgung_jahr = belastung_jahr - zinsanteil_jahr tilgung_jahr = mTilgung_jahr	Die Tilgung im Jahr. Für das Erste Jahr soll keine Tilgung anfallen: tilgung im jahr = 0; Anderenfalls soll die Tilgung berechnet werden: tilgung im jahr = belastung im jahr - zinsantal im jahr
Ermittle Belastung im Monat: Ermittle die monatliche Belastung mBelastung_mon = belastung_jahr / 12 belastung_mon = mBelastung_mon	Die Belastung im Monat. Entspricht 1/12 der jährlichen Belastung. belastung im monat= belastung im jahr/12;

<i>Ermittle Kontostand am Ende des Jahres:</i> 	<i>Der Kontostand am Ende des Jahres.</i> Im Ersten Jahr soll der Kontostand am Ende des Jahres dem Kreditbetrag entsprechen. In allen folgenden Jahren ergibt sich der Kontostand am Ende des Jahres aus dem Kontostand am Ende des letzten Jahres abzüglich der Tilgung im Jahr
<i>Berechne Zinsanteile kumuliert:</i> 	<i>Die kumulierten Zinsanteile.</i> Die wichtigste Frage für einen Kreditnehmer ist immer: „Was ist der Preis für den Kredit?“ Der Preis ergibt sich aus den Zinszahlungen, welche über die gesamte Laufzeit hinweg gezahlt werden müssen. Summiert man diese Beträge auf ergibt sich die Summe aller Zinsanteile.

7 PHP und Datenbanken

7.1 Einführung in Sessions

7.1.1 Informationsblatt

Thema: 	Sessions Autor: Christine Janischek Informationsblatt: Sessions
---	---

PHP Sessions

Mit PHP Sessions haben wir die Möglichkeit bestimmte Daten während einer Folge von Aufrufen auf einer Webseite festzuhalten und ggf. auf anderen Seiten weiterzuverarbeiten. Im Kontext von Webseiten die beabsichtigen Daten aus Relationalen Datenbanken zu nutzen (Daten einfügen, Daten auswählen, Daten verändern und Daten löschen), können wir auf den Einsatz von Sessions nicht verzichten.

Dem Besucher wird im Hintergrund eine Session-ID zugeordnet. Somit kann PHP den Besucher identifizieren. Diese Session ID wird entweder als Cookie, als verdeckte Formularfelder oder als Attributwert an die URL gehängt.

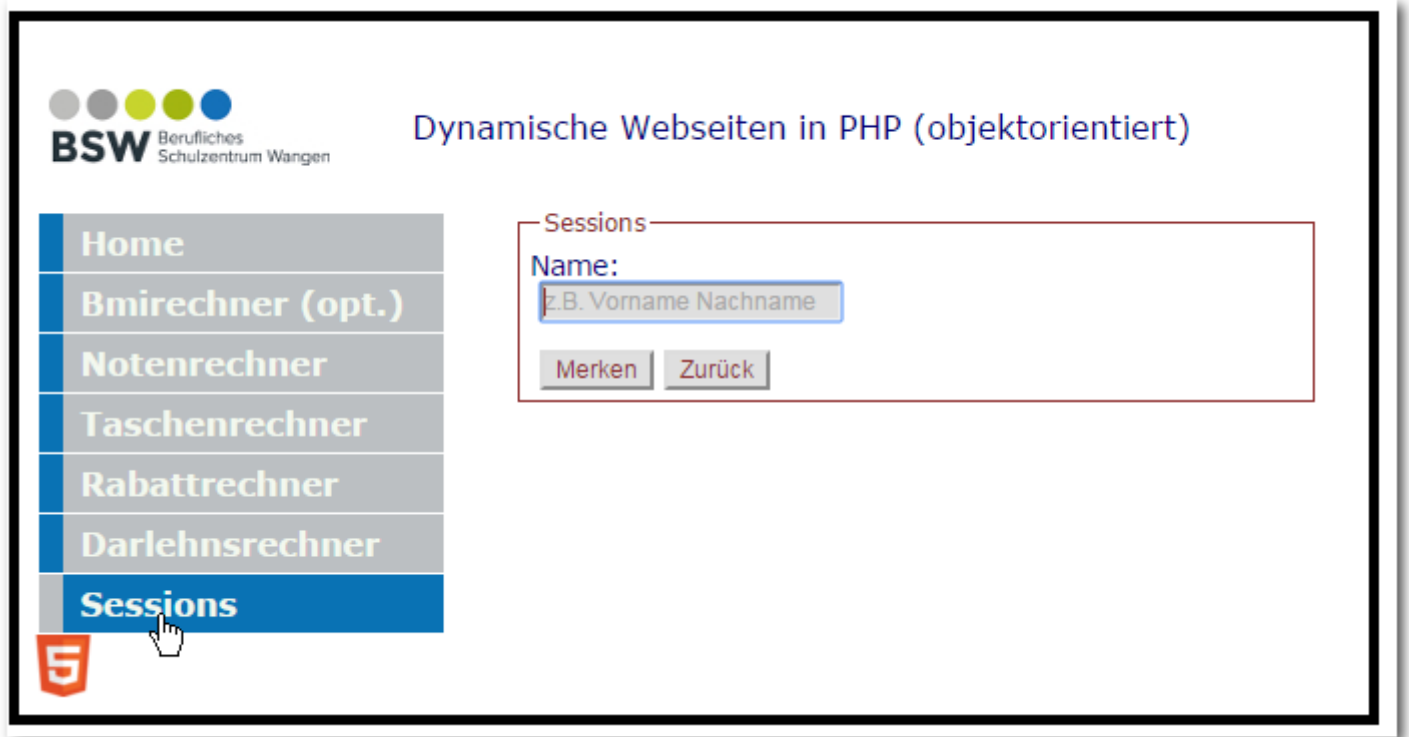
Zusatzinformationen können dazu in sogenannte Sessionvariable übernommen und ggf. weiterverwendet werden. Für die Handhabung von Sessions nutzen wir Heutzutage bestehende Bibliotheken sog. Frameworks.

[→ zurück zum Arbeitsblatt](#)

<pre><?php session_start(); ?></pre>	Mit session_start() ; sagen wir dem PHP Script, dass diese Seite mit Session arbeitet. Der Quellcode muss immer ganz oben stehen. Der Methodenaufwurf erfüllt zum einen den Zweck eine Session (Sitzung) zu starten und zum Anderen kann eine bestehende Sitzung fortgeführt werden.
<pre><?php \$_SESSION['name'] = "wert"; ?></pre>	Wir registrieren eine Session-Variable. Dazu weisen wir den Wert in einer Sessionvariable zu.
<pre><?php \$pName = \$_SESSION['name']; ?></pre>	Für den Fall, dass wir nach einem Seitenwechsel die Session fortsetzen, wird der Wert aus der Sessionvariable erneut in eine lokale Variable übermittelt.
<pre><?php session_destroy(); ?></pre>	Löschen der Sitzung. Globale Variablen und das Session-Cookie werden nicht gelöscht. Alle Session-Variablen werden mit dem Methodenaufwurf gelöscht: <pre>\$_SESSION = array();</pre>

7.1.2 Arbeitsblatt

Thema: 	Sessions Autor: Christine Janischek Arbeitsblatt: Sessions
---	--



Arbeitsauftrag:

1. Wir folgen dem informatischen Prinzip der Wiederverwendbarkeit und kopieren dazu den Inhalt aus dem vorgegebenen Projektverzeichnis in neues Projektverzeichnis.
2. Folgen Sie die folgenden Arbeitsanweisungen und nutzen Sie ergänzend die Hinweise auf dem [Informationsblatt](#).
3. Testen Sie das Projekt.
4. Helfen Sie Kollegen und dokumentieren Sie Ihre Vorgehensweise und Ergebnisse.

Arbeitsmaterialien und Übungsprojekte finden Sie im E-Learning:

→ [E-Learning OOP](#)

session.php

Ergänzen Sie den fehlenden Quellcode an entsprechender Stelle.

Verweisen Sie in der Form-Action auf die auszuführende Datei:

→ `session1.php`

Ergänzen Sie die Bezeichnung:

`<label for="tfName">Name:</label> <br />`

Ergänzen Sie das Texteingabefeld für den Namen:

`<input type="text" name="tfName" id="tfName" placeholder="z.B. Vorname Nachname" required="required" autofocus="autofocus" /> <br /><br />`

session1.php

Ergänzen Sie den fehlenden Quellcode an entsprechender Stelle.

Sie sollten die Sitzung starten:

→ `session_start();`

Eingabewert lesen:

`$pName = $_POST ['tfName'];`

Session registrieren:

`$_SESSION['username'] = $pName;`

Ergebnis ausgeben:

`echo "Ergebnis: <h5>Hallo " . $_SESSION['username'] . " <br />";`

Verweis auf die nächste Seite:

`echo "<a href=session2.php>Weiter</a>";`

Verweis zurück:

`echo "<a href=session1.php onClick='history.back()''>Zurück</a></h5>";`

Ergänzen Sie den fehlenden Quellcode an entsprechender Stelle.

Sie sollen die Session verwenden (fortsetzen):

`session_start(); //Ganz wichtig`

Eingaben merken und ausgeben Ergebnis: Du heißt immer noch Gast Abmelden session2.php	Übernehmen Sie den Wert aus der Session in ein lokale Variable: <pre>\$pName = \$_SESSION['username'];</pre> Ergebnis ausgeben: <pre>echo "Ergebnis: <h5>Du heißt immer noch ". \$_SESSION['username'].

";</pre> Verweisen Sie auf eine andere Seite: <pre>echo "Abmelden</h5>";</pre>
Eingaben merken und ausgeben Ergebnis: Sie sind erfolgreich abgemeldet! Erneut Anmelden session3.php	Ergänzen Sie den fehlenden Quellcode an entsprechender Stelle. Sie sollen die Session verwenden (fortsetzen): <pre>session_start(); //Ganz wichtig</pre> Übernehmen Sie den Wert aus der Session in ein lokale Variable: <pre>\$pName = \$_SESSION['username'];</pre> Löschen aller Session-Variablen: <pre>\$_SESSION = array();</pre> Löschen Sie die Daten der Sitzung: <pre>session_destroy();</pre> Text ausgeben: <pre>echo "Ergebnis: <h5>Sie sind erfolgreich abgemeldet!

";</pre> Verweis auf die Startseite der Session-Lektion: <pre>echo " Erneut Anmelden</h5>";</pre>

7.2 Einführung in die Verschlüsselung

7.2.1 Informationsblatt

Thema: 	Verschlüsselung Autor: Christine Janischek Informationsblatt: Ver- und Entschlüsseln (z.B von Passwörtern)
---	--

PHP Hash Funktion

Wer Zugangsdaten oder besonders vertrauliche Daten (Bankdaten) speichern möchte (z.B. in einer Datenbank), sollte diese nach Möglichkeit verschlüsseln.

Zusätzliche Informationen zu:

bcrypt

password_hash()

password_verify()

[→ zurück zum Arbeitsblatt](#)

<pre>password_hash(\$pPasswort.\$pTrick, PASSWORD_DEFAULT, array('cost'=>12))</pre>	Verschlüsseln: 1.Parameter Ein String der das Passwort enthält. Hier wird, aus Gründen der Sicherheit, dem Passwort (\$pPasswort) ein weiteres geheimes Wort (\$pTrick) angehängt. 2.Parameter Angabe des verwendeten Verschlüsselungsalgorithmus. 3.Parameter Die nicht zwingende Angabe (optional) für den Aufwand (Zeit, Variablen) des Algorithmus. Wird ggf durch eine Ganze Zahl festgelegt. Falls keine Angabe erfolgt ist der Standardwert 10. Die Funktion liefert als Rückgabe den Hashwert. → Dieser Wert wird in der DB gespeichert.
<pre>password_verify('passwort',\$hash)</pre>	Passwort prüfen: 1.Parameter Enthält das eingegebene Passwort mit dem angehängten geheimen Wort. 2.Parameter Enthält den in der Datenbank gespeicherten ggf. zugehörigen Hashwert. Die Funktion liefert als Rückgabewert 0 (falsch → Passwort konnte nicht verifiziert werden!) oder 1, also wahr (→ Passwort verifiziert), falls der Hashwert zum Eingabewert & Geheimwert passt.

7.2.2 Arbeitsblatt

Thema:

**Verschlüsselung**

Autor: Christine Janischek

Arbeitsblatt: Ver- und Entschlüsseln (z.B von Passwörtern)

Arbeitsauftrag:

1. Wir folgen dem informatischen Prinzip der Wiederverwendbarkeit und kopieren dazu den Inhalt aus dem vorgegebenen Projektverzeichnis in neues Projektverzeichnis.
2. Folgen Sie die folgenden Arbeitsanweisungen und nutzen Sie das [Informationsblatt](#).
3. Testen Sie das Projekt.
4. Helfen Sie Kollegen und dokumentieren Sie Ihre Vorgehensweise und Ergebnisse.

Arbeitsmaterialien und Übungsprojekte finden Sie im E-Learning:

→ [E-Learning OOP](#)

hash.php

Ergänzen Sie den fehlenden Quellcode an entsprechender Stelle.

Verweisen Sie in der Form-Action auf die auszuführende Datei:

→ `hash1.php`

Ergänzen Sie die fehlende Bezeichnung:

`<label for="tfPasswort">Passwort:</label><br />`

Ergänzen Sie das fehlende Texteingabefeld für das Passwort:

```
<input type="password" name="tfPasswort"
id="tfPasswort" placeholder="#####"
required="required"
autofocus="autofocus" /> <br /><br />
```

hash1.php

Ergänzen Sie den fehlenden Quellcode an entsprechender Stelle.

Sie sollten die Sitzung starten:

→ `session_start();`

Eingabewert lesen:

(simuliert) Formulardaten bei der Registrierung

```
$pName = $_POST ['tfName'];
$pPasswort = $_POST ['tfPasswort'];
```

Geheimer Wert ("pTrick") festlegen:

```
$pTrick = 'Angsthase!';
```

Erzeugen Sie mit Hilfe der PHP-Hash-Funktion einen Hashwert: Funktioniert nur ab PHP Version 5.5:

```
$pHash = password_hash($pPasswort.$pTrick,
PASSWORD_DEFAULT,array('cost'=>12));
```

Registrieren Sie die Session:

```
$_SESSION['username'] = $pName;
$_SESSION['password'] = $pPasswort;
$_SESSION['geheim'] = $pTrick;
```

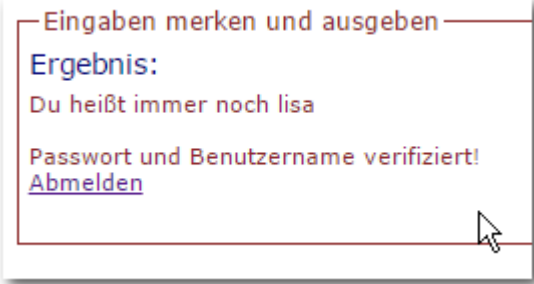
Ergebnis ausgeben:

```
echo "Ergebnis: <h5>Hallo "
.$_SESSION['username'].<br />";
```

Veranlassen Sie die Ausgabe des erzeugten Hash-Wertes für das Passwort:

```
echo "Für Dein Passwort haben wir folgenden
Hashwert festgelegt:<br />"
.$pHash."<br />";
```

	Verweis auf die nächste Seite: <pre>echo "Weiter

";</pre> Verweis zurück: <pre>echo "Zurück</h5>";</pre>
 hash2.php	Ergänzen Sie den fehlenden Quellcode an entsprechender Stelle. Sie sollen die Session verwenden (fortsetzen): <pre>session_start(); //Ganz wichtig</pre> Übernehmen Sie die Werte aus der Session in lokale Variable: <pre>\$pName = \$_SESSION['username']; \$pPasswort = \$_SESSION['password']; \$pTrick = \$_SESSION['geheim'];</pre> Erzeugen Sie einen Datencontainer der die Selektion der Daten (Benutzername, Passwort) aus der Datenbank vorübergehend simulieren soll: <pre>\$db_daten = array('uname' => 'lisa', 'pw' => '\$2y\$12\$FWf2TMp2kunBNVUTSB D6ey34A136FrSR2Yyd6xDCwV/MKJi6Cdu',);</pre> Erzeugen Sie den Quellcode für die Verifizierung und übernehmen Sie das Ergebnis in eine lokale Variable: <pre>\$valid = password_verify(\$pPasswort.\$pTrick, \$db_daten['pw']);</pre> Prüfen Sie ob die Verifizierung erfolgreich war und der richtige Benutzername eingegeben wurde: <pre>if(\$valid == 1 && \$pName == \$db_daten['uname']) { \$pMeldung = "Passwort und Benutzername verifiziert!"; //Verweis auf Abmeldeseite \$pMeldung = \$pMeldung . "
Abmelden</h5>"; } else{ \$pMeldung = "Passwort oder Benutzername konnte nicht verifiziert werden!";</pre>

	<pre>//Verweis auf Anmeldeseite \$pMeldung = \$pMeldung . "
 Erneut Anmelden</h5>"; session_destroy(); }</pre> Ergebnis ausgeben: <pre>echo "Ergebnis: <h5>Du heißt immer noch ". \$_SESSION['username'] . "

";</pre> Meldung ausgeben: <pre>echo \$pMeldung . "
";</pre> Verweisen Sie auf eine andere Seite: <pre>echo "Abmelden</h5>";</pre>
→ zum Informationsblatt hash3.php	<i>Ergänzen Sie den fehlenden Quellcode an entsprechender Stelle.</i> Sie sollen die Session verwenden (fortsetzen): <pre>session_start(); //Ganz wichtig</pre> Übernehmen Sie den Wert aus der Session in eine lokale Variable: <pre>\$pName = \$_SESSION['username'];</pre> Löschen aller Session-Variablen: <pre>\$_SESSION = array();</pre> Löschen der Sitzung: <pre>session_destroy();</pre> Text ausgeben: <pre>echo "Ergebnis: <h5>Sie sind erfolgreich abgemeldet!

";</pre> Verweis auf die Startseite der Session-Lektion: <pre>echo " Erneut Anmelden</h5>";</pre>

7.3 Einführung Datenbankzugriff

7.3.1 Informationsblatt

Thema: 	Datenbankzugriff Autor: Christine Janischek Informationsblatt: Datenbankzugriff
---	---

PHP Datenbankanbindung für das Gäste

Ausser der Datenbank benötigen wir eine PHP-Klasse (DBCon). Diese Klasse legen wir in unserer PHP-Bibliothek (lib.php) ab. Sie stellt die benötigten Eigenschaften und Verhaltensweisen für unsere Datenverbindungsobjekte zur Verfügung.

Voraussetzung für die Implementierung des Gästebuchs ist der Import der dazugehörigen Datenbank:

- freundedb_Struktur.sql
- freundedb_Daten.sql

Außerdem müssen der Webserver (Apache) und das Datenbanksystem MySQL über das Xampp-Software-Paket oder ggf. über das Startmenü der Digitalen Tasche (Informatikstick) gestartet werden.

DBCon	Klassenname
<pre># \$conn; # \$host; # \$pass; # \$user; # \$db; # \$query = ""; # \$ergebnis = array();</pre>	Attribute
<pre>+ DBCon() + get_con() + connect() + select(\$pSQL) + update(\$pSQL) + insert(\$pSQL,\$pTabelle) + delete(\$pSQL) + error() + execute()</pre>	Konstruktor & Methoden

Algorithmik, Code Reading

Was passiert im Quellcode. Wir analysieren den Quellcode zeilenweise...

dblogin1.php:

Der Benutzer gibt den Benutzername und sein Passwort ein. Wenn der Benutzer auf die Schaltfläche → Anmelden klickt werden die folgenden Schritte ausgeführt:

1. Die Sitzung wird gestartet
2. Aus dem eingegeben Passwort (pPasswort) und dem geheimen Wort

(pTrick) wird, wie zuvor auch, ein Hashwert erzeugt.

3. Alle Werte (pName, pPasswort und pTrick) werden an entsprechende Sessionvariable übergeben.
4. Zum testen lassen wir uns den erzeugten Hashwert auf der Benutzeroberfläche ausgeben.

dblogin2.php:

Mit einem Klick auf die Schaltfläche → Weiter werden dann die folgenden Schritte ausgeführt.

1. Die Sitzung wird fortgeführt
2. Die in den Sitzungsvariablen gespeicherten Werte werden an lokale Variablen übergeben.
3. Die Datenbankverbindung wird hergestellt. Dazu wird ein Datenverbindungsobjekt erzeugt und die Verbindung aufgebaut.
4. Das Datenbankverbindungsobjekt wird einer Sessionvariablen übergeben.
5. Die SQL-Auswahl-Abfrage wird erzeugt und an eine lokale Variable übergeben.
6. Die Abfrage wird mittels des Datenbankverbindungsobjektes auf der Datenbank ausgeführt, der Rückgabewert wird in die lokale Variable \$ergebnis übernommen. Das Ergebnis beinhaltet nun die Zugangsdaten aller Benutzer (ist eine Liste).
7. Schleife/Wiederholung: Schrittweise (zeilenweise) werden die Zugangsdaten dem Ergebnis entnommen
 - Werte der Zeile (Datensatz) werden in lokale Variablen/Attribute übergeben
 - Das Passwort wird verifiziert, der Rückgabewert der Verifizierung wird in eine lokale Variable übernommen (0 oder 1)
 - Fallunterscheidung:

1.Fall:

Wenn die Verifizierung erfolgreich und der Benutzername stimmig sind (Wert = 1)

- Wird die Meldung "Passwort verifiziert!" erzeugt
- Die Werte an die Sessionvariablen übergeben
- Eine positive Ausgabe erzeugt und der Wiederholvorgang wird abgebrochen
- Fallunterscheidung:

2.Fall:

Ansonsten, also wenn keine Übereinstimmung zustande kommt (Wert = 0)

- Wird die Meldung "Passwort konnte nicht verifiziert werden!" erzeugt.

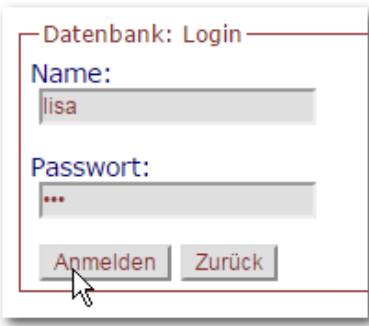
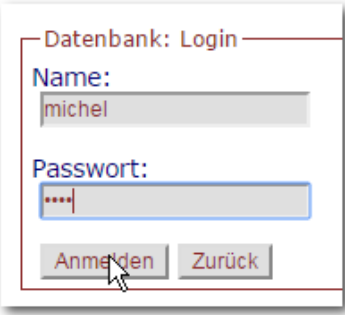
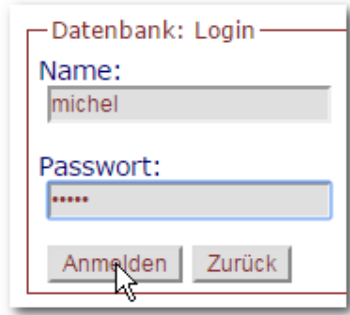
8. Ein Text mit der erzeugten Meldung wird auf der Benutzeroberfläche ausgegeben.

dblogin3.php:

Mit einem Klick auf die Schaltfläche → Abmelden werden dann die folgenden Schritte ausgeführt.

1. Die Sitzung wird gestartet bzw. fortgeführt.
2. Die in den Sitzungsvariablen gespeicherten Werte werden an lokale Variablen übergeben.
3. Daten der Sitzung werden gelöscht.
4. Die Sitzung wird gelöscht.
5. Ausgabe wird erzeugt.

Testfälle

		
Testfall 1: Eingabe Korrekt!	Testfall 2: Eingabe Korrekt!	Testfall 3: Eingabe Falsch!
Benutzername: lisa Passwort: 123  Lisa loggt sich richtig ein	Benutzername: michel Passwort: 1234  Michel loggt sich richtig ein	Benutzername: michel Passwort: 12345  Michel loggt sich falsch ein

Fragen zum Verständnis des Quellcodes:

1. Wie lauten die Quellcode-Anweisungen in PHP Anweisung, um eine Gästebuch-Sitzung (Session) zu starten bzw. fortzuführen?
2. Wie lautet die Quellcode-Anweisung in PHP Anweisung, um Werte an entsprechende Sitzungsvariablen, zu übergeben?
3. Wie lautet die Gegenoperation im Quellcode in PHP, um einen in einer Sitzungsvariablen gespeicherten Wert an eine lokale Variablen, zu übergeben?
4. Wie lauten die Quellcode-Anweisungen in PHP, um ein neues Datenbankverbindungsobjekt und den Aufbau der Datenbankverbindung zu erzeugen?
5. Wie lautet die Quellcode-Anweisungen in PHP, um das eingegebene Passwort, zu verifizieren?
6. Formulieren Sie in SQL die Auswahlabfrage zur Ermittlung der Zugangsdaten aller Benutzer des Gästebuchs!
7. Wie lautet die Meldung für den Fall, dass bei der Ermittlung der Zugangsdaten aller Benutzer des Gästebuchs keine Daten ermittelt werden konnten?
8. Wie lauten die Quellcode-Anweisungen in PHP, um die Daten der Sitzung abschließend zu löschen?

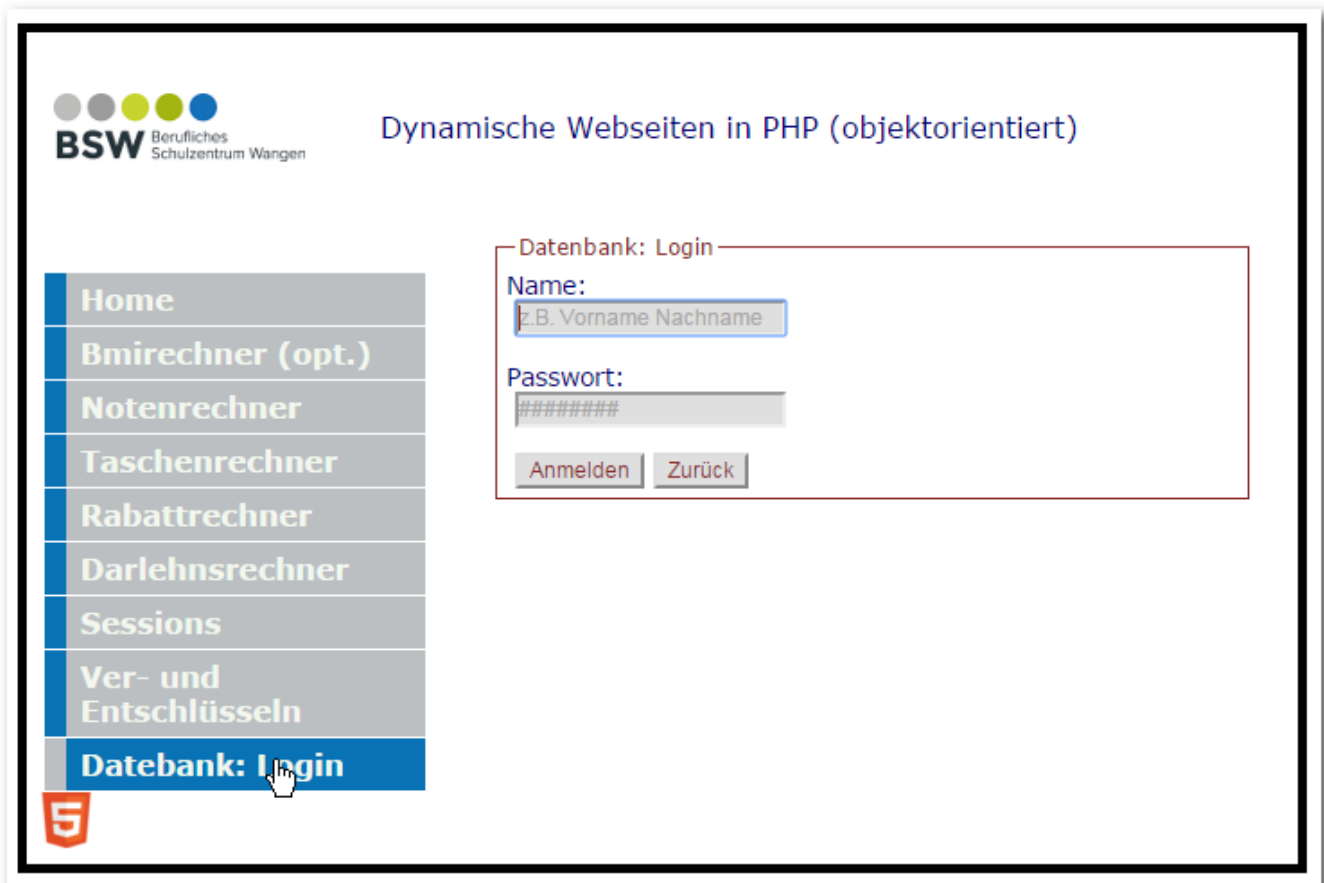
Zusatzaufgabe:

Studieren Sie die Datenbankarchitektur (freundedb). Wie müssen die SQL-Abfragen lauten, um:

- Posts (Beiträge) anzuzeigen?
- Post (Beitrag) löschen?
- Post (Beitrag) ändern?
- Post (Beitrag) einfügen?

7.3.2 Arbeitsblatt

Thema: 	Datenbankzugriff Autor: Christine Janischek Arbeitsblatt: Datenbankzugriff
---	--



Arbeitsauftrag:

1. Wir folgen dem informatischen Prinzip der Wiederverwendbarkeit und kopieren dazu den Inhalt aus dem vorgegebenen Projektverzeichnis in neues Projektverzeichnis.
2. Importieren Sie über Ihr Datenbank-Management-System (PHPmyadmin, MySQLWorkbench) die Struktur und Daten der Datenbank (freundedb).
3. Lesen Sie den [Anwendungsfall](#) aufmerksam durch.
4. Testen Sie das Projekt. (Testfälle)
5. Beantworten Sie im Anschluss die [Fragen](#).

Arbeitsmaterialien und Übungsprojekte finden Sie im E-Learning:

→ [E-Learning OOP](#)

7.4 Einführung Datenbankoperationen

7.4.1 Informationsblatt

Thema: 	Datenbankoperationen Autor: Christine Janischek Informationsblatt: Datenbankoperationen für das Gästebuch
---	---

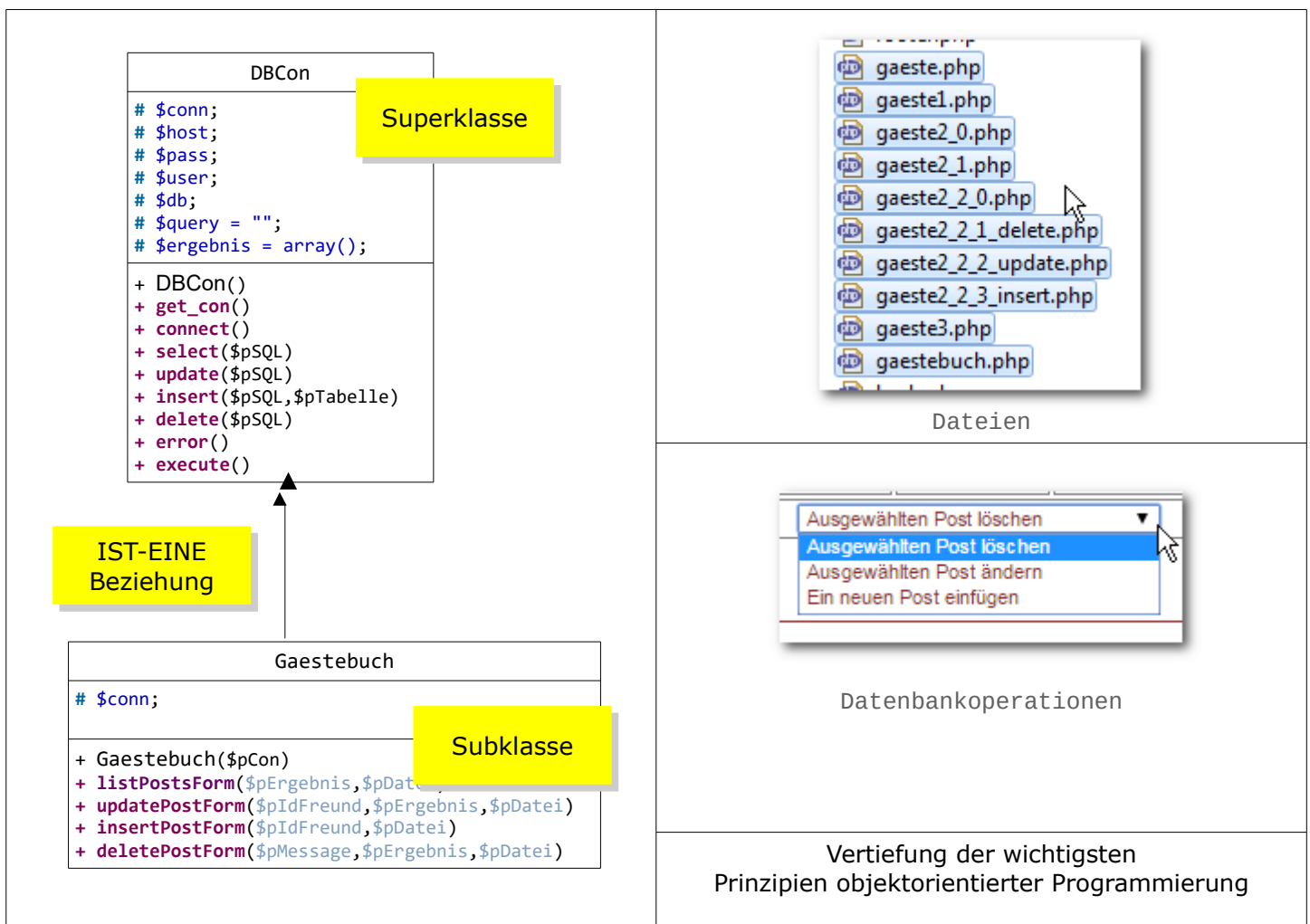
PHP Gästebuch

Voraussetzung für die Implementierung des Gästebuchs ist der Import der dazugehörigen Datenbank:

- freundedb_Struktur.sql
- freundedb_Daten.sql

Außerdem müssen der Webserver (Apache) und das Datenbanksystem MySQL über Xampp-Software-Paket oder alternativ über das Startmenü der Digitalen Tasche (Informatikstick) gestartet werden.

[→ zurück zum Arbeitsblatt](#)



	→ Abstraktion Verwendung von Klassenkonzepten zur Sicherstellung der Wiederverwendbarkeit und Erweiterbarkeit
	→ Kapselung Verwendung von Zugriffsmodifikatoren (z.B. private, public, protected,...) um den direkten Zugriff auf Werte regulieren zu können und Sicherheit zu gewährleisten.
	→ Vererbung Zur Sicherstellung der Wiederverwendbarkeit und Erweiterbarkeit von Eigenschaften und Verhaltensweisen von Objekten.
	→ Polymorphie Vielgestaltigkeit von Methoden (deren Implementierung). Um Flexibilität und Variabilität bezüglich der Implementierung von Verhaltensweisen zu gewährleisten

Anwendungsfälle Gästebuch		
		
Testfall 1: Eintrag löschen!	Testfall 2: Eintrag ändern!	Testfall 3: Eintrag einfügen!
Benutzername: lisa Passwort: 123 Datenbankoperation: Ausgewählter Post löschen ausloggen!	Benutzername: michel Passwort: 1234 Datenbankoperation: Ausgewählter Post ändern ausloggen!	Benutzername: lisa Passwort: 123 Datenbankoperation: Einen neuen Post einfügen ausloggen!

7.4.2 Arbeitsblatt

Thema: 	Datenbankoperationen Autor: Christine Janischek Arbeitsblatt: Gästebuch
---	---


Dynamische Webseiten in PHP (objektorientiert)

- Home
- Bmirechner (opt.)
- Notenrechner
- Taschenrechner
- Rabattrechner
- Darlehnsrechner
- Sessions
- Ver- und Entschlüsseln
- Datenbank: Login
- Datenbank: Gästebuch**

Datenbank: Das Gästebuch

Posts anzeigen

Datum	Gast ID	Gast Name	Post ID	Mitteilung	Auswahl
2015-11-22	1	lisa	8	Ich bin dabei!	☐
2015-11-22	1	lisa	9	Noch ein Versuch!	☐
2015-11-11	3	manuel	6	Wie wäre es wenn ein offizielles Freunde-Treffen organisieren?	☐
2015-11-10	2	michel	5	Na klar!	☐
2015-11-09	2	michel	2	Schön, dass es Dich gibt!	☐
2015-11-08	1	lisa	4	Meinst Du wir können uns bald wiedersehen? Hallo	☐

[Abmelden](#)

Anzeige aller Posts

Arbeitsauftrag:

Sie sollen die bestehende Anwendung optimieren!

1. Wir folgen dem informatischen Prinzip der Wiederverwendbarkeit und kopieren dazu den Inhalt aus dem vorgegebenen Projektverzeichnis in neues Projektverzeichnis.
2. Studieren Sie die Architektur und Implementierung der Anwendung aufmerksam. Testen Sie die Anwendung! Beachten und Nutzen Sie die Hinweise im [Informationsblatt](#).
3. Lokalisieren Sie die SQL-Abfragen und Wiederholstrukturen im Quellcode.
4. Versuchen Sie durch die Anwendung der informatischen Prinzipien „Kapselung“ und „Wiederverwendung“ den Quellcode zu optimieren.

Arbeitsmaterialien und Übungsprojekte finden Sie im E-Learning:

→ [E-Learning OOP](#)