



Vorbemerkungen:

Bei der erfolgten XAMPP-Installation des Apache Webservers wurde außer dem PHP-Modul auch ein MySQL-Datenbankserver auf dem Computer eingerichtet. Damit ist die Entwicklung dynamischer, datenbankunterstützter Webseiten mittels PHP und MySQL offline möglich.

Installation:

Nach besagter XAMPP-Installation befinden sich die installierten Programme im Verzeichnis C:\xampp . Möglicherweise ist auch die benutzereigene Installation in C:\apachefriends\xampp sinnvoll.

Konfiguration:

Zur bequemeren Programmierung mit PHP öffnen Sie die Datei C:\xampp\php\php.ini . Ersetzen Sie darin die Zeile

```
short_open_tag = Off
```

durch

```
short_open_tag = On
```

Dadurch ergibt sich die Möglichkeit die PHP – Programmierung nicht nur durch das vollständig korrekte Tag <?php sondern auch durch das nicht ganz so korrekte Kurztag <? eingeleitet werden kann.

Die Datenbankzugriffe können, falls keine besonderen Nutzer und Passwörter vergeben wurden, durch den durch Vorgabewert „root“ und leerem Passwort erfolgen.

Steuerung (Control-Panel):



Das Ein- und Ausschalten des Apacheservers sowie des MySQL-Zugriffs auf die Datenbanken erfolgt durch das XAMPP-Controlpanel im Verzeichniss

```
C:\xampp\xampp-control.exe
```

oder auch

```
C:\apachefriends\xampp\xampp-control.exe
```

Ohne eingeschalteten Apache-Server kann kein Abruf der Webseiten unter „localhost“ erfolgen.



Das in TryHTML benutzte Programm WinMySQLAdmin.exe, das mittels



geöffnet werden konnte, existiert in den neuen XAMPP-Versionen nicht mehr.



Eine Administration der Datenbanken kann über das Öffnen von PhpMyAdmin erfolgen.

PHP

Personal Homepage Tools

PHP ist eine in HTML eingebettete serverseitige Scriptsprache. Sie ermöglicht es, mit wenig Aufwand dynamische Webseiten zu erstellen.

Mit PHP entstehen neue einfache Möglichkeiten für die Arbeit mit Datenbanken: Für ein Gästebuch, einen Counter, für die eigene Statistik, für verschiedenste Auswertungen, Foren ... Das alles kann man selbstverständlich auch ohne Datenbank erstellen, mit einer geht es aber viel einfacher! Das aktuelle Thema im Internet ist "Shopping". Mit einer Datenbankanwendung lassen sich komplizierteste Strukturen und Verwaltungstools aufbauen, denn in PHP sind bereits Module für die Anbindung einer Datenbank mitgeliefert. Mit wenigen PHP-Befehlen binden Sie die Datenbank an Ihre Homepage an und schon ist die Dynamik da!

1. PHP-Programme tragen die Erweiterung .php
2. Jedes HTML-Programm kann auch statt der Erweiterung .html die Erweiterung .php bekommen. Zur Ansicht im Browser ist dann kein Apache-Server notwendig.
(Im Internet Explorer wählt man dazu „Datei öffnen“ -> Durchsuchen -> Beispiel.php öffnen -> ankreuzen „als Webordner öffnen“ -> (bei Nachfrage) wählen „als Webseite anzeigen“.)
3. PHP-Programme, die außer HTML auch PHP-Sprachelemente enthalten, müssen
 - bei der Triad-Version im Verzeichnis C:\apache\htdocs
 - bei der XAMPP-Version im Verzeichnis C:\xampp\htdocsoder in einem Unterverzeichnis gespeichert sein. Nun muss jedoch als URL aufgerufen werden localhost/Unterverzeichnis/Dateiname.php.
(Man beachte, dass nun ein Slash statt einem Backslash gesetzt wird.)
Beispiel: Die Datei C:\xampp\htdocs\hallo.php wird im Explorer aufgerufen mit URL localhost/hallo.php.
4. Zeilen mit PHP-Sprachelementen werden in die Tags

```
<?php ... ?>
<? ... ?>
<% ... %>
```

gefasst.
5. PHP-Anweisungen sind für den Aufrufer im Browser nicht sichtbar (auch nicht in Quelltext-Modus)- anders als bei Java-Script. Mit PHP-Anweisungen können keine Anwendungen auf dem Rechner des Clients gestartet werden.
6. PHP stellt über 1200 Funktionen bereit. (Mathematische Funktionen, Zugriff auf Dateien, HTTP-Funktionen, Array- und String-Behandlung, Bildverarbeitung, Datumsverarbeitung, Mail, Apache-Funktionen, Cookies usw...)
7. Hinter einer Anweisung steht ein Semicolon.
8. Eine Ausgabe wird mit der Anweisung „echo“ bewirkt.

```
echo „Hallo Welt!“ ;
```
9. Kommentare in PHP-Texten:

```
//Kommentar am Ende der Zeile
/* Kommentar
über mehrere Zeile */
```

(Erinnerung: <! Kommentar in HTML >)
10. Zu den „Ausdrücken“ in PHP gehören
Konstanten: 100
Variablen: \$Zahl1, \$Text1
(Achtung, bei Variablennamen wird zwischen Groß- und Kleinbuchstaben unterschieden „case-sensitiv“. Variablen werden durch ein vorangestelltes \$-Zeichen deklariert. Der Datentyp wird durch die erste Zuweisung bestimmt.)
Zuweisungen: \$Zahl1=\$Zahl2
Inkrementierungen: echo \$Zahl1++ (Erst Ausgabe, dann Erhöhung um 1)
echo ++\$Zahl1 (Erst Erhöhung um 1, dann Ausgabe)
\$Zahl3 += 10 (Addiere zum Wert von \$Zahl1 10 dazu)

Vergleiche: == gleich
 > größer
 >= größer gleich
 != ungleich
 < kleiner
 <= kleiner gleich

Konditionale Operatoren: `if (Ausdruck1) Ausdruck2; else Ausdruck3;`
 (Entspricht in PASCAL: `if Ausdruck1 then Ausdruck2 else Ausdruck3`)

Beispiel: `<? if (100<50)$erg="kleiner"; else $erg="größer";`
 `echo $erg,"
" ?>`

oder: `Ausdruck1 ? Ausdruck2 : Ausdruck3`

(Entspricht: `if Ausdruck1 then Ausdruck2 else Ausdruck3`)

Beispiel (alternativ): `<? $erg = (100<50)?"kleiner":"größer";`
 `echo $erg,"
" ?>`

11. Blöcke werden mit `{ ... }` geklammert. Entspricht `begin ... end` in PASCAL

12. Arrays werden wie folgt im Programmtext benutzt:

```
$Adresse[Nr]="1";
$Adresse[Name]="Gerd Müller";
$Adresse[Strasse]="Schwanenweg 27";
$Adresse[Ort]="München";
```

oder auch

```
$adresse1=array("Nr"=>"1",
               "Name"=>"Gerd Müller",
               "Strasse"=>"Schwanenweg 27",
               "Ort"=>"München");
```

Sie ähneln damit einem record in PASCAL.

13. Wiederholungsanweisungen (allgemeines und Beispiele):

a) while-Anweisung:

```
while (Ausdruck1): Anweisung endwhile
oder while (Ausdruck1){ Anweisung }
```

b) for -Anweisung:

```
<? for ($i=1; $i<5; $i++) {
    echo $i," ";
}
?>
oder<? for ($i=1,$j=10; $i<5 & $j>2; $i++,$j--) {
    echo "[",$i," ",$j,"] ";
}
?>
```

14. Cookies setzen und lesen (allgemeines und Beispiele):

a) setzen mit:

```
<? setcookie (name,value,date,path,domain,secure) ?>;
<? setcookie ("cook1","Hallo",time()+600,"","",0) ?>;
```

b) lesen mit

```
<? echo "Cookie '", $cook1, "' gelesen" ?>
```

16. Aufnahme von Variablen durch ein Folgeskript (XAMPP):

```
if (isset($_REQUEST['name'])){$name=$_REQUEST['name'];}
else $name="unbekannt";
if (isset($_REQUEST['email'])){$email=$_REQUEST['email'];}
else $email="";
if (isset($_REQUEST['kommentar'])){$kommentar=$_REQUEST['kommentar'];}
else $kommentar="kein Kommentar";
```

Eigene PHP/MySQL-Programme

Wir werden nun nacheinander die php/mysql-Befehle

- `mysql_connect()`
- `mysql_create_db()` und
- `mysql_db_query()`

kennenlernen. Ausführliche Informationen findet man im MySQL-Tutorial in der Karteikarte von Try HTML+PHP. Im XAMPP-Control-Panel muss Apache und MySQL eingeschaltet sein.

1. Eine Verbindung zum Datenbank-Server aufstellen:

Dazu gibt es mehrere Varianten:

```
if (mysql_connect("localhost","root","")) {
    echo "Verbindung hergestellt<BR>";}
else {echo "keine Verbindung<BR>";}

if ($id=mysql_connect("localhost","root","")) {
    echo "Verbindung hergestellt<BR>";}
else die ("keine Verbindung<BR>");
```

Probieren Sie es aus:

In TRY HTML+PHP *Datei / neu* wählen und dann in den Teil `<body> ... </body>` die entsprechenden 3 Zeilen eingeben. Bei Bedarf kann im *Tutorial MySQL* nachgelesen werden.

Wenn Sie dieses Script unter `connect.php` speichern, kann es zu Beginn eines jeden anderen Skripts mit `include(„connect.php“)` aufgerufen und ausgeführt werden.

2. Eine Datenbank erzeugen:

Bevor wir uns konkreten Aufgaben zuwenden, soll noch ausprobiert werden, wie eine Datenbank erzeugt wird. Es sollen zur weiteren Verwendung die Datenbank `wahl` und die Datenbank `shop` erzeugt werden. (Eine Datenbank entspricht übrigens unter Windows einem neu angelegten Ordner im Verzeichnis `C:\xampp\mysql\data` .

Benutzen Sie dazu

```
$Anweisung = "CREATE DATABASE wahl";
if (mysql_query($Anweisung)) {
```

mit Fehlermeldung wie im vorigen Beispiel. Beim Drücken von *aktualisieren* muss die Meldung

```
"keine neue Datenbank erstellt<BR>"
```

kommen, da ja bereits eine existiert.

Verbessern Sie die Zeilen mit einer Variablen `$Datenbank`. Bei der Fehlermeldung kann auch die Funktion `mysql_error()` aufgerufen werden.

Projekt Online-Umfrage

Damit die Getränke für den Schul-Kiosk bedarfsgerecht für die nächste Woche eingekauft werden können, sollen jedesmal in der Vorwoche durch eine Umfrage die Wünsche der Schüler ermittelt werden.

Getränke-Wahl für nächste Woche:

- Milch 0,25l
- Kakao 0,5l
- Cola 0,5l
- Capri Sonne 0,2l
- Bananenshake 0,5l
- Erdbeershake 0,5l
- Capuccino 0,2l

Abstimmen

Beim weiteren Vorgehen kann jedes der Scripte in jeweils einer eigenen Entwicklungsumgebung von Try HTML+PHP erstellt und getestet werden. Je nach Größe des Arbeitsspeichers können bis zu 5 Scripte in 5 verschiedenen Anwendungen von Try HTML+PHP gleichzeitig bearbeitet werden.

3. Schreiben Sie ein HTML- oder ein PHP-Script mit Namen `wahlform.htm` oder `wahlform.php` für die oben abgebildete Browser-Darstellung. Die Getränke-Liste kann natürlich beliebig verändert werden. Beim Drücken von *Abstimmen* soll das Script `wahlerf.php` (Abk. für Wahl-Erfassung) aufgerufen werden. Vorschlag für `wahlform.htm`

Getränke-Wahl für nächste Woche:

```
<form method="get" action="wahlerf.php">
  <input type="radio" name="trank" value="0"> Milch 0,25l<br>
  ...
```

```
  <input type="submit" >
</form>
```

Alternativ kann auch ein Array eingesetzt werden, dessen Elemente im Formular mit

```
<? $gname=array("Milch 0,25l", "Kakao 0,5l", "Cola 0,5l",
               "Capri Sonne 0,2l", "Bananenshake 0,5l",
               "Erdbeershake 0,5l", "Capuccino 0,2l"); ?>
```

So könnte eine Zeile aussehen:

```
<input type='radio' name='trank' value='0'> $gname[0]<br>
```

4. Bevor abgestimmt werden kann, muss in der bereits unter 2 erzeugten Datenbank `Wahl` eine Tabelle namens `getraenke` erzeugt werden, in der die einzelnen Abstimmungen gespeichert werden. (Das Programm soll `wahltab.php` heißen.)

Benutzen Sie dazu

```
$Tabelle="getraenke";
$Anweisung= "CREATE TABLE $Tabelle (
             Nr BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
             Name VARCHAR(100),
             Auswahl TINYINT)";
```

im Zusammenhang mit

```
if (mysql_db_query($Datenbank, $Anweisung)) {
```

oder auch

```
if (mysql_query($Anweisung)) {
```

und einer entsprechenden Meldung. Beim Drücken von *aktualisieren* muss die Meldung keine neue Tabelle erzeugt kommen, da ja bereits eine existiert.

In dieser Tabelle werden die Einträge automatisch fortlaufend nummeriert. Die Auswahl-Nummer wird im Feld *Auswahl* gespeichert. Der besseren Übersicht dient das Feld *Name*, in dem der Name des ausgewählten Getränks eingetragen wird.

Gelöscht wird die Tabelle übrigens mit

```
$Anweisung= "DROP TABLE $Tabelle";
if (mysql_db_query($Datenbank,$Anweisung)) {
```

einschließlich Meldung.

5. Nun kann das Programm Wahlerfassung *wahlerf.php* erstellt werden, mit dem die ausgewählten Getränke in die Tabelle eingetragen werden. Beim Aufruf findet man die Nummer des gewählten Getränks als Inhalt von *\$trank*. Nach der Abfrage

```
if (isset($_REQUEST['trank'])){$trank=$_REQUEST['trank'];}
if (isset($trank)){
```

kann die Eintragung in die Tabelle erfolgen. Sie erfolgt mittels

```
$gname=array("Milch 0,25l","Kakao 0,5l","Cola 0,5l",
            "Capri Sonne 0,2l","Bananenshake 0,5l",
            "Erdbeershake 0,5l","Capuccino 0,2l");

$Datenbank="wahl";
$Tabelle="getraenke";
$Anweisung= "INSERT INTO $Tabelle (Name,Auswahl)VALUES
            ('$gname[$trank]',$trank)";

if (mysql_db_query($Datenbank,$Anweisung)) {
```

einschließlich einer Meldung. Eine Abfrage, ob überhaupt ein Getränk gewählt wurde, also ob *AuswahlNr* gesetzt ist, ist noch nicht einmal nötig, da dann kein Eintrag erfolgt.

6. Nun kann *wahlform.htm* oder *wahlform.php* ausgeführt und die Abstimmung (mehrfach) durchgeführt werden. Jedesmal findet eine Eintragung in die Tabelle *getraenke* statt.

Da wir noch kein Auswertungs-Script geschrieben haben, müssen wir die Tabelleneinträge wie folgt beobachten:



Laden Sie nun in den Webbrowser PhpMyAdmin

localhost/phpmyadmin/index.php,

klicken Sie auf *Wahl* dann auf *Browse*. Anschließend können die Einträge angesehen werden.

7. Zuletzt wird nun ein eigenes Abfrage- und Ausgabe-Script geschrieben. Bei der Abfrage soll gezählt und die Ergebnisse entsprechend der in *Auswahl* gespeicherten Getränke-Nummer zusammengefaßt werden:

```
$Anweisung= "SELECT name,COUNT(*) AS anzahl from $Tabelle GROUP BY
auswahl";
if ($ergebnis1=mysql_db_query($Datenbank,$Anweisung)) {
einschließlich einer Meldung.
```

Die Ausgabe kann dann innerhalb einer Wiederholungsanweisung erfolgen:

```
while ($array=mysql_fetch_array($ergebnis1)) {
    $anzahl= $array['anzahl'];
    $name= $array['name'];
    echo "<tr>";
    echo "<td>$name: </td>";
    echo "<td>$anzahl x </td>";
    echo "</tr>";
}
```

Die Ausgabe kann in einfacher Form oder in Tabellenform erfolgen.

Mögliches Ergebnis:

Getränke-Auswertung

Verbindung hergestellt
einzelne Einträge gezählt

Getränke-Tabelle

Cola 0,5l:	1 x
Capri Sonne 0,2l:	4 x
Erdbeershake 0,5l:	1 x
Capuccino 0,2l:	1 x

Projekt Online-Shop

Damit die Schüler eigene Gegenstände verkaufen oder angebotene Gegenstände erwerben können, soll ein Online-Shop eingerichtet werden.

Die Datenbank `shop` wurde bereits angelegt. Nun müssen die Tabellen

- `artikel`, die die angebotenen Gegenstände aufführt,
- `kunde`, die die Kundenangaben enthält,
- `kaeufo`, die die getätigten Käufe protokolliert, also Artikel-Nr und Kunden-Nr speichert, neu angelegt werden.

8. Tabellen erzeugen

a) Tabelle `artikel` erzeugen (`shoptab.php`)

Die Tabelle `artikel` soll (mindestens) die Felder `Nr` (`NOT NULL AUTO_INCREMENT PRIMARY KEY`), `Name`, `Text` (für Zusatzinformationen), `Preis` und `Gruppennummer` (für späteres Zusammenfassen von Artikeln zu einer Gruppe wie z.B. Sportartikel, Gartengeräte o.ä.) enthalten. Erzeugen Sie die Tabelle; Legen sie dabei die jeweiligen Datentypen fest.

```
$Anweisung= "CREATE TABLE $Tabelle (
    Nr BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(30),
    Text ... ,
    Preis DECIMAL(8,2),
    GruppenNr BIGINT)";
```

Solange Sie sich nicht über den endgültigen Zustand der Tabelle sicher sind, sollten Sie sich die Möglichkeit offen halten, die Tabelle mit

```
$Anweisung= "DROP TABLE $Tabelle";
if (mysql_db_query($Datenbank,$Anweisung)) {
```

(einschl. Meldung) zu löschen.

b) Laden der Tabelle `artikel` mit Daten

Geben Sie in einer Datei `artikel.txt` eine Liste von Artikeln in folgender Form ein:

```
Tennisschläger,gut erhalten,45,4
Sporttasche,neuwertig,20,4
```

Fügen Sie noch einige andere Artikel hinzu.

Nun werden die Artikel aus der Text-Datei in die Tabelle eingelesen. Dazu verwenden Sie die Anweisung

```
$Anweisung= "LOAD DATA LOCAL INFILE
    'C:/xampp/htdocs/n-21RFB/Veranst3/artikel.txt'
    INTO TABLE $Tabelle FIELDS TERMINATED BY ','
    (Name,Text,Preis,GruppenNr)";
```

im Zusammenhang mit

```
mysql_db_query($Datenbank,$Anweisung)
```

einschließlich einer Fehlermeldung.

c) Tabelle `kunden` erzeugen (`shoptab1.php`)

Die Tabelle `kunden` soll (mindestens) die Felder `Nr` (`NOT NULL AUTO_INCREMENT PRIMARY KEY`), `Vorname`, `Nachname`, `Email` und `Adresse` enthalten. Erzeugen Sie die Tabelle; Legen sie dabei die jeweiligen Datentypen fest.

d) Tabelle `kaeufo` erzeugen (ebenfalls in `shoptab1.php`)

Die Tabelle `kaeufo` soll außer einer ID-Nr lediglich die Felder `ArtikelNr` (`NOT NULL`) und `KundenNr` (`NOT NULL`) enthalten. Erzeugen Sie die Tabelle; Legen sie dabei die jeweiligen Datentypen fest.

9. Es sollen nun in einem Formular `shopform.php` alle Daten aus der Tabelle `artikel` gelesen und ausgegeben werden. Als Besonderheit soll eine Referenz eingebaut werden, die bei der jeweiligen Auswahl die Artikel-Nummer an das Bestellungen-Script `shoptake.php` übergibt. Das Aussehen in Form einer HTML-Tabelle könnte wie folgt aussehen:

Verbindung hergestellt
Daten ausgewählt

Artikel-Tabelle

Nr	Name	Preis	Text	Warenkorb
1	Tennisschläger	45.00	gut erhalten	auswählen
2	Sporttasche	20.00	neuwertig	auswählen
3	CD-Laufwerk	15.00	40-fach	auswählen

Die Selektion aller Daten erfolgt mit der Anweisung:

```
$Anweisung="SELECT Nr,Name,Preis,Text
FROM artikel";
```

in Zusammenhang mit

```
if ($Daten = mysql_db_query($Datenbank,$Anweisung)) {
```

einschließlich Meldung.

Die Tabellenüberschriften ergeben sich nach

```
$Spalten=mysql_num_fields($Daten);
for ($Spalte=0;$Spalte<$Spalten;$Spalte++)
{echo "<td><b>" .mysql_field_name($Daten,$Spalte) . "</b></td>";}
echo "<th><b>Warenkorb</b></th>";
```

Die Tabelleninhalte und die Referenz ergeben sich nach:

```
while ($Datensatz=mysql_fetch_array($Daten,MYSQL_ASSOC)) {
    $Nr= $Datensatz['Nr'];
    $Name= $Datensatz['Name'];
    $Preis= $Datensatz['Preis'];
    $Text= $Datensatz['Text'];
    echo "<tr>";
    echo "<td> $Nr</td>";
    ...
    <td><a href='shoptake.php?Nr=$Nr'>auswählen</a></td>";
    echo "</tr>";
}
echo"</table>";
```

mit jeweiliger Ausgabe in einer Tabellenzeile.

10. Artikel und Kundenerfassung (shoptake.php)

Als nächstes muss dem potentiellen Käufer das ausgewählte Produkt noch einmal angezeigt werden (als Ergänzung könnte sogar ein Photo hinzugefügt werden). Des weiteren müssen nun die Kundendaten entgegen genommen werden.

Verbindung hergestellt
Daten ausgewählt

Warenkorb

Nr	Name	Preis	Text
2	Sporttasche	20.00	neuwertig

Kunde (bitte alles ausfüllen)

Vorname:

Nachname:

Email:

Adresse:

Die Erfassung der Daten erfolgt ähnlich wie bei `shopform.php`. Gelöscht werden muss jedoch in der Anweisung der `concat`-Zusatz mit Referenz. Hinzugefügt werden muss am Ende der Anweisung

```
$Anweisung="SELECT ...
FROM artikel WHERE Nr=$Nr";
```

Die Ausgabe der Daten erfolgt genau wie bei `shopform.php`.

Sodann wird ein HTML-Formular erzeugt mit dem Aufruf von `shopkauf.php`, falls der Bestellen-Schalter angeklickt wird, also mit

```
<form action="shopkauf.php" method="get">
```

und den Zeilen

```
<input type="hidden" Name="Nr" value=<? echo $Nr ?>>
<input type="submit" name "Form_ab" value="Bestellen">.
```

Mit der vorletzten Zeile wird erreicht, dass die Artikelnummer die bereits vom Script `shopform.php` übergeben wurde nun an das Script `shopkauf.php` weitergegeben wird.

Ausgeführt werden soll das gesamte Script nur, wenn die Variable `Nr` belegt ist, da sonst eine Weiterbearbeitung keinen Sinn macht. Dies wird geprüft mit

```
if (isset($_REQUEST['Nr'])){$Nr=$_REQUEST['Nr'];}
if (isset($Nr)){
    ...
    gesamtes Skript
    ...
}
```

11. Kundenerfassung speichern und Artikel kaufen (`shopkauf.php`)
Abschließend werden die Kundenangaben abgespeichert. Die dabei vergebene KundenNr wird zusammen mit der ArtikelNr in der Datei `kaeufer` gespeichert. Damit ist der Kauf besiegelt.

Die Kundenangaben werden mittels

```
$Anweisung="INSERT INTO $Tabelle (Vorname,Nachname,Email,Adresse)
VALUES ('$Vorname','$Nachname','$Email','$Adresse)";
if ($Daten = mysql_db_query($Datenbank,$Anweisung)) {
```

einschließlich Meldung in die Tabelle `kunden` eingefügt. An die nun automatisch vergebene Kundennummer kommt man nun mit

```
$KundenNr=mysql_insert_id();
```

Die Kaufdaten werden mittels

```
$Anweisung="INSERT INTO $Tabelle (ArtikelNr,KundenNr)
VALUES ('$Nr','$KundenNr)";
```

mit `mysql_db_query` einschließlich Meldung in die Tabelle `kaeufer` eingefügt.

Damit ist der Kauf getätigt.

Ausgeführt werden soll das gesamte Script nur, wenn die Variable `Nr` belegt ist, da sonst eine Weiterbearbeitung keinen Sinn macht. Dies wird geprüft mit

```
if (isset($_REQUEST['Nr'])){$Nr=$_REQUEST['Nr'];}
if (isset($Nr)){
    ...
    gesamtes Skript
    ...
}
```

Als Ergänzung könnte noch in der Tabelle `kunden` das Kaufdatum (timestamp) eingebaut werden.