



# Übungen zu PHP und MySQL

W. Rose  
(Version 3.0)

---

Inhalt

<b>Kapitel 1 - Installation vom Xampp</b>	<b>3</b>
<b>Installation des Servers</b>	<b>3</b>
<b>Konfiguration der Clients</b>	<b>3</b>
<b>Kapitel 2 - Erste Programme</b>	<b>4</b>
<b>Übung - echo</b>	<b>4</b>
<b>Übung - date</b>	<b>4</b>
<b>Kapitel 3 - Kontrollstrukturen</b>	<b>5</b>
<b>Theorie - Variablen</b>	<b>5</b>
<b>Übung – Variablen</b>	<b>5</b>
<b>Übung – If-Abfragen</b>	<b>5</b>
<b>Übung – Schleifen</b>	<b>6</b>
<b>Zusatz – Programmstrukturen</b>	<b>6</b>
<b>Theorie - Arrays</b>	<b>7</b>
<b>Übung – Arrays</b>	<b>7</b>
<b>Zusatz – Arrays</b>	<b>7</b>
<b>Kapitel 4 - Interaktives Arbeiten mit Formularen</b>	<b>8</b>
<b>Übung – Datenübertragung mit POST</b>	<b>8</b>
<b>Theorie – Superglobale Variablen</b>	<b>9</b>
<b>Übung – Datenübertragung mit GET</b>	<b>9</b>
<b>Theorie – Servervariable PHP_SELF</b>	<b>9</b>
<b>Theorie – Funktion include()</b>	<b>9</b>
<b>Übung – Menü mit GET</b>	<b>10</b>
<b>Zusatz – Vom Menü zur Webseite</b>	<b>10</b>
<b>Kapitel 5 - MySQL-Datenbanken</b>	<b>11</b>
<b>Theorie – Datenbanken</b>	<b>11</b>
<b>Übung – Datenbank erstellen</b>	<b>11</b>
<b>Kapitel 6 - Unser eigener Blog</b>	<b>12</b>
<b>Theorie – Blog</b>	<b>12</b>
<b>Übung – Blog</b>	<b>12</b>
<b>Kapitel 7 - Linkliste</b>	<b>14</b>
<b>Übung – Linkliste</b>	<b>14</b>

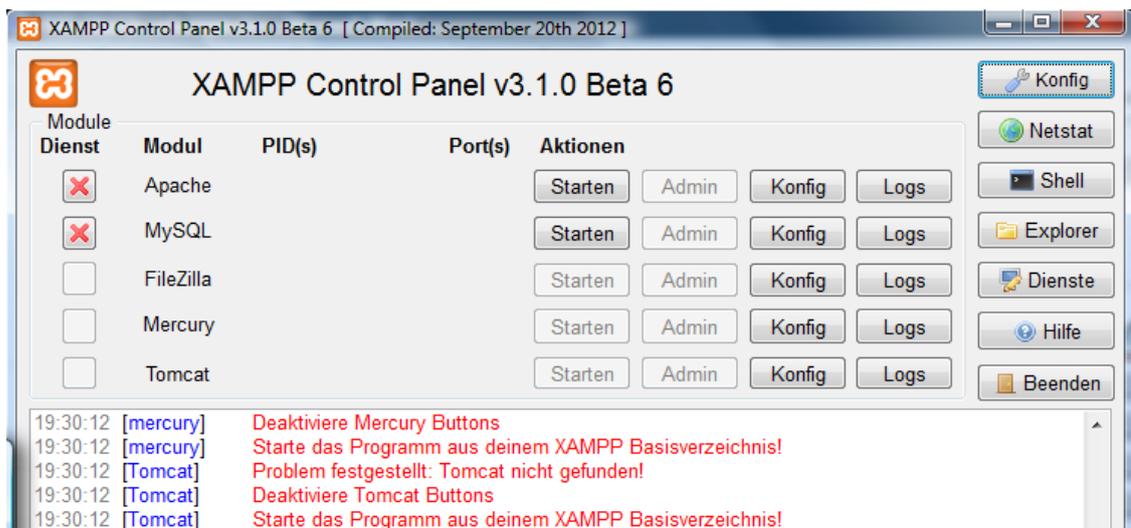
## Kapitel 1 - Installation vom Xampp

### Installation des Intranet-Servers

- ☒ Installiere die Datei *xampp-portable-win32-...-installer.exe* auf deinem **Speicherstick**.

Das Programm XAMPP installiert einen Apache-Webserver mit PHP-Unterstützung und eine MySQL-Datenbank. (Info unter <http://www.apachefriends.org/de/> ). Die Portable-Version kannst du später einfach vom Stick entfernen, wenn du sie nicht mehr gebrauchst. Dazu musst du nur den Ordner löschen.

- ☒ Öffne im Ordner xampp-portable die Datei xampp-control.exe. Es öffnet sich das XAMPP Control Panel. Starte Apache und MySQL.



- ☒ Öffne den Internet Explorer und gib im Adress-Fenster <http://localhost> (oder <http://127.0.0.1>) ein um deinen PC als lokalen Webserver zu nutzen.
- ☒ Wählen die Sprache „Deutsch“. Und klicke anschließend links im Menü auf „Status“. Dort müssen die Komponenten MySQL-Datenbank und PHP aktiviert sein.
- ☒ Öffne jetzt den Windows-Explorer. Gehe auf deinem Stick in das Verzeichnis „xampp-portable/htdocs“ und lege dort einen Ordner „uebungen“ an. In diesem Ordner erstellst du nun eine beliebige HTML-Datei, welche du „index.htm“ nennst.
- ☒ Gib nun in der Adress-Zeile des Internet-Explorers <http://localhost/uebungen/> ein.
- ☒ Du siehst jetzt deine Seite in deinem eigenen Internet (Intranet).

### Konfiguration der Clients

Um auf die Webserver deiner Mitschüler zuzugreifen, musst du ihre Adresse kennen:

- ☒ Um die Adresse deines Computers herauszufinden, gibst du „cmd“ in die Windows-Suche ein. Schreibe in das Fenster „ipconfig“. Du siehst jetzt deine IP-Adresse.
- ☒ Öffnen deinen Browser und gib die IP-Adresse des Computers, auf welchen du zugreifen möchtest, in der Adress-Zeile ein: <http://000.000.000.000/uebungen>
- ☒ Du siehst jetzt die Startseite des Webserver auf dem anderen Computer.

## Kapitel 2 - Erste Programme

Grundsätzlich benötigen auch PHP-Dateien ein vollständiges Grundgerüst incl. Doctype und XHTML. Für eine bessere Übersicht wird hier nur ein Minimalgrundgerüst benutzt.

### Übung - echo

✂ Erstelle folgende Datei mit einem Editor deiner Wahl.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>02_date</title>
  </head>
  <body>
    <p>
      Das heutige Datum:
      <?php
        echo date("D d.M.Y");
      ?>
    </p>
  </body>
</html>
```

✂ Speichere die Datei unter *02\_echo.php* in deinen Ordner „uebungen“. Achte darauf, dass der Dateityp beim Abspeichern „PHP“ bzw. „Alle Dateien (\*.\*)“ ist, sonst wird die Datei als HTML-Dokument gespeichert.

✂ Öffne deinen Browser und gib [http://localhost/uebungen/02\\_echo.php](http://localhost/uebungen/02_echo.php) ein.

Du kannst die Datei nicht direkt im Editor bzw. durch einen Doppelklick öffnen, sondern nur über den Webserver „<http://localhost>“.

### Übung - date

✂ Erstelle folgende kleine Datei, welche das aktuelle Datum ausgibt.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>02_echo</title>
  </head>
  <body>
    <p>Dies ist Text in HTML.</p>
    <?php
      echo "<p>Dies ist Text in PHP.</p>";
    ?>
  </body>
</html>
```

✂ Speichere die Datei unter *02\_date.php*.

✂ Öffne den Internet Explorer und gib [http://localhost/uebungen/02\\_date.php](http://localhost/uebungen/02_date.php) ein.

✂ Schau dir die beiden Programme näher an, was kannst du über die Einbindung von PHP aussagen:

---

---

## Kapitel 3 - Kontrollstrukturen

Die Übungen gehen in diesem Kapitel davon aus, dass schon über einige Vorkenntnisse über Programmiersprachen vorhanden sind. Die Grundstrukturen Variablen, Schleifen und IF-Abfragen werden daher nur kurz behandelt.

### Theorie - Variablen

Variablen müssen in PHP nicht extra deklariert werden. Auch eine explizite Typzuweisung fällt weg.

Variablen werden in PHP lediglich mit einem vorgestellten \$-Zeichen gekennzeichnet und können dann direkt verwendet werden.

### Übung – Variablen

✂ Erstelle folgende Datei mit drei Variablen. (*03\_variablen.php*)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>03_variablen</title>
  </head>
  <body>
    <?php
      $alter=36;
      $name="Klaus";
      $groesse=1.79;

      echo "$name ist $alter Jahre alt und $groesse m groß";

    ?>
  </body>
</html>
```

### Übung – If-Abfragen

If-Abfragen bieten in PHP neben dem optionalen else auch noch unbegrenzt viele optionale elseif. Weiter wird die Gleichheit eines Wertes durch ein == überprüft. Ein einzelnes = wäre eine Wertzuweisung.

✂ Erstelle folgende Datei (*03\_if.php*).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>03_if</title>
  </head>
  <body>
    <?php
      $gebdat=1971;
      $alter=date("Y")-$gebdat;
      if ($alter<20)
      {
        echo "Du bist Schüler";
      }
      else
      {
        echo "Du bist Lehrer";
      }

    ?>
  </body>
</html>
```

- ✗ Erstelle diese Datei mit einer verschachtelten IF-Abfrage (*03\_if-2.php*).

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>03_if</title>
  </head>
  <body>
    <p>
      <?php
        $a=100;
        $b=90;
        if ($a<$b)
        {
            echo "$a ist kleiner als $b";
        }
        elseif ($a==$b)
        {
            echo "$a ist gleich $b";
        }
        else
        {
            echo "$a ist gr als $b";
        }
      ?>
    </p>
  </body>
</html>

```

- ✗ Ändere die Variablen und gib auch mal Zeichenketten ein.

## Übung – Schleifen

- ✗ Erstelle folgendes Beispiel für eine for-Schleife. (*03\_for.php*)

```

<?php
for($i=0; $i<=10; $i++)
{
    echo "i: $i <br>";
}
?>

```

- ✗ Erstelle folgendes Beispiel für eine while-Schleife. (*03\_while.php*)

```

<?php
$i=32;
while ($i<127)
{
    $Zeichen = chr($i);
    echo "$Zeichen |";
    $i++;
}
?>

```

## Zusatz – Programmstrukturen

- ✗ Erstelle eine Datei, die die 6er-Reihe ausgibt.
- ✗ Erstelle eine Datei, die folgendes Ergebnis anzeigt. (*03\_Treppe.php*)

```

XX
XXX
XXXX
XXXXX
XXXXXX

```

Stringverknüpfung in PHP mit „.“  
Bsp: „Auto“ „Bahn“ → „Autobahn“

## Theorie - Arrays

Eine wichtige Datenstruktur von PHP ist das Array. Ein Array ist ein Feld bzw. eine Sammlung von Daten. Ein Array Schulfächer könnte beispielsweise die Elemente Englisch, Deutsch, Mathematik usw. haben.

Alle Programmiersprachen kennen Arrays in mind. zwei verschiedenen Arten:

- Numerische Arrays: Hier werden die Elemente mit Ganzzahlen indiziert, beginnend bei 0.
- Assoziative Arrays (Hash): hier wird jedem Element ein eigener Key zugewiesen.

## Übung – Arrays

✂ Erstelle folgende Datei mit einem numerischen Array. (*03\_numarray.php*)

```
<?php
//Flllen des Arrays
$schule=array("Deutsch","Mathematik","Englisch","Informatik","Musik");
//Hinzufügen eines Wertes
$schule[5]="Sport";
//Ändern eines Wertes
$schule[4]="Kunst";
//Ausgabe aller Werte
$i=0;
while ($i<count($schule))
{
    echo "$i: $schule[$i] <br>";
    $i++;
}
//Ausgabe eines Wertes
echo "Wert Nummer 3 ist $schule[3]";
?>
```

✂ Erstelle folgende Datei mit einem assoziativen Array, welcher dazu noch verschiedene Datentypen enthält. (*03\_assarray.php*)

```
<?php
//Flllen des Arrays
$noten=array("Deutsch"=>2,"Mathematik"=>4,"Religion"=>"abgemeldet");
//Hinzufügen eines Wertes
$noten["Englisch"]=4;
//Ausgabe aller Werte
foreach ($noten as $key=>$elem)
{
    echo "$key: $elem <br>";
}
//Ausgabe eines Wertes
echo "<br><br>Religionsnote: $noten[Religion]";
?>
```

## Zusatz – Arrays

- ✂ Erstelle ein Array, welches 10 die Namen von guten Liedern ausgeben kann.
- ✂ Erstelle ein Array, welches von 5 Personen den Namen und das Alter ausgeben kann.
- ✂ \*\*\* Erstelle ein Array, welches 5 Farbnamen enthält und bei der Ausgabe die Wörter in der richtigen Farbe ausgibt.

## Kapitel 4 - Interaktives Arbeiten mit Formularen

Grundlage für die Programmierung von interaktiven Webseiten sind HTML-Formulare.

### Übung – Datenübertragung mit POST

- ✂ Erstelle einen neuen Ordner *post*.
- ✂ Erstelle darin die beiden Dateien *eingabe.html*

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Eingabe</title>
  </head>
  <body>
    <form action="ausgabe.php" method="post">
      Name:      <input type="text" name="name"><br>
      Nachricht: <input type="text" name="nachricht"><br>
      Dringend: Ja<input type="radio" name="dringend" value="y"
checked="checked">
                Nein<input type="radio" name="dringend" value="n"><br>
      <input type="hidden" name="versteckt" value="geheim">
      <input type="submit" value="Abschicken">
    </form>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Ausgabe</title>
  </head>
  <body>
    <p>
      <a href="eingabe.html">Zur Eingabe</a><br>
      <?php
        $name=$_POST['name'];
        $nachricht=$_POST['nachricht'];
        $dringend=$_POST['dringend'];
        $versteckt=$_POST['versteckt'];
        echo "Ausgabe der Variablen: $name, $nachricht, $dringend,
$versteckt";
      ?>
    </p>
  </body>
</html>
```

- ✂ Ergänze die Dateien mit anderen Formulareingaben.

## Theorie – Superglobale Variablen

In PHP stehen vordefinierter Arrays zur Verfügung, die Variablen vom Webserver (gegebenenfalls), von der Umgebung und von Benutzereingaben enthalten. Diese Variablen sind automatisch global.

Drei superglobale Variablen sind für uns momentan interessant:

- `$_GET` Variablen, die meist mit Links an ein anderes Skript übertragen werden.
- `$_POST` Variablen, die dem Skript über HTTP POST (Formular) geliefert werden.
- `$_SERVER` Variablen, die vom Webserver gesetzt werden.

## Übung – Datenübertragung mit GET

Erstelle folgende Datei (04\_get.php)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Datenübertragung mit Get</title>
  </head>
  <body>
    <p>
      <a href="04_get.php?variable=Hallo">Datenübertragung</a>
      <?php
        @$variable=$_GET['variable'];
        if ($variable)
        {
          echo "<p>Die Variable ist $variable</p>";
        }
      ?>
    </p>
  </body>
</html>
```

@: Fehler-Kontroll-Operator

## Theorie – Servervariable PHP\_SELF

Die Servervariable PHP-SELF enthält den Namen des aktuellen ausgeführten Scripts. Die Variable wird dann genutzt, wenn eine Seite sich selbst wieder aufrufen soll, z.B. per Link.

## Theorie – Funktion include()

Mit der Anweisung `include()` wird eine andere Datei in ein Script eingebunden und ausgeführt.

Im Gegensatz zu Homepages mit reinen HTML, in welchen z.B. bei jedem Menüpunkt eine neue HTML-Seite geöffnet wird, ruft sich bei vielen PHP-Portalen die index-Seite immer wieder selbst auf und die unterschiedlichen Inhaltsseiten werden dann mit `include()` eingebunden.

Natürlich kann bei `include()` auch eine Pfadangabe eingegeben werden.

## Übung – Menü mit GET

- ✂ Erstelle einen neuen Ordner *get*.
- ✂ Erstelle darin die Datei *menue.php*

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>Get-Menü</title>
    <?php
      $PHP_SELF=$_SERVER['PHP_SELF'];
      @$menue=$_GET['menue']; // @ Behebt den Fehler beim ersten Aufruf
      if (!$menue)
      {
        $menue="eins";
      }
    ?>
  </head>
  <body>
    <!--Menue-->
    <p>
      <a href='<?php echo "$PHP_SELF?menue=eins '>;
        if ($menue=='eins')echo "<b>Home</b>"; else echo "Home";?></a>
      <a href='<?php echo "$PHP_SELF?menue=zwei '>;
        if ($menue=='zwei')echo "<b>Über mich</b>"; else echo "Über
mich";?></a>
    </p>
    <hr>
    <!--Inhalt-->
    <p>
      <?php
        if ($menue=='eins')include("eins.php");
        if ($menue=='zwei')include("zwei.php");
      ?>
    </p>
  </body>
</html>
```

- ✂ Du brauchst noch zwei Dateien *eins.php* und *zwei.php*. Diese enthalten lediglich einen kurzen Text ohne weiteres HTML.

- ✂ *eins.php*:

```
Dies ist meine Webseite.
```

- ✂ *Zwei.php*

```
Hier ist der Menüpunkt 2.
```

## Zusatz – Vom Menü zur Webseite

- ✂ Ergänze das Menü um weitere Seiten.
- ✂ \*\*\* Erweitere die Seite mit CSS.

# Kapitel 5 - MySQL-Datenbanken

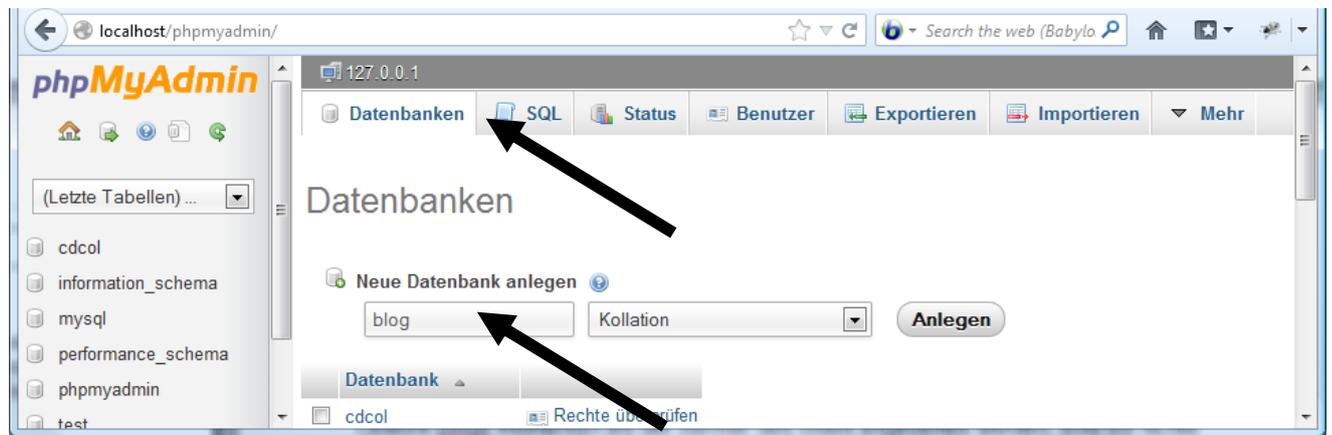
## Theorie – Datenbanken

Datenbanken dienen in Webanwendungen dazu, dynamische Inhalte anbieten können. Damit ist die Seite nicht mehr statisch, sondern kann den Bedürfnissen des Anwenders individuell angepasst werden. So werden in einem Webshop beispielsweise nur die Artikel angezeigt, die für den Benutzer interessant sind.

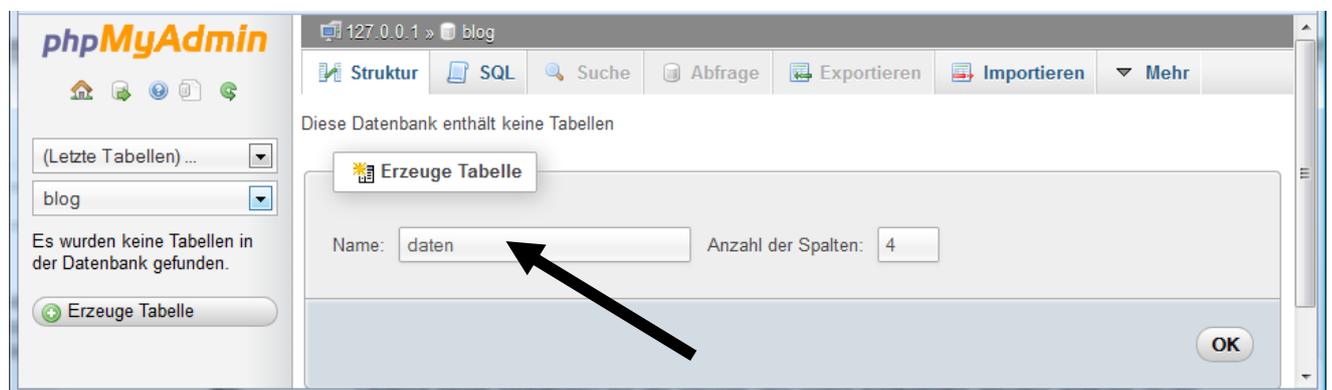
Datenbanken besitzen eine eigene Sprache SQL. Das Datenbank-Tool phpMyAdmin kann uns helfen, diese Sprache zu verstehen und mit Datenbanken umzugehen.

## Übung – Datenbank erstellen

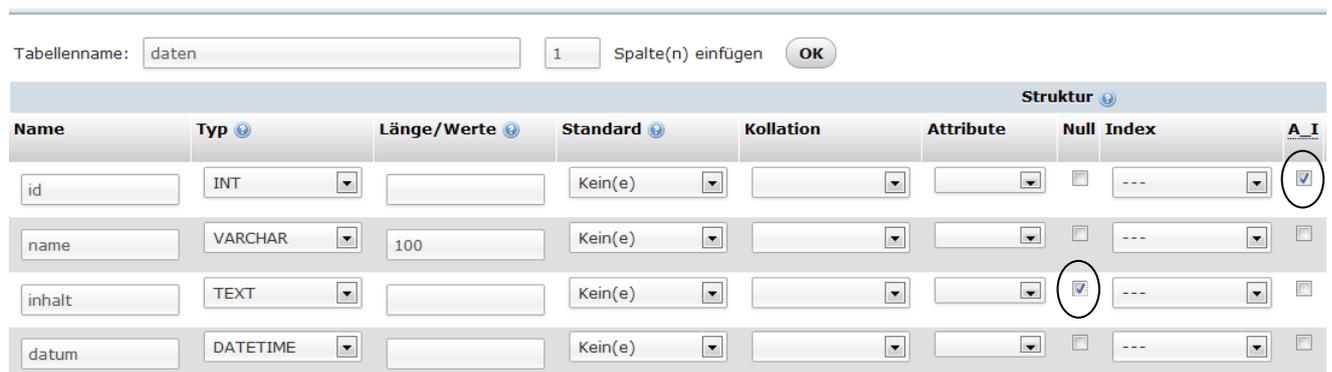
✂ Lege in phpMyAdmin eine neue Datenbank „blog“ an.



✂ Wähle links die Datenbank „blog“ und erstelle eine neue Tabelle mit 4 Feldern.



✂ Erstelle folgende Felder: (Achte auch auf die Haken!)



✂ Fertig ist die Datenbank. Jetzt müssen wir sie mit Inhalt füllen und wieder auslesen.

## Kapitel 6 - Unser eigener Blog

### Theorie – Blog

Ein Blog oder auch Weblog (Kombination aus „Web“ und „Log“) ist eine Webseite, die periodisch neue Einträge enthält, die chronologisch absteigend sortiert sind. Ein Blog beginnt also immer mit dem neuesten Eintrag.

### Übung – Blog

- ✂ Erstelle einen neuen Ordner `06_blog`.
- ✂ Erstelle darin die drei Dateien

#### *eingabe.html*

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <head>
    <title>Mein Blog</title>
  </head>
  <body>
    <form action="verarbeitung.php" method="post">
      <p>
        Name: <input type="text" name="name"><br>
        Nachricht: <input type="text" name="text"><br>
        <input type="submit" value="Abschicken">
      </p>
    </form>
  </body>
</html>
```

#### *verarbeitung.php*

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>Mein Blog</title>
  </head>
  <body>
    <p><a href="ausgabe.php">Zur Ausgabe</a></p>
    <?php
      $name=$_POST['name'];
      $text=$_POST['text'];

      $connect=mysql_connect("localhost","root","");
      mysql_select_db("blog");
      $res=mysql_query("INSERT INTO `daten` (`name`,`inhalt`,`datum`) VALUES
('{$name}', '{$text}', NOW( ));");
      if (! $res)
      {
        echo "Fehler beim Schreiben in der Datenbank";
      }
      mysql_close($connect);
    ?>
  </body>
</html>
```

*ausgabe.php*

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>Mein Blog</title>
  </head>
  <body>
    <p><a href="eingabe.html">Zur Eingabe</a></p>
    <hr>

    <?php
      $connect=mysql_connect("localhost","root","");
      mysql_select_db("blog");
      $res=mysql_query("select * from daten ORDER BY `id` DESC ");

      $num=mysql_num_rows($res);
      for ($i=0;$i<$num;$i++)
      {
        $nn=mysql_result($res,$i,"name");
        $nt=mysql_result($res,$i,"inhalt");
        $nd=mysql_result($res,$i,"datum");
        echo "<p>$nn schieb am $nd: <br>$nt<hr></p>";
      }
      mysql_close($connect);
    ?>
  </body>
</html>
```

- ☞ Du kannst deinen Blog natürlich noch graphisch verbessern und mit mehr Funktionen ausstatten

## Kapitel 7 - Linkliste

### Übung – Linkliste

Der Seitenaufbau ist ähnlich dem Blog. Du solltest die Seiten kopieren und dann anpassen.

- ✂ Erstelle einen neuen Ordner 07\_linkliste.
- ✂ Füge zur Datenbank blog eine neue Tabelle „links“ hinzu. Die Tabelle enthält die Felder
  - id (INT, AI)
  - adresse (VARCHAR, 255)
  - beschreibung (TEXT)
  - bewertung (VARCHAR, 6)
- ✂ Erstelle folgende Seite „eingabe.html“

Adresse:   
Beschreibung:   
Bewertung: \*  
Abschicke

```
Bewertung: <select name="bewertung">  
  <option value="*">*</option>  
  <option value="**">**</option>  
  <option value="***">***</option>  
  <option value="****">****</option>  
  <option value="*****">*****</option>  
</select>
```

- ✂ In der Datei „Verarbeitung.php“ musst du die Felder anpassen.
- ✂ Die Seite „ausgabe.html“ soll so aussehen. Auch hier musst du die Felder anpassen:

[Zur Eingabe](#)

Adresse	Beschreibung	Bewertung
<a href="http://www.google.de">www.google.de</a>	Suchmaschine	****