

## WEBAPPLIKATIONEN

Prof. Dr. Wolf-Fritz Riekert  
Hochschule der Medien (HdM) Stuttgart  
University of Applied Sciences

<mailto:riekert@hdm-stuttgart.de>  
<http://v.hdm-stuttgart.de/~riekert>

COPYRIGHT © W.-F. RIEKERT 17/04/08

- Lernziele:
  - ⇒ Fähigkeit zum Aufbau einer netzwerkfähigen Datenbank auf einem Datenbankserver.
  - ⇒ Anbindung einer MySQL-Datenbank an ein Access-Frontend mit Hilfe von ODBC.
  - ⇒ Entwicklung einer einfachen Webapplikation mit PHP und MySQL.
- Inhalte:
  - ⇒ Aufbau von Websites (Wiederholung/Steilkurs)
  - ⇒ Datenbanksystem MySQL
  - ⇒ Datenbankadministration mit phpMyAdmin
  - ⇒ Einführung in die Skriptsprache PHP
  - ⇒ Datenbankanbindung mit ODBC

WEBAPPLIKATIONEN

© W.-F. RIEKERT 17/04/08 S. 2

## AUFBAU VON WEBSITES

Wesentliches Ziel dieser Lehrveranstaltung ist der Aufbau einer datenbankgestützten Website mit Hilfe von PHP und MySQL.

Hierfür sind folgende Vorkenntnisse erforderlich:


- Webseitengestaltung, entweder direkt im HTML-Code oder mittels eines graphischen Editors (z.B. Frontpage, Dreamweaver)
- Übertragung von Dateien (insbesondere HTML-Dateien, Grafiken, PHP-Skripte) auf einen Webserver mit Hilfe von FTP oder SSH. (Alternativ: direkter Zugriff auf ein freigegebenes Verzeichnis auf dem Webserver)

Dies wird im Folgenden erklärt. Wer die Vorkenntnisse bereits besitzt, kann den folgenden Abschnitt überspringen.

WEBAPPLIKATIONEN

© W.-F. RIEKERT 17/04/08 S. 3

## HTML: HYPERTEXT MARKUP LANGUAGE

- Dokumentenbeschreibungssprache des WWW
- Legt primär die logische Struktur von Dokumenten fest: Überschriften, Absätze, Hervorhebungen, Tabellen, Links
- Die genaue Darstellung hängt vom Browser und vom Anzeigegerät ab.
- Mit sogenannten Style Sheets (Formatvorlagen) wird die genaue Darstellung (das Layout) festgelegt
- Gestaltung von HTML-Seiten
  - ⇒ im Quelltext, z.B. mit dem **Editor** , der unter **Start - Programme - Zubehör** zu finden ist, oder einem syntaxgesteuerten Editor wie **HTML-Kit** oder **Phase 5**
  - ⇒ oder mit spezialisiertem WYSIWYG (*What you see is what you get*)-Editor: **Mozilla Composer** (kostenfrei) **MS-Frontpage** oder **Dreamweaver** (kostenpflichtig)

WEBAPPLIKATIONEN

© W.-F. RIEKERT 17/04/08 S. 4

- Der aktuelle HTML-Standard ist XHTML (Extensible Hypertext Markup Language)
  - ⇒ Wenn im Folgenden die Rede von HTML ist, soll darunter immer der XHTML-Standard (meist in der Version XHTML 1.0 Transitional) gemeint sein und nicht die veralteten früheren HTML-Standards
- XHTML gehört zur Auszeichnungssprachenfamilie XML (Extensible Markup Language)
  - ⇒ Zur XML-Familie gehören auch andere Auszeichnungssprachen wie z.B. SVG (Vektorgraphik), SMIL (Streaming Media), MathML (Math Formeln)
  - ⇒ Extensible: XHTML kann mit anderen XML-Sprachen erweitert werden (gemischt gebraucht werden).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type"
          content="text/html; charset=iso-8859-1" />
    <title>Riekerts Homepage</title>
  </head>
  <body>
    <h1>Willkommen!</h1>
    
    <p>Wolf-Fritz <em>Riekert</em><br />
      <a href="http://www.hdm-stuttgart.de">
        HdM Stuttgart
      </a>
    </p>
  </body>
</html>
```



## EINIGE BEISPIELHAFTE HTML-ANWEISUNGEN

Funktion	Anfangskennung	Endekennung
HTML-Wurzel	<code>&lt;html&gt;</code>	<code>&lt;/html&gt;</code>
Kopfteil	<code>&lt;head&gt;</code>	<code>&lt;/head&gt;</code>
Metadaten	<code>&lt;meta ... /&gt;</code>	
Dokumenttitel	<code>&lt;title&gt;</code>	<code>&lt;/title&gt;</code>
Dokumentrumpf	<code>&lt;body&gt;</code>	<code>&lt;/body&gt;</code>
Überschrift Gr. 1	<code>&lt;h1&gt;</code>	<code>&lt;/h1&gt;</code>
Überschrift Gr. 2	<code>&lt;h2&gt;</code>	<code>&lt;/h2&gt;</code>
Absatz (paragraph)	<code>&lt;p&gt;</code>	<code>&lt;/p&gt;</code>
Zeilenwechsel (break)	<code>&lt;br /&gt;</code>	
Betont (emphasized)	<code>&lt;em&gt;</code>	<code>&lt;/em&gt;</code>
Hyperlink	<code>&lt;a href = "URL"&gt;</code>	<code>&lt;/a&gt;</code>
Graphik	<code>&lt;img src = "URL" /&gt;</code>	

## ERSTELLUNG VON WEBSEITEN

- Den Webseiten-Editor **Mozilla Composer** oder **Microsoft Frontpage** (oder anderen Web-Editor) starten
  - ⇒ eine **neue Webseite erstellen**
  - ⇒ Webseite im Dateisystem **speichern**
- **Vorschau** mit Web-Browser (Internet-Explorer, Mozilla Firefox, Netscape Navigator oder Opera)
- Seite mit **FTP** oder **SSH** auf den Webserver kopieren
- Ergebnis auf Webserver mit Web-Browser anschauen

### Literatur:

Stefan Münz: **SELFHTML - HTML-Dateien selbst erstellen**. Ein Skript rund um HTML, Javascript und Perl zum Download und Online-Lesen. <http://www.teamone.de/selfhtml/>

## WEBSEITEN MIT EINEM WEBSEITEN-EDITOR BEARBEITEN

- **Texte** nach Belieben schreiben. Formate ändern usw.
- **Bilder** einfügen
  - ⇒ (Gescannte oder mit Grafikeditor erstellte) GIF- oder JPEG-Grafik aus dem Dateisystem übernehmen
  - ⇒ oder aus dem Web (siehe nächste Folie)
- **Hyperlinks** einfügen
  - ⇒ Textbereich oder Grafik mit der Maus markieren
  - ⇒ mit Schaltfläche **Verknüpfung** URL festlegen
- **Sichern** mit Menü **Datei - Speichern unter**
  - ⇒ Als Namen für die Startseite verwendet man **index.html** (Unix/Linux) oder **default.htm** (Microsoft-Systeme).
  - ⇒ Weitere Webseiten erhalten beliebige Namen mit Endung **.html** (Unix/Linux) oder **.htm** (Microsoft).

## BILD AUS DEM WEB ÜBERNEHMEN (VORSICHT EVTL. COPYRIGHT!)

1. Rechter Mausklick

2. Menüaufruf „Grafik speichern unter ...“ bzw. „Save Image As ...“

3. Speicherziel festlegen: Gleiches Verzeichnis wie für die HTML-Dateien wählen.

## BILD EINFÜGEN (FRONTPAGE)

## ANBRINGEN EINER HYPERLINK-VERKNÜPFUNG (FRONTPAGE)

# ÜBERTRAGEN DER WEBSEITEN AUF DEN WEBSERVER

Wie kommen die erstellten Webseite(n) inkl. Grafiken auf den Webserver? Hierfür gibt es folgende Möglichkeiten:

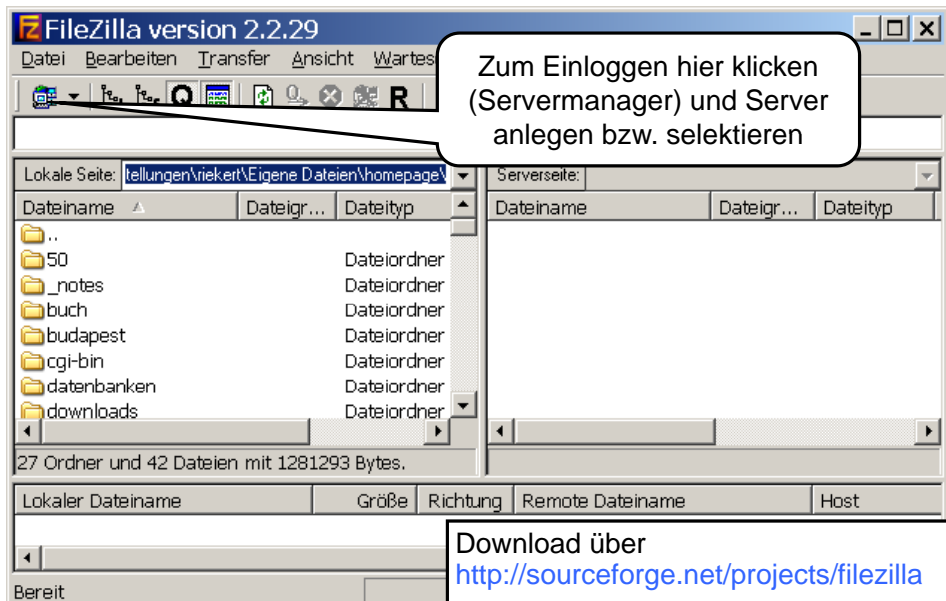
- Erstellen der Seite(n) und Grafiken auf einem lokalen PC im Rechnerpool bzw. zuhause und anschließendes Kopieren der Dateien mit dem **File Transfer Protocol (FTP)**
  - ⇒ Verwendung eines Windows-basierten FTP-Clients, z.B. **Filezilla**
  - ⇒ Verwendung des klassischen kommandobasierten **FTP-Clients** in der DOS-Eingabeaufforderung
  - ⇒ Eine sichere Übertragung ist mit Hilfe von **SSH** möglich.
- **Direktes Arbeiten** auf dem Dateisystem des Webserver bzw. Kopieren von Dateien dorthin mit Windows-Explorer
  - ⇒ Freigabe eines Server-Verzeichnisses im lokalen Netz
  - ⇒ **Verknüpfung mit FTP-Site** (ab Windows 2000)
  - ⇒ Arbeiten auf dem Server mit Hilfe des **HTML-Toolkits**

# FTP (FILE TRANSFER PROTOCOL)

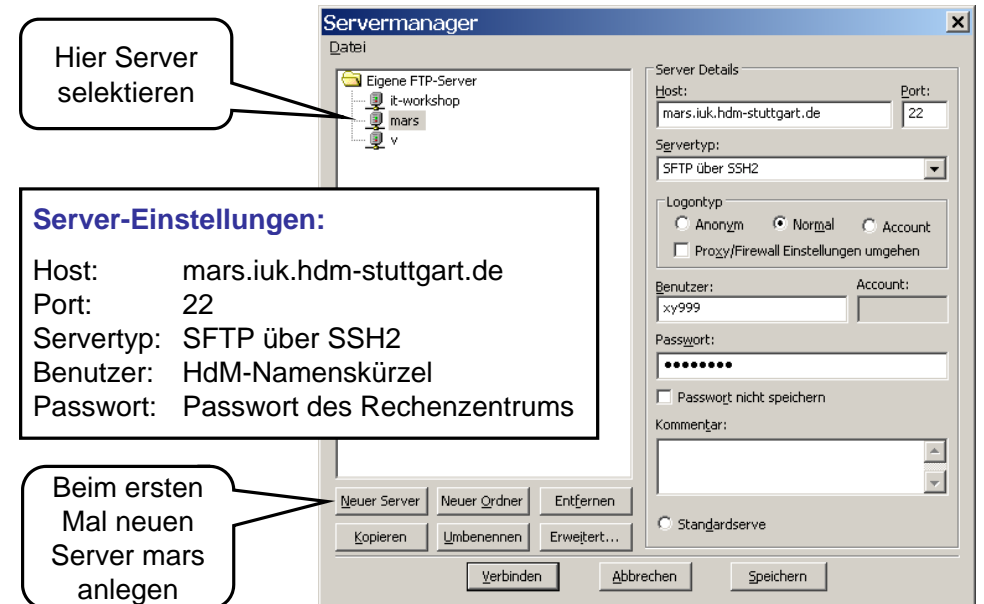
**FTP** (File Transfer Protocol): eines der ältesten Verfahren zur Übertragung von Dateien zwischen Computern im Internet

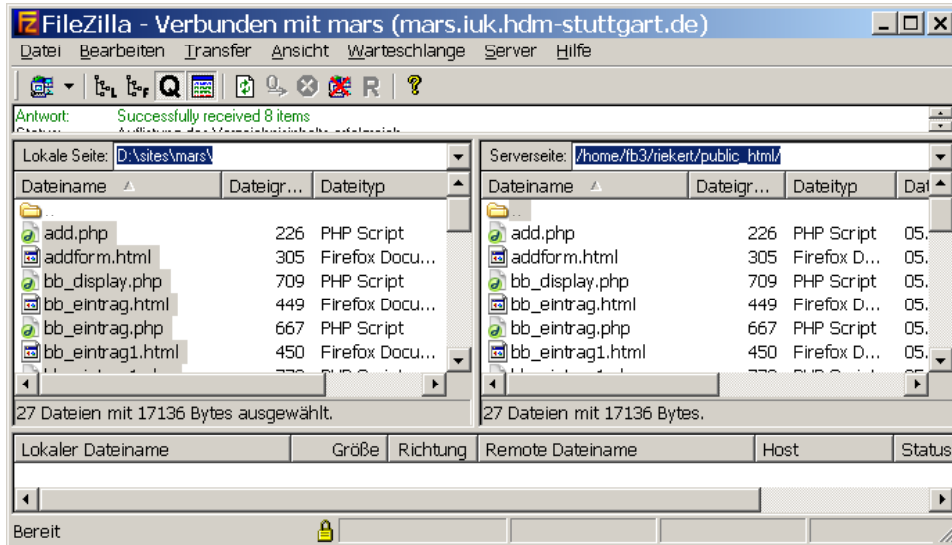
- Nutzung mit verschiedenen **FTP-Clients**:
  - ⇒ Windows-basierter FTP-Client Filezilla (hier erklärt)
  - ⇒ Klassischer, kommandobasierter FTP Client
  - ⇒ Windows Explorer mit FTP-Site-Verknüpfung ab Windows 2000
- Problem: Der normale FTP-Dienst verschlüsselt Passwörter und Daten nicht
- Abhilfe: Neuere verschlüsselte FTP-Dienste verwenden: FTP über SSL bzw. TLS oder SFTP über SSH2
- Der Übungsserver mars.iuk.hdm-stuttgart.de verwendet SFTP über SSH2

# FILEZILLA: WINDOWS-BASIRTER FTP-CLIENT



# FILEZILLA: SERVERMANAGER

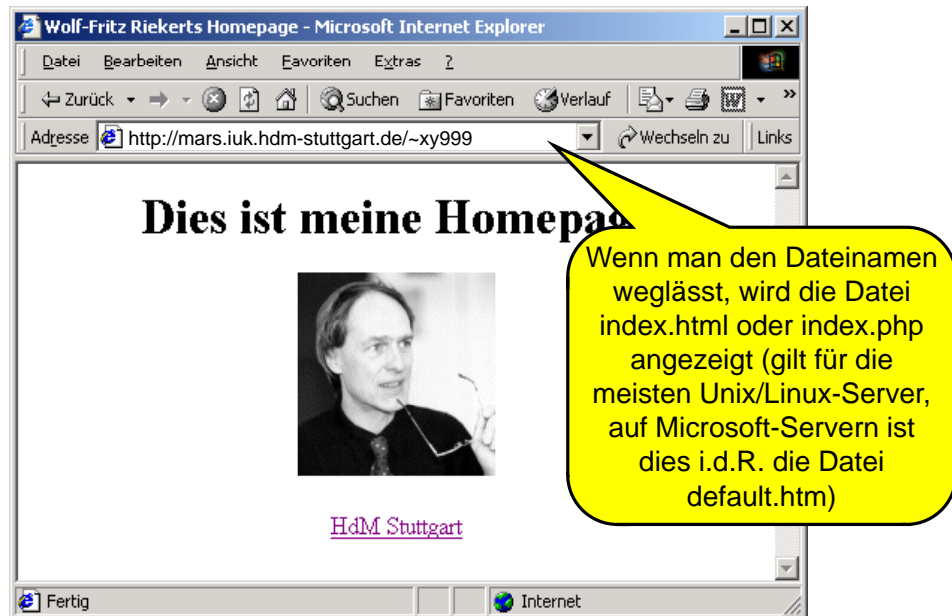




Dateien können durch Ziehen auf den Server kopiert werden.

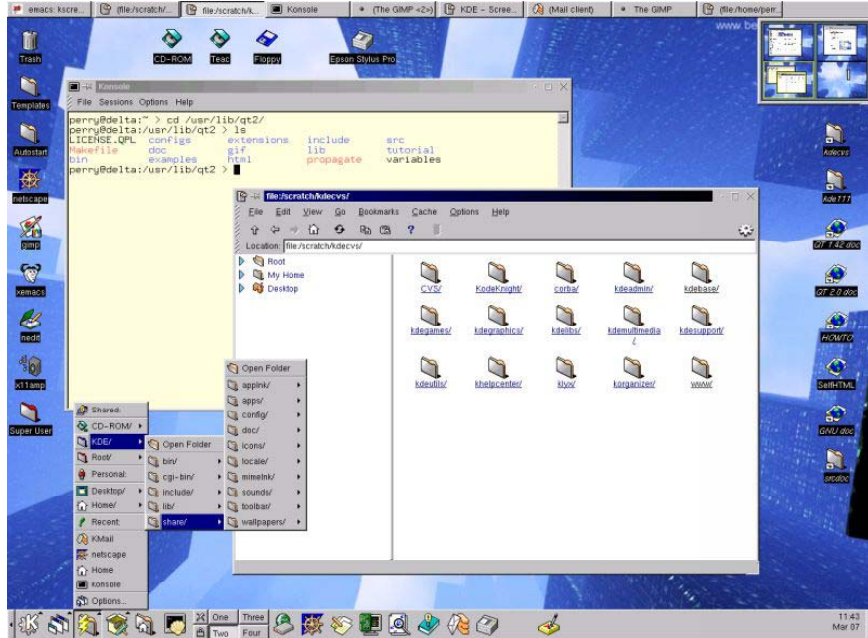
Die Webdateien (HTML-Seiten, Grafiken, PHP-Skripte) müssen auf ein Verzeichnis des Webserver kopiert werden, das für das Web freigegeben ist.

- Auf Linux/Unix-Systemen mit dem Apache-Webserver ist dafür i.d.R. das Unterverzeichnis `public_html` vorgesehen.
- Wenn `public_html` noch nicht vorhanden ist, kann es mit Filezilla erzeugt werden (rechter Mausklick – Verzeichnis erstellen).
- Dateien im Unterverzeichnis `public_html` können vom Web aus mit der URL <http://rechnernamen/~username/dateiname> angesprochen werden. Beispiele:
  - ⇒ <http://mars.iuk.hdm-stuttgart.de/~xy999/datei.html>
  - ⇒ <http://mars.iuk.hdm-stuttgart.de/~xy999/> erreicht die Datei `index.html` oder `index.php` in `public_html`.



- Familie von Multitasking- und Multiuserbetriebssystemen
- Frei verfügbar in Form der Unix-Variante Linux
- Beispiele für kostenpflichtige Unix-Betriebssysteme: Solaris (Sun), HP-UX (HP), AIX (IBM)
- Meist genutzt für Serversysteme
  - ⇒ Benutzung primär kommandoorientiert mittels Shell-Kommandointerpreter (ähnlich DOS-Interpreter)
- Aber auch für Clientsysteme geeignet
  - ⇒ graphische Benutzungsoberflächen, z.B. KDE Desktop Environment (KDE) oder Gnome Desktop Environment, auf der Basis des Fenstersystems X Window System (vergleichbar mit MS Windows)

# K DESKTOP ENVIRONMENT (KDE) GRAPHISCHE LINUX-OBERFLÄCHE



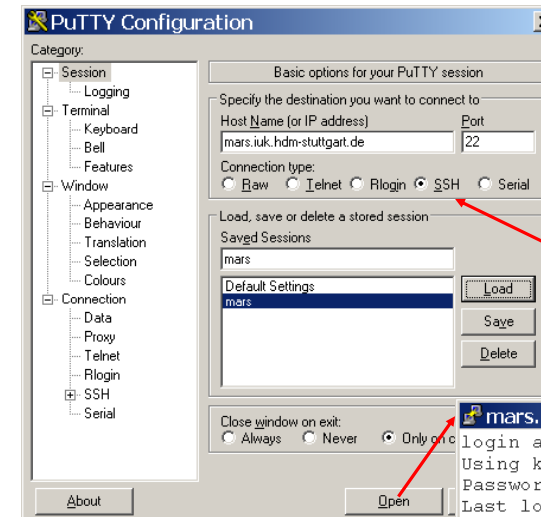
# UNIX/LINUX: UNTERSCHIEDE ZU WINDOWS / DOS

- Unix/Linux verfügt über graphische Benutzeroberflächen ähnlich Windows (z.B. KDE), auf Serversystemen wird jedoch meist kommandobasiert mit der „Shell“ gearbeitet.
- Bedienung des Shell-Kommandointerpreters ähnlich wie DOS-Kommandointerpreter unter Windows, aber anders lautende Befehle.
- Der Shell-Kommandointerpreter unterscheidet zwischen Groß- und Kleinschreibung.
- Trennsymbol für Pfadnamen „/“ statt „\“.

# BEFEHLE DER UNIX/LINUX-SHELL IM VERGLEICH ZU DOS-BEFEHLEN

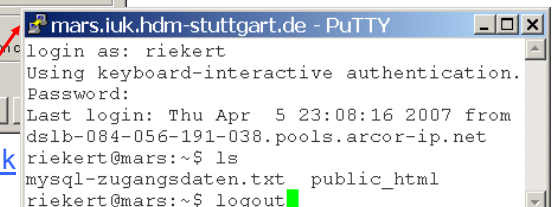
Unix	DOS	Erläuterung
<code>ls -al dir</code>		Auflistung der Dateien und Unterverzeichnisse des Arbeitsverzeichnisses ( <i>a = alle, l = lang</i> )
<code>cd</code>	<code>cd</code>	Verzeichnis wechseln
<code>pwd</code>	<code>cd</code>	Name des aktuellen Verzeichnisses ausgeben
<code>mv</code>	<code>ren</code>	Umbenennen von Dateien
<code>mv</code>	<code>move</code>	Verschieben von Dateien
<code>cp</code>	<code>copy</code>	Kopieren von Dateien
<code>rm</code>	<code>del</code>	Löschen von Dateien
<code>cat</code>	<code>type</code>	Datei ausgeben
<code>more</code>	<code>more</code>	Blättern
<code>mkdir</code>	<code>mkdir</code>	Erstellen von Verzeichnissen
<code>rmdir</code>	<code>rmdir</code>	Löschen von leeren Verzeichnissen
<code>man</code>	-	Manual (Online-Hilfe) ( <code>man Programmname</code> , z.B. Hilfe zum Manual: <code>man man</code> )

# PUTTY: VIRTUELLES TERMINAL AUF BASIS TELNET UND SSH



Mit einem „virtuellen Terminal“, z.B. PuTTY, können Betriebssystem-Befehle auf einem Unix/Linux-Server ausgeführt werden. Mars verwendet den sicheren SSH-Dienst anstelle des unverschlüsselten TELNET-Diensts.

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>



## BEISPIEL: EINE SHELL-SESSION AUF EINEM UNIX/LINUX-SERVER



- Virtuelles Terminal PuTTY starten, Connection Type = SSH
- Einloggen  
hostname=mars.iuk.hdm-stuttgart.de,  
user=xy999,  
password=\*\*\*\*\*
- Sie können nun UNIX-Kommandos eingeben, z.B.:
  - > **ls** *Verzeichnis anzeigen*
  - > **cd public\_html** *Verzeichnis wechseln*
  - > **more index.html** *Datei anzeigen*
  - > **cp index.html kopie.html** *Datei kopieren*
  - > **mkdir backup** *Verzeichnis anlegen*
  - > **mv kopie.html backup** *Datei verschieben*
  - > **cd backup** *Verzeichnis wechseln*
  - > **ls** *Verzeichnis anzeigen*
  - > **rm kopie.html** *Datei löschen*
  - > **logout** *Sitzung beenden*

WEBAPPLIKATIONEN

© W.-F. RIEKERT 17/04/08 S. 25

## CHMOD: SETZEN VON ZUGRIFFSBERECHTIGUNGEN



- Mit dem UNIX-Shell-Kommando **chmod** (*change mode*) werden Zugriffsberechtigungen für Dateien gesetzt
- Generelle Syntax: **chmod ugo+-rwx dateiname**
  - ⇒ Eine Auswahl der Buchstaben **ugo** steht für die betroffenen Benutzer: **u** (= *user*) steht für den Besitzer der Datei, **g** (= *group*) für die Gruppe, der er angehört, **o** (= *others*) für alle anderen Benutzer.
  - ⇒ +- bedeutet Rechte gewähren (+) bzw. entziehen (-)
  - ⇒ Eine Auswahl der Buchstaben **rwX** steht für die Zugriffsrechte: **r** (= *read*) steht für das Leserecht, **w** (= *write*) für das Schreibrecht und **x** (= *execute*) für das Recht, die Datei als Programm auszuführen.
- Beispiel:
  - ⇒ **chmod o+r index.html** *d.h. andere (others) dürfen (+) die Datei index.html lesen (read)*

WEBAPPLIKATIONEN

© W.-F. RIEKERT 17/04/08 S. 26

## MYSQL



- Relationales Datenbanksystem
- Frei erhältlich ([www.mysql.com](http://www.mysql.com))
- Läuft als Serverprozess (unter Unix/Linux und Windows)
- Keine graphische Benutzungsoberfläche wie MS Access
- Benutzung mittels Datenbanksprache SQL
- Gut funktionierende Schnittstelle zu PHP und anderen Programmiersprachen
- Fensterorientierte Bedienung über webbasiertes Administrationstool PHPMYADMIN
- Kommandobasierte Bedienung mit MYSQL-Monitor, aufrufbar von der Unix/Linux-Shell

WEBAPPLIKATIONEN

© W.-F. RIEKERT 17/04/08 S. 27

## MYSQL AUF MARS



Auf dem Server Mars ist ein MySQL-Datenbanksystem eingerichtet.

- Jeder Teilnehmer des Kurses hat eine Benutzungsberechtigung
- Die MySQL-Zugangsdaten stehen in einer Datei im Homeverzeichnis auf dem Server (ggf. Filezilla Anzeige aktualisieren)
  - ⇒ Benutzername
  - ⇒ Passwort (kann später geändert werden)
  - ⇒ Datenbank (für jeden Benutzer ist eine eingerichtet)
- Die Datenbank kann über das Tool PHPMYADMIN bearbeitet werden
- oder über ein Programm (z.B. in PHP) genutzt werden.

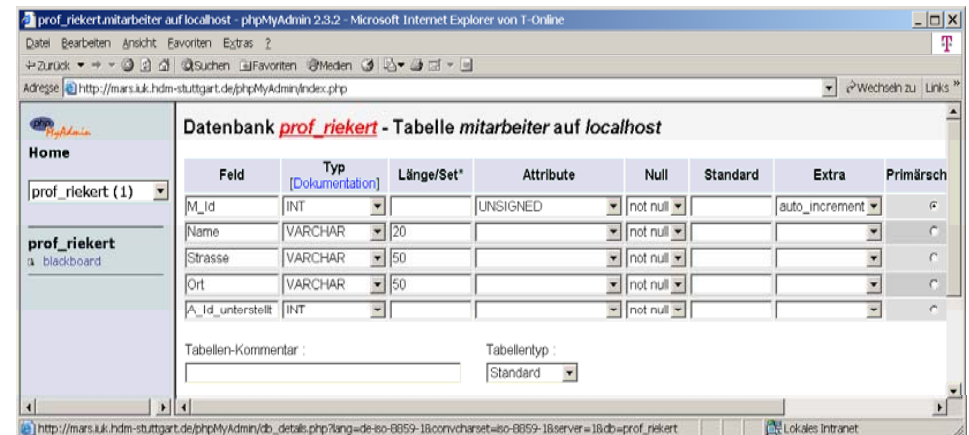
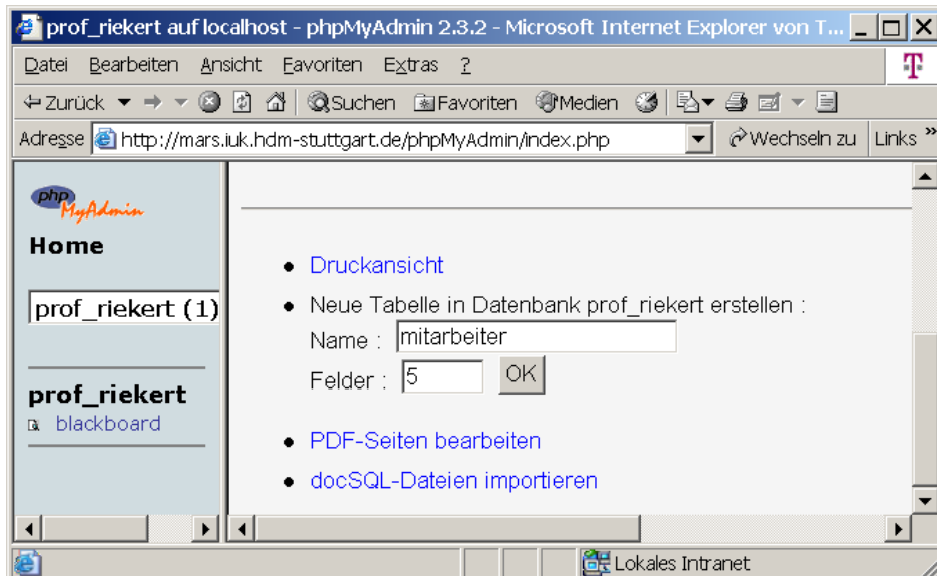
WEBAPPLIKATIONEN

© W.-F. RIEKERT 17/04/08 S. 28

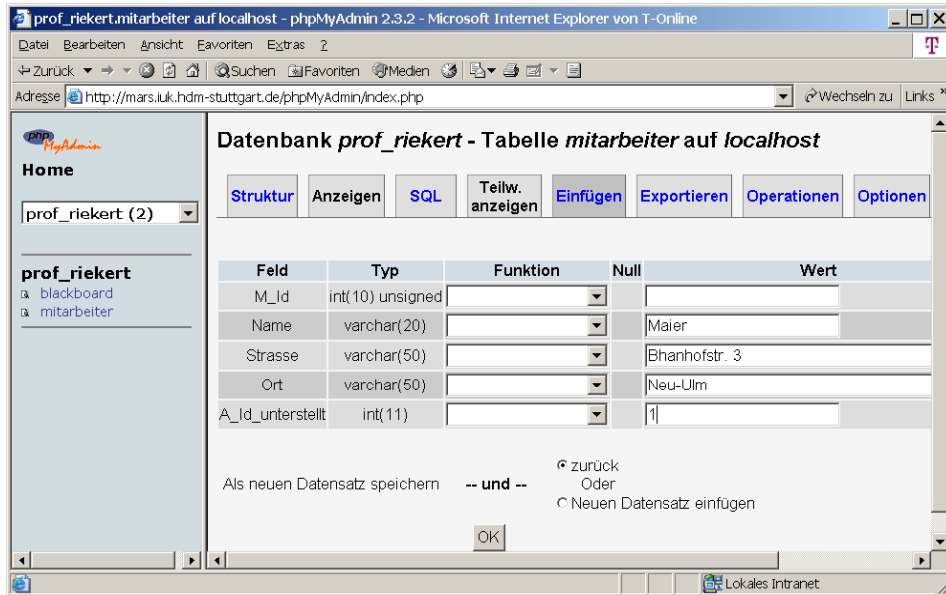
Das Administrationstool PHPMYADMIN erlaubt es die MySQL-Tabellen zu verwalten

- Webapplikation, erreichbar über Browser, meist auf selbem Server installiert wie MySQL selbst (auf Mars: <http://mars.iuk.hdm-stuttgart.de/phpmyadmin>)
- Einloggen mit
  - ⇒ Datenbank-Benutzername (z.B. **xx999**) und
  - ⇒ Datenbank-Passwort **\*\*\*\*\***
- Datenbank auswählen (z.B. **s\_xx999**) oder erstellen (Letzteres nicht für alle Nutzer erlaubt)
- Tabelle auswählen oder neue Tabelle erstellen
- Verschiedene Ansichten: Struktur („Entwurfsansicht“), Anzeigen (Inhalt), SQL (Abfrageeditor), Suche usw.

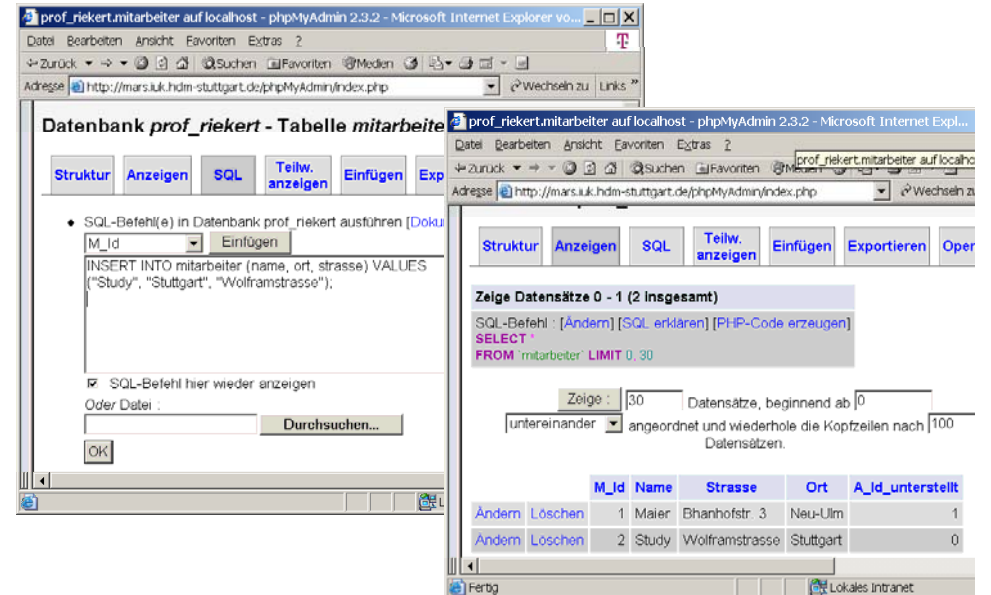
M_Id	Name	Strasse	Ort	A_Id_Unterstellt
1	Maier	Bahnhofstr. 3	Neu-Ulm	1
2	Study	Wolframstr. 32	Stuttgart	1
3	Schaufler	Marktplatz 6	Stuttgart	2
4	Schreiber	Schlossstr. 20	Böblingen	3
5	König	Schillerstr. 29	Ludwigsburg	2
6	Kerner	Kelterstr. 51	Esslingen	3
7	Einstein	Planckstr. 6	Karlsruhe	2
8	Walz	Industriestr. 4	Mannheim	1
9	Braun	Waldstr. 4	Leonberg	2
10	Filzer	Kohlstr. 45	Filderstadt	3
11	Graf	Goethestr. 9	Sindelfingen	3
12	Kaiser	Pfarrstr. 40	Stuttgart	1
13	Hacker	Zusestr. 200	München	4



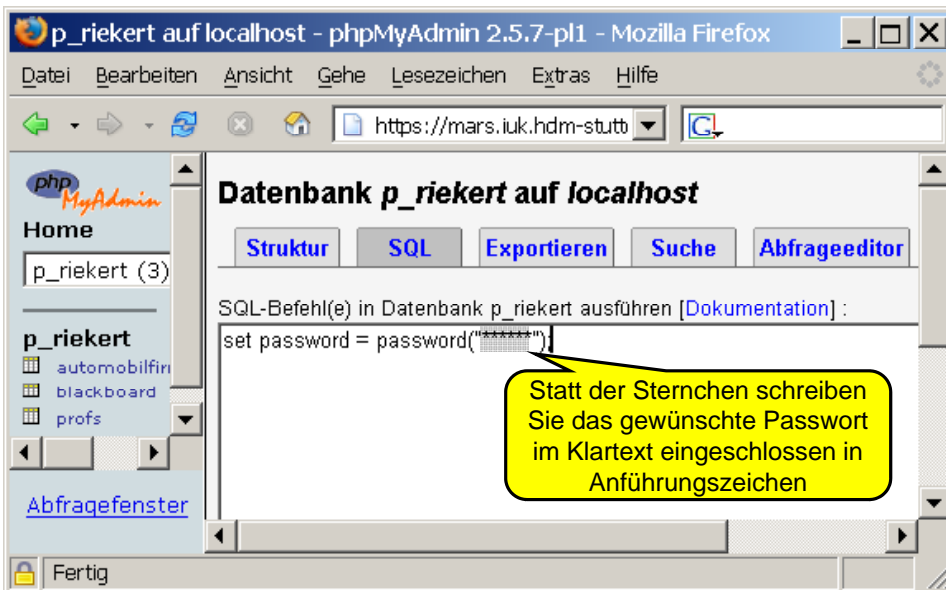
# PHPMYADMIN: TABELLE FÜLLEN



# PHPMYADMIN: EINFÜGEABFRAGE (INSERT)



# PHPMYADMIN: ÄNDERN DES PASSWORTS



# DER MYSQL-MONITOR

- Die „normale“ Bedienung von MySQL erfolgt über den „MySQL-Monitor“. (PHPMYADMIN ist nur eine komfortable Zusatzapplikation und nicht immer installiert.)
- Hierzu muss man sich mittels SSH oder Telnet auf dem Datenbankserver einloggen und das Kommando `mysql` aufrufen, wobei die Optionen `-u` (User), `-h` (Hostname) und `-p` (Datenbank) angegeben werden müssen:

```
mysql -u xy999 -h localhost -p s_xy999  
Enter password: *****
```

Falls sich die Datenbank auf dem Rechner befindet, auf dem man sich eingeloggt hat, wird „localhost“ als Hostname eingegeben.

- Anschließend können SQL-Kommandos und andere MySQL-Befehle eingegeben werden.

```
rieckert@mars:~$ mysql -u riekert -h localhost -p p_riekert
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3603 to server version: 4.0.21-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> select betreff,datum from blackboard where name="riekert";
+-----+-----+
| betreff          | datum          |
+-----+-----+
| Blackboard Relaunch | 20040817114649 |
| test am abend      | 20041011175813 |
+-----+-----+
2 rows in set (0.00 sec)

mysql> set password = password("*****");
```

- PHP: Skriptsprache zur serverseitigen Erzeugung von dynamischen Webseiten
- PHP ist eingebettet in HTML
  - ⇒ Dateinamenserweiterung .php statt .html
  - ⇒ Die festen Anteile einer Webseite werden weiterhin durch die bekannten HTML-Befehle erzeugt
  - ⇒ Die variablen Anteile werden durch die Sprachelemente der PHP-Skriptsprache erzeugt.
  - ⇒ Kennzeichnung der PHP-Skriptelemente durch besondere Tags
- PHP-Syntax ähnlich C/C++, Perl, Java oder Javascript
- Zentrale Website: <http://www.php.net>

- S. Münz: <Professionelle Websites>. Addison Wesley, 2005 (<http://www.professionelle-websites.de/>)
- R. Lerdorf, K. Tatro: *Programmieren mit PHP*. O'Reilly, 2002
- H. Williams, D. Lane: *Web-Datenbank-Applikationen mit PHP und MySQL*. O'Reilly 2002
- K. Yank: *PHP and MySQL. Schritt für Schritt zur datenbankgestützten Website*. Dpunkt-Verlag 2003
- W. J. Gilmore: *A Programmer's Introduction to PHP 4.0*. Apress, 2001 (<http://www.apress.com/book/download.html>)
- D. Enseleit, M. Hupp: SELFPHP. <http://www.selfphp.info/>
- PHP: Hypertext Processor*. <http://www.php.net/>
- PHP Documentation*. <http://www.php.net/docs.php>
- MySQL Documentation*. <http://www.mysql.com/documentation/>

- Mit einem Script-Tag:
  - ⇒ `<script language="php"> echo "Hallo"; </script>`
- XML-konforme Einbettung:
  - ⇒ `<?php echo "Hallo"; ?>`
- SGML-konforme Einbettung (erfordert Konfigurationseinstellung `short_open_tag`)
  - ⇒ `<? echo "Hallo"; ?>`
- Tags im Stil von Microsofts ASP (erfordert Konfigurationseinstellung `asp_tags`)
  - ⇒ `<% echo "Hallo"; %>`
  - ⇒ `<%= $variable >` als Abkürzung für `<% echo $variable %>`

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Beispiel</title>
  </head>
  <body>
    <?php
      echo "Hallo, ich bin ein PHP-Skript!";
    ?>
  </body>
</html>
```

- Von einem Internetbrowser aus wird eine URL mit Dateinamensendung .php abgerufen.
- Der Webserver ist so konfiguriert, dass er angeforderte Dateien mit Endung .php an den PHP-Interpreter übergibt. (Das geht mit Hilfe der so genannten CGI-Schnittstelle o.ä.)
- Der PHP-Interpreter verarbeitet die PHP-Dateien, indem er
  - ⇒ den eingebetteten PHP-Code ausführt und
  - ⇒ aus der PHP-Datei eine HTML-Seite erzeugt.
- Die erzeugte HTML-Seite sieht aus wie die PHP-Datei, nur sind die Teile innerhalb der PHP-Tags durch Text ersetzt, der durch eingebettete PHP-Ausgabeanweisungen generiert wurde, z.B. durch den Befehl `echo`.
- Diese HTML-Seite wird an den Webserver übergeben, der diese dann an den Internetbrowser zur Anzeige überträgt.

## PROGRAMMIEREN MIT PHP: ALLGEMEINE DEFINITIONEN

**Algorithmus:** Arbeitsanleitung zum Lösen eines Problems oder einer Aufgabe, die so präzise formuliert ist, dass sie im Prinzip auch von einem Computer ausgeführt werden kann.

**Programmablaufpläne** (Flussdiagramme) und **Struktogramme** (Nassi-Shneidermann-Diagramme) dienen zur graphischen Darstellung von Algorithmen.

**Programmiersprachen** dienen zur Formulierung von Algorithmen.

Ein in einer Programmiersprache formulierter Algorithmus heißt **Programm**.

In Form von Programmen können Algorithmen durch einen **Computer** ausgeführt werden.

## AUFGABE: DER QUADRATZAHLTTEST

Aufgabe:

Es ist ein Programm zu schreiben, das testet, ob eine vom Benutzer eingegebene Zahl eine Quadratzahl ist.

Beispiel:

<b>Benutzer:</b>	<b>9</b>
<b>System:</b>	<b>Quadratzahl</b>
<b>Benutzer:</b>	<b>3</b>
<b>System:</b>	<b>keine Quadratzahl</b>

# EIN ALGORITHMUS ZUR LÖSUNG DER AUFGABE

## Algorithmus

Parameter  $n$  übernehmen.  
 Der Reihe nach alle Quadratzahlen  $q$  bilden:  
 $q = 0, 1, 4, \text{ usw.}$   
 Weitermachen, solange  $q < n$  (kleiner als  $n$ ) ist.  
 Aufhören, wenn das nicht mehr der Fall ist.  
 Wenn nun  $n = q$  ist, war  $n$  eine Quadratzahl, sonst nicht.

## Beispiel 1

$n = 9$   
 $q = 0 \cdot 0 = 0$   
 $q = 1 \cdot 1 = 1$   
 $q = 2 \cdot 2 = 4$   
 $q = 3 \cdot 3 = 9$   
 Jetzt gilt  $q < n$  nicht mehr  
 $n = q$ ?  
 Ja!  $\Rightarrow n$  ist Quadratzahl

## Beispiel 2

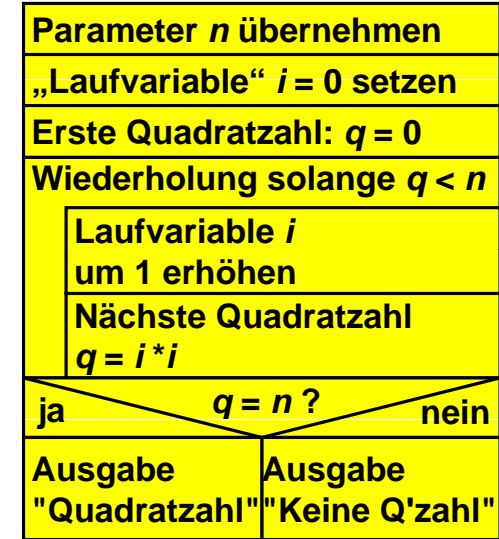
$n = 3$   
 $q = 0 \cdot 0 = 0$   
 $q = 1 \cdot 1 = 1$   
 $q = 2 \cdot 2 = 4$   
 Jetzt gilt  $q < n$  nicht mehr  
 $n = q$ ?  
 Nein!  $\Rightarrow n$  ist keine Q'zahl.

# UMSETZUNG DES ALGORITHMUS IN EIN PROGRAMM

## Algorithmus

Parameter  $n$  übernehmen.  
 Der Reihe nach alle Quadratzahlen  $q$  bilden:  
 $q = 0, 1, 4, \text{ usw.}$   
 Weitermachen, solange  $q < n$  (kleiner als  $n$ ) ist.  
 Aufhören, wenn das nicht mehr der Fall ist.  
 Wenn nun  $n = q$  ist, war  $n$  eine Quadratzahl, sonst nicht.

## Struktogramm



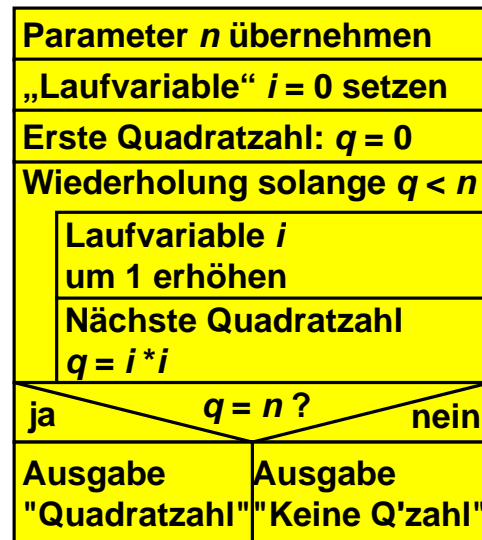
# AUFBAU VON STRUKTOGRAMMEN

Sequenz von Anweisungen

Schleife

Bedingte Anweisung

## Struktogramm

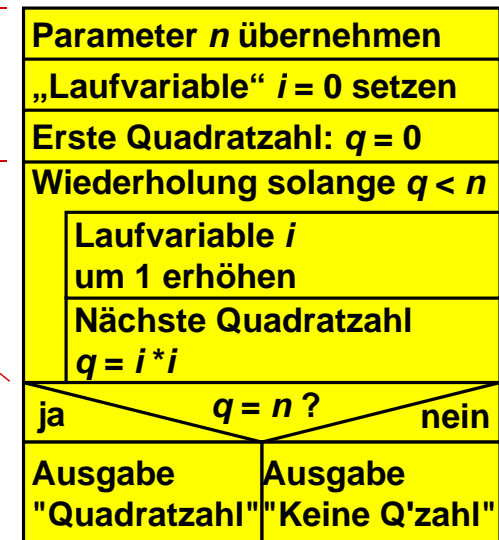


# UMSETZUNG DES STRUKTOGRAMMS IN EIN PHP-PROGRAMM

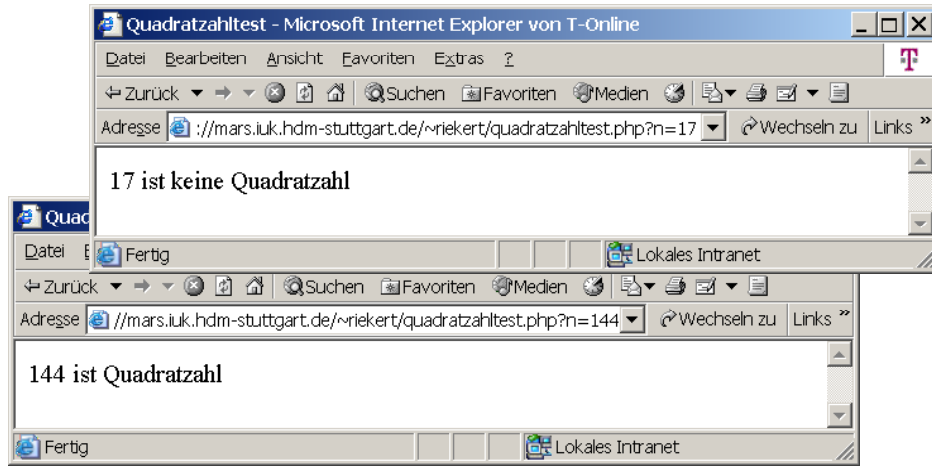
## PHP-Programm

```
$n = $_GET["n"];
$i = 0;
$q = 0;
while ($q < $n) {
    $i = $i + 1;
    $q = $i * $i;
};
if ($q == $n) {
    echo "Quadratzahl";
}
else {
    echo "Keine Q'zahl";
};
```

## Struktogramm



## PARAMETERÜBERGABE VIA URL

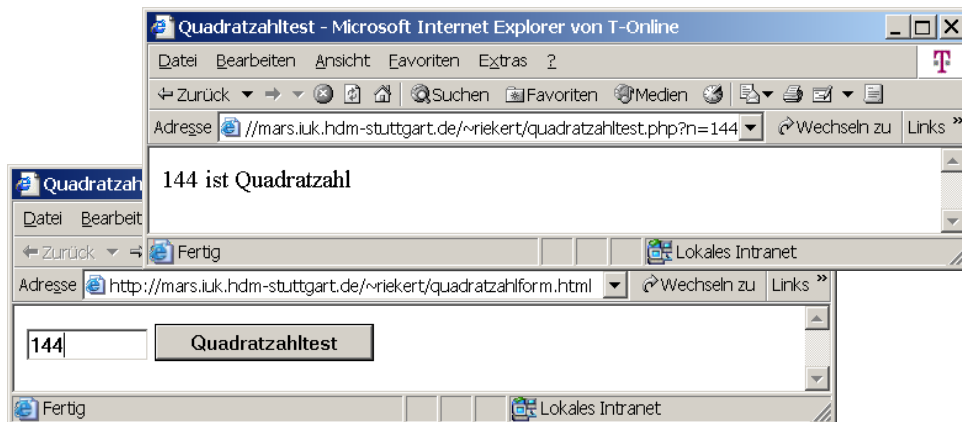


Übergabe des Parameters n an das Skript über die URL:  
Aufruf des Skripts über sogenannte GET-Methode.  
Zugriff auf den Parameter im PHP-Skript durch `$_GET["n"]`

## FORMULAR ZUR PARAMETERÜBERGABE

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Trans
"http://www.w3.org/TR/xhtml1/D
<html xmlns="http://www.w3.org
<head>
  <title>Quadratzahltest</title>
</head>
<body>
  <form action="quadratzahltest.php" method="get">
    <input type="text" name="n" size=10 />
    <input type="submit" value="Quadratzahltest" />
  </form>
</body>
</html>
```

## AUFRUF EINES SKRIPTS VIA FORMULAR

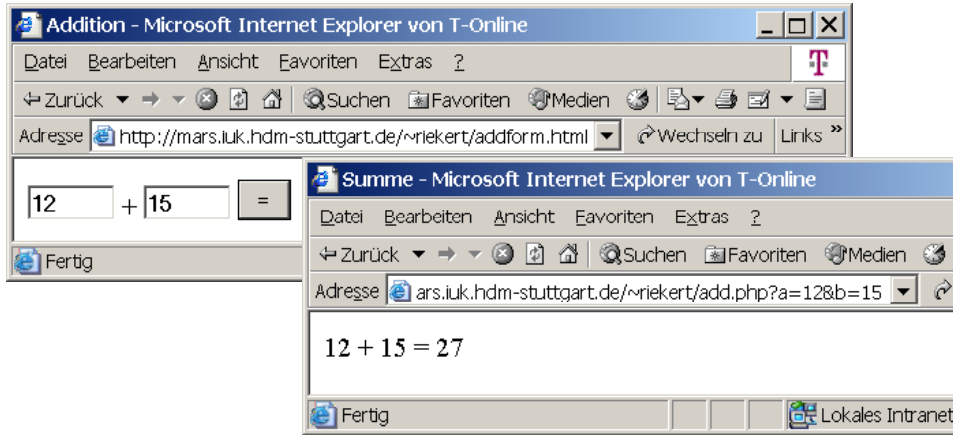


Die Option `method=get` bewirkt eine Parameterübergabe in der URL, doch das geht nur bei kurzen Parametern (< 128 Zeichen). Die Option `method=post` hat keine solche Beschränkung, dann werden aber die Parameter nicht mehr in der URL sichtbar und der Zugriff im PHP-Skript erfolgt dann durch `$_POST["n"]`

## ADDITION: FORMULAR MIT MEHREREN PARAMETERN

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/D
<html xmlns="http://www.w3.org
<head>
  <title>Addition</title>
</head>
<body>
  <form action="add.php" method=get>
    <input type="text" name="a" size="5" /> +
    <input type="text" name="b" size="5" />
    <input type="submit" value=" = " />
  </form>
</body>
</html>
```

# ADDITION: ADDITION ZWEIER ZAHLEN DURCH PHP-SKRIPT



Multiple Parameter werden mit dem &-Zeichen verkettet. Übergabe in der URL durch die Get-Methode:  
<http://domain/verzeichnis/skript.php?par1=wert1&par2=wert2> ...

# ADDITION: DAS PHP-SKRIPT

**Ausgabe des Skripts**

**Das PHP-Skript**

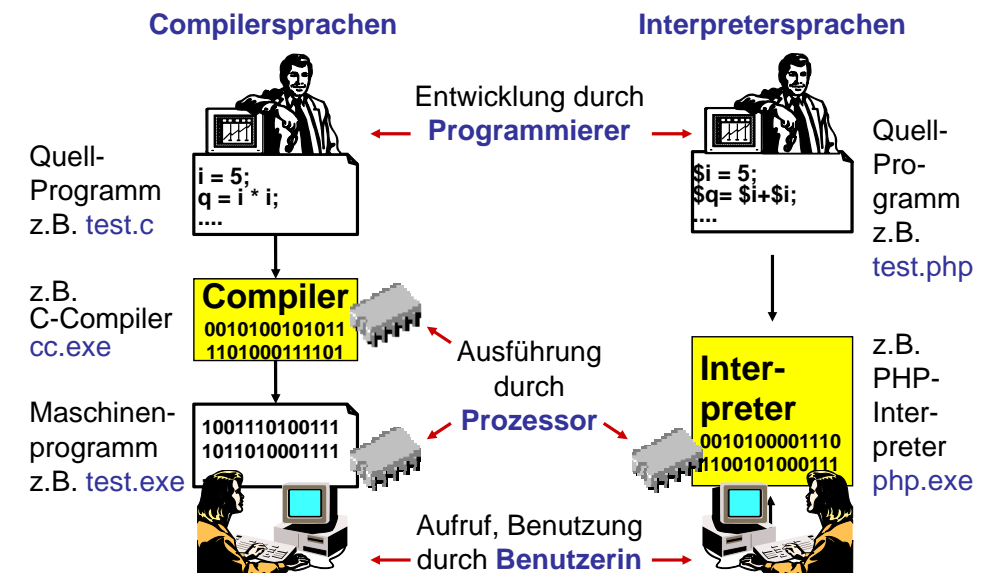
```
<!DOCTYPE HTML P
<html xmlns="http://w
<head>
<title>Summe</title>
</head>
<body>
<?php
    $a=$_GET["a"]; $b=$_GET["b"];
    echo $a . " + " . $b . " = " . ($a+$b);
?>
</body>
</html>
```

**Seitenquelltext der Ausgabe des Skripts**

# ARTEN DER AUSFÜHRUNG VON PROGRAMMEN

- Ein Prozessor (z.B. der Pentium-Prozessor) kann nur sogenannte **Maschinenprogramme** ausführen, diese bestehen aus einer Folge von Zahlencodes.
  - Menschen schreiben Maschinenprogramme in einer sog. **Assemblersprache**, dabei sind die Zahlencodes durch Namen von Maschinenbefehlen ersetzt.
  - Meist schreibt man Programme aber in **höheren Programmiersprachen** (z.B. C, Java, PHP)
    - ⇒ Sie werden dann entweder mit einem **Compiler** in Maschinenprogramme übersetzt und anschließend durch den Prozessor ausgeführt
    - ⇒ oder durch einen sog. **Interpreter** ausgeführt.
- Compiler und Interpreter sind selbst Programme.

# COMPILERSPRACHEN UND INTERPRETERSPRACHEN






- Der Compiler wird nur bei der Programmentwicklung gebraucht, im Betrieb läuft das übersetzte Programm ab.
- Deshalb hat der Compiler Zeit für aufwendige Programmüberprüfungen und Optimierungen.
- Compiler überprüfen Programme hinsichtlich „Vokabular“ und „Grammatik“, so dass viele Programmierfehler bereits bei der Compilierung entdeckt werden können.
- Durch Compiler übersetzte Programme sind Maschinenprogramme, die in der Regel sehr schnell und effizient ablaufen.
- Die Programmentwicklung mit Compilersprachen ist etwas mühevoll, da ein Programm nach jeder Änderung neu kompiliert werden muss.
- Wichtigstes Beispiel für Compilersprachen: C/C++

- Interpreterprogramme funktionieren nicht für sich alleine, sie benötigen zur Ausführung einen Interpreter.
- Da der Interpreter zur Laufzeit des Programms aktiv ist, hat er wenig Zeit für aufwendige Prüfungen. Fehlerhafte Programme „stürzen“ oft mit einer kurzen Meldung „ab“.
- Es gibt jedoch Programmentwicklungsumgebungen mit Editoren, die Syntaxüberprüfungen vornehmen.
- Interpretierte Programme sind deutlich langsamer als kompilierte, was mit den heutigen schnellen Computern allerdings kein großes Problem mehr darstellt.
- Die Programmentwicklung ist erleichtert, da Programme nach Änderungen sofort wieder gestartet werden können.
- Beispiele für Interpretersprachen: Visual Basic for Applications (VBA), Javascript, Perl, **PHP**




## ANWEISUNGEN

PHP-Scripts werden aus Anweisungen gebildet, die mit „;“ verkettet werden.

Elementare Anweisungen sind:

-  **Zuweisungen**, z.B.: `$i = $i + 1`
-  **Befehle**, z.B.: `echo $i`
-  **Unterprogrammaufrufe**, z.B.: `TitelAusgeben("Hi")`

Zusammengesetzte Anweisungen (Kontrollstrukturen) sind:

-  **Sequenzen (Blöcke)** von Anweisungen, verkettet mit „;“, geklammert durch „{}“  
z.B.: `{ $i = $i+1; $s = $s+$i }`
-  **Bedingte Anweisungen**, z.B.: `if`-Anweisung  
`if ($x>0) { $y=$x } else { $y=-$x }`
-  **Schleifen**, z.B.: `while`-Anweisung  
`while($i<10){ $erg = $erg+$i; $i=$i+1 }`

## BEDINGTE ANWEISUNGEN UND SCHLEIFEN

### Bedingte Anweisung:

```
if (Bedingung)
    {Anweisung; Anweisung; ...}
else
    {Anweisung; Anweisung; ...}
```

Bedingung	
ja	nein
Anw.	Anw.
Anw.	Anw.

Der Else-Zweig kann auch entfallen, sofern er nicht gebraucht wird.

### Schleife:

```
while (Bedingung)
    {Anweisung; Anweisung; ...}
```

while (Bedingung)	
Anw.	Anw.
Anw.	Anw.

## VARIABLEN

- In Variablen können einfache Werte gespeichert werden.
- Auch Arrays und Hashes (zusammengesetzte Daten) werden in normalen Variablen gespeichert.
- In PHP sind die Werte typisiert und nicht die Variablen (anders als in Perl oder Java)
- Deshalb beginnen die Namen aller Variablen mit dem Dollarzeichen (\$)

```
$hausnummer = 32; $strasse = "Wolframstrasse";  
$woche = array("mo","di","mi","do","fr","sa");  
// z.B. Formulareingaben in Hash uebernehmen:  
$telefonnr[$name] = $nummer;
```

## WERTE

In PHP gibt es folgende Typen von Werten:

- **Skalartypen** (einfache Datentypen)
  - ⇒ **int** (ganze Zahlen)
  - ⇒ **double** (Dezimalbrüche)
  - ⇒ **string** (Texte)
- **Strukturierte Typen** (zusammengesetzte Datentypen)
  - ⇒ **Array** (Listen von Werten), hierzu zählen auch Hashes
  - ⇒ **Object** (hier nicht erklärt)

```
$i = 100;  
$pi = 3.1415;  
$fh = "HdM";
```

```
$sprache = array("Java","Perl");  
$sprache[2] = "PHP";  
$telefon = array("ra" => 192, "ri" => 185);  
echo $sprache[0] . ": Tel." . $telefon["ra"];  
⇒ Java: Tel.192
```

## STRINGLITERALE (1)

Stringlitterale dienen zur Angabe fester Strings (Zeichenketten). Sie können mit "doppelten" oder 'einfachen' Anführungszeichen dargestellt werden.

In "doppelten" Anführungszeichen lassen sich Sonderzeichen mit Hilfe des Gegenschrägstriches (Backslash) angeben.

- \"** Anführungszeichen
- \\** Gegenschrägstrich (Backslash)
- \n** Zeilenvorschub (Newline)
- \\$** Dollarzeichen
- \033** Sonderzeichen mit Oktalcode 33 (ESC)
- \x7F** Sonderzeichen mit Hexcode 7F (DEL)

```
echo "Der \"zwanghafte\nProgrammierer\"";  
⇒ Der "zwanghafte  
Programmierer"
```

## STRINGLITERALE (2)

- In Stringlitteralen, die durch doppelte Anführungszeichen eingeschlossen sind, können auch Werte von Variablen eingefügt werden (In Perl nennt man das „Interpolation“):  

```
echo "$faktor1 * $faktor2 = ";  
echo $faktor1 * $faktor2;
```
- In Stringlitteralen, die durch 'einfache' Anführungszeichen eingeschlossen sind, werden alle Zeichen unverändert übernommen, mit Ausnahme des Gegenschrägstriches und des einfachen Anführungszeichen. Diese müssen folgendermaßen dargestellt werden:
  - \'** Anführungszeichen
  - \\** Gegenschrägstrich (Backslash)

## AUSDRÜCKE

Zahlenwerte lassen sich mit Hilfe von Rechenoperatoren (Grundrechenarten: + - \* /, Divisionsrestbildung: %, Potenzbildung: \*\*) zu Ausdrücken kombinieren:

```
$umfang = 2 * ($laenge + $breite);  
$neunerrest = $wert % 9;  
$xquadrat = $x ** 2;
```

Strings lassen sich mit Hilfe des Verkettungsoperators . zusammenfügen:

```
$text = "Gesamtpreis = " . 2*3  
$text .= " Euro"; // Anfüegen an die Variable  
echo $text;  
⇒ Gesamtpreis = 6 Euro
```

## TYPWANDLUNG

- Arithmetische Operatoren verknüpfen Zahlen miteinander, Stringoperatoren verknüpfen Strings.
- Erforderlichenfalls wird automatisch gewandelt
  - ⇒ Für Stringoperatoren werden Zahlen in gleichlautende Strings gewandelt
  - ⇒ Für arithmetische Operatoren werden Strings automatisch in Zahlen gewandelt. Wenn diese mit Zahlen beginnen, so werden diese als Ergebnis genommen. Falls nicht, wird 0 als Wert angenommen!
- Besser ist es, die Typwandlungsoperatoren zu verwenden: (int), (double), (string), (array), (object), z.B.  
\$n = 3; echo (double) \$n / 2.0;  
⇒ 1.5

## VERGLEICH VON WERTEN

Für Zahlen und Strings werden dieselben Vergleichsoperatoren verwendet:

```
if ($hausnummer == 34)  
    {print "Nebengebäude"}  
  
if ($str == 'Wolframstrasse')  
    {$hausmeister = "Rueber"}
```

Falls die Datentypen ungleich sind, werden diese automatisch gewandelt, und zwar in Richtung Zahlen! Ein String, der nicht mit Ziffern beginnt, wird dann als 0 interpretiert!

==	gleich
!=	nicht gleich
<	kleiner als
<=	kleiner oder gleich
>	größer als
>=	größer oder gleich

**Achtung! Nicht verwechseln:**  
== Vergleich  
= Zuweisung

## ARRAYS ALS WERTE

Arrays (Listen) sind ein besonderer Datentyp, der mehrere Werte aufnehmen kann:

- Arrays werden mit der Funktion `array()` angelegt.
- Für den Zugriff auf einzelne Elemente verwendet man das \$-Zeichen sowie einen numerischen Index in [eckigen Klammern], der ab 0 zählt.

```
$woche = array('mo', 'di', 'mi', 'do', 'fr', 'sa');  
echo "$woche[1]\n";  
⇒ di  
$woche[6] = 'so';
```

## HASHS

Die Indizes von Arrays müssen nicht notwendigerweise Zahlen sein. Man kann hierfür auch beliebige Strings verwenden.

- Dadurch umfassen Arrays auch die Funktion eines „Hash“ (im Sinne von Perl)

```
$translate = array('gut' => 'good', 'schlecht' => 'bad');  
print "I feel " . $translate['schlecht'] . "\n";
```

⇒ I feel bad

```
$translate['sehr'] = 'very';  
print „PHP is" . $translate['sehr'] . $translate['gut'] . "\n";
```

⇒ PHP is very good

## AUFGABE: SCHWARZES BRETT

Erstellen Sie mit phpMyAdmin in MySQL eine Tabelle mit dem Namen **blackboard**, die als schwarzes Brett dient, mit folgenden Merkmalen:

- id = Nummer des Aushangs, Primärschlüssel
- betreff = Titel des Aushangs, max. 80 Zeichen
- name (maximal 20 Zeichen)
- datum = Datum der Eingabe
- text = eigentlicher Text des Aushangs
- pin = PIN oder Passwort
- aktiv = Aushang ist „aktiv“ geschaltet

## DEFINITION DER TABELLE BLACKBOARD

Feld	Typ [Dokumentation]	Länge/Set*	Attribute	Null	Standard	Extra
id	INT	10	UNSIGNED	not null		auto_increment
betreff	VARCHAR	80		not null		
name	VARCHAR	20		not null		
datum	TIMESTAMP	14		null	NULL	
text	TEXT			not null		
pin	INT	10	UNSIGNED ZEROFILL	not null	0000000000	
aktiv	TINYINT	3	UNSIGNED	not null	0	

## AUSGABE ALLER „AUSHÄNGE“ AUF DEM SCHWARZEN BRETT

```
<?php  
if (mysql_connect("localhost","riekert","*****") AND  
    mysql_select_db("prof_riekert")) {  
  
    $abfrage =  
        "SELECT * FROM blackboard";  
    $ergebnis =  
        mysql_query($abfrage);  
    $zeilenanzahl =  
        mysql_num_rows($ergebnis);  
    $zeilennummer = 0;  
  
    while ($zeilennummer < $zeilenanzahl) {  
        $betreff =  
            mysql_result($ergebnis,$zeilennummer,'betreff');  
        echo $betreff . "<br />\n";  
        $zeilennummer=$zeilennummer+1;  
    }  
}  
  
else {echo "Keine Verbindung zu Datenbank moeglich!";}  
?>
```

Schwarzes Brett  
Informationssysteme  
Wellensittich  
Schlüssel verloren

## AUSGABE DER AUSHÄNGE ALS TABELLE

### Aufgabe:

- Wie könnte man die Aushänge als Tabelle ausgeben?
- Wie könnte man den Betreff jeweils anklickbar machen?

Betreff	Name
<a href="#">Informationssysteme</a>	riekert
<a href="#">Wellensittich</a>	xy007
<a href="#">Schlüssel verloren</a>	schussel

### Lösungsansatz:

- Zunächst mal eine Tabelle mit HTML gestalten nach Anleitung in SELFHTML (<http://selfhtml.teamone.de/html/>) oder mit Hilfe eines HTML-Editors (z.B. Frontpage).
- Dann nach diesem Muster eine PHP-Seite entwickeln.
- Der Betreff sollte als Hyperlink ausgebildet werden:  
[http://mars.iuk.hdm-stuttgart.de/~riekert/bb\\_display.php?id=1](http://mars.iuk.hdm-stuttgart.de/~riekert/bb_display.php?id=1)  
[http://mars.iuk.hdm-stuttgart.de/~riekert/bb\\_display.php?id=2](http://mars.iuk.hdm-stuttgart.de/~riekert/bb_display.php?id=2)  
usw. Allerdings muss die id durch PHP eingefügt werden.

## EERSTELLEN VON TABELLEN MIT HTML

```
<html>
<head>
<title>Tabellenbeispiel</title>
</head>
<body>
  <table border="1">
    <tr>
      <th> Id </th> <th> Betreff      </th> <th> Name </th>
    </tr>
    <tr>
      <td> 1 </td> <td> WG              </td> <td> Paul </td>
    </tr>
    <tr>
      <td> 2 </td> <td> Skateboard </td> <td> Susi </td>
    </tr>
  </table>
</body>
</html>
```

Id	Betreff	Name
1	WG	Paul
2	Skateboard	Susi

siehe auch: <http://selfhtml.teamone.de/html/tabellen/>

## AUFBAU DES PHP-SKRIPTS: ÄUSSERER RAHMEN

```
<table border="1">
  <tr> <th> Betreff </th> <th> Name </th> </tr>
<?php
if (mysql_connect("localhost","riekert","*****") AND
    mysql_select_db("prof_riekert")) {

  $abfrage = "SELECT * FROM blackboard";
  $ergebnis = mysql_query($abfrage);
  $zeilenanzahl = mysql_num_rows($ergebnis);
  $zeilennummer = 0;

  while ($zeilennummer < $zeilenanzahl) {
    Hier Ausgabe der Tabellenzeilen, siehe nächste Folie
  }
}
else {echo "Keine Verbindung zu Datenbank moeglich!";}
?>
</table>
```

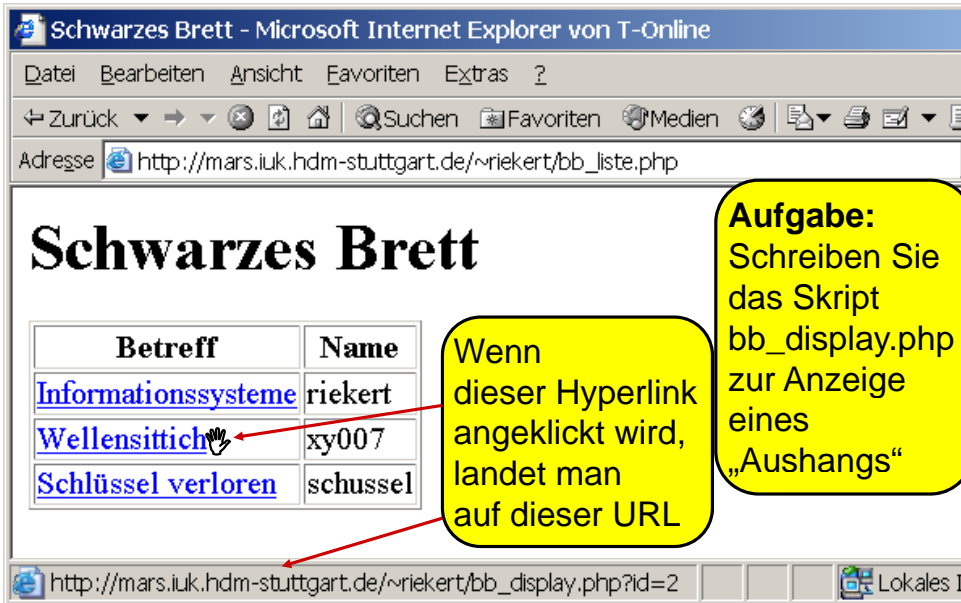
## AUFBAU DES PHP-SKRIPTS: AUSGABE DER TABELLENZEILEN

```
$id =
  mysql_result($ergebnis,$zeilennummer,'id');
$betreff =
  mysql_result($ergebnis,$zeilennummer,'betreff');
$name =
  mysql_result($ergebnis,$zeilennummer,'name');

echo "<tr><td><a href=\"bb_display.php?id=" . $id .
  "\" >" .
  $betreff .
  "</a> </td><td>" . $name . "</td></tr>\n";

$zeilennummer=$zeilennummer+1;
```

Betreff	Name
<a href="#">Informationssysteme</a>	riekert
<a href="#">Wellensittich</a>	xy007
<a href="#">Schlüssel verloren</a>	schussel



Weiterentwicklung der Anwendung alleine oder im Team von 2 Personen und Bereitstellung auf Mars:

- Anzeige einer Liste aller Aushänge inkl. Betreff etc.
- Detaildarstellung eines einzelnen Aushangs mit komplettem Text (beim Anklicken des Betreffs)
- Formular zur Neueingabe von Aushängen
- zweite, über eine Fremdschlüsselbeziehung verknüpfte Tabelle, die in der Webapplikation sinnvoll genutzt wird, z.B.
  - ⇒ Studiengangnamen, die einem Beitrag zugeordnet werden können
  - ⇒ Namen der Autoren der Aushänge.

Ähnliche Aufgabenstellungen nach Rücksprache möglich.

Genauere Anforderungen auf meinen Webseiten

<http://v.hdm-stuttgart.de/~riekert/lehre/cnwa-anforderungen.html>

- Bei Neueingabe: automatische Erzeugung und Versand einer PIN an die Emailadresse [name@iuk.hdm-stuttgart.de](mailto:name@iuk.hdm-stuttgart.de). Erst nach Eingabe dieser PIN kann der Aushang aktiv geschaltet werden
- Anzeigefunktion soll nur aktive Aushänge anzeigen
- Möglichkeit der Bearbeitung und des Löschens von Aushängen (erfordert Angabe der PIN)
- Feedbackformular: Versand einer Email an den Ersteller des Aushangs ([name@iuk.hdm-stuttgart.de](mailto:name@iuk.hdm-stuttgart.de))
- Suchfunktionen, Sortierfunktionen

- Riskant: Vertrauen auf korrekte Formularparameter. Jedes Skript kann auch von einem „selbstgebastelten“ Formular mit inkorrekten Parametern aufgerufen werden.
- Riskant: Vertrauen auf korrekt gesetzte versteckte Formularfelder und korrekt gesetzte Cookies. Auch diese können manipuliert werden.
- Riskant: Ungeprüfte Weiterverwendung von Formularparametern zur Erzeugung von Webseiten, SQL-Kommandos und Betriebssystemkommandos oder zur Verwendung als Dateinamen.
- Problematisch aber bei Webhostern üblich: Mehrere Websites auf demselben Server (denn Webserverprozess hat Zugriff auf alle Webverzeichnisse, auch die fremder Websites)

## PROBLEME DURCH UNGEPRÜFTE FORMULARVARIABLE (HTML)

- Variablen aus Formularen werden manchmal auf Webseiten ausgegeben
- Probleme entstehen, wenn der User HTML-Code ins Formular eingibt.
- Abhilfe: HTML-Kommandos entfernen mit Funktion `strip_tags()` oder nur im Quelltext sichtbar machen mit `html_entities()`:

Name: Riekert  
Betreff: Test  
Text des Aushangs:  
<img src=http://M.HdM-Stuttgart.de/~riekert/  
Aushaengen!  
Fertig



## HTML-CODE AUS AUSGABE ENTFERNEN

### `htmlentities()`

#### PHP-Code:

```
<?php  
$str = "<b>fett</b>";  
echo htmlentities($str);  
?>
```

#### Ausgabe im HTML-Code:

```
&lt;b&gt;fett&lt;/b&gt;
```

#### Anzeige im Browser:

```
<b>fett</b>
```

HTML-Code wird angezeigt,  
aber nicht ausgeführt

### `Strip_tags()`

#### PHP-Code:

```
<?php  
$str = "<b>fett</b>";  
echo strip_tags($str);  
?>
```

#### Ausgabe im HTML-Code:

```
fett
```

#### Anzeige im Browser:

```
fett
```

HTML-Code wird ignoriert

## PROBLEME DURCH UNGEPRÜFTE FORMULARPARAMETER (SQL)

```
$text=$_POST["text"];
```

```
$abfrage = "INSERT INTO blackboard (name,betreff, text)  
VALUES ('$name','$betreff','$text')";
```

Was passiert, wenn für \$text Folgendes eingegeben wird?

```
blabla"); DELETE from blackboard;
```

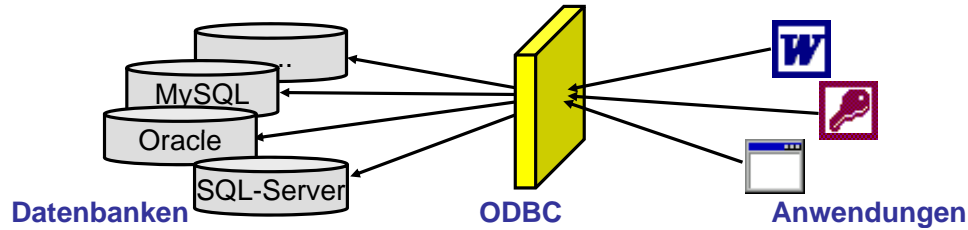
Möglicherweise wird der DELETE-Befehl ausgeführt:

Abhilfe:

- Quotes (") unwirksam machen durch Funktion `addslashes()`
- Oder PHP mit Option "magic quotes" konfigurieren (ist auf Mars der Fall)

## MULTIPLE WEBSITES UND SAFE MODE

- Normalerweise ist der PHP-Quellcode nicht sichtbar
  - ⇒ User sehen nur den generierten HTML-Code
  - ⇒ Sicherheitsgewinn („security by obscurity“)
- Aber: PHP-Skripte werden serverseitig vom Webserver-Prozess ausgeführt. Dieser Prozess hat Zugriff auf alle PHP-Skripte im Quelltext.
  - ⇒ Sicherheitsproblem auf Servern, die mehrere Websites hosten.
  - ⇒ Ein Skript auf einer Website kann Dateien auf einer anderen Website öffnen, lesen und interpretieren.
- Abhilfe (für Webmaster): PHP im „Safe Mode“ konfigurieren. Der PHP-Interpreter verbietet dann solche Zugriffe



- ODBC: Von Microsoft definierte Schnittstelle zu beliebigen relationalen Datenbanken (auch Nicht-Microsoft-Produkte)
- Kann von beliebigen Anwendungen unter Windows genutzt werden.
- Beispiele:
  - ⇒ Serienbrief mit MS-Word, Adressen werden aus einer beliebigen Datenbank geholt
  - ⇒ Access-Benutzungsoberfläche als Frontend für eine beliebige Serverdatenbank

- Um eine bestimmte Datenbank als so genannte ODBC-Datenquelle ansprechen zu können, muss ein geeignetes Treiberprogramm installiert sein.
  - ⇒ für MySQL wird der MySQL ODBC Driver benötigt, Download über [www.mysql.com](http://www.mysql.com) bzw. direkt unter <http://dev.mysql.com/downloads/connector/odbc/>
- Einrichtung im ODBC-Datenquellen-Administrator (erreichbar über Systemsteuerung – Verwaltung) oder in Access (Öffnen ODBC-Datenquelle)
  - ⇒ Hinzufügen einer neuen Datenquelle für den betreffenden Treiber (z.B. MySQL ODBC Driver)
  - ⇒ Vergabe eines Datenquellennamens (DSN = Data Source Name) und Festlegung der DSN-Konfiguration (Eigenschaften der Datenbankverbindung)
- Nutzen der Datenquelle aus einer dafür geeigneten Anwendung, z.B. aus dem MS-Access-Frontend

## EINRICHTEN EINER ODBC-SCHNITTSTELLE

### Startmenü:

Einstellungen

Systemsteuerung

Verwaltung

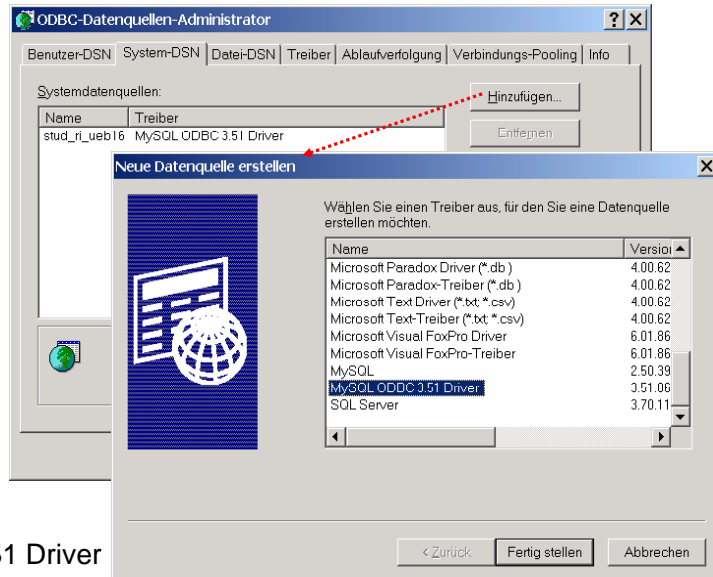
Datenquellen (ODBC)

System-DSN (es geht auch: Benutzer-DSN)

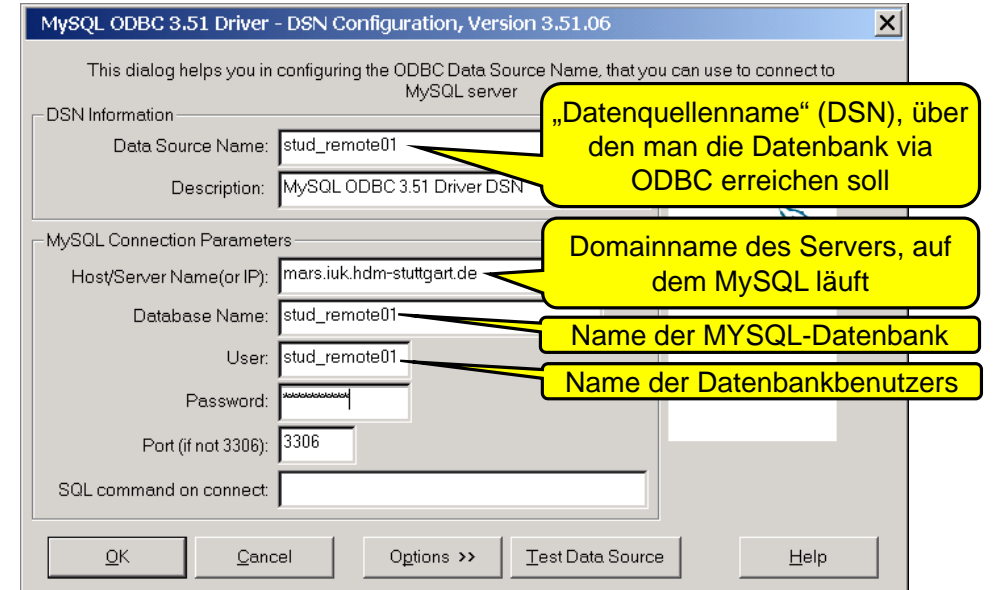
Hinzufügen

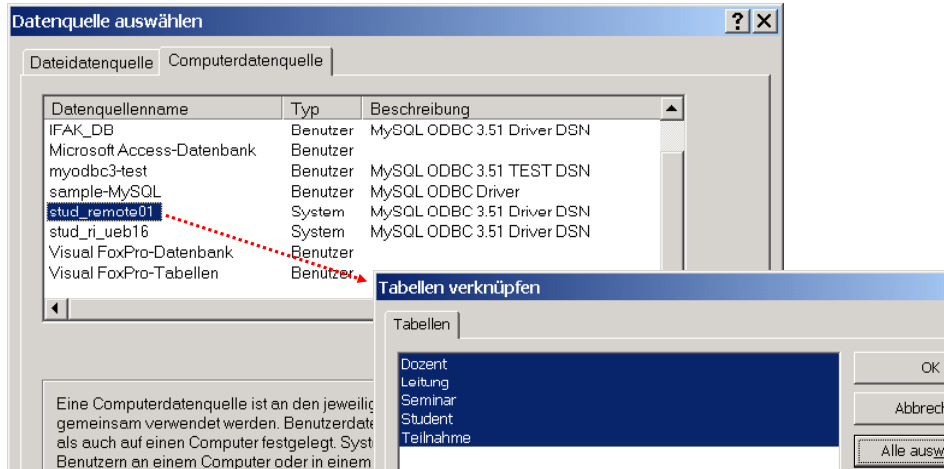
ODBC-Treiber auswählen

hier z.B. MySQL ODBC 3.51 Driver

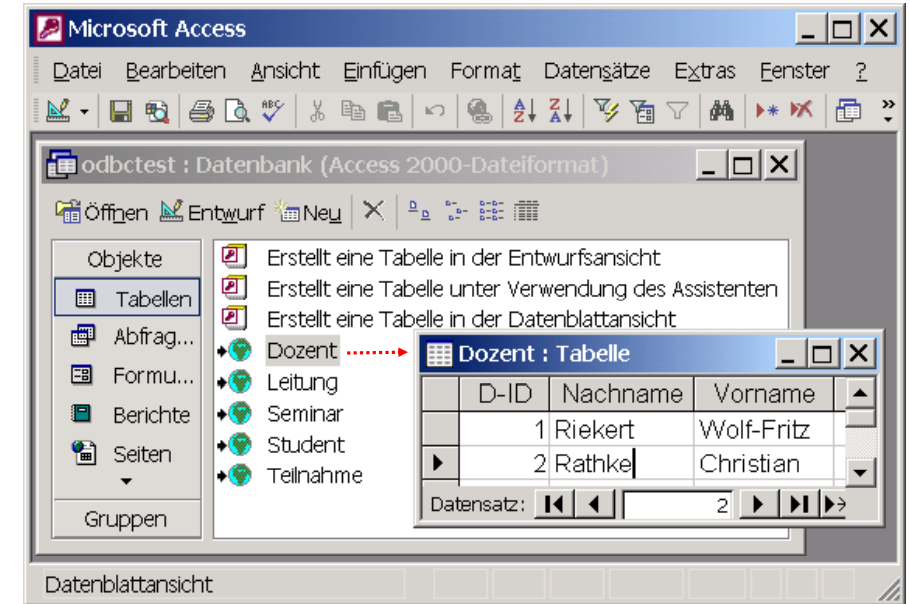


## EINRICHTEN EINER ODBC-DATENQUELLE FÜR MYSQL





In MS-Access: Datei – Externe Daten - Tabellen Verknüpfen  
Dateityp = ODBC Databases – Computerdatenquellen –  
Datenquelle auswählen – Tabellen verknüpfen



Der ODBC-Zugang zu MySQL erlaubt auch den Mehrbenutzerzugriff. Damit eventuelle Zugriffskonflikte entdeckt werden können, sind Modifikationen in der MySQL-Datenbank erforderlich:

- In jeder Tabelle muss ein Feld mit dem Namen timestamp eingerichtet werden
- Das Feld erhält den Datentyp TIMESTAMP
  - ⇒ Wirkung: Bei jeder Änderung eines Datensatzes wird der Wert dieses Felds auf die aktuelle Systemzeit gesetzt.
  - ⇒ Dadurch kann erkannt werden, ob ein gelesener Datensatz vor dem Zurückschreiben durch einen anderen Benutzer verändert wurde.

Auf den folgenden Seiten finden Sie Informationen zu alternativen Tools für die Erstellung von Webapplikationen

- SSH Secure Shell Client: Sichere Alternative zu Telnet, ähnliche Funktionalität wie SSH-Tool putty
- SSH File Transfer Client: Sichere Alternative zu FTP, dieselbe Funktionalität ist auch in Filezilla enthalten
- HTML-Kit: FTP-basiertes Tool mit integriertem HTML-Editor, erlaubt das Bearbeiten von HTML-Dateien „auf dem Server“. Beim Öffnen einer Datei erfolgt ein Download, beim Sichern ein Upload.

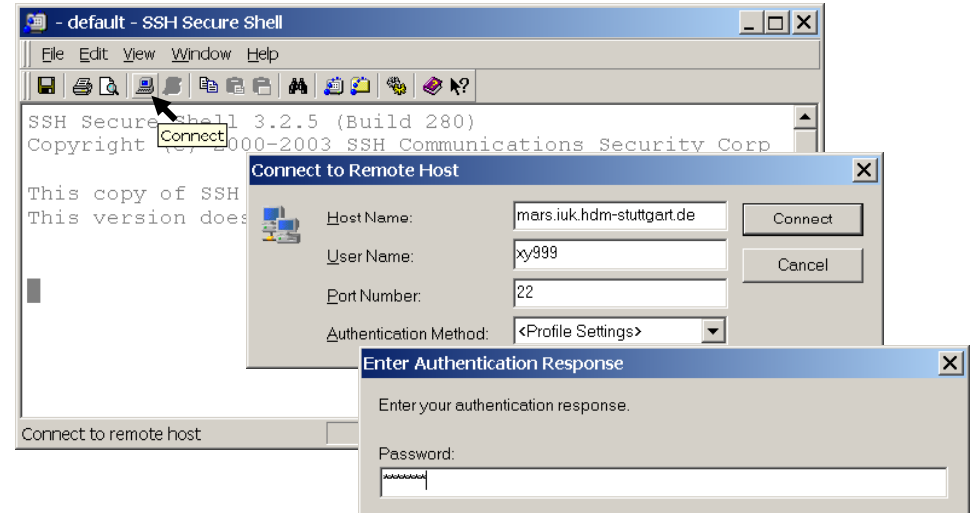
## SECURE SHELL (SSH)

Secure Shell (SSH) nennt man einen Internetdienst und das zugehörige Protokoll mit folgenden Eigenschaften:

- SSH benutzt eine verschlüsselte Datenübertragung mittels SSL (Secure Socket Layer). Die Datenübertragung ist also sicher.
- SSH ermöglicht die Übertragung von Betriebssystemkommandos an einen Serverrechner, ähnlich wie der Telnet-Dienst.
- SSH ermöglicht den Austausch von Dateien zwischen einem Client- und einem Serverrechner, ähnlich wie FTP.
- SSH Communications Security Corp ([www.ssh.com](http://www.ssh.com)) bietet zwei frei erhältliche Clients an: den Telnet-artigen Secure Shell Client und den FTP-ähnlichen Secure File Transfer Client. Die Benutzungsoberfläche des File Transfer Client ähnelt der des Windows-Explorer.

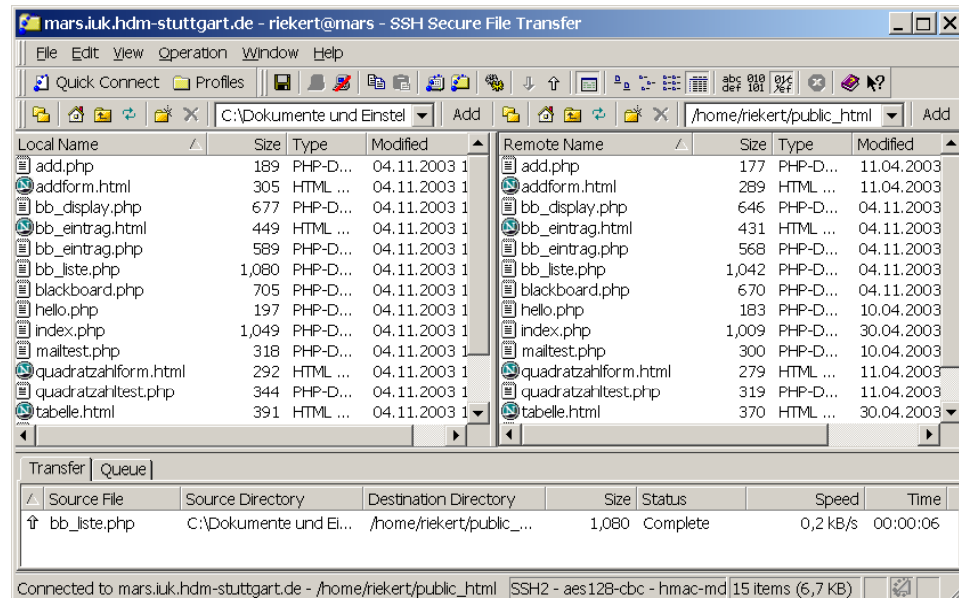


## DER SSH SECURE SHELL CLIENT

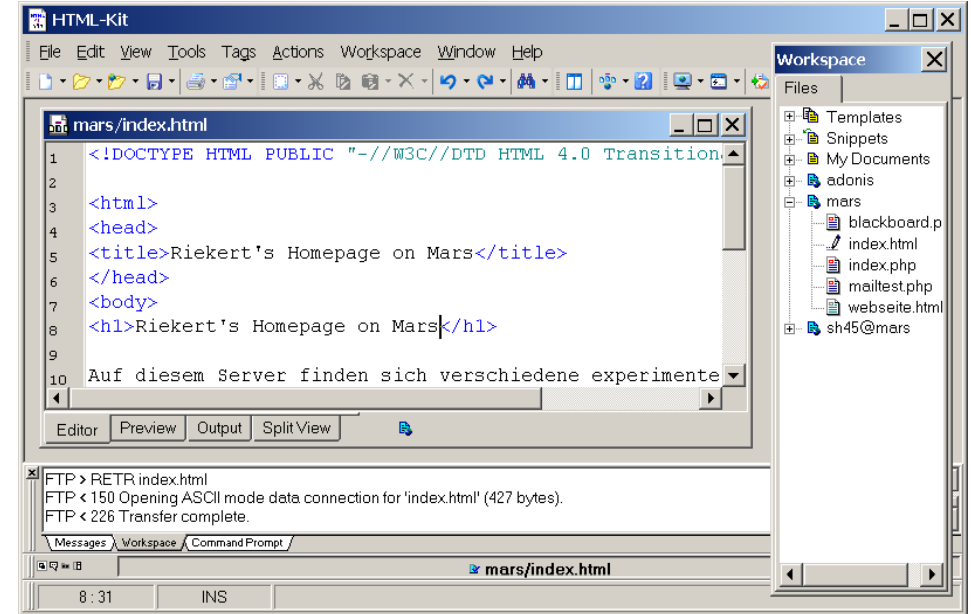


Zum Eingeben von Shell-Kommandos stellt man mit dem SSH Client (ggf. auch mit Telnet) eine Verbindung zum Server her.

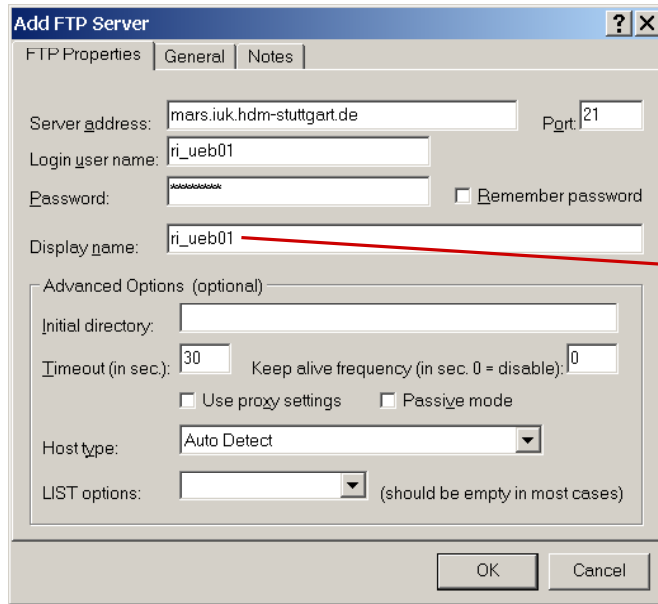
## SECURE FILE TRANSFER CLIENT



## HTML-KIT: HTML/PHP-EDITOR MIT INTEGRIERTEM FTP-SERVER



## HTML-KIT: NEUES FTP-VERZEICHNIS HINZUFÜGEN



Mit der Funktion Add FTP-Server (Menu Workspace) kann ein neues FTP-Verzeichnis hinzugefügt werden.

## HTML-KIT: BEARBEITEN VON DATEIEN AUF DEM WEBSERVER

- Mit den HTML-Kit kann man Dateien (meist HTML- oder PHP-Dateien) bearbeiten, die auf dem Webserver liegen.
- Das Dateisystem auf dem Server sieht man in einem Explorer-ähnlichen Fenster, dem „Workspace“.
- Dateien bearbeitet man, indem man sie im Workspace doppelklickt.
- Neue Dateien erzeugt man, indem man den betreffenden Ordner im Workspace selektiert und anschließend mit der rechten Maustaste *New – Create File – Blank HTML Page* selektiert
- Mit Hilfe der Download-Funktion (rechte Maustaste) können Dateien auf den Arbeitsplatz-PC heruntergeladen werden
- Mit Hilfe der Upload-Funktion können Dateien vom Arbeitsplatz-PC auf den Webserver hochgeladen werden.

## HTML-KIT: BEARBEITEN VON DATEIEN AUF DEM WEBSERVER

