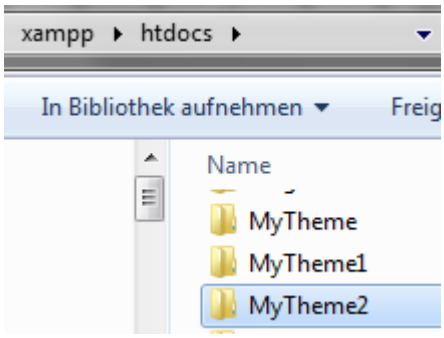
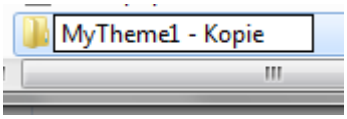
– Anleitung –

1 Formulare mit PHP erstellen und auswerten



Wechseln Sie wie zuvor in das htdocs-Verzeichnis. Kopieren Sie das gerade fertiggestellte Verzeichnis >> MyTheme1<<:



Und geben Sie dem Verzeichnis den Namen >>MyTheme2<<

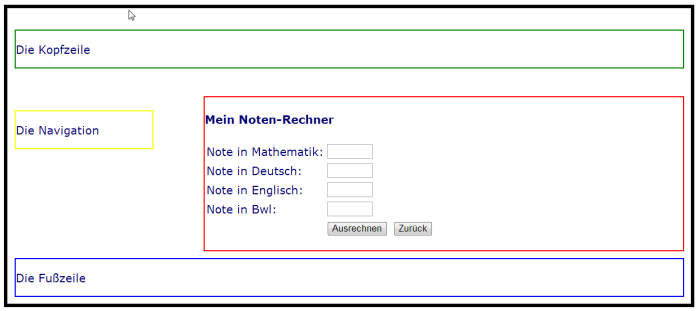
```
<div id="content">
        <form action="note1.php" method="post">
            <h4>Mein Noten-Rechner</h4>
            <table>
              <tr>
                <td>Note in Mathematik:</td>
                <td><input type='text' name='tfMathe' size="5" maxlength="5"  /></td>
              </tr>
              <tr>
                <td>Note in Deutsch:</td>
                <td><input type='text' name='tfDeutsch' size="5" maxlength="5"  /></td>
              </tr>
              <tr>
                <td>Note in Englisch:</td>
                <td><input type='text' name='tfEnglisch' size="5" maxlength="5"  /></td>
              </tr>
              <tr>
                <td>Note in Bwl:</td>
                <td><input type='text' name='tfBwl' size="5" maxlength="5"  /></td>
              </tr>
              <tr>
                <td></td>
                <td><input type="submit" value="Ausrechnen" name='durchschnittAusrechnen'/>
                    <?php echo"<input type='button' value='Zur&uuml;ck' onClick='history.back()' />" ?></td>
              </tr>
            </table>
        </form>
</div>
```

Der Inhaltsbereich >>content.php<< soll nun um ein einfaches Formular erweitert werden.

Öffnen Sie dazu die Datei >>content.php<< mit einem einfachen Editor und ergänzen Sie in der div-Box >>content<< den folgenden Quellcode für das Formular wie nebenstehend angezeigt.

Prüfen Sie ihre Eingaben auf Fehler und speichern Sie das Ergebnis (STRG +S).

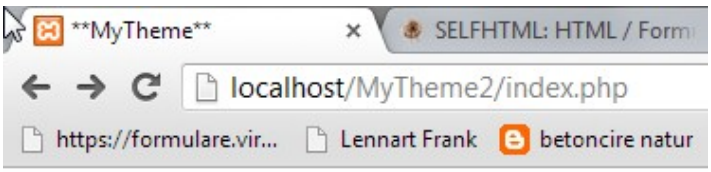
Hinweis:
Formulare bestehen aus Komponenten, wie beispielsweise Texteingabefeldern und Schaltflächen. In SELFHTML finden Sie dazu jede Menge weitere Beispiele (http://de.selfhtml.org/html/formulare/).



Testen Sie nun ob die >>View<< des Formulars korrekt implementiert wurde. Mit >>View<< ist das Darstellung des Fomulars gemeint.

Um die Ereignisse, >>Die Auswertung des Formulars<< kümmern wir uns etwas später.

CC BY NC SA



Wenn der Webserver gestartet ist können wir das Ergebnis im Internetbrowser testen.

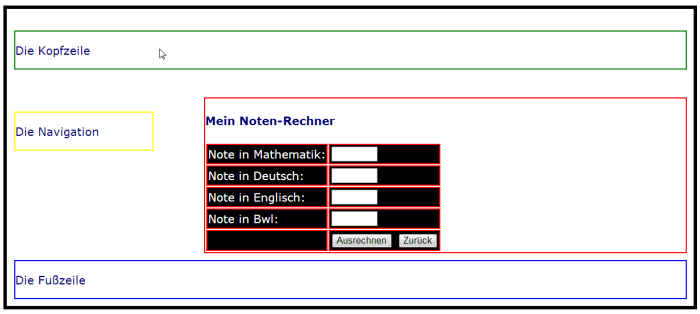
Öffnen Sie dazu den Internetbrowser >>Firefox<< und geben Sie den Pfad zur gerade erstellen Datei an. Geben Sie dazu den in der Grafik angezeigten Pfad ein.

```
#content{
        border:2px solid red;
        float: right;
        width: 70%;
        height:auto;

}

#content form td{
        border:2px solid red;
        background-color: black;
        color: white;
}
```

Wir wollen das Formular hübscher gestalten und müssen dazu das Stylesheet >>style.css<< erweitern.

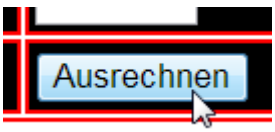
Öffnen Sie dazu mit einem einfachen Editor die Datei >>style.css<<.

Ergänzen Sie das Stylesheet an der Stelle wo die div-Box >>content<< deklariert ist (darunter) wie angezeigt.



Testen Sie wie zuvor erneut die >>View<<.

Das Ergebnis sollte sein, dass die Zellen in der Formular-Tabelle rot umrandet sind, die Hintergrundfarbe schwarz und die Schriftfarbe weiß dargestellt wird.



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

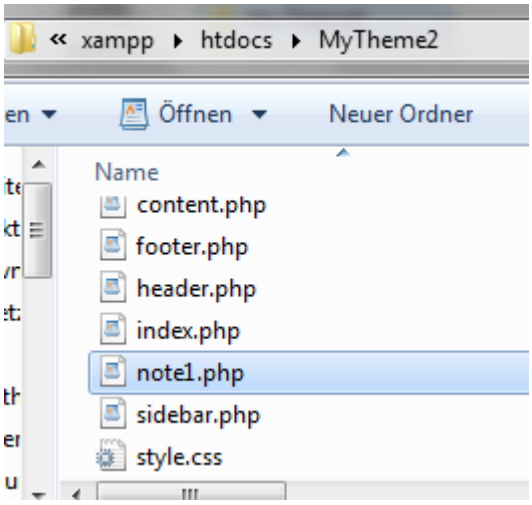
Diese Auswertung kann eine Seitenbeschreibungssprache HTML nicht leisten. Wir nutzen deshalb die Programmiersprache PHP dafür.

CC BY NC SA

Im Moment rufen Wir mit dem Klick auf die Schaltfläche >>Ausrechnen<< eine Datei >>note1.php<< auf (siehe Form-Action in der Datei >>content.php<<):

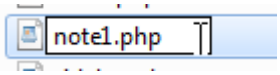
```
<form action="note1.php" method="post">
        <h4>Mein Noten-Rechner</h4>
        <table>
          <tr>
```

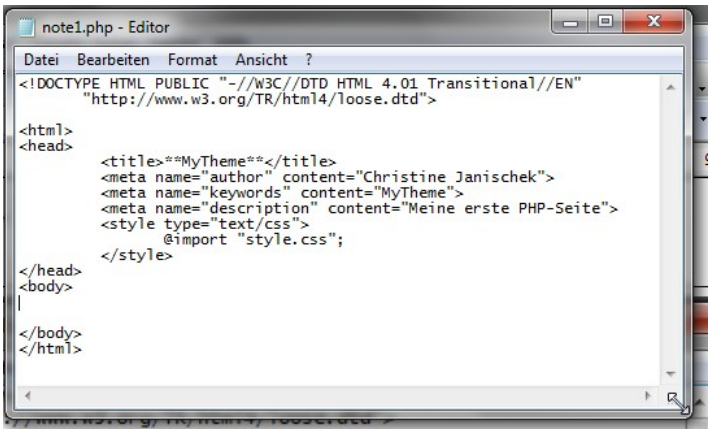
Diese Datei existiert aber noch nicht, deshalb erscheint auch die nebenstehende Fehlermeldung.



Wir werde 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 >>note1.php<<.





Öffnen Sie die Datei >>note1.php<< mit einem einfachen Editor. Die Datei ist wie erwartet noch leer!

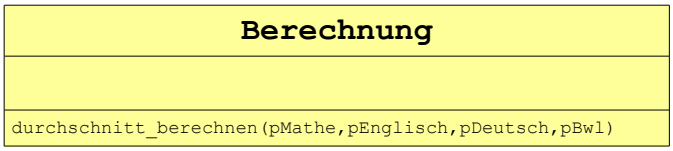
Ergänzen Sie das HTML-Grundgerüst. Kopieren Sie dazu den Inhalt einer bereits erzeugten PHP-Datei (z.B. der Datei >>index.php<<) und entfernen Sie den Quellcode aus dem <body>...</body>.

CC BY NC SA

```
<body>
        <div id="content">
                <p align="center" >
                        <?php
                        //Eingaben lesen
                                $pMathe = $_POST['tfMathe'];
                                $pDeutsch = $_POST['tfDeutsch'];
                                $pEnglisch = $_POST['tfEnglisch'];
                                $pBwl = $_POST['tfBwl'];

                        ?>
                </p>
        </div>
</body>
```

Wir lesen nun im Ersten Schritt die Eingabewerte aus den Texteingabefeldern (tfMathe, tfEnglisch, tfDeutsch, tfBWL) und übernehmen diese Werte in die PHP-Parameterattribute (pMathe, pEnglisch, pDeutsch, pBWL).

Ergänzen Sie dazu im <body>...</body> den angezeigten Quellcode.



UML-Klasse: Berechnung

Egal in welcher Programmiersprache wir heutzutage programmieren lautet das Prinzip >>Objektorientierung<<. Im Großen und Ganzen geht es dabei darum, Ordnung zu halten, d. h. den Quellcode auf eine bestimmte Art und Weise zu strukturieren.

Das Prinzip stellt u. a. sicher, dass wir die Anwendung später problemlos erweitern können und Quellcodebestandteile wiederverwenden können.

Wir halten uns von Anfang an dieses Prinzip und werden es in den nächsten Schritten erstmals anwenden.

```
<body>
        <div id="content">
                <p align="center" >
                        <?php
                        //Eingaben lesen
                                $pMathe = $_POST['tfMathe'];
                                $pDeutsch = $_POST['tfDeutsch'];
                                $pEnglisch = $_POST['tfEnglisch'];
                                $pBwl = $_POST['tfBwl'];

                        //Für die Objektorientierng
                        class Berechnung{

                                public function berechne_durschnitt($pMathe,$pDeutsch,$pEnglisch,$pBwl) {
                                        $pErgebnis = ($pMathe +$pDeutsch + $pEnglisch + $pBwl)/4;
                                        return $pErgebnis;
                                }

                        }
                        ?>
                </p>
        </div>
</body>
```

Erweitern Sie dazu den gerade eingefügten Quellcode um den noch fehlenden Teil.

```
<body>
        <div id="content">
                <p align="center" >
                        <?php
                        //Eingaben lesen
                                $pMathe = $_POST['tfMathe'];
                                $pDeutsch = $_POST['tfDeutsch'];
                                $pEnglisch = $_POST['tfEnglisch'];
                                $pBwl = $_POST['tfBwl'];

                        //Für die Objektorientierng
                        class Berechnung{

                                public function berechne_durschnitt($pMathe,$pDeutsch,$pEnglisch,$pBwl) {
                                        $pErgebnis = ($pMathe +$pDeutsch + $pEnglisch + $pBwl)/4;
                                        return $pErgebnis;
                                }

                        }
                        //Verarbeitung
                        $durchschnitt = new Berechnung();
                        $ausgabe = $durchschnitt -> berechne_durschnitt($pMathe,$pDeutsch,$pEnglisch,$pBwl);
                        ?>
                </p>
        </div>
</body>
```

Wir nutzen nun den für die Strukturierung notwendigen Quellcode.

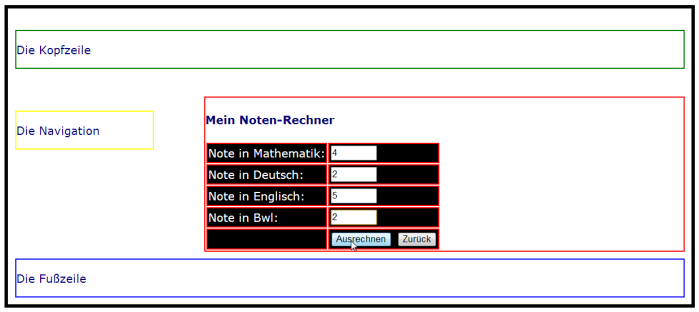
Ergänzen Sie dazu den Quellcode für die Verarbeitung der Formulardaten.

CC BY NC SA

```
<body>
        <div id="content">
                <p align="center" >
                        <?php
                        //Eingaben lesen
                                $pMathe = $_POST['tfMathe'];
                                $pDeutsch = $_POST['tfDeutsch'];
                                $pEnglisch = $_POST['tfEnglisch'];
                                $pBwl = $_POST['tfBwl'];

                        //Für die Objektorientierng
                        class Berechnung{

                                public function berechne_durschnitt($pMathe,$pDeutsch,$pEnglisch,$pBwl) {
                                        $pErgebnis = ($pMathe +$pDeutsch + $pEnglisch + $pBwl)/4;
                                        return $pErgebnis;
                                }

                        }

                        //Verarbeitung
                        $durchschnitt = new Berechnung();
                        $ausgabe = $durchschnitt -> berechne_durschnitt($pMathe,$pDeutsch,$pEnglisch,$pBwl);

                        //Ausgabe
                        echo $ausgabe;
                        ?>
                </p>
        </div>
</body>
```

Abschließend möchten Wir das Ergebnis auch ausgeben.

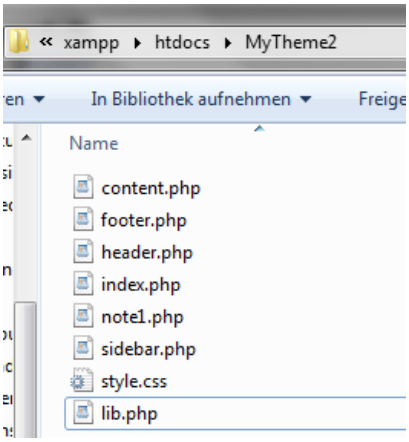
Ergänzen Sie dazu den Quellcode für die Ausgabe des Ergebnisses.



Testen Sie das Formular. Geben Sie Noten in das Formular ein und klicken Sie auf die Schaltfläche >>Ausrechnen<<.

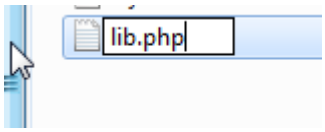
Im Ergebnis sollte die Ausgabe so aussehen:

3.25



Weil wir künftig unsere Seite um weitere Berechnungen (BMI, Umfang, Flächeninhalt,...) erweitern möchten und diese auf unterschiedlichen Unterseiten nutzen möchten, lagern wir den Quellecode der Klasse in eine extra Datei aus.

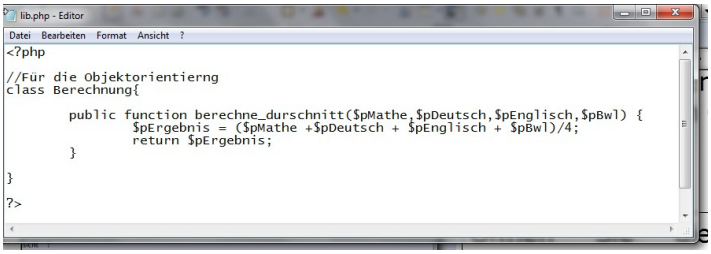
Erzeugen Sie dazu eine neue Datei >>lib.php<<




```
//Für die Objektorientierng
class Berechnung{

        public function berechne_durschnitt($pMathe,$pDeutsch,$pEnglisch,$pBwl) {
                $pErgebnis = ($pMathe +$pDeutsch + $pEnglisch + $pBwl)/4;
                return $pErgebnis;
        }

}
```

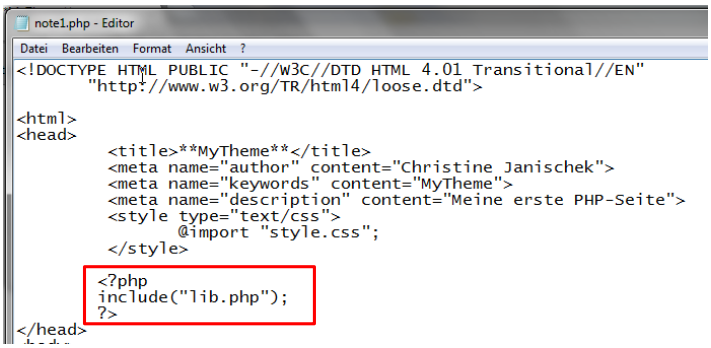
Schneiden Sie den für die Objektorientierung notwendigen Quellcode aus.

CC BY NC SA



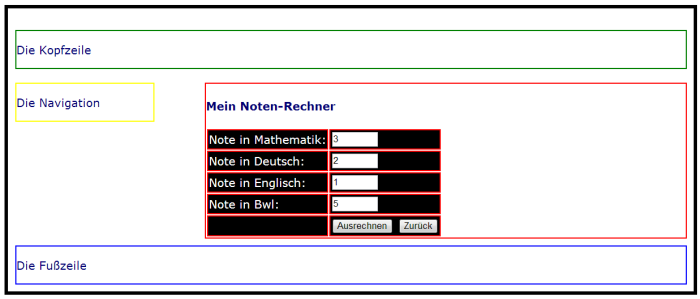
Öffnen Sie die noch leere Datei >>lib.php<< und fügen Sie den ausgeschnittenen Quellcode ein und ummanteln Sie die Klasse mit dem <?php … ?>-Tag wie angezeigt.

Speichern Sie die Änderungen in beiden Dateien (note1.php und lib.php).

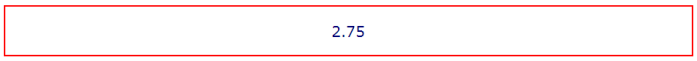


Damit wir die berechnende Methode in der Datei >>note1.php<< trotz Auslagerung nutzen können, müssen wir ähnlich wie beim Stylesheet darauf verweisen.

Ergänzen und speichern Sie dazu im <head>...</head> den notwendigen PHP-include-Befehl.



Testen Sie die Anwendung erneut. An der Funktionalität sollte sich nichts geändert haben.



Mit der vorgenommen Auslagerung können wir nun die Berechnende Methode >>berechne_durchschnitt()<<, immer wieder und an unterschiedlichen Stellen, nutzen.

CC BY NC SA