

# Dynamische Webprogrammierung

## Unterrichtsdokumentation

Modulname:	Dynamische Webprogrammierung in der Sek 2
------------	---

Stand: 23. Apr 2020



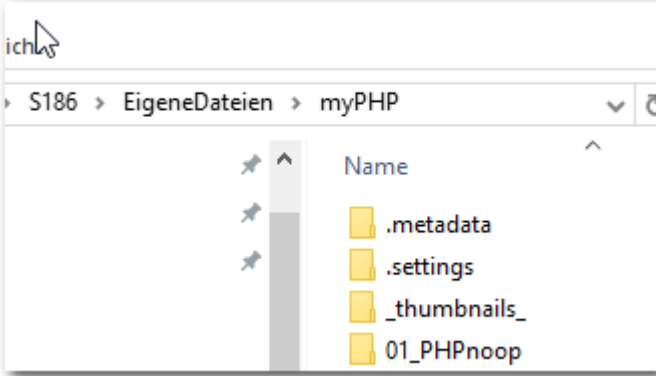
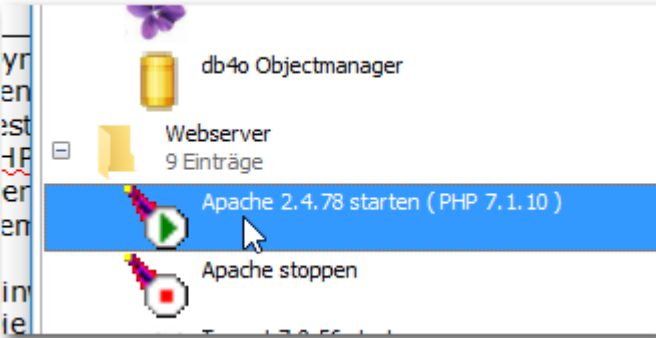
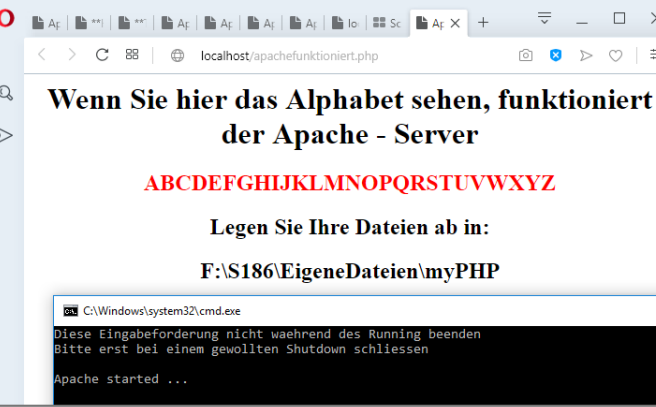
© **BS-Wangen**

## Inhaltsverzeichnis

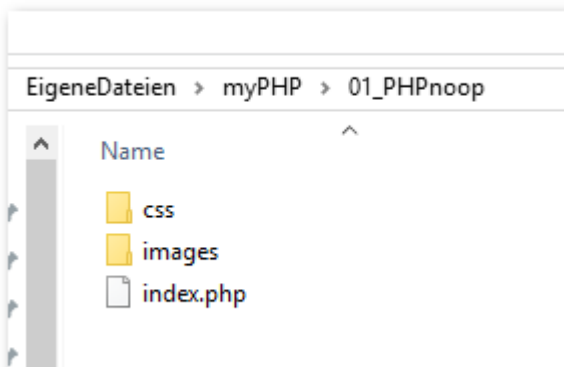
1 Dynamische Webprogrammierung.....	3
1.1 Einführung.....	3
1.2 Hallo Welt.....	4
1.3 Dynamisches Layout: Grundgerüst einer Webseite.....	6
1.4 Formulare: Der Bmirechner.....	9
1.5 Formulare: Der Notenrechner.....	13
1.6 Kontrollstrukturen.....	16
1.7 Kontrollstrukturen: Der Taschenrechner.....	18
1.8 Kontrollstrukturen: Der Rabattrechner.....	23

# 1 Dynamische Webprogrammierung

## 1.1 Einführung

	<p><i>Projektdateien speichern.</i></p> <p>Dynamische Webseiten werden zum großen Teil in PHP umgesetzt. Um die Seiten testen und ausführen zu können, wird ein PHP-Interpreter benötigt. Der Interpreter läuft in der Regel auf einem (lokalen) Webserver.</p> <p>Hinweis: Die PHP-Projekte müssen deshalb im www-Verzeichnis liegen. Wenn die Digitale Tasche verwendet wird zeigt dieses Verzeichnis auf den Ordner → myPHP.</p> <p>Den Webserver muss man zuvor starten.</p>
	<p><i>Webserver starten.</i></p> <p>Dazu in der Digitalen Tasche den Webserver Apache starten.</p> <p>Der lokale Server ist i.d. R. Unter der IP-Adresse 127.0.0.1 verfügbar.</p>
	<p><i>Startklar!</i></p> <p>Wenn der Webserver erfolgreich gestartet wurde öffnet sich erst ein schwarzes Konsolenfenster und anschließend der Browser Opera mit der nebenstehend angezeigten Seite.</p> <p>Jetzt können wir programmieren!!</p> <p><b>Das schwarze Konsolenfenster muss geöffnet bleiben.</b></p>

## 1.2 Hallo Welt



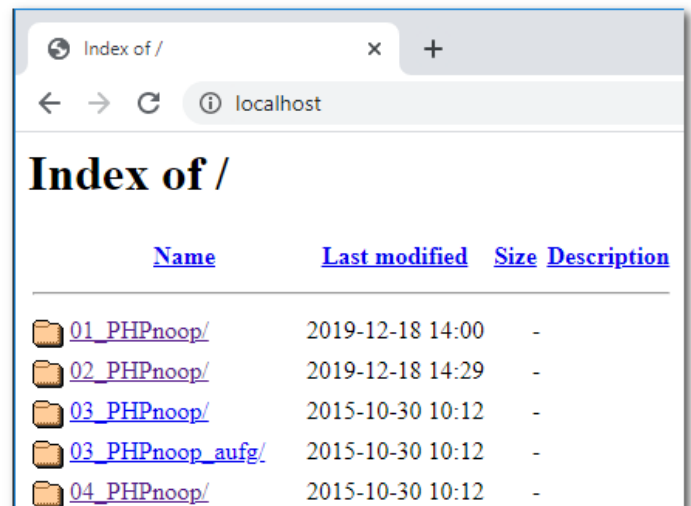
### Projektstruktur.

#### Hinweis:

Achten Sie darauf, dass ihr Projekt im www-Verzeichnis liegt → Digitale Tasche → EigeneDateien → myPHP

Der Webserver „Apache“ muss gestartet sein!

Geben Sie im Browser die Adresse ein: <http://localhost/>



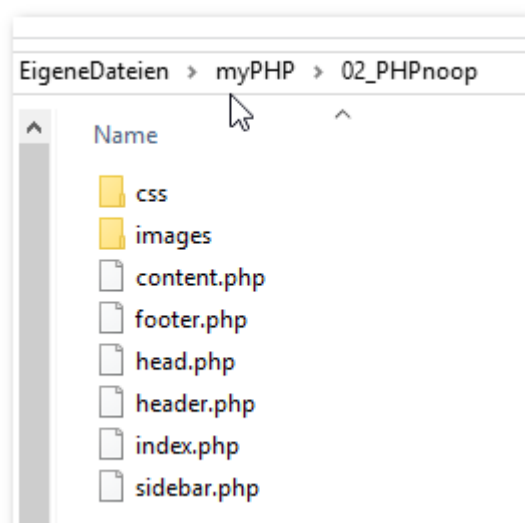
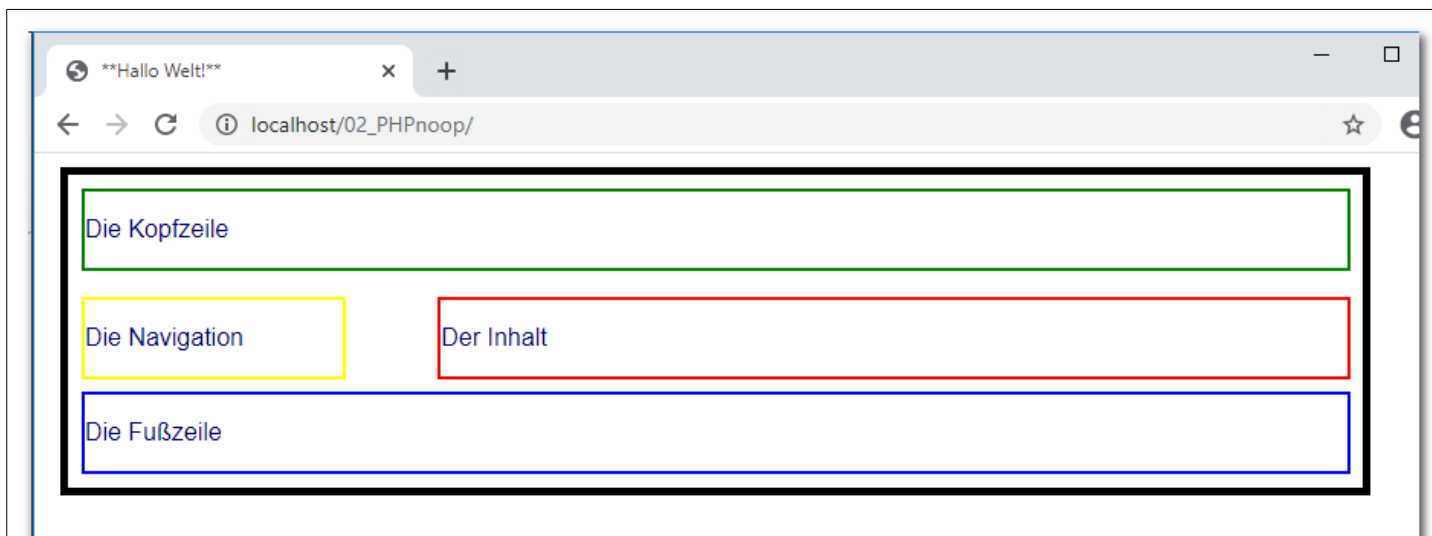
```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>**Hallo Welt!**</title>
5     <meta name="author" content="Ihr Name">
6     <meta name="keywords" content="Hallo, Welt">
7     <meta name="description" content="Meine erste PHP-Seite">
8     <style type="text/css">
9       @import "css/styles.css";
10    </style>
11  </head>
12  <body>
13    <?php
14      echo "Hallo sch&ouml;ne Welt!"
15    ?>
16  </body>
17 </html>
```

### Einfache Ausgaben in PHP.

Werden mit dem echo-Befehl erzeugt:

```
<?php
echo "Hallo sch&ouml;ne Welt!"
?>
```

### 1.3 Dynamisches Layout: Grundgerüst einer Webseite



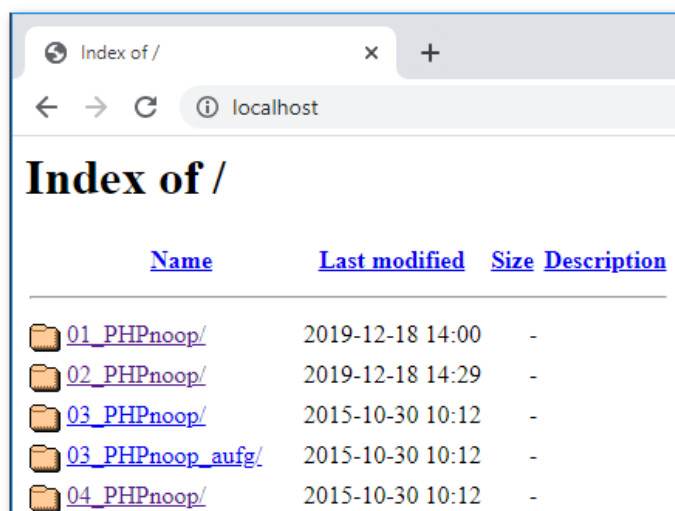
#### Projektstruktur.


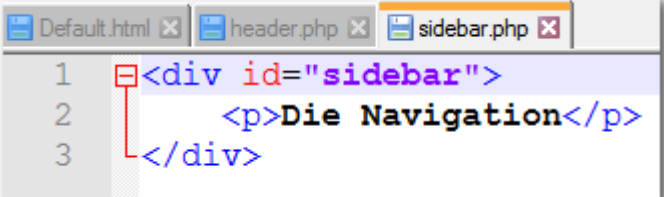
##### Hinweis:

Achten Sie darauf, dass ihr Projekt im www-Verzeichnis liegt → Digitale Tasche → EigeneDateien → myPHP

Der Webserver „Apache“ muss gestartet sein!

Geben Sie im Browser die Adresse ein: <http://localhost/>



	<p><i>Prinzip Divide and Conquer.</i></p> <p>→ Teile und Herrsche</p> 
	<p><i>Der Kopfzeile</i></p> <p><input type="checkbox"/> <i>header.php</i></p>
	<p><i>Die Navigation</i></p> <p><input type="checkbox"/> <i>sidebar.php</i></p>
	<p><i>Der Inhalt</i></p> <p><input type="checkbox"/> <i>content.php</i></p>
	<p><i>Die Fußzeile</i></p> <p><input type="checkbox"/> <i>footer.php</i></p>

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>**Hallo Welt!**</title>
5     <meta name="author" content="Ihr Name">
6     <meta name="keywords" content="Hallo, Welt">
7     <meta name="description" content="Meine erste PHP-Seite">
8     <style type="text/css">
9       @import "css/styles.css";
10    </style>
11  </head>
```

Der Kopf.

*head.php*

Im Kopf des HTML Dokuments wird u.a. auf das verwendete Stylesheet verwiesen.

```
1 <?php
2   include ("head.php");
3   ?>
4 <body>
5   <div id="wrapper">
6     <?php
7       include ("header.php");
8     ?>
9     <?php
10      include ("sidebar.php");
11    ?>
12    <?php
13      include ("content.php");
14    ?>
15    <?php
16      include ("footer.php");
17    ?>
18  </div>
19 </body>
20 </html>
```

Zusammenführung aller Seitenbestandteile.

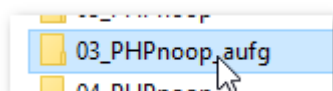
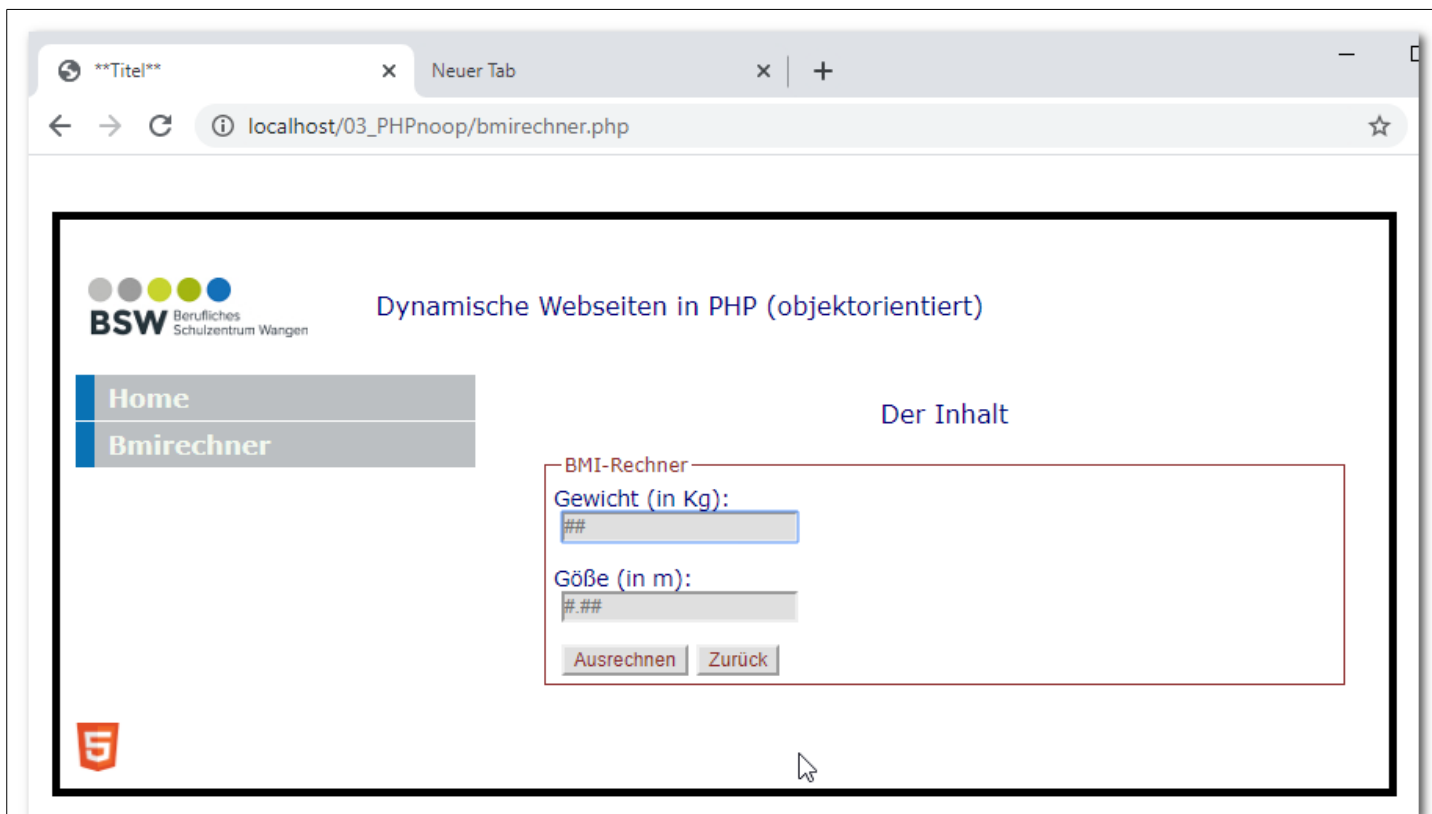
*index.php*

Testen

[http://localhost/02\\_PHPnoop/](http://localhost/02_PHPnoop/)



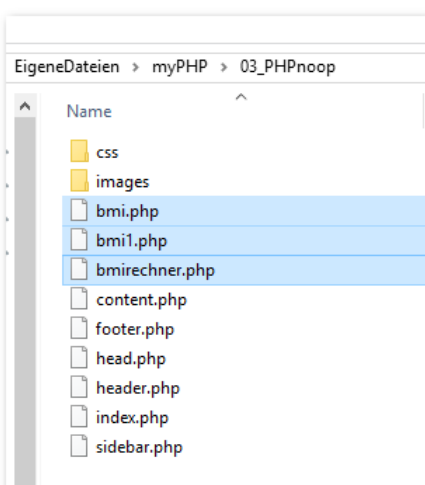
### 1.4 Formulare: Der Bmirechner



Ausgangsprojekt.


→ 03\_PHPnoop\_aufg

Kopieren Sie das Projekt, fügen Sie es ein und benennen Sie es um → 03\_PHPnoop



Tätigkeiten:

1. Sidebar erweitern
2. index.php kopieren, einfügen und in bmirechner umbenennen
3. content.php kopieren, einfügen und in bmi umbenennen
4. in der Datei bmirechner auf die → bmi.php verweisen
5. in die Datei bmi.php das Formular einfügen
6. eine neue Datei bmi1.php anlegen und den Ablaufplan für die Anwendung festlegen.

 <pre> 1 2 3 4 &lt;div id="sidebar"&gt; 5     &lt;ul&gt; 6         &lt;li&gt;&lt;a href="index.php"&gt;Home&lt;/a&gt;&lt;/li&gt; 7         &lt;li&gt;&lt;a href="bmirechner.php" target="_parent"&gt; 8             Bmirechner&lt;/a&gt;&lt;/li&gt; 9     &lt;/ul&gt; 10 &lt;/div&gt; 11                 </pre>	<p>7. Testen...</p> <p>Die Sidebar</p>
 <pre> 1 &lt;?php 2     include ("head.php"); 3     ?&gt; 4 5 &lt;body&gt; 6 7     &lt;div id="wrapper"&gt; 8         &lt;?php 9             include ("header.php"); 10            ?&gt; 11 12            &lt;?php 13                include ("bmi.php"); 14            ?&gt; 15 16            &lt;?php 17                include ("sidebar.php"); 18            ?&gt; 19 20            &lt;?php 21                include ("footer.php"); 22            ?&gt; 23        &lt;div&gt; 24 25    &lt;/body&gt; 26 &lt;/html&gt; 27                 </pre>	<p>Die Zusammenführung der Dateien</p> <p>→ bmirechner.php</p>
 <pre> 1 &lt;div id="content"&gt; 2     &lt;p&gt;Der Inhalt&lt;/p&gt; 3     &lt;form name="bmirechnerformular" method="post" action="bmi.php"&gt; 4         &lt;fieldset&gt; 5             &lt;legend&gt; BMI-Rechner&lt;/legend&gt; 6             &lt;label for="tfGewicht"&gt; 7                 Gewicht (in Kg): 8             &lt;/label&gt; &lt;br /&gt; 9             &lt;input type="text" name="tfGewicht" id="tfGewicht" 10                placeholder="###" required="required" 11                autofocus="autofocus" /&gt;&lt;br /&gt; 12 13            &lt;label for="tfGroesse"&gt; 14                Größe (in m): 15            &lt;/label&gt; &lt;br /&gt; 16            &lt;input type="text" name="tfGroesse" id="tfGroesse" 17                placeholder="#.##" required="required" /&gt;&lt;br /&gt; 18 19            &lt;input type="submit" value="Ausrechnen" name="bmiAusrechnen"/&gt; 20            &lt;?php echo &lt;input type="button" value="zurück" onClick='history.back()' /&gt; 21        &lt;/fieldset&gt; 22    &lt;/form&gt; 23 &lt;/div&gt;                 </pre>	<p>Das Formular</p> <p>→ bmi.php</p>

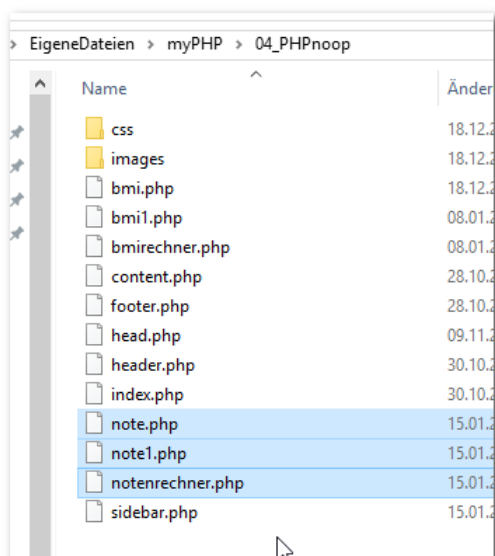
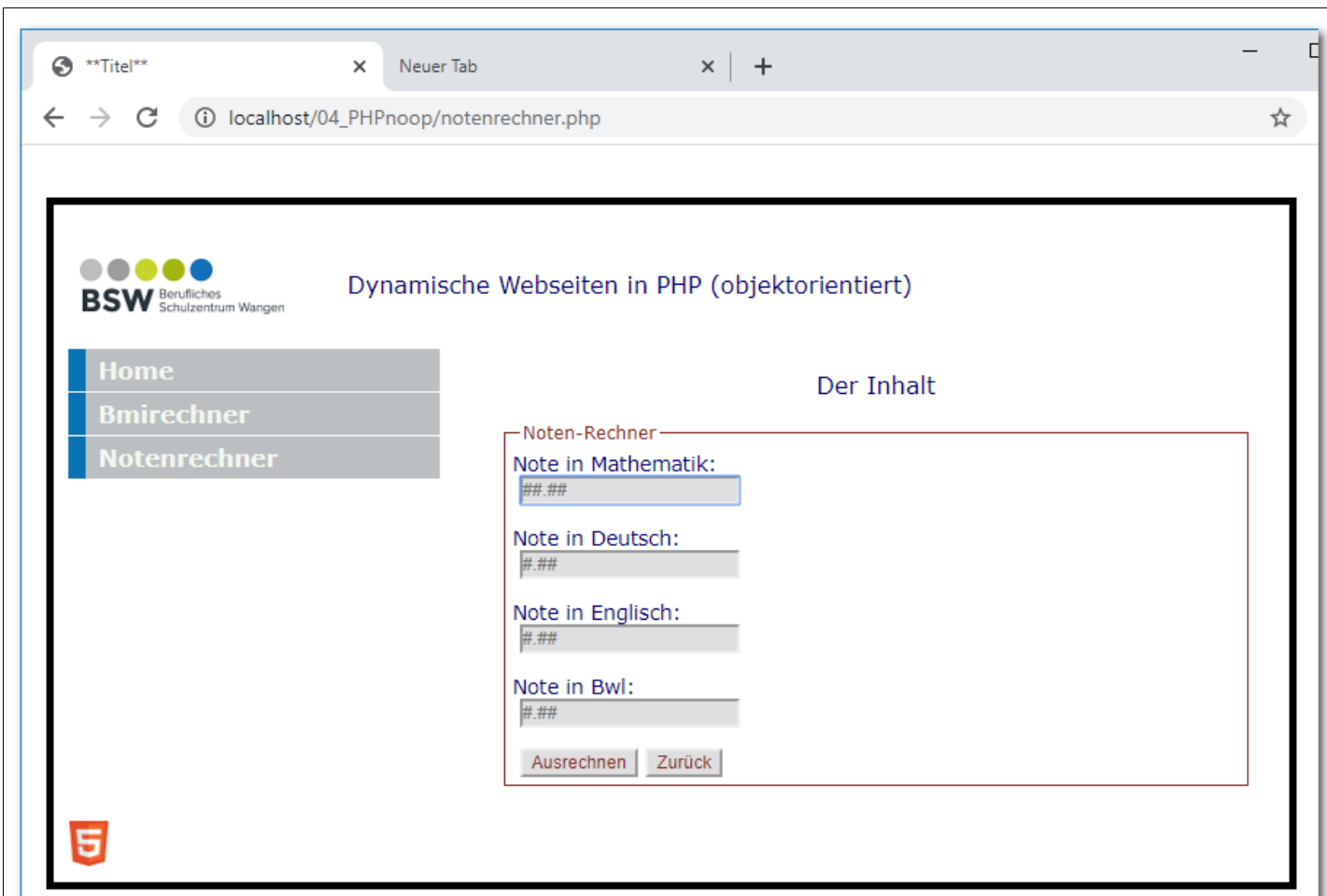
```
1 <?php
2 include ("head.php");
3 ?>
4 <body>
5 <div id="wrapper">
6 <?php
7 include ("header.php");
8 ?>
9 <div id="content">
10 <form>
11 <fieldset>
12 <legend>Bmi berechnen</legend>
13 <?php
14 //Eingaben lesen und in Variablen übernehmen
15 $pGewicht = $_POST['tfGewicht'];
16 $pGroesse = $_POST['tfGroesse'];
17
18 //Funktion/Methode definieren
19 function berechne_bmi ($pGroesse,$pGewicht) {
20     ///###Eingaben###
21     $mGewicht = $pGewicht;
22     $mGroesse = $pGroesse;
23
24     ///###Verarbeitung###
25     $mBmi = $mGewicht/ ($mGroesse*$mGroesse);
26     return $mBmi;
27 }
28
29 //Methoden- oder Funktionsaufruf: Die Berechnung
30 $ausgabe = berechne_bmi ($pGroesse,$pGewicht);
31
32 //Ausgaben konfigurieren
33 echo "Gewicht (in Kg):<h5>" . $pGewicht . "</h5>";
34 echo "Gröuml;sslig:e (in m):<h5>" . $pGroesse . "</h5>";
35 echo "Ergebnis:" . round ($ausgabe, 2) ;|
36 ?>
37 </fieldset>
38 </form>
39 </div>
40 <?php
41 include ("sidebar.php");
42 ?>
43
44 <?php
45 include ("footer.php");
46 ?>
47 </div>
48
49 </body>
```

### Der Ablaufplan der Anwendung

→ bmi1.php

- Prinzip:
- E**ingabe
- V**erarbeitung
- A**usgabe

## 1.5 Formulare: Der Notenrechner

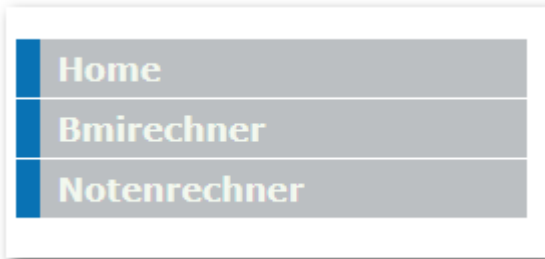


### Projektstruktur.

Hinweis:  
Achten Sie darauf, dass ihr Projekt im www-Verzeichnis liegt → Digitale Tasche → EigeneDateien → myPHP

Der Webserver „Apache“ muss gestartet sein!

Geben Sie im Browser die Adresse ein:  
<http://localhost/>



### Die Sidebar

```

1
2
3
4
5
6
7
8
9
10
11
<div id="sidebar">
  <ul>
    <li><a href="index.php">Home</a></li>
    <li><a href="bmirechner.php" target="_parent">Bmirechner</a></li>
    <li><a href="notenrechner.php" target="_parent">Notenrechner</a></li>
  </ul>
</div>

```

→ sidebar.php



### Das Formular

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
<div id="content">
  <p>Der Inhalt</p>
  <form name="notenrechnerformular" method="post" action="note1.php">
    <fieldset>
      <legend>Noten-Rechner</legend>
      <label for="tfMathe">
        Note in Mathematik:
      </label> <br />
      <input type="text" name="tfMathe" id="tfMathe"
        placeholder="###" required="required" autofocus="autofocus" /><br />
      <label for="tfDeutsch">
        Note in Deutsch:
      </label> <br />
      <input type="text" name="tfDeutsch" id="tfDeutsch"
        placeholder="###" required="required" /><br />
      <label for="tfEnglisch">
        Note in Englisch:
      </label> <br />
      <input type="text" name="tfEnglisch" id="tfEnglisch"
        placeholder="###" required="required" /><br />
      <label for="tfBwl">
        Note in Bwl:
      </label> <br />
      <input type="text" name="tfBwl" id="tfBwl"
        placeholder="###" required="required" /><br />
      <input type="submit" value="Ausrechnen" name="noteAusrechnen" />
      <?php echo "input type='button' value='Zurück' onClick='history.back()' />" ?>
    </fieldset>
  </form>
</div>

```

→ note.php

```

<fieldset>
  <legend>Notendurchschnitt berechnen</legend>
  <?php
    $pMathe = $_POST['tfMathe'];
    $pDeutsch = $_POST['tfDeutsch'];
    $pEnglisch = $_POST['tfEnglisch'];
    $pBwl = $_POST['tfBwl'];

    function berechne_notendurchschnitt($pMathe,$pDeutsch,$pEnglisch,$pBwl) {
      #####Eingaben####
      $mMathe = $pMathe;
      $mDeutsch = $pDeutsch;
      $mEnglisch = $pEnglisch;
      $mBwl = $pBwl;

      #####Verarbeitung####
      $mDurchschnitt = ($mMathe+$mDeutsch+$mEnglisch+$mBwl)/4;
      return $mDurchschnitt;
    }

    //Methoden- oder Funktionsaufruf: Die Berechnung
    $ausgabe = berechne_notendurchschnitt($pMathe,$pDeutsch,$pEnglisch,$pBwl);

    echo "Mathematiknote:<h5>". $pMathe. "</h5>";
    echo "Deutschnote:<h5>". $pDeutsch. "</h5>";
    echo "Englischnote:<h5>". $pEnglisch. "</h5>";
    echo "Bwlnote:<h5>". $pBwl. "</h5>";
    echo "Ergebnis:". round($ausgabe,2);
  <?>
</fieldset>

```

### Der Ablaufplan.

→ note1.php

### Hinweis:

Im Anschluss an die Definition der benötigten Funktionen, folgt für die gewünschte Ausgabe, der Aufruf (Funktions- oder Methodenaufruf) der Funktion(en) und die echos für die Anzeige des Ergebnisses.

```
notenrechner.php x
1  <?php
2      include ("head.php");
3  ?>
4
5  <body>
6
7      <div id="wrapper">
8          <?php
9              include ("header.php");
10             ?>
11
12             <?php
13                 include ("note.php");
14             ?>
15
16             <?php
17                 include ("sidebar.php");
18             ?>
19
20             <?php
21                 include ("footer.php");
22             ?>
23             <div>
24
25         </div>
26     </body>
27 </html>
```

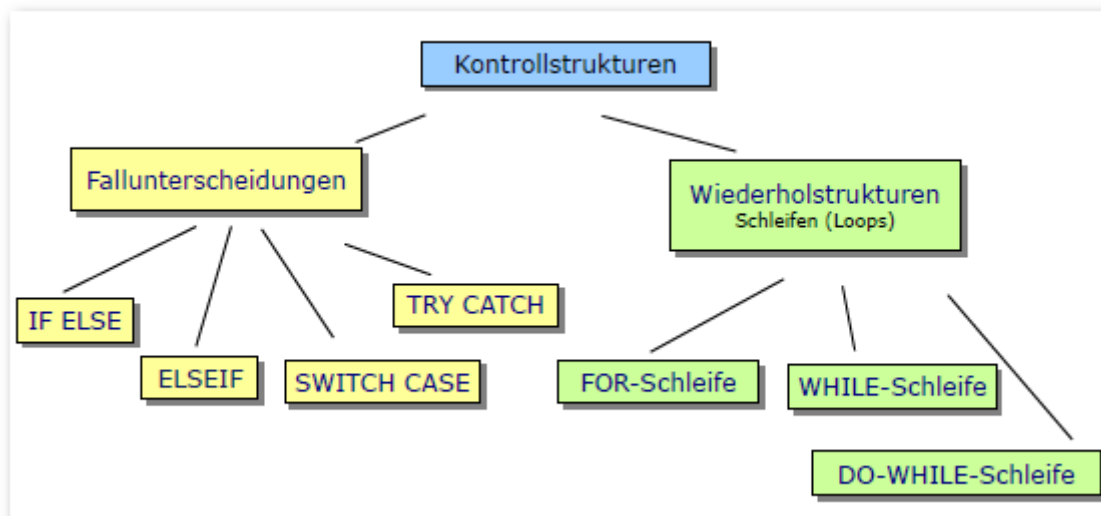
## Zusammenführung der Dateien

→ notenrechner.php

## Testen

[http://localhost/04\\_PHPnoop/notenrechner.php](http://localhost/04_PHPnoop/notenrechner.php)

## 1.6 Kontrollstrukturen



### Überblick zu allen Kontrollstrukturen

Hinweis: Wenn Ihr die Kontrollstrukturen in der Klassenarbeit nennen sollt müsst Ihr diese Zeichnung erstellen!!!! (auch die Wiederholstrukturen)

#### IF ELSE

```

if(Bedingung){
    //Anweisung(en)
}else{
    //weitere Anweisung(en)
}
    
```

**Hier Beispiel für den Notenrechner → Prüfung des Notendurchschnitts einfügen!**

#### ELSE IF

```

if(Bedingung1){
    //Anweisung(en)
}elseif(Bedingung2){
    //weitere Anweisung(en)
}
    
```

```

//weitere Anweisung(en)
}elseif(Bedingung3){
    //weitere Anweisung(en)
}
}
    
```

Hier angewendet für die Ermittlung des Rabattsatzes beim Rabattrechner:

```

//Funktion/Methode definieren
function ermittle_rabattsatz($pMenge){
    $mMenge = $pMenge;

    if($mMenge >= 150){
        $mRabattsatz = 12;
    }elseif($mMenge >= 100){
        $mRabattsatz = 10;
    }elseif($mMenge >= 50){
        $mRabattsatz = 8;
    }elseif($mMenge >= 20){
        $mRabattsatz = 6;
    }else{
        $mRabattsatz = 0;
    }

    return $mRabattsatz;
}
    
```

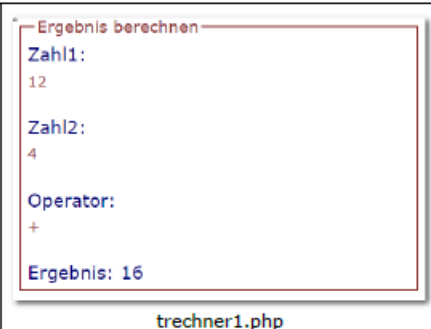
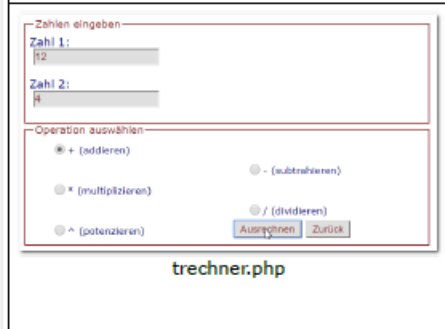
## SWITCH CASE

```
switch(Ausdruck){  
  
    case ausdruck_1:  
        //Anweisung(en)  
        break;  
  
    case ausdruck_2:  
        //Anweisung(en)  
        break;  
  
    case ausdruck_3:  
        //Anweisung(en)  
        break;  
  
    default:  
        //Anweisung(en)  
}
```

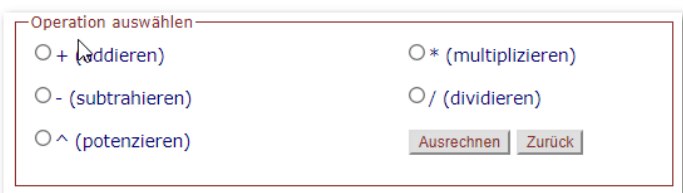
```
function pruefe($pOperator) {  
    #####EINGABE#####  
    $mOperator = $pOperator;  
  
    #####Verarbeitung###  
  
    //hier Reihenfolge der Operatoren berücksichtigen  
    switch($mOperator){  
        case 1:  
            $mOperator = "+";  
            break;  
        case 2:  
            $mOperator = "-";  
            break;  
        case 3:  
            $mOperator = "*";  
            break;  
        case 4:  
            $mOperator = "/";  
            break;  
        case 5:  
            $mOperator = "^";  
            break;  
        default:  
            echo "Bitte wählen Sie eine Rechenoperation aus!";  
    }  
    return $mOperator;  
}
```



### 1.7 Kontrollstrukturen: Der Taschenrechner



Im Formular waren neu die Radiobuttons:



trechner.php

```

<fieldset>
  <legend>Operation ausw&auml;hlen</legend>
  <div id="left">
    <input type="radio" name="rbOperator" value="1"/>+ (addieren)<br/><br/>
    <input type="radio" name="rbOperator" value="2"/>- (subtrahieren)<br/><br/>
    <input type="radio" name="rbOperator" value="5"/>^ (potenzieren)<br/><br/>
  </div>
  <div id="right">
    <input type="radio" name="rbOperator" value="3"/>*(multiplizieren)<br/><br/>
    <input type="radio" name="rbOperator" value="4"/>/ (dividieren)<br/><br/>
    <input type="submit" value="Ausrechnen" name="Ausrechnen"/>
  </div>
  <?php echo"<input type='button' value='Zur&uuml;ck'
</div>
</fieldset>
    
```

Hilfsfunktionen zur Berechnung:  
addieren, subtrahieren, dividieren, potenzieren, multiplizieren

Beispiele:

für die Addition:

```
//Hilfsfunktionen
function addieren($pZahl1, $pZahl2){
    $mZahl1 = $pZahl1;
    $mZahl2 = $pZahl2;

    //hier vergessener Semicolon
    $mErgebnis = $mZahl1 + $mZahl2;

    return $mErgebnis;
}
```

für die Potenz:

```
function potenzieren($pZahl1, $pZahl2){
    $mZahl1 = $pZahl1;
    $mZahl2 = $pZahl2;

    $mErgebnis = pow($mZahl1, $mZahl2);

    return $mErgebnis;
}
```

→ pow(\$zahl1, \$zahl2) ist eine PHP-Funktion aus der Bibliothek

Prüfung des gewählten Rechenoperators und Übernahme eines entsprechenden Zeichens für die Rechenoperation.

→ siehe auch Verwendung der Kontrollstruktur SWITCH CASE

```
//Verhaltensweisen: Berechnende Funktion Methode//
function pruefe($pOperator) {
    #####EINGABE#####
    $mOperator = $pOperator;

    #####Verarbeitung#####

    //hier Reihenfolge der Operatoren berücksichtigen
    switch($mOperator){
        case 1:
            $mOperator = "+";
            break;
        case 2:
            $mOperator = "-";
            break;
        case 3:
            $mOperator = "*";
            break;
        case 4:
            $mOperator = "/";
            break;
        case 5:
            $mOperator = "^";
            break;
        default:
            echo "Bitte wählen Sie eine Rechenoperation aus!";
    }

    return $mOperator;
}
```

	<p>Erläuterung: Prüfe anhand der Eingabevariable (Eingabewert) pOperator.</p> <p>Übernehme den Eingabewert in eine lokale Variable mOperator.</p> <p>Für den Fall, dass mOperator den Wert 1 hat. Überschreibe mOperator mit dem Zeichen „+“. Breche die Prüfung ab, wenn dieser Fall eingetroffen ist.</p> <p>Für den Fall, dass mOperator den Wert 2 hat. Überschreibe mOperator mit dem Zeichen „-“. Breche die Prüfung ab, wenn dieser Fall eingetroffen ist.</p> <p>Für den Fall, dass mOperator den Wert 3 hat. Überschreibe mOperator mit dem Zeichen „*“. Breche die Prüfung ab, wenn dieser Fall eingetroffen ist.</p> <p>Für den Fall, dass mOperator den Wert 4 hat. Überschreibe mOperator mit dem Zeichen „/“. Breche die Prüfung ab, wenn dieser Fall eingetroffen ist.</p> <p>Für den Fall, dass mOperator den Wert 5 hat. Überschreibe mOperator mit dem Zeichen „^“. Breche die Prüfung ab, wenn dieser Fall eingetroffen ist.</p> <p>Für den Fall, dass keine Eingabe erfolgt ist, tritt der Standardfall (Default) ein. Es erfolgt die Ausgabe: "Bitte wählen Sie eine Rechenoperation aus!"</p> <p>Gebe mOperator zurück (Rückgabewert).</p>
--	--

Berechne anhand der Eingabewerte für Zahl1, Zahl2 und pOperator

```
function berechne($pZahl1,$pZahl2,$pOperator) {  
  #####EINGABE#####  
  $mOperator = $pOperator;  
  $mZahl1 = $pZahl1;  
  $mZahl2 = $pZahl2;  
  
  ####Verarbeitung####  
  switch($mOperator){  
    case 1:  
      $mErgebnis = addieren($mZahl1,$mZahl2);  
      break;  
    case 2:  
      $mErgebnis = subtrahieren($mZahl1,$mZahl2);  
      break;  
    case 3:  
      $mErgebnis = multiplizieren($mZahl1,$mZahl2);  
      break;  
    case 4:  
      $mErgebnis = dividieren($mZahl1,$mZahl2);  
      break;  
    case 5:  
      $mErgebnis = potenzieren($mZahl1,$mZahl2);  
      break;  
    default:  
      echo"Bitte w&auml;hlen Sie eine Rechenoperation aus!";  
  }  
  return $mErgebnis;  
}
```

Erläuterung:

Berechne anhand der Eingabewerte für Zahl1, Zahl2 und pOperator.

Übernehme die Eingabewerte in die lokalen Variablen mOperator, mZahl1, mZahl2.

Für den Fall, dass mOperator den Wert 1 hat. Führe die Hilfsfunktion „addieren“ aus und fange das Ergebnis in der lokalen Variable mErgebnis auf.


Für den Fall, dass mOperator den Wert 2 hat. Führe die Hilfsfunktion „subtrahieren“ aus und fange das Ergebnis in der lokalen Variable mErgebnis auf. Breche die Prüfung ab, wenn dieser Fall eingetroffen ist.

Für den Fall, dass mOperator den Wert 3 hat. Führe die Hilfsfunktion „multiplizieren“ aus und fange das Ergebnis in der lokalen Variable mErgebnis auf. Breche die Prüfung ab, wenn dieser Fall eingetroffen ist.

Für den Fall, dass mOperator den Wert 4 hat. Führe die Hilfsfunktion „dividieren“ aus und fange das Ergebnis in der lokalen Variable mErgebnis auf. Breche die Prüfung ab, wenn dieser Fall eingetroffen ist.

	<p>Für den Fall, dass mOperator den Wert 5 hat. Führe die Hilfsfunktion „potenzieren“ aus und fange das Ergebnis in der lokalen Variable mErgebnis auf. Breche die Prüfung ab, wenn dieser Fall eingetroffen ist.</p> <p>Für den Fall, dass keine Eingabe erfolgt ist, tritt der Standardfall (Default) ein. Es erfolgt die Ausgabe: "Bitte wählen Sie eine Rechenoperation aus!"</p> <p>Gebe mErgebnis zurück (Rückgabewert).</p>
Testen:	

### 1.8 Kontrollstrukturen: Der Rabattrechner



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:  
 $rabattbetrag = betrag / 100 * rabattsatz;$

Des Zahlungsbetrags:  
 $zahlungsbetrag = betrag - rabattbetrag;$

Rabatt berechnen

Betrag: 100 €

Menge: 110

Rabattsatz: 10%

Rabattbetrag: 10 €

Ergebnis (Zahlungsbetrag): 90 €

**Angewendete Prinzipien der Informatik:**

- Teile und Herrsche
- Wiederverwendungsprinzip

**Wir teilen das komplexe Problem in die Teilprobleme:**

- Rabattsatz ermitteln
- Rabattbetrag berechnen
- Zahlungsbetrag berechnen

<pre>//Funktion/Methode definieren function ermittle_rabattsatz(\$pMenge) {     \$mMenge = \$pMenge;      if(\$mMenge &gt;= 150) {         \$mRabattsatz = 12;     }elseif(\$mMenge &gt;= 100) {         \$mRabattsatz = 10;     }elseif(\$mMenge &gt;= 50) {         \$mRabattsatz = 8;     }elseif(\$mMenge &gt;= 20) {         \$mRabattsatz = 6;     }else{         \$mRabattsatz = 0;     }      return \$mRabattsatz; }</pre>	<p>Erläuterung: Ermittle den Rabattsatz anhand der eingegebenen Menge (pMenge).</p> <p>Übernehme den Eingabewert in eine lokale Variable mMenge.</p> <p>Für den Fall, dass mMenge größer oder gleich 150 ist. Übernehme für den Rabattsatz den Wert 12.</p> <p>Für den Fall, dass mMenge größer oder gleich 100 ist. Übernehme für den Rabattsatz den Wert 10.</p> <p>Für den Fall, dass mMenge größer oder gleich 50 ist. Übernehme für den Rabattsatz den Wert 8.</p> <p>Für den Fall, dass mMenge größer oder gleich 20 ist. Übernehme für den Rabattsatz den Wert 6.</p> <p>Ansonsten übernehme für den Rabattsatz den Wert 0.</p> <p>Gebe den ermittelten Rabattsatz zurück.</p>
<pre>function berechne_rabattbetrag(\$pBetrag, \$pRabattsatz) {     \$mBetrag = \$pBetrag;     \$mRabattsatz = \$pRabattsatz;      \$mRabattbetrag = \$mBetrag/100 * \$pRabattsatz;      return \$mRabattbetrag; }</pre>	<p>Erläuterung: Berechne anhand des Eingabewertes (pBetrag) und des ermittelten Rabattsatzes.</p> <p>Übernehme dazu die Werte in lokale Variable mBetrag und mRabattsatz.</p> <p>Berechne anhand der vorgegebenen Berechnung: rabattbetrag = betrag/100 * rabattsatz;</p> <p>und fange das Ergebnis in der Variablen mRabattbetrag auf.</p> <p>Gebe den berechneten Rabattbetrag zurück.</p>

```
function berechne_zahlungsbetrag($pBetrag, $pRabattbetrag){  
    $mBetrag = $pBetrag;  
    $mRabattbetrag = $pRabattbetrag;  
  
    $mZahlungsbetrag = $mBetrag - $mRabattbetrag;  
  
    return $mZahlungsbetrag;  
}
```

**Erläuterung:**

Berechne anhand des Eingabewertes (pBetrag) und des ermittelten Rabattbetrag.

Übernehme dazu die Werte in lokale Variable mBetrag und mRabattbetrag.

Berechne anhand der vorgegebenen Berechnung:

zahlungsbetrag = betrag - rabattbetrag;

und fange das Ergebnis in der Variablen mZahlungsbetrag auf.

Gebe den berechneten Zahlungsbetrag zurück.