

PHP-Tutorial - Teil 24

MySQL und PHP:

Sie kennen jetzt die grundlegenden (My)SQL-Befehle und können Tabellen anlegen, Zeilen einfügen oder löschen und Abfragen vornehmen. Um diese Befehle in einem PHP-Script zu verwenden, gibt es einige spezielle MySQL-Funktionen in PHP.

Folgende Funktionen brauchen Sie beim Umgang mit MySQL und PHP:

mysql_connect:

mysql_connect öffnet eine Verbindung zum Datenbank-Server. Der Funktion können Sie insgesamt drei Argumente übergeben:

```
int mysql_connect ([hostname [,benutzername [,passwort]])
```

Das erste Argument ist der Hostname (beim Offline-Testen "localhost"), das Zweite und Dritte der Benutzername und das Passwort (beide beim Offline-Testen leer).

Wenn Sie Ihre Scripte später auch online stellen möchten, empfiehlt es sich diese Angaben als globale Variablen in einer (Config-)Datei festzuhalten, damit Sie die Scripte schnell "online-tauglich" machen können.

Die Angaben erhalten Sie von Ihrem Webespace-Anbieter.

Die Funktion gibt im Erfolgsfall (Verbindung hergestellt) eine Verbindungskennung zurück, die Sie in einer Variable speichern sollten.

mysql_select_db:

Mit mysql_select_db wählen Sie eine Datenbank aus. Die Funktion muss den Namen der Datenbank und kann eine Verbindungskennung übergeben bekommen.

```
int mysql_select_db (string Datenbankname [, int Verbindungskennung])
```

Die Funktion gibt im Erfolgsfall (Datenbank ausgewählt) true, sonst false zurück.

mysql_query:

mysql_query nimmt eine Anfrage bzw. einen Befehl entgegen, die Sie z.B. vorher in einer Variablen gespeichert haben. Zusätzlich kann noch die Verbindungskennung übergeben werden.

```
int mysql_query (string Anfrage [, int Verbindungskennung])
```

War die Anfrage erfolgreich wird true zurückgegeben. Andernfalls wird false zurückgegeben (z.B. bei syntaktisch falschem Befehl). Haben Sie einen Select-Anfrage übergeben, wird eine Ergebniskennung zurückgegeben, die Sie in einer Variablen speichern sollten. Die Anfrage sollte nicht mit einen Semikolon enden.

mysql_affected_rows:

Haben Sie mysql_query eine Delete-, Insert- oder Update-Anfrage übergeben, können Sie mit dieser Funktion herausfinden, wie viele Datensätze verändert bzw. betroffen sind. Zusätzlich können Sie die Verbindungskennung übergeben. Übergeben Sie keine Kennung wird die aktuell offene Verbindung benutzt.

```
int mysql_affected_rows ([int Verbindungskennung])
```

mysql_num_rows:

Wenn mysql_query eine Select-Anfrage übergeben haben, können Sie mit mysql_num_rows herausfinden, wie viele Datensätze im Ergebnis enthalten sind. Sie müssen die Ergebniskennung der vorherigen mysql_query-Funktion übergeben.

```
int mysql_num_rows (int Ergebniskennung)
```

mysql_fetch_object:

Mit der Funktion `mysql_fetch_object` kommen Sie an die eigentlichen Inhalte des Ergebnisses einer Select-Anfrage. Die Funktion benötigt die Ergebniskennung der `mysql_query`-Funktion und kann noch ein zusätzliches Argument (den Ergebnistyp) übergeben bekommen. Den Ergebnistyp benötigen Sie nicht, deshalb werde ich nicht näher darauf eingehen.

```
object mysql_fetch_object (int Ergebniskennung [, int Ergebnistyp])
```

Die Funktion gibt ein Object zurück. Wie Sie an die Datensätze in diesem Objekt kommen, wird in der nächsten Lektion erklärt.

mysql_close:

Mit `mysql_close` können Sie die Verbindung zum Datenbank-Server beenden. Dies ist besonders sinnvoll bei größeren Scripten, sobald die Verbindung nicht mehr benötigt wird. Die Verbindung wird nach Beendigung des Scriptes immer geschlossen. Optional kann man die Verbindungskennung übergeben, die geschlossen werden soll.

```
int mysql_close ([int Verbindungskennung])
```

PHP-Tutorial - Teil 25

MySQL und PHP - Das Gästebuchbeispiel:

Nachdem sich die letzten Lektionen mehr mit dem Erlernen von SQL und den dazugehörigen PHP-Funktionen beschäftigt haben, beginnen wir in dieser Lektion mit dem Gästebuch-Beispiel.

Tabelle anlegen:

Als erstes benötigen Sie die Tabelle 'gaestebuch'. Falls Sie sie gelöscht haben, hier der SQL-Befehl, mit dem Sie die Tabelle wieder erstellen können. Diesen Befehl können Sie in das SQL-Fenster von phpMyAdmin eingeben oder im MySQL-Monitor (Sie finden diese Befehl auch als SQL-Datei im Beispiel zu dieser Lektion).

```
CREATE TABLE daten.gaestebuch (  
  datum datetime,  
  name varchar(50),  
  titel varchar(40),  
  eintrag text  
);
```

Formular zum Einfügen eines Gästebucheintrages erstellen:

Dieser Teil hat nichts mit PHP zu tun, vielmehr sollen Sie ein Formular erstellen, mit dem der Benutzer dem Gästebuch einen Eintrag hinzufügen kann. Das Formular könnte so aussehen (Sie finden das Formular auch im Beispiel zu dieser Lektion):

```
<html>  
<head>  
<title>Mein Gästebuch</title>  
</head>  
<body>  
<form action="einfuegen.php3" method="post">  
<table border="1">  
<tr>  
<td><b>Name:</b></td>  
<td><input type="text" name="name" maxlength="50"></td>  
</tr>  
<tr>  
<td><b>Titel des Eintrages:</b></td>  
<td><input type="text" name="titel" maxlength="40"></td>  
</tr>  
<tr>  
<td><b>Eintrag:</b></td>  
<td><textarea name="eintrag" cols="30" rows="5"></textarea></td>  
</tr>  
<tr>  
<td align="center" colspan="2">  
<input type="submit" value="Eintragen">  
<input type="reset" value="Löschen">  
</td>  
</tr>  
</table>  
</form>  
</body>  
</html>
```

Speichern Sie das Formular unter dem Dateinamen "gaestebuch.php3".

So sieht das obige Formular im Browser aus:

Name:	<input type="text"/>
Titel des Eintrages:	<input type="text"/>
Eintrag:	<input type="text"/>
<input type="button" value="Eintragen"/> <input type="button" value="Löschen"/>	

Verbinden mit der Datenbank:

Legen Sie eine neue Datei mit dem Namen "einfuegen.php3" an. In diesem Script soll das Einfügen eines neuen Eintrags erfolgen (deshalb verweist obiges Script auf diese Datei).

Als erstes müssen Sie eine Verbindung mit der Datenbank öffnen. Dafür ist die Funktion 'mysql_connect' zuständig. Öffnen Sie Ihren Texteditor und schreiben Sie folgenden Zeilen:

```
<html>
<head>
<title>Eintrag hinzuf&uuml;gen</title>
</head>
<body>
<?php

$db=mysql_connect("localhost","","");
mysql_select_db("daten", $db);

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

Wenn Sie beim Ausführen des Scriptes keine Fehlermeldung, sondern nur einen weißen Bildschirm sehen, ist die Verbindung zur Datenbank erfolgreich hergestellt worden. Die Funktion 'mysql_select_db' wählen Sie Ihre Datenbank "daten" aus. Erst nach Auswählen einer Datenbank können Sie die dort befindlichen Tabellen ansprechen.

SQL in PHP - Daten einfügen:

Erweitern Sie die Datei 'einfuegen.php3' um folgende Zeilen:

```
<html>
<head>
<title>Eintrag hinzuf&uuml;gen</title>
</head>
<body>
<?php

$db=mysql_connect("localhost","","");
mysql_select_db("daten", $db);
mysql_query("
    INSERT INTO gaestebuch
    (datum, name, titel, eintrag)
    VALUES
    (NOW(''), '$name', '$titel', '$eintrag')
    ", $db);
```

```

if(mysql_affected_rows($db)==1)
    echo "Vielen Dank f&uuml;r Ihren Eintrag!<br>",
        "<a href=\"gaestebuch.php3\">Zur&uuml;ck</a>";
else
    echo "Beim Eintragen ist ein Fehler aufgetreten.<br>",
        "<a href=\"javascript:history.back();\">Zur&uuml;ck</a>";

?>
</body>
</html>

```

Nach Herstellen der Verbindung mit der Datenbank können Sie mit der Funktion 'mysql_query' Befehle in den Tabellen ausführen, in diesem Fall ein INSERT-Befehl zum Einfügen eines neuen Gästebucheintrages. Bis zu dem Wort "VALUES" kennen Sie den Befehl bereits; da die einzufügenden Werte bei jedem Gästebuchbenutzer anders sind, stehen in der Klammer für die Werte Variablen. Diese Variablen kommen aus dem vorher ausgefüllten Formular, deshalb müssen die Namen der Eingabefelder im Formular und die Variablennamen völlig identisch sein. Um dem Eintrag immer ein aktuelles Datum hinzuzufügen, gibt es ein spezielles MySQL-Attribut "NOW()". Im Gegensatz zu den anderen Werten steht dieses nicht in Hochkommata.

Um dem Besucher anzuzeigen, ob der Eintrag gelungen ist folgt noch eine kurze if-Bedingung. Die in diesem Zusammenhang verwendete Funktion 'mysql_affected_rows' ermittelt die Anzahl der betroffenen Zeilen beim Einfügen von Werten in die Tabelle (dieser Wert sollte natürlich immer 1 sein!).

PHP-Tutorial - Teil 26

Gästebucheinträge anzeigen:

Besucher können zwar Einträge zum Gästebuch hinzufügen, nur erscheinen die Einträge nirgendwo. Dazu erweitern wir die Datei "gaestebuch.php3" um folgende Zeilen:

```
<html>
<head>
<title>Mein Gästebuch</title>
</head>
<body>
<h2>Mein Gästebuch</h2>
<a href="#neuereintrag">Eintrag hinzufügen</a><br><br>
<?php

$db=mysql_connect("localhost","","");
mysql_select_db("daten", $db);
$result=mysql_query("SELECT * FROM gaestebuch ORDER BY datum DESC LIMIT
30", $db);
for($i=0; $i<mysql_num_rows($result); $i++)
{
    $gaestebuch[$i]=mysql_fetch_object($result);
}
echo "<table border=\"0\">";
for($i=0; $i<mysql_num_rows($result); $i++)
{
    echo "<tr>",
        "<td><b>#",
        $i+1,
        ": ",
        $gaestebuch[$i]->titel,
        "</b></td>",
        "</tr>",
        "<tr>",
        "<td>",
        nl2br($gaestebuch[$i]->eintrag),
        "</td>",
        "</tr>",
        "<tr>",
        "<td>Eintrag von <b>",
        $gaestebuch[$i]->name,
        "</b> am <b>",
        substr($gaestebuch[$i]->datum, 8, 2),
        ".",
        substr($gaestebuch[$i]->datum, 5, 2),
        ".",
        substr($gaestebuch[$i]->datum, 0, 4),
        " ",
        substr($gaestebuch[$i]->datum, 10),
        "</b></td>",
        "</tr>",
        "<tr><td><hr></td></tr>";
}
echo "</table><br><br>";

?>
<a name="neuereintrag"><h3>Neuer Eintrag:</h3></a>
<form action="einfuegen.php3" method="post">
<table border="1">
```

```

<tr>
  <td><b>Name:</b></td>
  <td><input type="text" name="name" maxlength="50"></td>
</tr>
<tr>
  <td><b>Titel des Eintrages:</b></td>
  <td><input type="text" name="titel" maxlength="40"></td>
</tr>
<tr>
  <td><b>Eintrag:</b></td>
  <td><textarea name="eintrag" cols="30" rows="5"></textarea></td>
</tr>
<tr>
  <td align="center" colspan="2">
    <input type="submit" value="Eintragen">
    <input type="reset" value="L&ouml;schen">
  </td>
</tr>
</table>
</form>
</body>
</html>

```

Erläuterungen:

Zuerst wird eine Verbindung mit der Datenbank hergestellt (`mysql_connect`) und die Datenbank "daten" ausgewählt (`mysql_select_db`)

Zum Auslesen der Einträge wird der SQL-Befehl

```
SELECT * FROM gaestebuch ORDER BY datum DESC LIMIT 30
```

verwendet. Es werden alle Spalten (*) aus der Tabelle "gaestebuch" ausgelesen und nach dem Datum aufsteigend (`ORDER BY datum DESC`) sortiert. Davon werden die ersten 30 (und somit neuesten) Einträge ausgewählt.

In der folgenden Schleife werden die Ergebnisse in den Array `$gaestebuch` eingelesen. Dabei läuft die Schleife solange wie Einträge gefunden wurden (also maximal 30 mal).

Die nächste Schleife läuft ebenfalls solange wie Einträge gefunden wurden und gibt dabei vor dem Titel (`$gaestebuch[$i]->titel`), die Nummer des Eintrages aus (`$i+1`).

Danach wird der eigentliche Eintragstext ausgegeben (`n12br($gaestebuch[$i]->eintrag)`). Die Funktion `n12br` wandelt dabei neue Zeilen im Eintragstext in `
`-Tags um.

Unter dem Eintragstext steht noch der Name des Eintragerstellers und die Uhrzeit, wann der Eintrag erstellt wurde. Die Uhrzeit wird dabei ein wenig verändert, damit Sie im deutschen Zeitformat angezeigt wird.

Wenn Sie diese Datei gespeichert haben, sehen Sie folgende Ausgabe im Browser (vorher müssen Sie natürlich einen oder mehrere Einträge hinzugefügt haben):

Mein Gästebuch

[Eintrag hinzufügen](#)

#1: Der erste Gästebucheintrag

Dies ist der erste Gästebucheintrag.

Dieser Eintrag rutscht immer weiter nach unten je mehr neue Einträge hinzukommen.

Eintrag von **Christian Felken** am **06.01.2002 11:46:20**

Neuer Eintrag:

Name:	<input type="text"/>
Titel des Eintrages:	<input type="text"/>
Eintrag:	<input type="text"/>
<input type="button" value="Eintragen"/> <input type="button" value="Löschen"/>	

Das Gästebuch ist jetzt praktisch fertig. Sie können die Formatierungen natürlich beliebig anpassen.

In der nächsten Lektion wird ein kleines Administrations-Script zum Verwalten der Einträge erstellt.


```

        "</tr>",
        "<tr>",
        "<td>",
        nl2br($gaestebuch[$i]->eintrag),
        "</td>",
        "</tr>",
        "<tr>",
        "<td>Eintrag von <b>",
        $gaestebuch[$i]->name,
        "</b> am <b>",
        substr($gaestebuch[$i]->datum, 8, 2),
        ".",
        substr($gaestebuch[$i]->datum, 5, 2),
        ".",
        substr($gaestebuch[$i]->datum, 0, 4),
        " ",
        substr($gaestebuch[$i]->datum, 10),
        "</b></td>",
        "</tr>",
        "<tr><td><hr></td></tr>";
    }
    echo "</table><br><br>";
}
else
{
    echo "Login:<br>",
        "<form action=\"admin.php3\" method=\"post\">",
        "User: <input type=\"text\" name=\"user\"><br>",
        "Passwort: <input type=\"password\" name=\"password\"><br>",
        "<input type=\"submit\" name=\"login\" value=\"Login\">";
}
?>
</body>
</html>

```

Erläuterungen:

Die erste if-Abfrage überprüft, ob die eingegebenen Login-Daten (User und Passwort) korrekt sind. Wenn nicht, wird der else-Teil ausgeführt: das Login-Formular.

Bei richtigem User und Passwort wird der Inhalt des Gästebuch angezeigt, genauso wie es der Gästebuchbesucher sieht. Einzige Ausnahme: ein kleiner Link neben jedem Titel des Eintrages zum Löschen des Eintrages.

Nach dem Klick auf den Löschen-Link wird die if-Abfrage 'if(\$delete=="ja")' ausgeführt. Es wird versucht den Eintrag zu löschen und bei Erfolg wird eine kurze Meldung ('Eintrag wurde gelöscht!') angezeigt. Danach werden wieder die restlichen Einträge angezeigt.

Wenn Sie diese Datei gespeichert haben, sehen Sie folgende Ausgabe im Browser (vorher müssen Sie natürlich einen oder mehrere Einträge hinzugefügt haben):

Mein Gästebuch

[Eintrag hinzufügen](#)

#1: Der erste Gästebucheintrag

Dies ist der erste Gästebucheintrag.

Dieser Eintrag rutscht immer weiter nach unten je mehr neue Einträge hinzukommen.

Eintrag von **Christian Felken** am **06.01.2002 11:46:20**

Neuer Eintrag:

Name:	<input type="text"/>
Titel des Eintrages:	<input type="text"/>
Eintrag:	<input type="text"/>
<input type="button" value="Eintragen"/> <input type="button" value="Löschen"/>	