

- Das World Wide Web
- Grundelemente von HTML
- Gestalten mit Stylesheets
- Multimedia-Daten
- Datenbanken und SQL
- Numerische Berechnungen mit Matlab
- Aufgaben
- Anhang

Peter Junglas 27.4.2012

Inhaltsverzeichnis

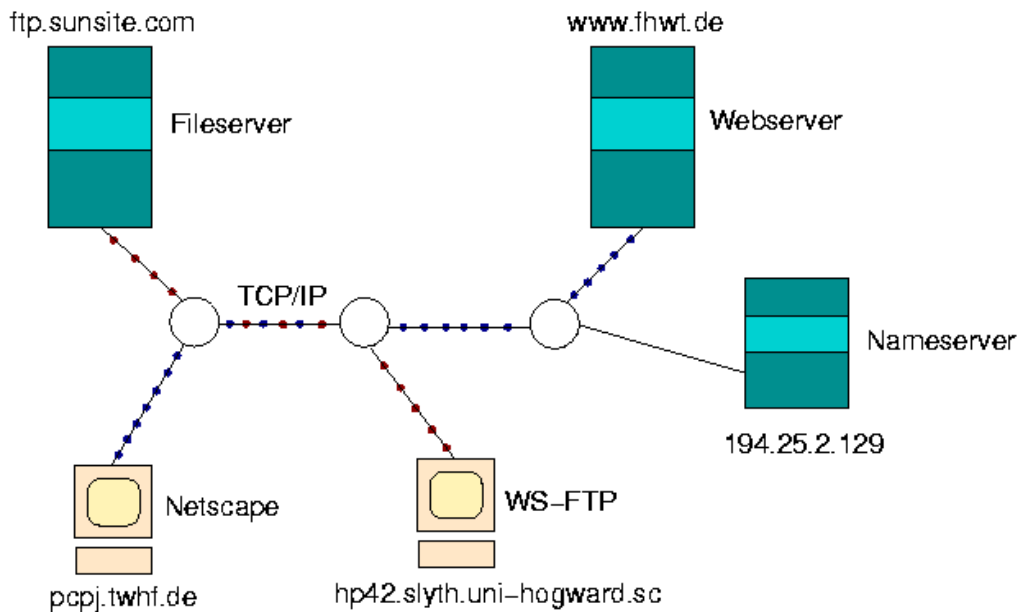
Übersicht

- Das World Wide Web
- Grundelemente von HTML
 - Allgemeine Regeln von HTML
 - Grundgerüst einer HTML-Seite
 - Text- und Absatzelemente
 - Verweise
 - Tabellen
 - Formulare
- Gestalten mit Stylesheets
- Multimedia-Daten
 - Einführung
 - Farben
 - Bilder
- Datenbanken und SQL
 - Datenbank-Managementsysteme
 - Aufbau von Tabellen
 - Grundlagen von SQL
 - Arbeiten mit Tabellen
 - Datenbankentwurf
 - Abfragen in Datenbanken
 - Einfache Abfragen
 - Abfragen über mehrere Tabellen
 - Komplizierte Abfragen
- Numerische Berechnungen mit Matlabc
 - Grundfunktionen von Matlab
 - Vektoren
 - Matrizen
 - Lösen von Gleichungen
 - Berechnung bestimmter Integrale
 - Integration von Differentialgleichungen
- Aufgaben
 - Aufgabe 1
 - Aufgabe 2
 - Aufgabe 3
 - Aufgabe 4
 - Aufgabe 5
 - Aufgabe 6
 - Lösung von Aufgabe 6
 - Aufgabe 7
 - Lösung von Aufgabe 7
 - Aufgabe 8
 - Lösung von Aufgabe 8
 - Aufgabe 9
 - Lösung von Aufgabe 9
 - Aufgabe 10
 - Lösung von Aufgabe 10

- Aufgabe 11
 - Lösung von Aufgabe 11
- Aufgabe 12
 - Lösung von Aufgabe 12
- Anhang
 - Literatur
 - Nachweise
 - HTML-Beispiele
 - HTML-Code von simple.html
 - HTML-Code von text.html
 - HTML-Code von block.html
 - HTML-Code von list.html
 - HTML-Code von table.html
 - HTML-Code von forms1.html
 - HTML-Code von forms2.html
 - HTML-Code von multimedia.html
 - HTML-Code von multimedia2.html
 - CSS-Beispiele
 - austen.css (cssdemo.html)
 - ohne Stylesheet (cssdemo01.html)
 - austen02.css (cssdemo02.html)
 - austen03.css (cssdemo03.html)
 - austen04.css (cssdemo04.html)
 - austen05.css (cssdemo05.html)
 - austen06.css (cssdemo06.html)
 - austen07.css (cssdemo07.html)
 - Multimedia-Beispiele
 - SQL-Beispiele
 - Beispiel-Datenbank
 - Ergebnisse der SQL-Abfragen für die Beispiel-Datenbank
 - Matlab-Beispiele
 - zeichneFunktion.m
 - demo1.dat
 - trapez1.m
 - trapez2.m
 - trapez.m
 - glocke.m
 - radioaktiv.m
 - radiokette.m
 - erzwungen.m
 - erzwungenp.m
 - exercise7.m
 - ex8_1.dat
 - exercise8.m
 - exercise9.m
 - exercise10.m
 - exercise10a.m
 - kreisprozess.dat
 - exercise11.m
 - exercise12.m
 - laminar.m

- turbulent.m
- freifall.m

- Internet:



Rechner

- identifiziert durch Nummern (**IP-Adressen**)
- stellen Informationen/Dienste zur Verfügung (**Server**)
- nehmen Dienste in Anspruch (**Clients**)

physikalisches Netzwerk

- Leitungen
- Vermittlungsknoten

Protokoll

- Sprache der Rechner
- wichtigstes im Internet: TCP/IP

Netzdienste

- Email
- Datei-Übertragung (FTP)
- Diskussionsforen (News)
- Arbeiten auf entfernten Rechnern (Telnet)
- vernetzte Informationen (WWW)
- div. Infrastruktur-Dienste (Namensdienst, Zeitdienst, ..)

- Protokolle:

Sprachen zur Kommunikation von Rechnern/Anwendungen auf verschiedenen Ebenen in Schichten übereinander

FTP
TCP
IP
Ethernet

FTP (**File Transfer Protocol**)

- erlaubt das Holen und Senden von Dateien
- enthält Kommandos zum Auflisten von Verzeichnissen
- ermöglicht die Identifikation eines Benutzers (Username und Password)

TCP (**Transmission Control Protocol**)

- schickt eine Nachricht zuverlässig an einen Zielrechner
- zerlegt sie dazu in kleine Pakete
- sammelt und sortiert eintreffende Paket am Zielrechner
- setzt daraus die gesamte Nachricht zusammen
- holt ggf. verlorene Pakete auf anderem Weg

IP (**Internet Protocol**)

- schickt ein Paket an einen Zielrechner
- sucht dazu einen Weg durch das Netzwerk

Ethernet

- schickt Daten auf einer Ethernet-Leitung zu einer anderen Netzwerkkarte
- sorgt dafür, dass immer nur eine Nachricht auf einmal auf dem "Draht" liegt

untere Schichten für Anwender in der Regel unsichtbar

- IP-Adresse:

weltweit eindeutige Nummer aus 4 Byte

oft geschrieben als 4 Dezimalzahlen, etwa

205.166.250.94

erste zwei oder drei Nummern oft für bestimmte Teilnetze/Institutionen

- Rechner-Namen (**Domain-Namen**):

Klartext-Namen, zur Vereinfachung für den Benutzer

aufgebaut als

RECHNERNAME . UNTERNETZ . NETZ . HAUPTNETZ

RECHNERNAME der eigentliche Rechner

UNTERNETZ ggf. eine (auch mehrfache) Untergliederung von NETZ

HAUPTNETZ eine globale Einteilung aller angeschlossenen Teilnetze

Hauptnetze (Top-Level-Domains) für Länder und einige Organisationen, z.B.

Top-Level-Domain	Bedeutung
de	Deutschland
fr	Frankreich
uk	Großbritannien
jp	Japan
edu	Hochschule/Forschungseinrichtung in den USA
gov	US-Regierung
com	Firmen (auch weltweit)

Umsetzung von Namen in IP-Adressen durch **Nameserver**

- **WWW (World Wide Web):**

spezieller Dienst im Internet

Aufgabe: möglichst einfacher Transport vernetzter Informationen

Kernelemente

- http-Protokoll (**Hypertext Transfer Protocol**) zur Übertragung von Daten von einem Rechner (**Web-Server**) zum Rechner des Anwenders
- Sprache HTML (**HyperText Markup Language**) zur Beschreibung des Inhalts eines Dokuments incl. Querbezügen zu anderen Dokumenten
- Programm (**Browser**) zum Anzeigen von HTML-Dokumenten und zum einfachen Verfolgen von Querverweisen

- **URL (Uniform Resource Locator):**

im ganzen Internet eindeutige Bezeichnung eines Dokuments

Grundaufbau

PROTOKOLL://RECHNERNAME/LOKALER_NAME

PROTOKOLL Verfahren zur Übertragung des Dokuments (http, ftp)

RECHNERNAME Domainname des Servers, der das Dokument bereithält

LOKALER_NAME Informationen über den Ort des Dokuments auf dem Server

Beispiel

`http://www.peter-junglas.de/fh/index.html`

- Allgemeine Regeln von HTML
- Grundgerüst einer HTML-Seite
- Text- und Absatzelemente
- Verweise
- Tabellen
- Formulare

Allgemeine Regeln von HTML



- HTML (**HyperText Markup Language**):

strukturierte Texte mit Verweisen und eingebetteten Multimedia-Elementen

beschreibt logische Bestandteile eines Dokuments z.B.

- Überschrift
- Absatz
- Tabelle

legt nicht das Aussehen fest

- z.B. Farben, Fonts, Einrückungen
- werden in **Style Sheets** vereinbart
- einige (veraltete) HTML-Elemente durchbrechen dieses Prinzip

verwendet reinen (ASCII-)Text

- leicht lesbar und verarbeitbar
- unabhängig von Rechnertyp und Betriebssystem
- allgemein offene Definition

- Ein erstes Beispiel:

[simple.html](#) (HTML-Quelle)

HTML-Text der Seite vom Browser aus anzeigen (Ansicht/Seiten-Quelltext)

logische Elemente der Seite

- Überschrift
- drei Absätze
- ein hervorgehobenes Element (*nicht*)
- ein Verweis
- ein eingefügtes Bild (Trennlinie)
- eine Adresse mit Email-Verweis

- Auszeichnungselemente (**Tags**):

Beispiele:

`body` Textteil des Dokuments

`p` Absatz

`em` betonter Text

`a` Verweis

beschreiben ein logisches Element

werden in spitze Klammern gesetzt

umschließen (meistens) das bezeichnete Element in der Form

```
<TAG>element</TAG>
```

können groß oder klein geschrieben werden

lassen sich verschachteln

- Attribute:

Beispiele

- `src="juhu.gif"`
- `href="http://heritage.stsci.edu/"`
- `alt="Ruler"`

enthalten zusätzliche Informationen zu einem Tag

stehen in der öffnenden Tag-Klammer

haben die Form

```
NAME="WERT"
```

- Stilangaben:

in der Beschreibungssprache CSS (**Cascading Style Sheets**)

können Tags als Style-Attribut mitgegeben werden

```
<p style="color:#ff2080;">
```

mehr dazu **später**

seit langem veraltet:

- Stil-Tags, etwa `` oder ``
- besondere Stilattribute, z.B. `bgcolor` und `text` bei `BODY`

- Dokumententext:

am einfachsten in ASCII

Zeilenumbrüche und mehrfache Leerzeichen werden ignoriert

Umlaute und Sonderzeichen in der Form

```
&NAME;
```

einige Beispiele

Sonderzeichen	HTML-Bezeichnung
Ä Ö Ü	<code>&Auml;</code> <code>&Ouml;</code> <code>&Uuml;</code>
ä ö ü ß	<code>&auml;</code> <code>&ouml;</code> <code>&uuml;</code> <code>&szlig;</code>
æ å é	<code>&aelig;</code> <code>&aring;</code> <code>&eacute;</code>
< > & "	<code>&lt;</code> <code>&gt;</code> <code>&amp;</code> <code>&quot;</code>
geschütztes Leerzeichen	<code>&nbsp;</code>

alternativ Angabe des Zeichensatzes im Header

- `<meta http-equiv="content-type" content="text/html; charset=ISO-8859-15">`
- Zeichensatz ISO-8859-15 enthält die Sonderzeichen vieler westeuropäischer Sprachen
- deutsche Umlaute können dann direkt eingegeben werden

- Zeichensatz UTF-8 enthält Zeichen fast aller Sprachen (incl. chinesisch)

Grundgerüst einer HTML-Seite



- Schema jeder HTML-Datei:

Aufbau gemäß HTML-Standard

```
DOKUMENTENTYP
<html>
  <head>
    HEADER
  </head>
  <body>
    TEXTKÖRPER
  </body>
</html>
```

- Dokumententyp

verwendete HTML-Version

benutzter Standard

Sprache der Tags (immer en = englisch)

Beispiel

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
```

- Header mit Informationen über das Dokument, z.B.

Titel

verwendeter Zeichensatz (s.o.)

verwendetes Stylesheet

Informationen für Suchmaschinen

- Textkörper

eigentlicher Inhalt des Dokuments

enthält nach striktem HTML-Standard keinen Text ohne umschließende Tags

- Kontrolle des HTML-Codes

Browser sind sehr fehlertolerant

Darstellung bei Fehlern stark browser-abhängig

Kontrolle auf Standard unter <http://validator.w3.org/>

- Aufgaben:

Aufgabe 1

Text- und Absatzelemente

- Textelemente:

Beispiel (HTML-Quelle)

beschreiben logische Bedeutung eines Textteils

Beispiele

Tag	Bedeutung	Darstellung (meistens)
EM	betont	kursiv
CITE	Zitat	engerückt, kursiv
CODE	Programmtext	nicht-proportionaler Font
SUB	Index	kleiner, tiefgestellt
SUP	Exponent	kleiner, hochgestellt

- Einbinden von Bildern:

mit IMG-Tag, ohne Endtag

```

```

wird wie ein großer Buchstabe behandelt (also als Textelement)

mögliche Bildquellen

- Datei im gleichen Verzeichnis ("photo.jpg")
- Datei in anderem Verzeichnis ("../bilder/photo.gif")
- Datei direkt aus dem Netz ("http://www.bilder.de/photo.jpg")
- Copyrights beachten!

width, height

- angezeigte Größe des Bildes in Pixel
- u.U. verzerrte Darstellung
- Ladezeit nur von der Originalgröße abhängig!

alt

- alternativer Text (Vorschau oder für Textbrowser)
- seit HTML 4 (strict) obligatorisch!

- Absatzelemente:

Beispiel (HTML-Quelle)

beschreiben Elemente zur Gliederung

Grundelement <P> (**Paragraph**) normalerweise durch Zeilenumbruch und Leerzeile dargestellt

Positionierung (links-, rechtsbündig, zentriert) möglich

- mit (veraltetem) Attribut

```
align="left" (oder "right", "center")
```

- mit Style-Attribut

`style="text-align:right;"`

Zeilenumbruch erzwingen mit `
` (ohne End-Tag)

vorformatierter Text mit `<PRE>`

Überschriften mit `<H1>`, `<H2>` ... `<H7>`

- Listen:

Beispiel (HTML-Quelle)

ganze Liste

- unnummeriert ``
- nummeriert ``

Elemente der Liste

- jeweils mit `` eingeleitet
- Endtag `` kann entfallen

Darstellung

- eingerückt
- jedes Element in einzelner Zeile
- vor jedem Element Listensymbol (Kreis, Kästchen) oder Nummer

Schachtelung

- Liste darf nicht direkt in einem Absatz `<P>` stehen
- jedes Listenelement darf Absätze enthalten

Stilarten

- Listensymbol oder Art der Nummerierung durch Style-Element `list-style-type`
- Werte für ``: `disc`, `circle`, `square`, `none`
- Werte für ``: `decimal`, `lower-roman`, `upper-roman`, `lower-alpha`, `upper-alpha`, `none`

- Aufbau eines Verweises:

mit `<A>`-Tag (**Anchor**)

```
<a href="Verweisziel">Verweistext</a>
```

Darstellung

- normalerweise Verweistext unterstrichen und anklickbar
- kann durch Stilangaben geändert werden

Verweisziel

- WWW-Adresse (`http://www.nasa.gov/`)
- lokaler Dateiname (`../bilder/image.gif`)
- Email-Adresse (`mailto:papst@vatikan.va`)
- URLs mit anderen Protokollen (`ftp, telnet, news`)

- Einsatz von Verweisen:

Navigation innerhalb einer Homepage

- Blättern, zur Hauptseite, Inhaltsverzeichnis,...
- am besten immer gleichartig
- häufig mit Frames oder Javascript-Techniken

interne oder externe Querverweise

Kontaktaufnahme (Email-Verweis)

wichtig:

- aussagekräftiger Verweistext
- Groß-/Kleinschreibung bei Dateinamen beachten
- immer '/' als Trennzeichen für Verzeichnisse, nicht '\'

- Grundstruktur:

<TABLE> umschließt ganze Tabelle

<TR> bezeichnet eine Zeile (**Table Row**)

<TD> umschließt ein einzelnes Feld (Spalte) in einer Zeile (**Table Data**)

Beispiel

```
<table>
  <tr>
    <td> 1. Zeile, 1. Spalte</td>
    <td> 1. Zeile, 2. Spalte</td>
  </tr>
  <tr>
    <td> 2. Zeile, 1. Spalte</td>
    <td> 2. Zeile, 2. Spalte</td>
  </tr>
</table>
```

optional <TH> (**Table Header**) statt <TD>

- für erste Zeile (Spaltenüberschriften)
- meistens fett dargestellt

- Gestaltung von Tabellen:

Beispiel (HTML-Quelle)

Rahmenbreite festlegen

```
<table border="4">
```

Abstand der Elemente von den Gitterlinien

```
<table cellpadding="20">
```

Breite der Tabelle

- automatisch nach Textinhalt
- Größe in Pixel festlegbar durch

```
<table width="100">
```

- Größe relativ zum Browser-Fenster mit

```
<table width="60%">
```

- width auch für einzelne Spalten am besten mit `colgroup`
- Größe gilt nur, wenn Platz für Tabelleninhalt ausreicht

Höhe einer Zeile

alt: <tr height="200">

neu: <tr style="height:200px;">

horizontale Ausrichtung

- `align="left"` ("`right`", "`center`")
- bei `TD` und `TH` für ein Element
- bei `TR` für eine ganze Zeile

vertikale Ausrichtung

- `valign="top"` ("`middle`", "`bottom`")
- wie `align` bei `TD`, `TH` und `TR`

Zentrieren einer ganzen Tabelle auf einer Seite

- mit Stilelementen `margin-left:auto; margin-right:auto;`
- beachte entsprechenden [Selfhtml-Artikel](#)

- Einsatz von Formularen:

grundlegende Eingabe-Möglichkeit in HTML

enthält Eingabeelemente sowie beliebige weitere HTML-Elemente zur Gestaltung

Beispiel [forms1.html](#) ([HTML-Quelle](#))

- Definition eines Formulars:

Tag `<form>` umschließt ganzes Formular

`action`-Parameter: Wohin mit den Daten?

mögliche Aktionen

- als Email verschicken (`mailto:peter@zuhause.de`)
- an auswertende Seite (CGI, PHP) übergeben

`method`-Parameter: Wie werden die Daten verschickt?

- **get**: kodiert in der URL
z.B. `http://peter.de/echo.php?user=klaus&password=juhu`
auf kleine Datenmengen beschränkt
- **post**: als eigener Datenblock
wird i.f. ausschließlich verwendet

Button zum Abschicken (`submit`) löst Aktion aus

- Einzeilige Text-Eingabefelder:

definiert durch Tag `<input>`

kein Ende-Tag

Parameter `size`

- Länge des Feldes in Zeichen

Parameter `name`

- Name des Feldes
- wichtig für auswertendes Programm

Parameter `value`

- zur Vorbelegung des Textes

Passwort-Felder

- zeigen eingegebene Zeichen nur als * an
- erzeugt mit `type="password"`

- Buttons zum Abschicken und Zurücksetzen:

ebenfalls mit `<input>` erzeugt

Beschriftung mit Parameter `value`

Abschicken aller Formular-Daten

- `type="submit"`
- muss in einem Formular auftauchen (sinnvollerweise)

Zurücksetzen aller Formular-Daten

- `type="reset"`
- sollte in einem Formular auftauchen

- Ein komplizierteres Beispiel:

[forms2.html](#) (HTML-Quelle)

enthält die wichtigsten Formular-Elemente

übersichtliche Anordnung

- Auswahllisten:

Listen zur Auswahl eines oder mehrerer Elemente

definiert durch `<select> ... </select>`

Listeneinträge mit `<option>` Listentext

Parameter `size`

- Zahl der gleichzeitig sichtbaren Elemente
- Rest ggf. über Scrollbar erreichbar
- `size = 1` → aufklappbare Liste (**drop-down**)

Parameter `multiple`

- mehr als ein Element darf ausgewählt werden
- Default: nur ein Element

Parameter `name`

- Name für das auswertende Programm
- bei `multiple` für Auswertung mit PHP `[]` (= mehrere Werte) anhängen

- Checkboxes:

Knöpfe zur (Mehrfach-)Auswahl

definiert durch `<input>`-Tag und Attribut `type="checkbox"`

Parameter `name`

- Name für das auswertende Programm
- für Auswertung mit PHP `[]` anhängen
- alle Checkboxes mit gleichem Namen bilden eine Gruppe

Parameter `value`

- Wert, der für gedrückten Knopf übergeben wird

- Radiobuttons:

Menge von Knöpfen, von denen genau einer gedrückt ist (außer evtl. am Anfang)

definiert durch `<input>`-Tag und Attribut `type="radio"`

Parameter `name` und `value` wie bei Checkboxes

Parameter `name` ohne `[]`

- Textfelder:

Eingabefeld für mehrzeilige Texte

definiert durch `<textarea></textarea>`

erzeugt eigenen Absatz

Parameter `rows` und `cols`

- Zahl der Zeilen und Spalten
- bei größerem Eingabetext automatisch Scrollbars

- Bemerkungen zur Formatierung:

Anordnung der Elemente durch normale HTML-Anweisungen

häufiges Hilfsmittel: unsichtbare Tabellen (ohne Rand)

andere Möglichkeit z.B. Gruppierung mit horizontalen Linien (`<hr>`)

- Aufgaben:

Aufgabe 2

- Stylesheet:

Definition von Formatierungseigenschaften

eigene Sprache CSS als Ergänzung zu HTML

W3C-Standard

i. a. gute Unterstützung durch die Browser

- Internet-Explorer hinkt deutlich hinterher

Grundstruktur eines Stilelements

- `Selektor { Eigenschaft: Wert; }`
- Selektor gibt an, auf welche HTML-Elemente der Stil angewendet wird

hier nicht behandelt u.a.

- Details des Box-Modells zur pixelgenauen Formatierung
- Layouts mit CSS (ohne Tabellen)

- Einbinden von Stil-Elementen:

direkt an einen HTML-Tag

- `<TAG style="font-size: 10pt;">`
- unschön, da Inhalt und Stil vermischt werden

in den Header

```
<style type="text/css">
  /* Das ist ein Kommentar */
  li {font-size: 10pt;} /* kleine Fontgröße für Listenelemente */
</style>
```

in eine eigene Datei

- `<link rel="stylesheet" type="text/css" href="meinstil.css">`
- kann für mehrere HTML-Dateien verwendet werden
- → einheitliches Aussehen

- Adressieren mit dem Selektor:

ein oder mehrere Tags

- `li, p, h1 {font-size: 12pt;}`
- gilt für alle ``, `<p>`, `<h1>`

verschachtelte Tags

- `td li {color: red;}`
- gilt nur für `` innerhalb von `<td>`
- kann auch tief verschachtelt sein

Klassen

- `h3.menue {border: solid #908070;}`
- wirkt auf alle Elemente `<h3 class="menue">`

Pseudoklassen

- `a:link {color: #660000;}`
- `a:visited {color: #990000;}`

Individualformate

- `td#portrait {width: 253px;}`
- wirkt auf das Element `<td id="portrait">`
- eine id darf nur einmal im Dokument vorkommen

beliebig kombiniert

- `tr#footer td.odd { /* ... */ }`
- wirkt auf ein `<td class="odd">` innerhalb eines `<tr id="footer">`

• Einige Eigenschaften:

Fonts

- `font-family`, z.B. 'Times New Roman', 'Nimbus Roman No9', serif
- immer Alternativen angeben (als letztes serif, sans-serif o.ä.)
- `font-size`, `font-weight`

Masseinheiten für Längen

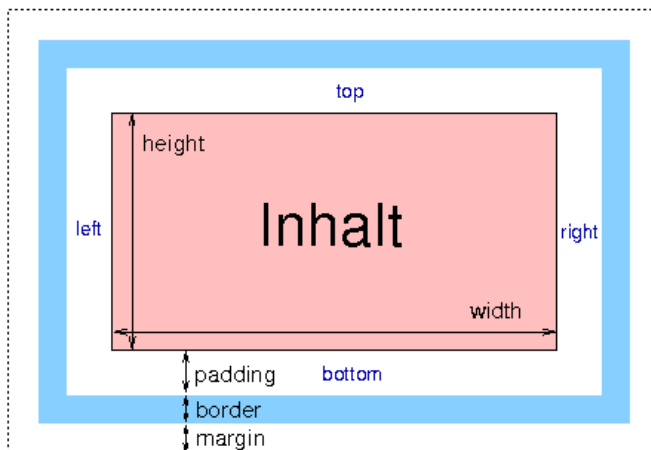
px	Pixel
pt	Typographischer Punkt
mm	Millimeter
em	Breite des 'M' im aktuellen Font
%	Prozent

Farben

- `color`, `background-color`
- Farbnamen (black, blue), RGB-Werte #RRGGBB

Ränder:

- `padding`, `border-width`, `margin`,



- einzeln angebar (`margin-left: 10px;`)
- ein Wert für alle (`border-width: 3mm;`)
- alle vier Werte auf einmal (`padding: 2px 3px 2px 5px;`)
- Reihenfolge der vier Werte: top, right, bottom, left ("TRouBLE")

Textumfluss (um einen Block herum)

- `float:left` bzw. `float:right`
- `clear:left` bzw. `clear:right` startet Bereich unter dem Float-Block

• Beispielseite:

[cssdemo.html](#) mit Stylesheet [austen.css](#)

typische Aufteilung

- Kopfzeile
- Fusszeile
- globale Navigationsleiste oben
- lokale Navigationsmenü links
- Service-Spalte rechts
- Hauptbereich mit normalem Text + Info-Kasten

Seite [ohne CSS](#)

Struktur der Seite

- Tabellen rot, Listen blau, Überschriften grün, `div/span` schwarz
- erzeugt durch einfaches Stylesheet [austen02.css](#)

Attribut `cellspacing` in `table`

- ist in HTML strict erlaubt
- durch CSS-Elemente (u.a. `border-spacing`) ersetzbar
- `border-spacing` in IE7 immer noch nicht unterstützt

- Allgemeine Formate ([austen03.css](#)):

spezieller Font (Verdana, falls möglich, aber jedenfalls ohne Serifen)

Standardfarben schwarz/weiß (überschreibt ggf. Browser-Defaults)

keinen Platz verschenken bei Body, Tabellen und -feldern

Tabellen auf 100% Breite

Inhalt der Tabellenfelder nach oben (statt mittig)

Farben der Links an verwendetes Farbschema anpassen

Überschrift im Textteil formatieren

- Text (Größe, Gewicht, Farbe), Abstand und unteren Rand

[Ergebnis](#)

- Gestaltung des Kopfbereichs ([austen04.css](#)):

Id's vergeben zur eindeutigen Kennzeichnung von Elementen

- `title` (Kopfbereich links, Titel)
- `portrait` (Kopfbereich rechts, Bild)

Titelbereich

- vertikal mittig
- Hintergrundbild + Farbe
- Fontart und -größe
- Abstand nach links

Bild

- `td`-Feld an Bild anpassen (damit linke Spalte den Rest bekommt)
- Bild nach unten rücken (sonst Platz für Unterlänge)

[Ergebnis](#)

- Gestaltung der Navigationsleiste ([austen05.css](#)):

Ids einführen

- `content-top` (Navigationsleiste)
- `content-left` (Navigationsleiste, links)
- `content-links` (Navigationsleiste, Verweise)

Formatierungen

- Fonts und Farben
- vertikal mittig ausgerichtet
- Abstand zwischen den Links

`padding` bei `a`, nicht bei `td`

Ergebnis

- Gestaltung von Haupttext, Randspalten und Fußzeile ([austen06.css](#)):

Ids einführen

- `leftside` (Menü links)
- `content` (Haupttext)
- `rightside` (Service-Spalte rechts)
- `footer` (Fußzeile)
- `feedback` (Footer, Kontakt)
- `author` (Footer, Name)
- `date` (Footer, Änderungsdatum)

linke Spalte

- Breite und Hintergrundfarbe
- Art der gesamten Liste (`list-style-type`, `list-style-position`)
- Font, Unterstreichung (`border-bottom`) und Platz der einzelnen Elemente

Haupttext

- Abstand
- Font

rechte Spalte

- Breite
- Font und Padding von Tabellen-Elementen
- Spaltenüberschrift (`h3`)

Fußzeile

- Text mittig, klein
- oben Rand, etwas Platz
- links Farbe an Spalte angepasst
- `date/author`-Selektoren kompliziert, um Vorrang-Reihenfolge zu garantieren

Ergebnis

- Besonderheiten ([austen07.css](#)):

`cite` (Zitat rechts)

- Textblock mit Rahmen
- Überschrift dicht am Text

`class` statt `id`, wenn etwas mehrmals pro Seite vorkommen kann

Klasse `infobox`

- eigene Box mit fester Breite und Abständen

- wird links vom normalen Text umflossen

Klasse `rightside-head` zur Gestaltung der Überschriften rechts

Klassen `even/odd` zur Unterscheidung von geraden/ungeraden Tabellenzeilen

Klasse `preis` zur Heraushebung von Preisen

Ergebnis

- Aufgaben:

Aufgabe 3

📍 Einführung

📍 Farben

📍 Bilder

- Wichtige Multimedia-Typen:

Bilder (JPG, GIF, PNG, TIFF)

Klänge (WAV, MP3, MIDI)

Videos (MPEG, AVI, Quicktime, OggTheora)

Animationen und Programme (Java-Applet, Flash)

3D-Modelle (VRML)

- Anzeige im Browser:

[Beispiel](#) mit MPEG-Video ([HTML-Quelle](#))

grundsätzlich drei Möglichkeiten

- Darstellung durch den Browser selber (einige Bildformate)
- Anzeige im Browser mit Plugins
- Anzeige in externem Fenster durch Hilfsprogramme

einfügen in HTML mit Tag `<object>`

- erst ab HTML 4.0
- Bedeutung der Attribute im Internet-Explorer komplett "nicht-kanonisch", d. h. inkompatibel.

noch häufig zu sehen: mit (veraltetem) Tag `<embed>`

einfacher in HTML 5: [Beispiel](#) mit Ogg Theora ([HTML-Quelle](#))

- MIME-Typ:

Information über die Art des Multimedia-Elements

im Tag angegeben oder durch Datei-Endung festgelegt

legt die Interpretation der Bytes fest

- so wie der Zeichensatz bei Textdateien
- z.B. bedeutet das Byte `0xe4` ä (in ISO8859-1, westeuropäisch), φ (in ISO8859-5, kyrillisch) oder δ (in ISO8859-7, griechisch)

einige Beispiele

Typ	Beschreibung	Endung
image/png	PNG image	png
image/tiff	TIFF image	tiff, tif
audio/wav	Microsoft wave file	wav
audio/mpeg3	MPEG audio	mp3
audio/midi	MIDI audio file	midi, mid
video/mpeg	MPEG animation	mpeg, mpg, mpe
video/ogg	Ogg Theora Video	ogg
video/quicktime	Quicktime animation	mov, qt
video/msvideo	AVI animation	avi

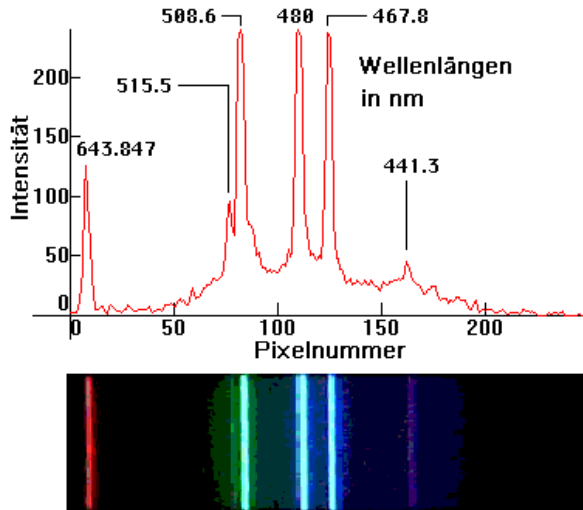
[offizielle Liste](#)

- Physikalische Basis:

Lichtwellen verschiedener Wellenlängen werden von Oberflächen reflektiert oder gestreut

Anteil der verschiedenen Wellenlängen kann als Spektrum dargestellt werden

Beispiel: Spektrum einer Niederdruck-Cadmiumdampfampe

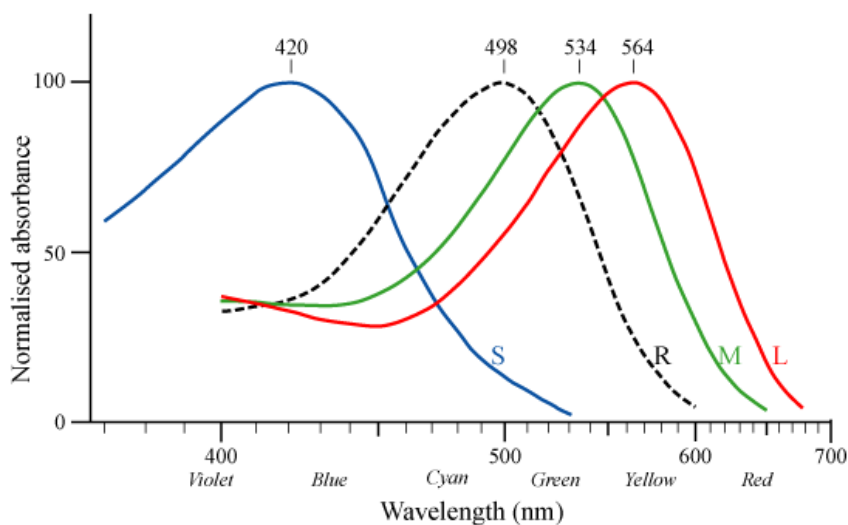


- Physiologische Basis:

zwei Typen von lichtempfindlichen Zellen in der Netzhaut

- **Stäbchen:** besonders empfindlich, keine Farbunterscheidung, vor allem zum Dämmerungssehen
- **Zäpfchen:** drei Arten mit unterschiedlichen spektralen Empfindlichkeiten, als Rot-, Grün- und Blau-Rezeptor bezeichnet

Empfindlichkeitskurven der Zäpfchen



Menschen unterscheiden etwa 1 Million Farben

bei Säugetieren häufig nur 2 Zapfentypen, bei anderen Wirbeltieren und Insekten häufig 4 Typen

- Farbmodelle:

Farbmodell = Methode, Farben als Zahlenwerte abzuspeichern
 meistens mit drei Parametern (\cong Zahl der verschiedenen Zäpfchenarten!)

RGB-Modell

- aus der additiven Farbmischung (Bildröhre)
- Helligkeiten jeweils für Rot, Grün und Blau



- häufig jeweils 8 bit (0 .. 255) für die drei Werte
- \rightarrow insgesamt $256^3 = 16$ Mio. Farben



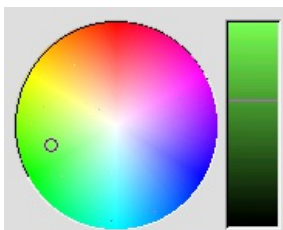
CMYK-Modell

- Basis der subtraktiven Farbmischung (Drucker)
- Grundfarben: Cyan, Magenta, Yellow, Black (K = "Key" aus der Drucktechnik)
- Schwarz theoretisch aus CMY mischbar, ist aber in der Praxis farbstichig, daher eigener Kanal



HSB-Modell

- aus dem Farbkreis abgeleitet



- beschrieben durch

Parameter	Bedeutung	Werte
Hue	Winkel im Farbkreis	0 .. 360
Saturation	Farbsättigung	0 .. 255
Brightness	Helligkeit	0 .. 255

YCbCr-Modell

- unterscheidet Grundhelligkeit Y, Anteil der Farbe in blau/gelb-Richtung Cb und Anteil in rot/türkis-Richtung Cr
- Auge viel empfindlicher für Helligkeitsvariationen als für Änderungen in Cb oder Cr



- Basis für Bildkompression: Y mit mehr Bit abspeichern als Cb, Cr
- verwendet bei digitalem Fernsehen, JPEG-Bildformat, MPEG-Videoformaten (z. B. für DVD)

- Farbtabellen:

Tabelle fester Länge mit Auswahl an Farben

nützlich bei beschränkter Farbanzahl (ältere Grafikkarten)

Farben der Tabelle oft an das Bild anpassbar, z.B. 256 freie Farben bei GIF-Bild

Farbe angeben durch Nummer in der Tabelle → Reduktion des Speicherbedarfs

Beispiel GIF-Bild

- Farbe in RGB \cong 24 Bit
- stattdessen Nummer in der Tabelle \cong 8 Bit
- also Platzreduktion um Faktor 3
- dazu kommt noch der Platz für die Tabelle selbst (256 x 24 Bit)

- Digitalisierung von Bildern:

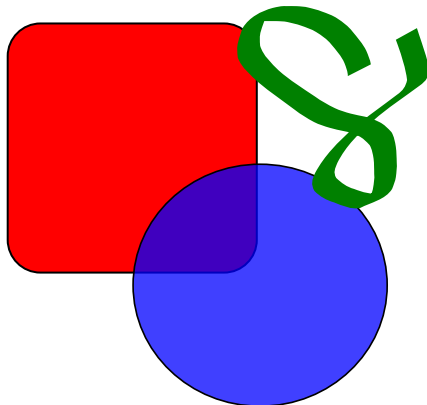
Zerlegung in rechteckige Bereiche (**Pixel**) mit einem mittleren Farbwert
technisch realisiert mit Scanner oder Digitalkamera

einfachste Abspeicherung

- RGB-Farbwerte für jedes Pixel
- alle Pixel hintereinander
- Header mit Breite und Höhe
- wird verwendet beim TIFF-Format (u.a.)

Alternative für Graphiken: Vektorgrafik

- Objekte (Polygonzüge, Kreise, Splines etc.) und ihre Eigenschaften (z. B. Farbe, Strichdicke, Füllungsmuster) werden abgespeichert
- Formate z.B. PS (Postscript), CDR (Corel Draw)
- Standardformat im Internet: **SVG** (Scalable Vector Graphics)
- wird von allen aktuellen Browsern außer Internet Explorer direkt unterstützt
- Plugin für Internet Explorer verfügbar
- Vorteil: optimale Darstellung in jeder Auflösung



- GIF-Format:

von CompuServe eingeführt

gut geeignet für Graphiken und Zeichnungen, weniger für Fotos

reduziert Farben auf 256 Werte aus freier Farbtabelle

folgende Komprimierung nicht verlustbehaftet

enthaltenen Daten

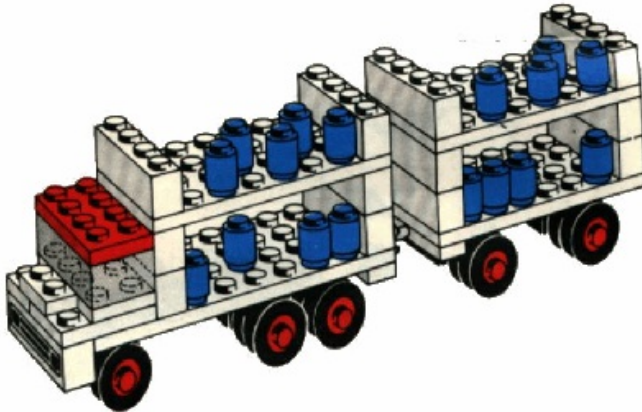
- Verwaltungs-Informationen (Größe des Bilds, Kommentare etc)
- Farben der Farbtabelle
- Bild: für jeden Punkt die Farbnummer

Prinzip der (LZW-)Komprimierung

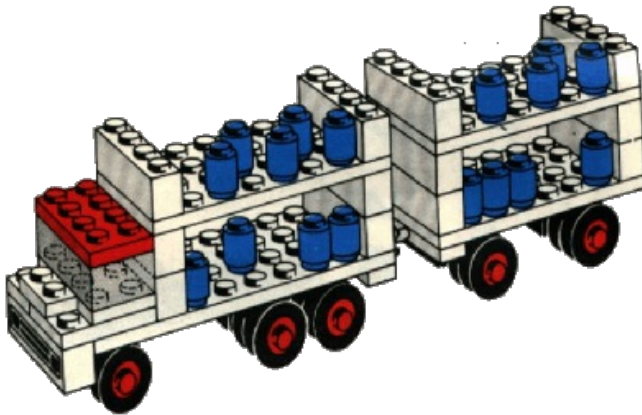
- häufig auftauchende Muster suchen

- Muster in extra Tabelle abspeichern
- im Bild statt Muster nur Nummer in der Mustertabelle

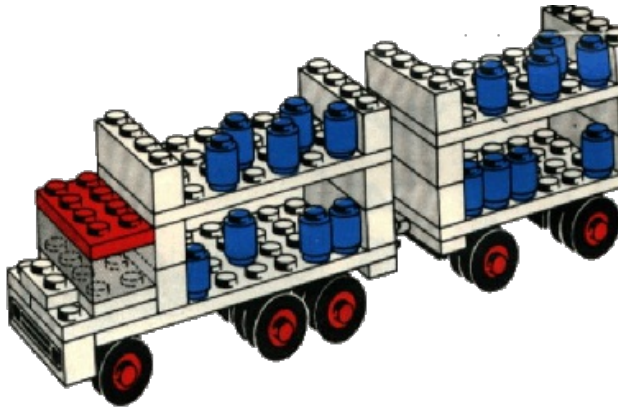
Spezialitäten



Interlacing = Bild wird erst grob geladen, dann genau



Transparenz = Durchscheinen des Hintergrunds



Animation = Folge von Bildern in einer Datei als Filmchen

- JPEG-Format:

für Bilder mit Echtfarben

besonders geeignet für Fotos und Farbverläufe

Basis der MPEG-Filmkodierung

verlustbehaftete Komprimierung

- nutzt Eigenschaften des Auges
- bei hoher Komprimierung sichtbare Fehler ("Artefakte")



(Qualität 75%, 15.4 kB)



(Qualität 20%, 5.7 kB)



(Qualität 2%, 1.8 kB)

Prinzip der Komprimierung

- Wandeln in YCbCr-Farben, Y mit mehr Bit abspeichern als CbCr
- Bild zerlegen in 8x8-Blöcke
- in jedem Block kleine "hochfrequente" Anteile weglassen
- Rest mit verlustfreier Methode komprimieren

- PNG-Format:

Nachfolger von GIF

kann größere Farbtabelle und Echtfarben

verlustfreie Komprimierung

erlaubt Transparenz

- Aufgaben:

Aufgabe 4

- Datenbank-Managementsysteme
- Aufbau von Tabellen
- Grundlagen von SQL
- Arbeiten mit Tabellen
- Datenbankentwurf
- Abfragen in Datenbanken

Datenbank-Managementsysteme



- Datenbank:

Sammlung von Daten an zentraler Stelle
verschiedene Organisationsformen
relationale Datenbank $\hat{=}$ Menge von Tabellen

- Datenbank-Managementsystem (**DBMS**):

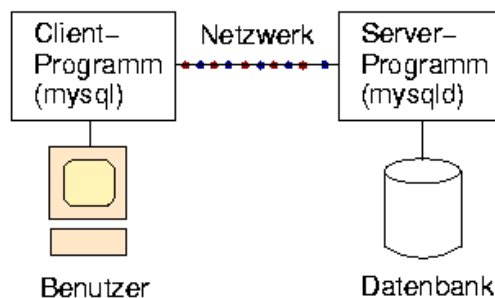
Programm zur Verwaltung von (großen) Datenbanken

viele Systeme auf dem Markt, z.B.

- Oracle
- Sybase
- Ingres
- DB2
- MySQL (Open Source, im Kurs verwendet)

erlaubt einheitlichen, netzwerkweiten Zugriff auf Daten

- nach außen immer gleiches Vorgehen für alle Daten
- internes Speicherschema versteckt
- Client-Server-Architektur



gewährleistet Konsistenz der Daten, z.B. durch

- Typen mit erlaubten Bereichen (Wert für Datumsfeld muss gültig sein)
- Kontrolle beim Einfügen von Daten (Kundennummer muss zu einem Kunden gehören)
- Kontrolle beim Löschen von Daten (Kunde löschen \rightarrow sein Konto wird auch gelöscht)

garantiert Sicherheit und Vertraulichkeit

- eigene Benutzerverwaltung
- Anmelden etwa mit

```
mysql -h meinrechner -u meinuser -p <RET>  
meinpasswort
```

- ausgefeiltes System von Zugriffsrechten
- z.B. Rechte zum Erstellen, Ändern, Ansehen oder Löschen von Daten

- SQL ("**Structured Query Language**"):

Sprache zur Definition und Manipulation von Daten

Sprache zur Definition und Manipulation von Daten

beschreibt Aufgabe, nicht den Lösungsweg (nicht-prozedural)

Syntax an normales Englisch angelehnt

standardisiert durch ISO

- grundlegende Version: SQL92 (oder SQL2)
- SQL:1999 (SQL3) u.a. mit objektorientierten Erweiterungen
- neu: SQL:2003 u.a. mit XML-Unterstützung

Aufbau von Tabellen



- Ein erstes Beispiel:

autor	titel	seiten	verlag	jahr	auflage	isbn
Hering	Physik für Ingenieure	744	Springer	1999	7	3540661352
Stöcker	Taschenbuch der Physik	1092	Harri Deutsch	2000	4	3817116284
Halliday	Fundamentals of Physics Extended	1198	Wiley	1997	5	471105597
Pitka	Physik - Der Grundkurs	426	Harri Deutsch	1999	1	3817115768

- Datensatz:

Zeile in einer Tabelle, z.B.

autor	titel	seiten	verlag	jahr	auflage	isbn
Stöcker	Taschenbuch der Physik	1092	Harri Deutsch	2000	4	3817116284

Beschreibung eines Objekts (**Entität**) durch Werte für die interessierenden Eigenschaften (**Attribute**)

Attribute haben Wertebereiche (**Domänen**), z.B.

Attribut	Bereich
autor, titel, verlag	Zeichenkette
seiten	ganze Zahl
jahr	gültige Jahreszahl
auflage	kleine ganze Zahl
isbn	10stellige ganze Zahl

spezieller Wert **NULL** = Hilfwert bei unbekanntem oder undefinierten Wert

- Tabelle:

Menge von Datensätzen

Reihenfolge der Datensätze beliebig

keine doppelten Datensätze (in der Regel)

- Schlüssel:

jeder Datensatz beschreibt ein festgelegtes Objekt (ein bestimmtes Buch, eine bestimmte Person, Produkt, etc)

ein oder mehrere Attribute (**Schlüssel**) legen Objekt fest

Beispiele:

Objektart	Schlüssel
Buch	ISBN
Auto	Kennzeichen
Vorlesungsraum	Gebäudenummer + Raumnummer

Anforderungen an Schlüssel

- Eindeutigkeit (verschiedene Objekte haben verschiedene Schlüssel)
- Definiertheit (jedes Objekt hat einen Schlüsselwert != NULL)
- Konstanz (jedes Objekt hat immer denselben Schlüsselwert)

natürliche Schlüssel häufig ungeeignet

z.B. Vorname und Name

- nicht eindeutig (Hans Müller)
- nicht definiert (direkt nach der Geburt)
- nicht konstant (Namensänderungen, etwa bei Hochzeit)

Aushilfe: künstliche Schlüssel wie ISBN, Kundennummer, Matrikelnummer etc.

Grundlagen von SQL



- Allgemeine Syntax:

Schlüsselwörter unabhängig von Klein-/Großschreibung

bei Namen für Tabellen, Attribute etc. Klein-/Großschreibung relevant!

Befehle über mehrere Zeilen möglich

Ende eines Befehls mit ;

i.f. Großbuchstaben für Schlüsselwörter

- Datenbank in SQL:

erzeugen mit SQL-Kommando

```
CREATE DATABASE datenbank;
```

danach zum Benutzen "in Datenbank wechseln" mit

```
USE datenbank;
```

erspart bei Tabellen Angabe des vollen Namens `datenbank.tabelle`

Liste aller vorhandenen Tabellen in der ausgewählten Datenbank:

```
SHOW TABLES;
```

Liste aller vorhandenen Datenbanken mit

```
SHOW DATABASES;
```

- Datentypen für Attribute:

viele verschiedene Typen definiert

einige Beispiele

Name	Bedeutung
CHAR (n)	Zeichenkette mit genau n Zeichen
VARCHAR (n)	Zeichenkette variabler Länge mit höchstens n Zeichen
INTEGER	ganze Zahl mit Standardbereich
DECIMAL (p, n)	Zahl mit p Stellen und n Nachkommastellen
DATE	Datum (Format "YYYY-MM-DD")
YEAR	Jahreszahl

Arbeiten mit Tabellen



- Erzeugen von Tabellen:

Syntax

```
CREATE TABLE tabelle (feld1 typ1, feld2 typ2, ...);
```

zusätzliche Parameter bei CREATE

NOT NULL	darf nicht den Wert NULL haben
UNIQUE	Werte für alle Datensätze verschieden
PRIMARY KEY	zentraler Zugriffsschlüssel erfordert NOT NULL impliziert UNIQUE
AUTO_INCREMENT	automatisches Hochzählen des Index für PRIMARY KEY nur bei Integer-Datentypen

Beispiel

```
CREATE TABLE Buch
  (autor      varchar(40),
  titel      varchar(255),
  isbn       decimal(10,0) NOT NULL,
  PRIMARY KEY(isbn)
  );
```

- Löschen von Tabellen:

Syntax

```
DROP TABLE tabelle;
```

- Information über Tabellen:

Syntax

```
SHOW COLUMNS FROM tabelle;
```

andere Form

```
DESCRIBE tabelle;
```

- Einfügen von Daten:

Syntax

```
INSERT INTO tabelle (feld1, feld2 ..)
  VALUES(wert1, wert2 ...);
```

Beispiel

```
INSERT INTO Buch (isbn, autor, titel)
  VALUES(1236548792, "Paul Panther", "Der rosa Riese");
```

fehlende Felder bekommen NULL-Werte

- Beispiel Bücher-Datenbank:

Titel	Autor	Seiten	Verlag	Jahr	Auflage	ISBN	Schlagwörter
Physik für Ingenieure	Hering, Martin, Stohrer	744	Springer	1999	7	3540661352	Physik, Lehrbuch, Mechanik, Quantenmechanik, Thermodynamik
Taschenbuch der Physik	Stöcker	1092	Harri Deutsch	2000	4	3817116284	Physik, Lehrbuch, Mechanik, Quantenmechanik, Thermodynamik
Fundamentals of Physics Extended	Halliday, Resnick, Walker	1198	Wiley	1997	5	471105597	Physik, Lehrbuch, Mechanik, Quantenmechanik, Thermodynamik
Physik - Der Grundkurs	Pitka, Bohrmann, Stöcker, Terlecki	426	Harri Deutsch	1999	1	3817115768	Physik, Lehrbuch, Mechanik, Thermodynamik
Einführung in die Thermodynamik	Cerbe, Hoffmann	NULL	Carl Hanser	1999	12	3446211101	Lehrbuch, Thermodynamik
Thermodynamik für Ingenieure	Langeheinecke, Jany, Sapper	NULL	Vieweg	1999	2	3528147857	Lehrbuch, Thermodynamik
Relationale Datenbanken und SQL	Matthiesen, Unterstein	352	Addison-Wesley	2000	2	3827315581	Lehrbuch, SQL, Datenbank
MySQL & mSQL	Yarger, Reese, King	512	O'Reilly	2000	1	3897211637	SQL, Datenbank, Handbuch

- Probleme mit der Tabelle:

Felder mit mehreren Einträgen (Autor, Schlagwort)

- → kann nicht nach einzeltem Autor oder Schlagwort durchsucht werden

hohe Redundanz

- gleiche Begriffe an mehreren Stellen (Schlagwörter, Autoren, Verlag)
- → vergrößerter Speicherbedarf
- → Gefahr von Inkonsistenzen (z.B. durch Schreibfehler)

Datenbanken brauchen klaren Entwurf!

- Entity-Relationship-Modell (**ERM**):

Verfahren zum systematischen Entwurf von Datenbanken

Entity

- eindeutig identifizierbares Objekt
- beschrieben durch eine Menge von Attributen
- Typ des Objekts = Liste der Attributnamen und -typen
- Entitäten ergeben Tabellen

Vergleich mit objektorientierter Programmierung

Tabelle \triangleq Klasse
 Attribut \triangleq Datenfeld
 Datensatz \triangleq Objekt

Relationship

- Beziehung zwischen Entitäten
- kann selbst Eigenschaften (Attribute) haben
- kann zwei oder mehr verschiedene Arten von Entitäten verknüpfen
- kann verschiedene Vielfachheiten haben, z.B.

1:1	ein Objekt A mit einem Objekt B
1:N	ein Objekt A mit mehreren Objekten B
N:M	ein A mit mehreren B und ein B mit mehreren A

- ein Objekt möglicherweise mit keinem anderen verknüpft
- Notation z.B.

1:0..* = ein Objekt A mit gar keinem, einem oder mehreren B verknüpft

graphische Darstellung z.B. als Klassendiagramm (**UML-Diagramm**)

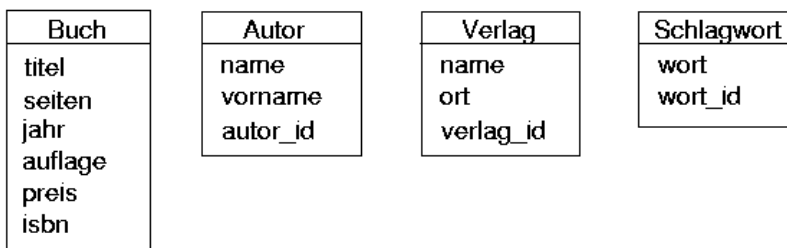


- Anwendung auf Beispiel:

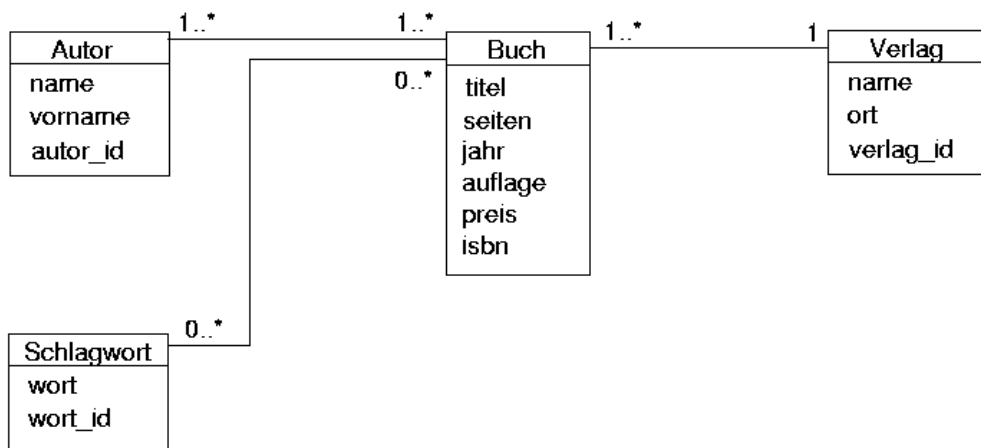
Entitäten sind

- Buch
- Autor
- Verlag
- Schlagwort

Attribute



Beziehungen zwischen den Entitäten



z.B. Beziehung zwischen Buch und Verlag

- ein Buch hat genau einen Verlag (1)
- ein Verlag hat ein oder mehrere Bücher (1..*)

einige Designentscheidungen

- nur Autoren aufgeführt, für die Bücher in der Datenbank existieren
- zu einem Schlagwort kann evtl. noch kein Buch vorhanden sein

- Umsetzung der Relationen:

unterschiedlich nach Grundtypen 1:1, 1:N, N:M

bei 1:N

- Schlüssel der 1-Seite als Attribut zur Tabelle der N-Seite hinzufügen (**Fremdschlüssel**)
- im Beispiel: Tabelle `Buch` bekommt Attribut `verlag_id`

bei 1:1

- beide Tabellen zu einer mit allen Attributen zusammenfassen
- häufig durch Hinzufügen der Attribute einer Tabelle zur anderen (z.B. Telefon-Nr. zu Autor)

bei N:M

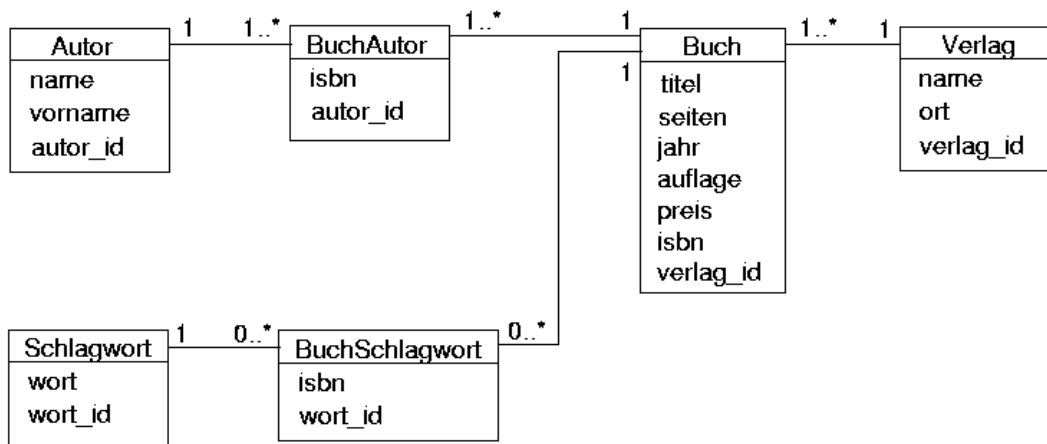
- neue Tabelle für die Relation einführen (**Verbindungsentität**)
- enthält Fremdschlüssel beider Partner
- Primärschlüssel ist häufig Paar der Fremdschlüssel
- enthält manchmal weitere Attribute

im Beispiel:

- neue Tabelle `BuchAutor`
- Attribute `autor_id` und `isbn`
- bilden zusammen Primärschlüssel
- anlegen mit `PRIMARY KEY (autor_id, isbn)`
- einen Eintrag für jede Autor/Buch-Kombination
- analog für `BuchSchlagwort`

- Fertige Bücher-Datenbank:

als Klassendiagramm



Tabellen mit Beispieldaten

- Aufgaben:

Aufgabe 5

- Einfache Abfragen
- Abfragen über mehrere Tabellen
- Komplizierte Abfragen

Einfache Abfragen

- **SELECT-Kommando:**

grundlegendes Kommando für Abfragen (**Queries**)

sehr vielfältig

große Zahl an optionalen Klauseln

Grundabfrage zur Anzeige aller Daten einer Tabelle

```
SELECT * FROM tabelle;
```

- **Auswahl von Spalten (Projektion):**

nicht alle Angaben in einer Tabelle für die aktuelle Abfrage relevant

Auswahl über Liste von gewünschten Attributen

```
SELECT attribut1, attribut2, .. FROM tabelle;
```

Ausgabe kann sortiert werden mit der Zusatzklausel

```
ORDER BY attribut1, attribut2, ..
```

Default: aufsteigende Reihenfolge, absteigend mit

```
ORDER BY attribut DESC
```

Beispiel

```
SELECT titel, seiten, jahr FROM Buch
ORDER BY jahr, seiten DESC;
```

Resultat

- **Auswahl von Zeilen (Selektion):**

mit der **WHERE**-Klausel

viele gängige Operationen

Vergleiche	<, >, <=, =, !=
Logik	AND OR NOT
Arithmetik	+, -, *, /, %
LIKE	String-Vergleich mit Wildcards % (beliebig viele Zeichen) _ (genau ein Zeichen)

Beispiel

```
SELECT titel, seiten, jahr FROM Buch
WHERE jahr >= 2000;
```

Resultat

Beispiel

```
SELECT titel, seiten, jahr FROM Buch
WHERE titel LIKE "%Physi%";
```

Resultat

Abfragen auf NULL

- mit `attribut=NULL` immer leer
- stattdessen mit `attribut IS NULL`

- Funktionen im `SELECT`:

bei Auswahl-Attributen oder in `WHERE`-Bedingung

große Anzahl vordefiniert, z.B.

numerische Funktionen	SIN, LOG
String-Funktionen	SUBSTRING, CONCAT, LENGTH, TRIM
Datums-Funktionen	NOW, WEEK, DATE_ADD

Beispiel: Umrechnung in Dollar

```
SELECT titel, preis/2.04 FROM Buch
WHERE titel LIKE "%Physics%";
```

Resultat

- Weitere Anwendungen von `WHERE`:

Ändern von Daten

```
UPDATE tabelle SET attribut1=wert1, ...
WHERE bedingung;
```

Löschen von Daten

```
DELETE FROM tabelle WHERE bedingung;
```

Abfragen über mehrere Tabellen



- **SELECT mit mehreren Tabellen (Join):**

- mehrere Tabellen angeben

```
SELECT attribut1, attribut2, ...  
FROM table1 JOIN table2 ...
```

ohne weitere Einschränkung alle Kombinationen aus allen Tabellen

Beispiel

```
SELECT titel, name FROM Buch JOIN Verlag;
```

liefert 70 Einträge (10 Bücher x 7 Verlage)

- Verknüpfung der Tabellen durch Übereinstimmung der Schlüssel

```
SELECT titel, name FROM Buch INNER JOIN Verlag  
ON Buch.verlag_id = Verlag.verlag_id;
```

bei gleichen Attributbezeichnungen Verknüpfung automatisch (**Natural Join**)

```
SELECT titel, name FROM Buch NATURAL JOIN Verlag;
```

Resultat

- bessere Attributbezeichnungen bei Joins mit AS, z.B.

```
SELECT titel, name AS verlag  
FROM Buch NATURAL JOIN Verlag  
ORDER BY verlag;
```

Resultat

- **Gruppierung:**

weitere Klausel für SELECT

```
GROUP BY attribut1, attribut2, ..
```

fasst Datensätze mit gleichen Werten für Attribute zusammen

übrige (nicht-gleiche) Attribute brauchen Funktionen zum Kombinieren

wichtigste Funktionen

COUNT	Anzahl der zusammengefassten Werte
SUM	Summe der jeweiligen Attribute
AVG	Durchschnitt
MAX, MIN	Maximum, Minimum

- **Beispiele:**

Wieviele Bücher gibt es für jedes Jahr, wie groß ist der Gesamtpreis pro Jahr?

```
SELECT COUNT(*), SUM(preis), jahr FROM Buch
GROUP BY jahr;
```

Resultat

Wieviele Bücher gibt es von jedem Verlag?

```
SELECT name as verlag, COUNT(*)
FROM Buch NATURAL JOIN Verlag
GROUP BY name;
```

Resultat

Komplizierte Abfragen



- Einige beispielhafte Fragestellungen:
 - Bestimme den Gesamtpreis aller Bücher der einzelnen Verlage.
 - Bestimme zu jedem Schlagwort die Zahl der passenden Bücher
 - Bestimme für jeden Autor alle Bücher, an denen er mitgeschrieben hat.
 - Ermittle die Namen aller Autoren, mit denen Stöcker zusammen ein Buch verfasst hat.
- Vorgehensweise bei komplizierten Abfragen:
 - alle benötigten Tabellen (für Attribute und benötigte Relationen) zusammensuchen
 - graphisch: Klassenkästchen mit benötigten Attributen nebeneinander stellen
 - benötigte Verknüpfungen über die Fremdschlüssel formulieren
 - graphisch: entsprechende Schlüssel verbinden und mit Relation (=, !=, Wert) beschriften
 - bei = ggf. NATURAL JOIN benutzen
 - bei Gruppierung: zu gruppierende Attribute bestimmen, Zusammenfassungsfunktionen bei den anderen
 - graphisch: zu gruppierende Attribute einkreisen, an alle anderen Funktion schreiben
 - Feinarbeit: Umbenennung von Attributen, Sortierung
- Anwendung bei Beispielfrage 1:

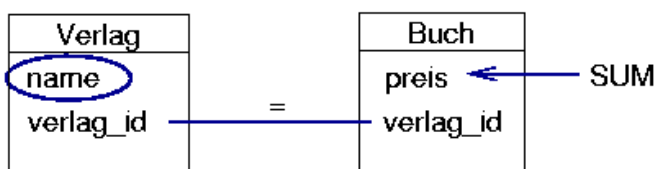
benötigt werden

Tabelle	Attribute
Verlag	name, verlag_id
Buch	preis, verlag_id

verknüpft wird über Gleichheit der `verlag_id`-Schlüssel (NATURAL JOIN)

gruppiert wird über `name`, über `preis` wird summiert

graphisch also



Feinschliff

- Spalten umbenennen in `verlag` und `gesamtpreis`
- absteigend sortieren nach `gesamtpreis`

SQL-Kommando

```
SELECT name AS verlag , SUM(preis) AS gesamtpreis
```

```
FROM Buch NATURAL JOIN Verlag  
GROUP BY verlag  
ORDER BY gesamtpreis DESC;
```

Resultat

- Aufgaben:

Aufgabe 6

- Grundfunktionen von Matlab
- Vektoren
- Matrizen
- Lösen von Gleichungen
- Berechnung bestimmter Integrale
- Integration von Differentialgleichungen

Grundfunktionen von Matlab



- Matlab:

Umfangreiches Programmpaket für alle Arten numerischer Berechnungen

Hersteller: [The MathWorks Inc.](#)

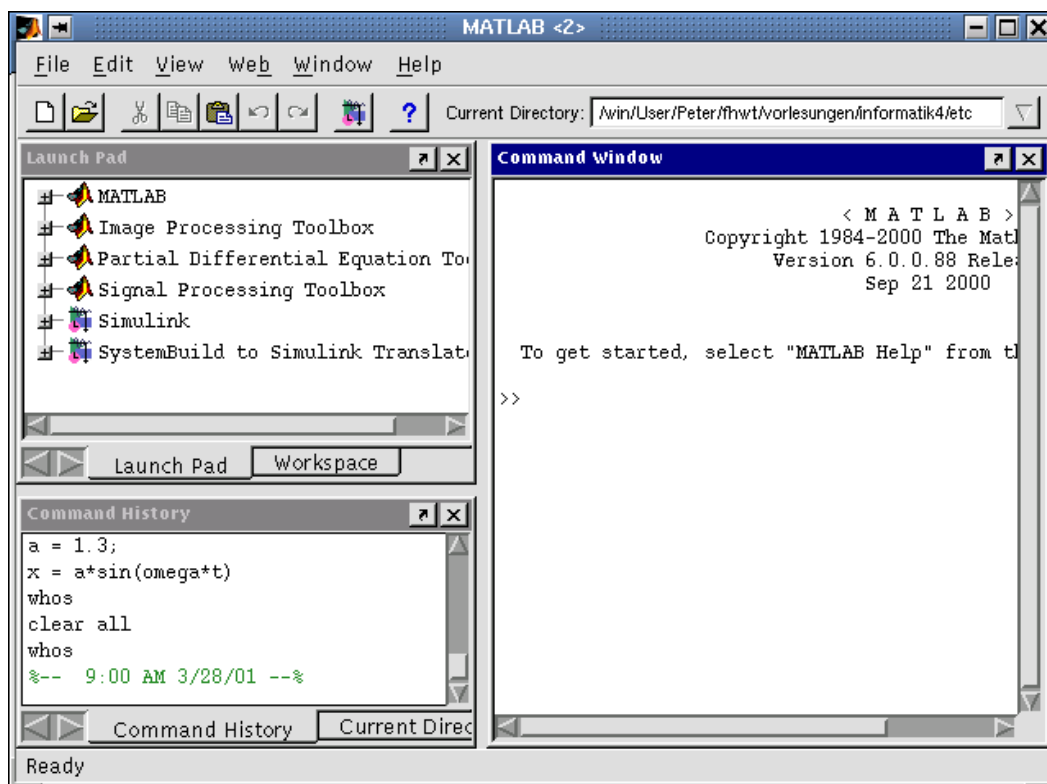
viele Zusatzpakete für spezielle Anwendungen, auch von Drittfirmen

bei uns (PC-Pool Diepholz) installiert

Paket	Zweck
Simulink	graphisches Simulations-Tool
Signal Processing Toolbox	Signalverarbeitung, Filter, ..
Image Processing Toolbox	Bildbearbeitung
PDE Toolbox	Partielle DGLs (einfach)
FEMLAB	Partielle DGLs (vielseitig, 3D, ..)
	von Comsol

- Bedienung:

graphische Oberfläche mit mehreren Unter-Fenstern



Kommandofenster zur Eingabe von Matlab-Befehlen am Prompt >>

Fenster mit den letzten Befehlen (**Command History**)

Fenster mit wichtigen Hinweisen und Spezialfunktionen (**Launch Pad**)

- Hilfe:

Informationen zu einer Funktion

```
>> help NAME
```

Fenster mit Informationen zu den wichtigsten Funktionen, nach Sachgruppen gegliedert

```
>> helpwin
```

viele komplette Beispiele

```
>> demo
```

Help-Menü

- öffnet Help-Fenster mit Suchfunktionen
- enthält u.a. alle Handbücher(70 cm im Regal!)

- Matlab als Taschenrechner:

Zahlen und Operationen direkt eingeben

```
>> 12.567 * 31.34 / (17 + 4) - 1.88e1
```

```
ans =
```

```
-0.0452
```

die üblichen Operatoren + - * / ^ ("hoch")

alle gängigen Funktionen und viele mehr, z.B.

- sqrt, exp, log, log10
- sin, cos, tan, asin, acos, atan
- sinh, cosh, atanh, ...
- rem, floor, ceil
- pi

Wiederholung mit "Pfeiltaste rauf"

- Genauigkeit:

Anzeige

- Standard: 4 Nachkommastellen bzw. e-Notation
- alle Dezimalen mit

```
>> format long
```

normalerweise mit **double**-Zahlen

- "fast" 16 Dezimalen
- Exponent bis ± 308

daher rechnerische Fehler

```
>> 1 - 5 * 0.2
```

```
ans =
```

```
0
```

```
>> 1 - 0.2 - 0.2 - 0.2 - 0.2 - 0.2
```

```
ans =
```

```
5.551115123125783e-17
```

- Variablen:

"Speicher" mit Namen

```
>> g = 9.81
```

```
g =
```

```
9.8100
```

Namen aus Buchstaben, Zahlen und Unterstrich _

Klein- und Großbuchstaben verschieden

Unterdrücken der Ausgabe mit abschließendem ;

```
>> T = 0.3;  
>> omega = 2*pi/T;  
>> t = 2.75;  
>> a = 1.3;  
>> x = a*sin(omega*t)
```

```
x =
```

```
1.1258
```

Variable `ans` für unbenannte Ergebnisse

Liste aller Variablen

```
>> whos
```

Name	Size	Bytes	Class
T	1x1	8	double array
a	1x1	8	double array
ans	1x1	8	double array
b	1x1	8	double array
g	1x1	8	double array
omega	1x1	8	double array
t	1x1	8	double array
x	1x1	8	double array

Grand total is 8 elements using 64 bytes

Löschen aller Werte (und Freigabe des Speichers) mit

```
>> clear all
```

- Grundoperationen mit Vektoren:

Folge von Zahlen, eingegeben als

```
>> a = [1 2 3 4 5 6]
```

```
a =
```

```
1 2 3 4 5 6
```

einfache Addition, Subtraktion, Verknüpfung mit Zahlen

```
>> b = 2*a - 1
```

```
b =
```

```
1 3 5 7 9 11
```

viele Standardfunktionen arbeiten elementweise mit Vektoren

```
>> c = log(b)
```

```
c =
```

```
0 1.0986 1.6094 1.9459 2.1972 2.3979
```

elementweise Multiplikation, Division und Potenz mit `.*`, `./`, `.^`

```
>> a.^3
```

```
ans =
```

```
1 8 27 64 125 216
```

- Vektor-Operationen:

viele spezielle Funktionen, z.B.

max	größter Wert
min	kleinster Wert
sum	Summe aller Werte
mean	Mittelwert
std	Standard-Abweichung
sort	sortierter Vektor

Umwandlung von Zeilenvektor in Spaltenvektor (**Transposition**)

```
>> a'
```

```
ans =
```

```
1  
2  
3  
4  
5
```

Skalarprodukt = (Matrix-)Produkt von Zeilenvektor mit Spaltenvektor

```
>> b * a'
```

```
ans =
```

```
161
```

- Teile von Vektoren auswählen:

einzelnes Element mit Klammern

```
>> a(3)
```

```
ans =
```

```
3
```

Elemente i bis j mit $i:j$

```
>> a(3:5)
```

```
ans =
```

```
3     4     5
```

bis zum Schluss mit `end`

```
>> a(4:end)
```

```
ans =
```

```
4     5     6
```

nur jedes k -te Element mit $i:k:j$

```
>> a(1:2:6)
```

```
ans =
```

```
1     3     5
```

$i:k:j$ ist selbst ein Vektor (sogar mit Dezimalzahlen)

```
>> 1:0.4:3
```

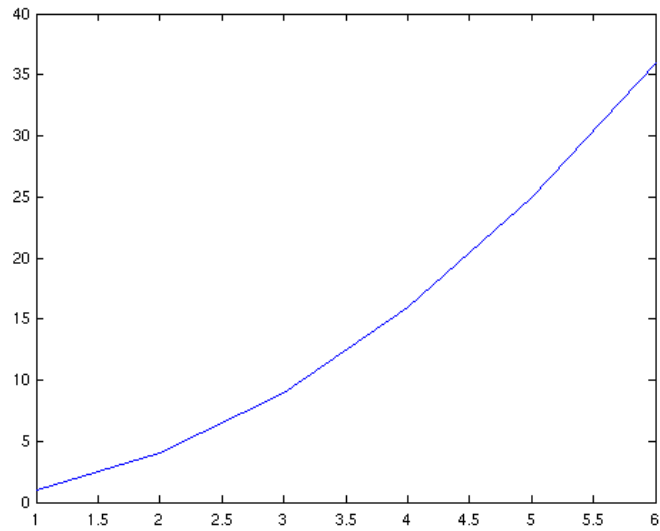
```
ans =
```

```
1.0000    1.4000    1.8000    2.2000    2.6000    3.0000
```

- Plotten von Vektoren:

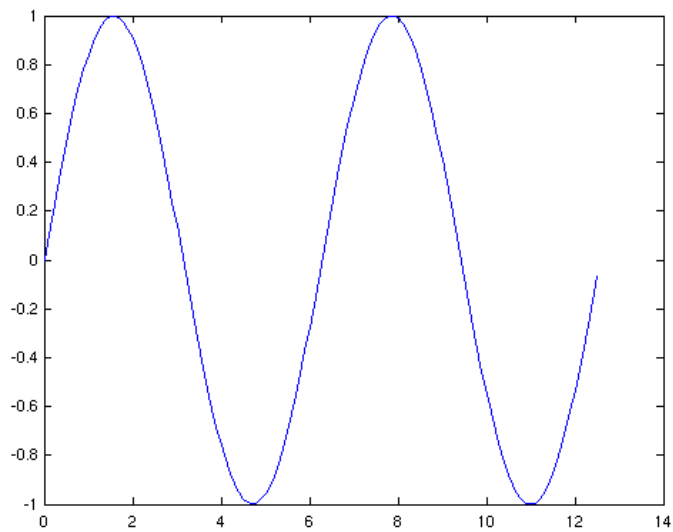
einfach mit `plot`

```
>> plot(a.*a)
```



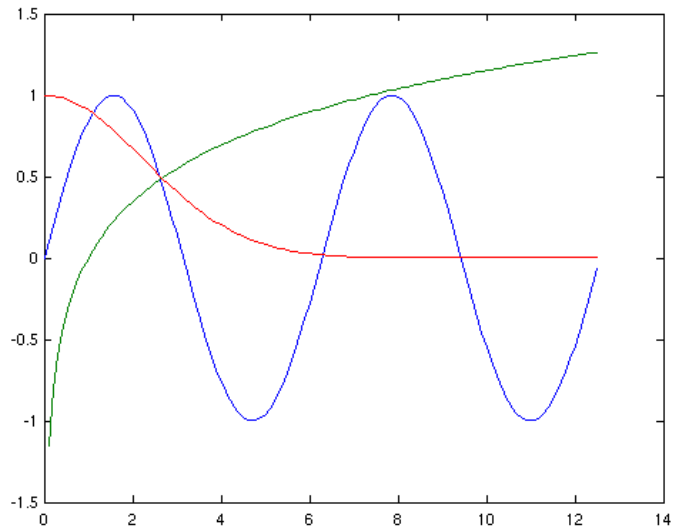
mit zwei Vektoren als x- und y-Koordinaten

```
>> t = 0:0.1:4*pi;
>> y = sin(t);
>> plot(t,y)
```



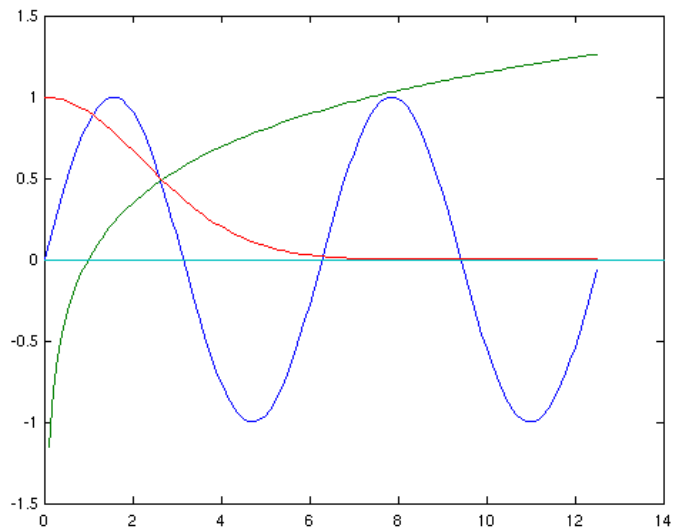
auch mehrere Vektoren möglich, jeweils als x- und y-Koordinaten

```
>> y2 = 0.5*log(t);
Warning: Log of zero.
>> y3 = exp(-0.1*t.*t);
>> plot(t, y, t, y2, t, y3)
```



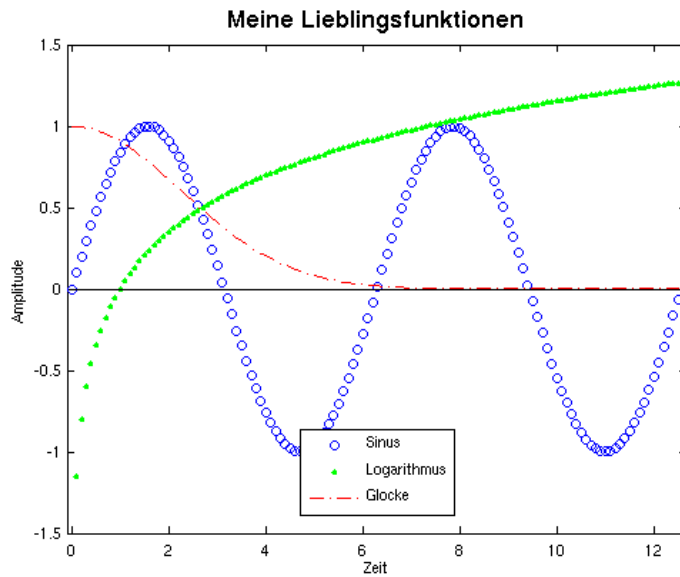
x-Achse als Datensatz mit zwei Punkten

```
>> plot(t, y, t, y2, t, y3, [0 4*pi], [0 0]);
```



zahllose Optionen und weitere Funktionen, s. help plot

```
>> plot(t, y, 'bo', t, y2, 'g.', t, y3, 'r-.', [0 4*pi], [0 0], 'k-')
>> title('Meine Lieblingsfunktionen', ...
        'FontSize', 14, 'FontWeight', 'bold')
>> xlabel('Zeit');
>> ylabel('Amplitude');
>> legend('Sinus', 'Logarithmus', 'Glocke', 'Location', 'Best');
>> xlim([-0.1, 4*pi + 0.1])
```



- Arbeiten mit Skripten

Editor öffnen mit `File/New/M-File`

Kommandos genauso eingeben wie im Command Window

abspeichern als `NAME.m`, wobei `NAME` nur Buchstaben, Ziffern oder Unterstrich `_` enthält und mit Buchstaben beginnt

Beispiel `zeichneFunktion.m`

ausführen mit Taste `<F5>` oder durch Eingabe von `NAME` im Command Window

- Aufgaben:

[Aufgabe 7](#)

[Aufgabe 8](#)

- Grundfunktionen:

rechteckiges Zahlenschema, eingeben mit

```
>> a = [1 2 3; 4 5 6; 7 8 9; 10 11 12]
```

```
a =
```

```
     1     2     3
     4     5     6
     7     8     9
    10    11    12
```

elementweise Funktionen wie bei Vektoren

erster Index ist Zeilenindex

```
>> a(2,3)
```

```
ans =
```

```
6
```

Auswahlfunktionen wie bei Vektoren

```
>> a(3:4, 2:3)
```

```
ans =
```

```
     8     9
    11    12
```

Auswahl ganzer Zeilen oder Spalten mit :

```
>> a(3,:) 
```

```
ans =
```

```
     7     8     9
```

```
>> a(:,2)
```

```
ans =
```

```
     2
     5
     8
    11
```

Multiplikation von Matrizen untereinander und mit Vektoren

```
>> a = [1 2 3; 4 5 6; 7 8 9];  
>> b = [0 1 2; -1 0 1; -2 -1 0];  
>> v = [2 0 -1]'
```

```
v =
```

```
    2  
    0  
   -1
```

```
>> a*v
```

```
ans =
```

```
   -1  
    2  
    5
```

```
>> a*b
```

```
ans =
```

```
   -8   -2    4  
  -17   -2   13  
  -26   -2   22
```

Vektorfunktionen auf Matrizen liefern Vektoren

```
>> sum(a)
```

```
ans =
```

```
    12    15    18
```

- Einige Matrixfunktionen:

Transposition

```
>> a'
```

```
ans =
```

```
    1    4    7  
    2    5    8  
    3    6    9
```

Matrizen aus Nullen oder aus Einsen

```
>> zeros(2,2)
```

```
ans =
```

```
    0    0  
    0    0
```

```
>> ones(2,2)
```

```
ans =
```

```
    1    1  
    1    1
```

Matrizen aus Zufallszahlen zwischen 0 und 1

```
>> rand(3,3)
```

```
ans =
```

```
    0.7680    0.7889    0.2140  
    0.9708    0.4387    0.6435  
    0.9901    0.4983    0.3200
```

- Ein- und Ausgabe:

alle aktuellen Variablen abspeichern (in Datei `matlab.mat`)

```
>> save
```

wieder einladen mit

```
>> load
```

abspeichern der Matrix `a` in ASCII-Format in Datei `juhu.dat`

```
>> save -ascii 'juhu.dat' a
```

einladen

```
>> load -ascii 'juhu.dat'
```

liefert Ergebnis in Matrix `juhu`

viele Funktionen für ausgefeilte Formatierungen

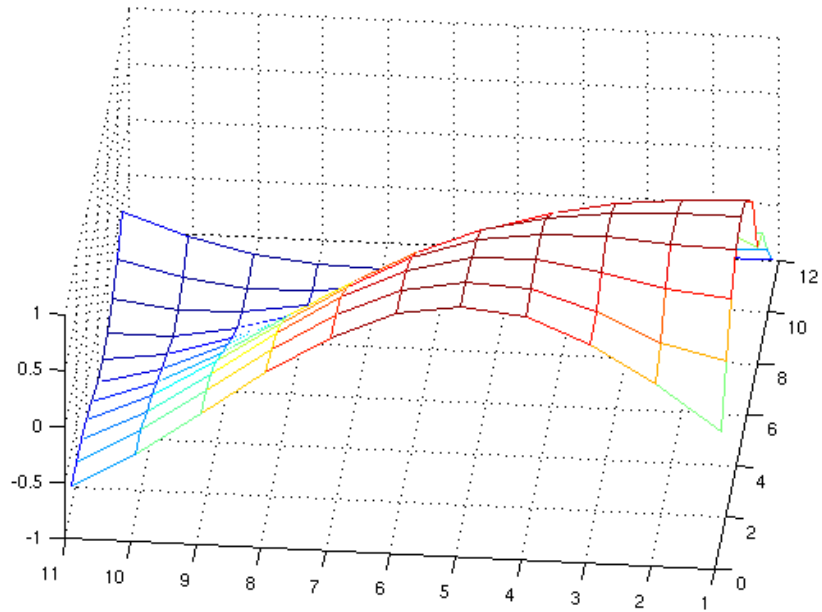
- Darstellung von Matrizen:

Demo-Matrix aus `demo1.dat` laden

```
>> load -ascii 'demo1.dat'
```

Darstellung als 3d-Gitter

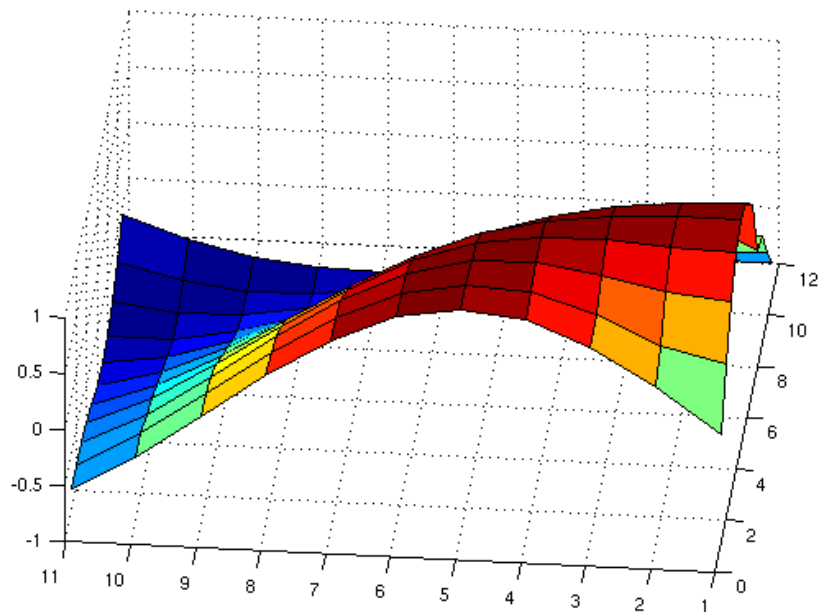
```
>> mesh(demo1)
```



Änderung der Ansicht durch Lupen- und Drehfunktionen

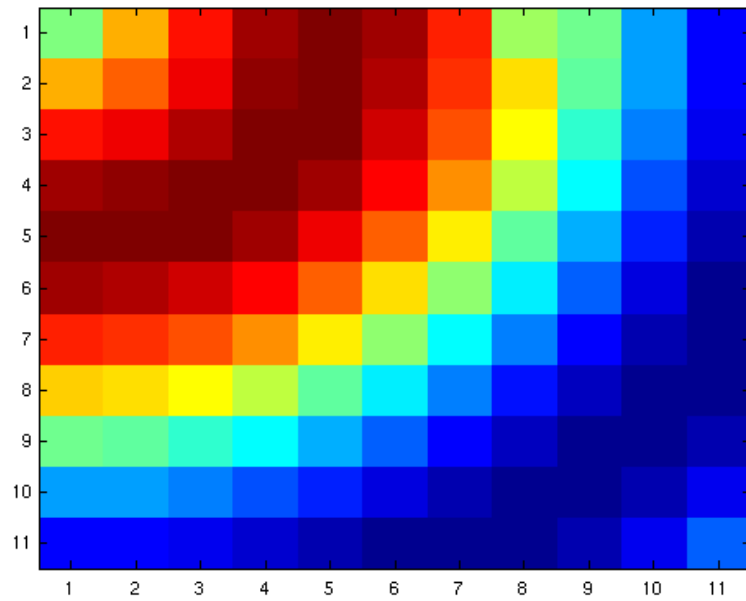
Darstellung als Oberfläche

```
>> surf(demo1)
```



Darstellung als Bild

```
>> imagesc(demo1)
```



Farben gemäß aktueller Farbtabelle

Lösen von Gleichungen



- Lineare Gleichungssysteme:

betrachten Beispiel

$$2x - y + 3z = 1$$

$$4x - 2y + z = 5$$

$$3y + z = -3$$

in Matrixform

$$A x = r$$

wobei

$$A = \begin{pmatrix} 2 & -1 & 3 \\ 4 & -2 & 1 \\ 0 & 3 & 1 \end{pmatrix} \quad r = \begin{pmatrix} 1 \\ 5 \\ -3 \end{pmatrix}$$

- Lösung in Matlab:

für das Beispielsystem

```
>> A = [2 -1 3; 4 -2 1; 0 3 1]
```

```
A =
```

```
     2     -1     3
     4     -2     1
     0      3     1
```

```
>> r = [1 5 -3]'
```

```
r =
```

```
     1
     5
    -3
```

```
>> x = A \ r
```

```
x =
```

```
     1.0000
    -0.8000
    -0.6000
```

Probe:

```
>> A*x - r
```

```
ans =
```

```
1.0e-15 *  
-0.3331  
0  
0
```

und genauso auch z.B. für ein 1000x1000-System

```
>> A=rand(1000);  
>> r = rand(1000,1);  
>> x = A \ r;
```

Lösung braucht etwa 10 s auf 450 MHz-PC

Probe

```
>> error = max(abs(A*x-r))
```

```
error =
```

```
2.0761e-13
```

- **Überbestimmte Systeme:**

mehr Gleichungen als Unbekannte

i.a. keine Lösung

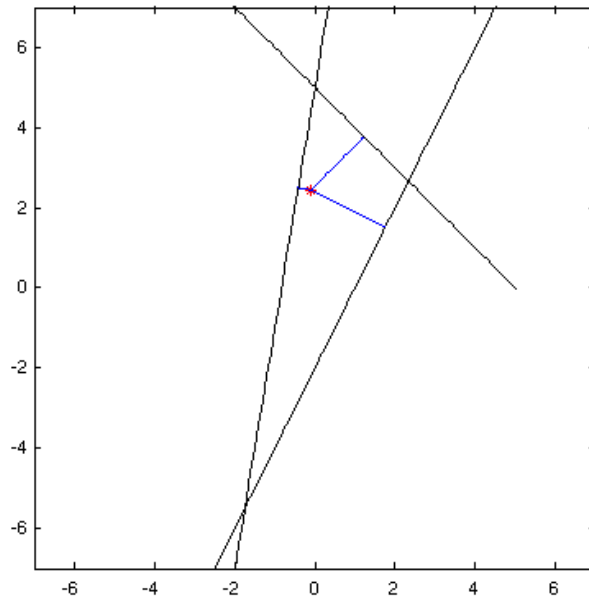
gesucht: Lösung mit geringstem Fehler

in Matlab auch einfach mit

```
>> x = A \ r;
```

graphisch bei 2 Unbekannten

- Gleichung $ax + by = r$ ist Gerade
- mehr als 2 Geraden schneiden sich (i.a.) nicht in einem Punkt
- gesuchter Punkt hat geringste Summe der Abstände



- Nullstellen von Polynomen:

Beispiel 2. Grades

$$2x^2 - 2x - 24 = 0$$

in Matlab Polynom als Vektor der Koeffizienten

```
>> poly1 = [2 -2 -24];
```

alle Lösungen mit der Funktion `roots`

```
>> roots(poly1)
```

```
ans =
```

```
    4
   -3
```

geht auch bei komplexen Lösungen, etwa $x^2 + 1 = 0$

```
>> poly2 = [1 0 1];
```

```
>> roots(poly2)
```

```
ans =
```

```
    0 + 1.0000i
    0 - 1.0000i
```

klappt auch bei höherer Ordnung

```
>> poly3 = [1 -8 20 -10 -25 22];
```

```
>> roots(poly3)
```

```
ans =
```

```
3.7059
-1.0781
2.1861 + 0.8527i
2.1861 - 0.8527i
1.0000
```

- Lösung "beliebiger" Gleichungen:

Beispiel: Löse die Gleichung

$$e^{-0.05x^2} = \sin(x)$$

Umformulierung: Finde Nullstellen der Funktion

$$f(x) = \sin(x) - e^{-0.05x^2}$$

Funktion `fzero` braucht als Argumente Funktion und Startwert

Funktion direkt hinschreiben als "anonyme Funktion"

```
>> F = @(x) sin(x) - exp(-0.05*x.^2)
```

```
F =
```

```
@(x) sin(x) - exp(-0.05*x.^2)
```

irgendeinen (?) Startwert nehmen und ab geht's

```
>> x0 = 0;
```

```
>> x = fzero(F, x0)
```

```
x =
```

```
1.1968
```

andere Startwerte liefern andere Lösungen

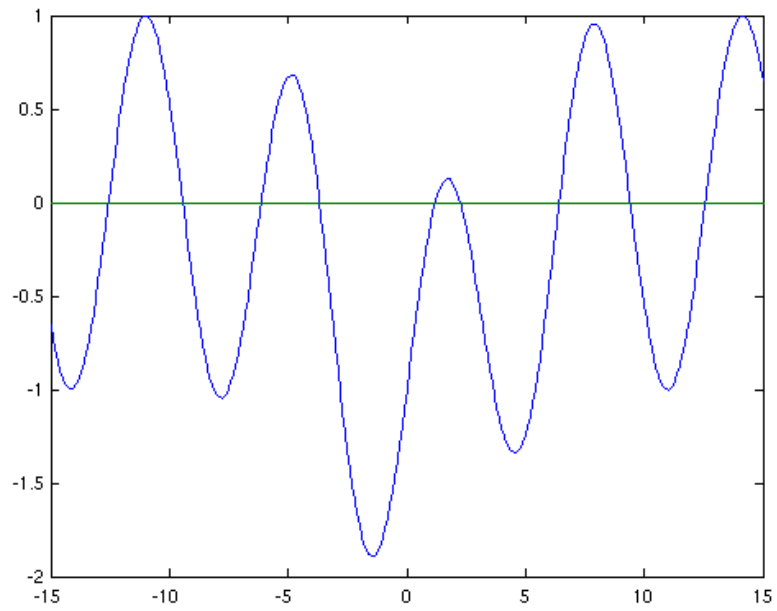
x0	x
0	1.1968
1	1.1968
2	2.2537
3	2.2537
4	2.2537
5	6.4116

wieviele Lösungen gibt es überhaupt?

graphisch Überblick gewinnen

```
>> x = -15:0.1:15;
```

```
>> plot(x, F(x), [-15 15], [0 0])
```



damit Struktur der Lösungsmenge und gute Startwerte für `fzero`

- Aufgaben:

Aufgabe 9

Aufgabe 10

Berechnung bestimmter Integrale

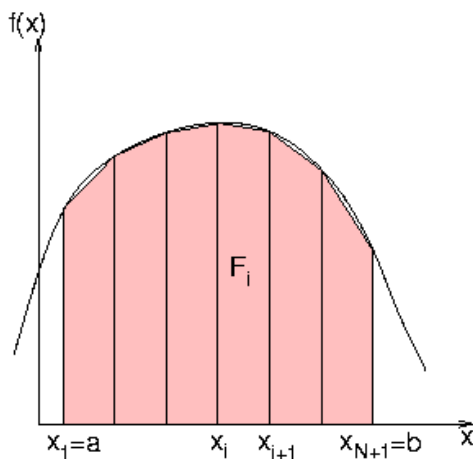


- Trapezregel:

sehr einfaches Verfahren zur numerischen Integration

Prinzip

- Unterteilung des Integrationsintervalls in N Teile
- Funktion in jedem Intervall durch Strecke nähern
- Fläche der sich ergebenden Trapeze addieren



in Formeln

- Zahl der Intervalle N vorgeben
- Breite h der Intervalle

$$h = (b - a)/N$$

- Fläche F_i eines Trapezes

$$F_i = h \frac{(f(x_i) + f(x_{i+1}))}{2}$$

- Integral damit

$$\begin{aligned} \int_a^b f(x) dx &\approx \sum_{i=1}^N F_i \\ &= h \left[\frac{1}{2} f(x_1) + \sum_{i=2}^N f(x_i) + \frac{1}{2} f(x_{N+1}) \right] \end{aligned}$$

Beispiel i.f.

$$I = \int_0^{\pi/2} \sin(x) dx$$

Berechnung von I in Matlab

```
>> N = 5;  
>> a = 0;  
>> b = pi/2;  
>> h = (b-a)/N;
```

```
>> x = a:h:b;
>> y = sin(x);
>> I = h*(0.5*y(1) + sum(y(2:N)) + 0.5*y(N+1))
```

```
I =
```

```
0.9918
```

ganz genauso für N = 100 ergibt (mit `format long`)

```
I = 0.99997943823961
```

- Eigene Funktionen in Matlab:

einfachste Form

- alle Kommandos in eine Datei `trapez1.m`
- mit `trapez1 (Dateiname)` aufrufen
- zur Erklärung Kommentare einfügen (von `%` bis zum Zeilenende)
- Problem: alle Parameter direkt in Datei ändern

als Funktion

- alle Kommandos in Datei `trapez2.m`
- erste Zeile definiert Funktionsnamen und Argumente

```
function trapez2(a, b, N)
```

benutzen mit

```
>> trapez2(0, pi/2, 1000)
```

```
ans =
```

```
0.99999979438323
```

Kommentar am Anfang wird Hilfetext

```
>> help trapez2
```

```
Integration mit der Trapezregel
```

```
Argumente
```

```
    Grenzen a, b
```

```
    Zahl der Intervalle N
```

```
Ergebnis
```

```
    Integral des Sinus von a bis b,
```

```
    genaehert mit Trapezregel mit N Intervallen
```

andere Funktion als Sinus

- Funktion als neues Argument `func` (s. `trapez.m`)
- ersetze `sin(x)` durch `func(x)`
- aufrufen mit Funktionszeiger ("function handle"): `@FUNKTIONSNAME`

```
>> trapez(@cos, 0, pi, 100)
```

```
ans =
```

```
8.370884395220717e-17
```

beliebige Funktion integrieren

- gewünschte Funktion als eigene Matlab-Funktion
- z.B. Fehlerintegral e^{-x^2} (**glocke.m**)

```
>> glocke(1)
```

```
ans =
```

```
0.36787944117144
```

damit in trapez

```
>> trapez(@glocke, 0, 10, 1000)
```

```
ans =
```

```
0.88622692545276
```

- Bessere Integrationsmethoden:

Funktion durch Parabeln statt Geraden annähern (**Simpson-Methode**)

Intervalle nicht gleich groß, sondern geschickt anpassen

- unzählige Verfahren
- kann auch mit schwierigen Integranden fertigwerden
- nicht möglich bei Tabellenfunktionen

Matlabfunktion `quad` berechnet Integral auf relative Genauigkeit $1e-6$

```
>> quad(@glocke, 0, 10)
```

- Aufgaben:

Aufgabe 11

- Beispiele für Differentialgleichungen:

radioaktiver Zerfall

- Anzahl N der Kerne mit Zerfallskonstante λ

$$\frac{dN}{dt} = -\lambda N$$

radioaktive Zerfallskette

- Teilchensorte 1 zerfällt in Sorte 2 mit λ_1
- Sorte 2 zerfällt weiter mit λ_2

$$\frac{dN_1}{dt} = -\lambda_1 N_1$$

$$\frac{dN_2}{dt} = \lambda_1 N_1 - \lambda_2 N_2$$

angeregte Schwingung mit viskoser Reibung

- Masse m schwingt an Feder mit Federkonstante c
- Reibung ist proportional zur Geschwindigkeit v
- äußere zeitabhängige Erregerkraft $F(t)$

$$m\ddot{x} + b\dot{x} + cx = A\cos(\omega t)$$

- Lösungsverfahren von Euler:

einfachstes (und ungenauestes) Verfahren

Grundidee

- Ableitung ersetzen durch (endliche) Differenzen

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

- ergibt normale Gleichungen für f

im Beispiel "radioaktiver Zerfall"

$$\frac{N(t+h) - N(t)}{h} = -\lambda N(t)$$

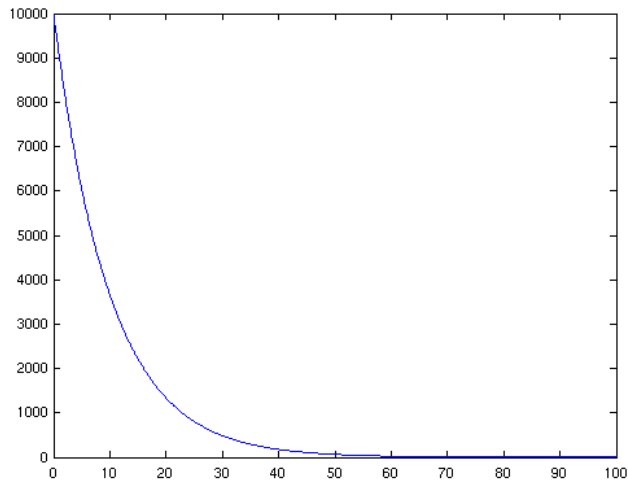
$$\Rightarrow N(t+h) = (1 - \lambda h)N(t)$$

- bei gegebenem Startwert $N(0)$ schrittweise lösen
- hier sogar direkt auflösbar

$$N(nh) = (1 - \lambda h)^n N(0)$$

in Matlab

```
>> N0 = 10000;           % Anfangsmenge
>> lambda = 0.1;        % Zerfallskonstante
>> h = 0.1;             % Schrittweite
>> n = 0:1000;          % 1000 Schritte
>> t = h*n;             % Zeitwerte
>> N = N0*(1-lambda*h).^n; % Werte fuer N(t)
>> plot(t, N);
```



- Lösungsverfahren in Matlab:

viele sehr weitentwickelte Verfahren (**ode solver**)

einige Solver für schwierige (**steife**) Differentialgleichungen

Auswahl des Solvers u.a. abhängig von

- Genauigkeitsforderung
- "Glattheit" der Gleichungen
- Schwierigkeit der Gleichung (steif oder nicht steif)

Daumenregel

- erste Wahl: `ode45`
- falls der versagt oder zu lange braucht: `ode15s`
- wenn der auch nicht klappt: Experten fragen

- Vorgehensweise am Beispiel "radioaktiver Zerfall"

Gleichungen schreiben als $y' = f(t, y)$

$$y' = -\lambda y$$

rechte Seite als Matlab-Funktion definieren (`radioaktiv.m`)

Solver aufrufen

```
>> [t,y] = ode45(@radioaktiv,[0 100],[10000]);
```

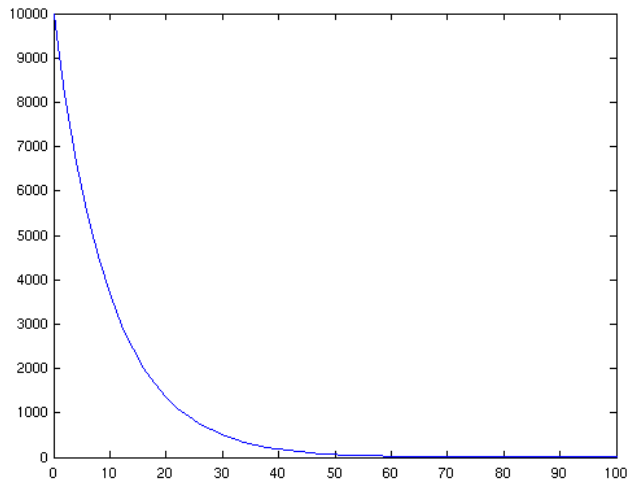
Parameter

- betrachtetes Zeitintervall [0 100]
- Anfangswert [10000]

Ergebniswerte y und zugehörige Zeiten t

Ergebnis in gewohnter Form plotten

```
>> plot(t, y);
```



- Beispiel "Zerfallskette":

System von Differentialgleichungen

wie vorher, aber alle Größen als Vektoren

Gleichungen als $y' = f(t, y)$

$$y_1' = -\lambda_1 y_1$$

$$y_2' = \lambda_1 y_1 - \lambda_2 y_2$$

rechte Seiten als Matlab-Funktion [radiokette.m](#)

Anfangsbedingungen z.B.

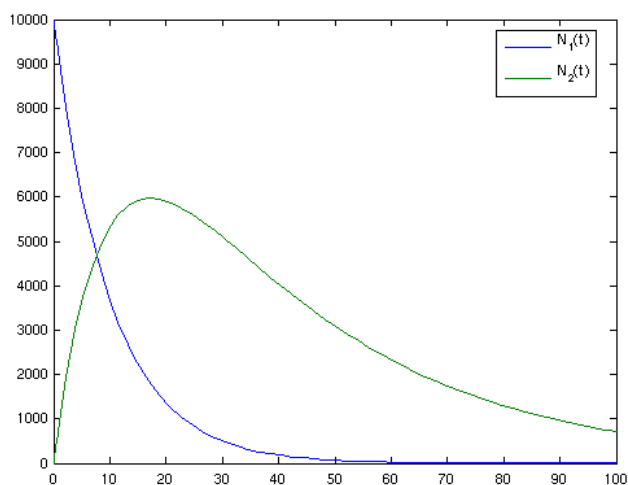
- 10000 Teilchen von Sorte 1
- keine Teilchen von Sorte 2

Lösen

```
>> [t,y] = ode45(@radiokette,[0 100],[10000 0]);
```

Spalten 1, 2 von y enthalten Zeitwerte von y_1, y_2 , daher plotten mit

```
>> plot(t, y(:,1), t, y(:,2))
>> legend('N_1(t)', 'N_2(t)', 0);
```



- Beispiel "angeregte Schwingung mit viskoser Reibung"

Problem: Gleichung zweiter Ordnung

Trick: Geschwindigkeit \dot{x} als zweite Variable einführen

damit Umschreiben in 2 Gleichungen 1. Ordnung

$$y_1' = y_2$$

$$y_2' = -\frac{b}{m}y_2 - \frac{c}{m}y_1 + \frac{A}{m}\cos(\omega t)$$

rechte Seiten als Matlab-Funktion `erzwungen.m`

Anfangsbedingungen z.B. Masse in Ruhe am Ursprung

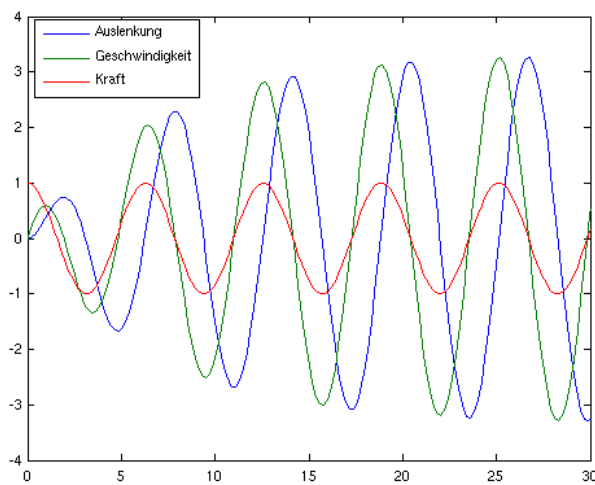
Lösen

```
>> [t, y] = ode45(@erzwungen, [0 30], [0 0]);
```

Darstellung

- zweite Spalte ist Geschwindigkeit
- zusätzlich äußere Anregung plotten

```
>> F = cos(t);
>> plot(t, y(:,1), t, y(:,2), t, F)
>> legend('Auslenkung', 'Geschwindigkeit', 'Kraft', 'Location', 'NorthWest');
```



- Übergabe von Parametern:

Problem:

- Parameter (m, b, c, A, omega) fest in Funktion `erzwungen`
- sollen direkt angebar sein

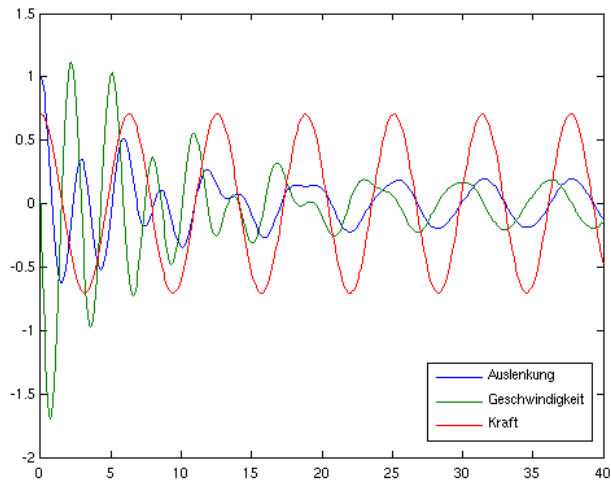
Parameter als zusätzliche Argumente der Funktion (`erzwungenp.m`)

Erzeugen einer Hilfsfunktion `myfunc`, die nur von t und y abhängt

```
>> myfunc = @(t, y) erzwungenp(t, y, 1, 0.286, 4.68, 0.706, 1);
```

Aufruf des Solvers mit der Hilfsfunktion

```
>> [t, y] = ode45(myfunc, [0 40], [1 0]);
```



- Aufgaben:

Aufgabe 12

Aufgaben



Bei den Aufgaben handelt es sich um kleine Projekte, die jeweils in Gruppen von 3 bis 4 Personen bearbeitet werden sollen.

- 📍 Aufgabe 1
- 📍 Aufgabe 2
- 📍 Aufgabe 3
- 📍 Aufgabe 4
- 📍 Aufgabe 5
- 📍 Aufgabe 6
- 📍 Aufgabe 7
- 📍 Aufgabe 8
- 📍 Aufgabe 9
- 📍 Aufgabe 10
- 📍 Aufgabe 11
- 📍 Aufgabe 12

Aufgabe 1



- Erstellen Sie in Gruppen von drei bis vier Personen eine kleine Homepage aus mehreren Seiten zu einem Thema Ihrer Wahl. Sie soll im Semester durch diverse Elemente erweitert werden; wählen Sie also ein hinreichend ergiebiges Thema.
- Einige Vorschläge bei Ideenmangel:
 - Der Kurs MB999 an der FHWT
 - Wohnen in Diepholz (oder sonstwo)
 - Meine Briefmarkensammlung
 - Saurier in Museen der Umgebung
 - Haltung des Zwergkrallenfroschs
 - **nicht:** Meine Lieblingsautos (hatte ich gefühlte 297 mal!)
- Benutzen Sie an Software und Hardware, was Ihnen an der FH oder privat bzw. im Internet zur Verfügung steht.
- Einhalten des HTML Standards 4.01-Strict ist obligatorisch!

Aufgabe 2



- Ergänzen Sie Ihre Seiten um Elemente zur Kommunikation mit dem Benutzer. Setzen Sie geeignete Formular-Elemente ein und wählen Sie zum Testen als Wert des `action`-Feldes immer meine [allgemeine echo-Seite](#)

Aufgabe 3



- Entfernen Sie alle Stil-Attribute aus Ihren bisherigen HTML-Seiten und erstellen Sie stattdessen ein globales Stylesheet für Ihr Projekt. Versehen Sie dabei alle Seiten mit einem ansprechenden einheitlichen Layout.

Aufgabe 4



- Überprüfen Sie, ob die von Ihnen eingesetzten Bilder zweckdienliche Formate verwenden. Versuchen Sie, durch geeignete Bearbeitung der Bilder bzw. der Format-Eigenschaften die Größe der Bilder möglichst weitgehend zu reduzieren. Erwägen Sie auch, kleinere Vorschaubilder einzusetzen.
- Ergänzen Sie Ihre Seiten durch multimediale Elemente (zumindest ein Klang- und ein Video-Element). Achten Sie darauf, dass die Seiten sowohl im Firefox als auch im Internet Explorer angezeigt werden; installieren Sie ggf. benötigte Plugins.
- Erstellen Sie weitere Multimedia-Dateien (keine Bilder) oder verändern Sie aus dem Netz geladene Dateien für Ihre Zwecke. Suchen Sie dafür geeignete Programme.

Aufgabe 5



- Entwickeln Sie eine Datenbank, die Bestellvorgänge enthält. Eine Bestellung geht von einem Kunden aus und enthält eine Menge von Artikeln. Im einzelnen:
 - a. Überlegen Sie sich die auftretenden Entitäten und Attribute und stellen Sie ein Klassendiagramm auf, das die Beziehungen aufzeigt.
 - b. Entwickeln Sie daraus eine Menge von Tabellen, wobei ggf. einige Tabellen aus Relationen dazukommen.
 - c. Implementieren Sie die Tabellen und fügen Sie einige charakteristische Datensätze ein.
 - d. Formulieren Sie einige wichtige Fragen an die Datenbank und entwickeln Sie die zugehörigen SQL-Kommandos.

Aufgabe 6



- Entwickeln Sie SQL-Kommandos zur Beantwortung der folgenden Fragestellungen und testen Sie sie an der Beispiel-Datenbank:
 - a. Bestimmen Sie zu jedem Schlagwort die Zahl der passenden Bücher.
 - b. Bestimmen Sie für jeden Autor die Titel aller Bücher, an denen er mitgeschrieben hat.
 - c. Erstellen Sie eine Liste aller Autoren und der Verlage, bei denen sie Bücher veröffentlicht haben, sortiert nach Namen der Autoren. Achten Sie darauf, dass keine Einträge mehrmals vorkommen.
 - d. Ermitteln Sie die Namen aller Autoren, die Bücher zum Schlagwort "Thermodynamik" geschrieben haben, mit bzw. ohne Angabe der Buchtitel, jeweils sortiert nach Autornamen.
 - e. Ermitteln Sie die Namen aller Autoren, mit denen Stöcker zusammen ein Buch verfasst hat.
- Lösung

Aufgabe 7



- Untersuchen Sie das Phänomen der Schwebung, indem Sie Kurven plotten für die Funktionen

$$y_1 = \sin(\omega_1 t)$$

$$y_2 = \sin(\omega_2 t)$$

und ihre Summe $s = y_1 + y_2$. Wählen Sie zunächst die Werte

$$\omega_1 = 1.0$$

$$\omega_2 = 1.1$$

und untersuchen Sie dann das Verhalten bei Veränderung der Frequenzen. Wie groß ist jeweils die Schwebungsfrequenz?

- Lösung

Aufgabe 8



- Der Wert für die spezifische Wärmekapazität c_p von Wasser bei $T_1 = 23.1 \text{ °C}$ ist gesucht.
 - a. Die Datei `ex8_1.dat` enthält Ergebnisse von Messungen von $c_p(T)$. Auf einen Temperaturwert (in °C) folgt dabei immer der dazugehörige c_p -Wert (in $\text{kJ}/(\text{kg K})$). Fitten Sie eine Gerade, eine quadratische Parabel und eine Parabel 4. Grades an die Messwerte und ermitteln Sie so Näherungswerte für $c_p(T_1)$.
 - b. Plotten Sie die Messwerte zusammen mit den Fit-Kurven. Welche erscheint Ihnen am vertrauenswürdigsten?
- Hinweis: Die folgenden Funktionen könnten hilfreich sein:

`load, polyfit, polyval`

- Lösung

Aufgabe 9



- Finden Sie alle Lösungen der Gleichung

$$\tan(x) = x \ln(0.1 x)$$

im Intervall $[0, 10]$.

- Lösung

Aufgabe 10



- Vier Peilstationen befinden sich an den Orten

$$x_1 = (2.7, 2.5)$$

$$x_2 = (36.1, 5.5)$$

$$x_3 = (28.4, 31.9)$$

$$x_4 = (4.0, 24.6)$$

Sie messen jeweils die Richtung zu einem Störsender und geben dafür folgende Werte an (Abweichung in Grad zur Nordrichtung = positive y-Richtung, gegen den Uhrzeigersinn):

$$\alpha_1 = -64.5^\circ$$

$$\alpha_2 = 62.0^\circ$$

$$\alpha_3 = 147.6^\circ$$

$$\alpha_4 = -122.7^\circ$$

Welches ist die wahrscheinlichste Position des Störsenders ?

- Hinweise:

Berücksichtigen Sie zuerst nur zwei Peilstationen. Stellen Sie die zugehörigen Geradengleichungen auf, schreiben Sie sie als lineares 2x2-Gleichungssystem um und lösen Sie es mit Matlab.

Fügen Sie nun die zwei weiteren Gleichungen hinzu und lösen Sie erneut (diesmal im Sinne kleinster Quadrate).

- Lösung

Aufgabe 11



- Um die Leistung eines Verbrennungsmotors zu bestimmen, wurden für einen Zyklus des Kreisprozesses der Druck bei verschiedenen Kolbenständen (Volumina) gemessen. Die Tabelle `kreisprozess.dat` enthält diese Daten in der Form von drei Spalten mit Werten

$$V [\text{cm}^3] \quad p_{\text{unten}}[\text{bar}] \quad p_{\text{oben}} [\text{bar}]$$

wobei p_{unten} bzw. p_{oben} jeweils der untere bzw. obere Druck beim Volumen V sind.

- a. Stellen Sie den Kreisprozess graphisch im pV -Diagramm dar.
 - b. Bestimmen Sie die Leistung (für den einen gemessenen Zylinder), wenn die Drehzahl bei der Messung $n = 3000$ U/min betrug.
- Hinweis:

Zur Integration kann die Funktion `trapz` verwendet werden.

- Lösung

Aufgabe 12



- Berechnen Sie die Bahn $x(t)$ eines Körpers, der unter Reibungseinfluss frei fällt. Seine Bewegungsgleichung lautet

$$m \ddot{x} = -m g + F_R(x)$$

mit

- $F_R(x) = -b_1 \dot{x}$ (laminare Reibung)
- $F_R(x) = -b_2 \dot{x}^2 \text{sign}(\dot{x})$ (turbulente Reibung)
- Plotten Sie beide Kurven in einem Diagramm, zusammen mit der Bahn bei reibungslosem freien Fall. Plotten Sie ebenso die drei Geschwindigkeitskurven.
- Werte:

$$m = 75 \text{ kg}$$

$$g = 9.81 \text{ m/s}^2$$

$$b_1 = 5 \text{ kg/s}$$

$$b_2 = 0.1 \text{ kg/m}$$

$$x_0 = 5000 \text{ m}$$

$$v_0 = 0$$

- Lösung

- 📍 Literatur
- 📍 Nachweise
- 📍 HTML-Beispiele
- 📍 CSS-Beispiele
- 📍 Multimedia-Beispiele
- 📍 SQL-Beispiele
- 📍 Matlab-Beispiele

1. P. Junglas: cliXX Webapplikationen.
Verlag Harri Deutsch, 2004, ISBN: 3-8171-1713-2
2. Münz/Nefzger: HTML & Web-Publishing Handbuch
Franzis-Verlag, 2002, ISBN: 3-7723-7515-4
auch [online](#) verfügbar!
3. Beschreibung des HTML4-Standards beim [W3-Consortium](#)
<http://www.w3.org/TR/html4/>
4. E. Meyer: Eric Meyer on CSS
New Riders Publishing, 2002, ISBN: 978-3827268952
5. P. Henning: Taschenbuch Multimedia
Hanser Fachbuch, 4. Auflage 2007, ISBN: 978-3446409712
6. M. Kofler: MySQL
Addison-Wesley, 2. Aufl. 2003, ISBN: 3-8273-2046-1
7. Yarger/Reese/King: MySQL & mSQL
O'Reilly, 2000, ISBN: 3-8972-1163-7
8. Matthiesen/Unterstein:Relationale Datenbanken und SQL
Addison-Wesley, 2. Aufl. 2000, ISBN: 3-8273-1558-1
9. Eirund/Kohl: Datenbanken leicht gemacht
Teubner, 2000, ISBN: 3-5190-2644-9
10. Angermann, Beuschel, Rau, Wohlfarth: Matlab - Simulink- Stateflow
Oldenbourg, 2. Aufl 2003, ISBN: 3-486-27377-9

Nachweise



- [apollo15.mpeg](#) und [apollo15.ogg](#) basieren auf [Video Apollo_15_feather_and_hammer_drop.ogg](#) aus der freien Enzyklopädie [Wikipedia](#) und ist gemeinfrei. Es stammt von der NASA.
- [Bild 03](#) basiert auf dem [Bild Cd_Niederdruck_Spektrum.png](#) aus der freien Enzyklopädie [Wikipedia](#) und ist gemeinfrei. Der Urheber des Bildes ist User:Herbertweidner.
- [Bild 04](#) basiert auf dem [Bild Cone-response.png](#) aus der freien Enzyklopädie [Wikipedia](#) und steht unter der [GNU-Lizenz für freie Dokumentation](#). Der Urheber des Bildes ist User:Maxim Razin. Es basiert auf Bowmaker J.K. and Dartnall H.J.A., "Visual pigments of rods and cones in a human retina." J. Physiol. 298: pp501-511 (1980).

- 📄 [simple.html](#) (HTML-Code)
- 📄 [text.html](#) (HTML-Code)
- 📄 [block.html](#) (HTML-Code)
- 📄 [list.html](#) (HTML-Code)
- 📄 [table.html](#) (HTML-Code)
- 📄 [forms1.html](#) (HTML-Code)
- 📄 [forms2.html](#) (HTML-Code)
- 📄 [multimedia.html](#) (HTML-Code)
- 📄 [multimedia2.html](#) (HTML-Code)

simple.html



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Praktische Informatik 1</title>
  </head>

  <body style="background-color:#0fffa7;color:#2339ff;">
    <h1>Die kleine Homepage</h1>

    <p>Jeder fängt mal klein an - so auch diese Seite.</p>

    <p style="color:#ff2080;">
      Über Geschmack lässt sich bekanntlich <em>nicht</em>
      streiten. Völlig unbestreitbar ist aber, dass ich bei der Gestaltung
      dieser Seite keinen bewiesen habe.</p>

    <p>Eine meiner Lieblingsseiten zeigt <a href="http://heritage.stsci.edu/">
      das Schönste vom Hubble-Teleskop</a>.
    </p>

    <p></p>

    <address>Viele Grüße von
      <a href="mailto:peter@peter-junglas.de">Peter</a></address>
  </body>
</html>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=ISO-8859-15">
    <title>Textelemente</title>
  </head>

  <body>
    <h1>Textelemente</h1>

    <p>Die richtige <em>Betonung</em> ist wichtig, vor allem bei diesem
berühmten Gedicht:</p>
    <p>
      <cite> Zu Dionys, dem Tyrannen schlich Damon, den Dolch im Gewande</cite>
    </p>

    <p>Weniger wichtig ist die Betonung bei diesem Werk der Prosa-Literatur:
    </p>
    <code> for (i=0;i<100;i++) {System.out.println("&quot;Du sollst dem Lehrer
zuhören("&quot;);} </code></p>

    <p>Im folgenden altgriechischen Gedicht liegt die Betonung ganz klar auf
den Höhen und Tiefen</p>
    <p><cite> a<sup>2</sup> + b<sup>2</sup> = c<sup>2</sup></cite></p>
  </body>
</html>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=ISO-8859-15">
    <title>Absatzelemente</title>
  </head>

  <body>
    <h1>Absatzelemente</h1>

    <h2>Deutschstunde (von Heinz Friedlich Fr&uuml;hling)</h2>

    <p>Wer wagt es, Rattersmann oder Knipp,<br>
      zu schlauchen in diesen Tund?</p>
    <p> es folgt ... <br><br><br><br><br><br>
      g&auml;hnende Lehre</p>

    <h2>Sportstunde</h2>
    <p>Zum Aufwachen ein wenig Gymnastik:</p>
    <p style="text-align:right;">Wir drehen den Kopf alle mal nach links,</p>
    <p style="text-align:center;"> dann schauen wir nach vorne.</p>
    <p style="text-align:left;">und schliesslich nach rechts.</p>

    <h2>Informatik</h2>
    <p>Wir wiederholen vom letzten Mal:</p>
    <pre>
    for (i=0; i &lt; 100; i++) {
      System.out.println(&quot;Du sollst dem Lehrer zuh&ouml;ren&quot;);
    }
    </pre>
  </body>
</html>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=ISO-8859-15">
    <title>Listen</title>
  </head>

  <body>
    <h1>Listen</h1>

    <P>Liste mit disc:</P>
    <UL STYLE="list-style-type:disc">
      <LI> Taschentuch</LI>
      <LI> Gebetbuch</LI>
      <LI> Kamm</LI>
    </UL>

    <P>Liste mit square:</P>
    <UL STYLE="list-style-type:square">
      <LI> Kinder</LI>
      <LI> K&uuml;che</LI>
      <LI> Kirche</LI>
    </UL>

    <P>gemischte Liste:</P>
    <UL>
      <LI STYLE="list-style-type:square"> Welt</LI>
      <LI STYLE="list-style-type:none"> Weites</LI>
      <LI STYLE="list-style-type:circle"> Warten</LI>
    </UL>

    <P>Z&auml;hlen mit Buchstaben:</P>
    <OL STYLE="list-style-type:lower-alpha">
      <LI> Schafft</LI>
      <LI> Arbeits-</LI>
      <LI> Pl&auml;tze</LI>
    </OL>

    <P>Und jetzt wirts verr&uuml;ckt:</P>
    <OL>
      <LI STYLE="list-style-type:decimal"> Ich</LI>
      <LI STYLE="list-style-type:lower-roman"> Bin</LI>
      <LI STYLE="list-style-type:upper-alpha"> M&uuml;de</LI>
    </OL>
  </body>
</html>
```

table.html



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=ISO-8859-15">
    <title>Tabellen</title>
  </head>

  <body>
    <h1>Tabellen</h1>

    <p>Eine Tabelle, die sich breit macht:</p>
    <table border="12" cellpadding="20">
      <tr>
        <td> 1. Zeile, 1. Spalte</td>
        <td> 1. Zeile, 2. Spalte</td>
      </tr>
      <tr>
        <td> 2. Zeile, 1. Spalte</td>
        <td> 2. Zeile, 2. Spalte</td>
      </tr>
    </table>

    <p>Eine Tabelle mit Ordnung</p>
    <table border="2" width="60%">
      <tr>
        <td align="left">links</td>
        <td align="left">rechts</td>
      </tr>
      <tr>
        <td align="center" style="height:30px;">links</td>
        <td align="center">rechts</td>
      </tr>
      <tr>
        <td align="right" style="height:50px;">links</td>
        <td align="right">rechts</td>
      </tr>
    </table>

    <p>Eine unauff&auml;llige Tabelle mit H&ouml;hen und Tiefen</p>
    <table width="60%">
      <tr>
        <td style="height:50px;" valign="top">Schau</td>
        <td valign="middle">nach</td>
        <td valign="bottom">rechts:</td>
        <td></td>
      </tr>
      <tr>
        <td>wohin?</td>
        <td>wohin?</td>
        <td>wohin?</td>
        <td> <em>Ach da!</em> </td>
      </tr>
    </table>
```

```
</tr>
<tr>
  <td style="height:50px;" valign="bottom">Schau</td>
  <td valign="middle" style="height:50px;">nach</td>
  <td valign="top">rechts:</td>
  <td></td>
</tr>
</table>
</body>
</html>
```

forms1.html



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Einfaches Formular</title>
  </head>
  <body style="background-color:#d2ffc8">
    <h1>Einfaches Formular</h1>

    <p>Willkommen zum Online-Kurs. Bitte melden Sie sich an:</p>

    <form action="echo.php" method="post">
      <p>
Benutzer: <input name="user" size="20" value="blabla"><br>
Passwort: <input type="password" name="passwort" size="20"><br>
<input type="submit" value="Anmelden">
<input type="reset" value="Zurücksetzen">
      </p>
    </form>

  </body>
</html>
```

forms2.html



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Java - die Klausur</title>
  </head>
  <body style="background-color:#d2ffc8">
    <h1>Java - die Klausur</h1>

    <p>Willkommen zur Java-Klausur. Bitte beantworten Sie die folgenden
    Fragen:</p>

    <form action="echo.php" method="post">
      <table cellpadding="10">
        <tr valign="top">
<td>Welche Eigenschaften hat Java:<br>
        <select name="java_eigenschaften[]" size="3" multiple>
          <option>objektorientiert</option>
          <option>leicht verständlich</option>
          <option>modern</option>
          <option>internet-fähig</option>
          <option>weit verbreitet</option>
        </select></td>
        <td>Wie wird ein Java-Programm (normalerweise) übersetzt:<br>
        <select name="compiler" size="1">
<option>mit einem Java-Compiler in Maschinsprache</option>
          <option>mit einem Java-Compiler in einen universellen Bytecode</option>
          <option>zeilenweise von einem Interpreter</option>
          <option>gar nicht, Java ist eine Skriptsprache</option>
          <option>gar nicht, der Prozessor versteht Java direkt</option>
        </select></td>
</tr>
        <tr valign="top">
          <td>Wie lautet die erste aufgerufene Methode eines Java-Programms:<br>
          <input type="radio" name="methode" value="Program">program <br>
          <input type="radio" name="methode" value="System">system <br>
          <input type="radio" name="methode" value="Main">main
          </td>
          <td>Wie gibt man in Java eine Zeichenkette s aus:<br>
          <input type="radio" name="output" value="cpp">cout &lt;&lt; s; <br>
          <input type="radio" name="output" value="java">System.out.print(s);<br>
          <input type="radio" name="output" value="fortran">write (2, '(A)'), s<br>
          <input type="radio" name="output" value="c">printf(&quot;%s&quot;, s);
          </td>
</tr>
        <tr valign="top">
          <td>Welche der folgenden Klassen sind in Java vordefiniert: <br>
          <input type="checkbox" name="klassen[]" value="1">System<br>
          <input type="checkbox" name="klassen[]" value="2">Input<br>
          <input type="checkbox" name="klassen[]" value="3">Double<br>
          <input type="checkbox" name="klassen[]" value="4">Complex<br>
          <input type="checkbox" name="klassen[]" value="5">Image
          </td>
          <td>Schreiben Sie eine Java-Methode, die Fibonacci-Zahlen
          berechnet:<br>
          <textarea name="java_programm" rows="10" cols="40">

```

```
int fibo(int n) {
    // Ihr Text hier
}

    </textarea>
    </td>
</tr>
    <tr valign="top">
        <td align="right"><input type="submit" value="Absenden">
</td>
        <td align="left"><input type="reset" value="Zurücksetzen">
        </td>
    </tr>
</table>
</form>

</body>
</html>
```

multimedia.html



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Praktische Informatik 1</title>
  </head>
  <body>
    <h1>Kino mit Plugin</h1>
    <p><object data="../multimedia/apollo15.mpeg" type="video/mpeg"
      width="768" height="576">
      Ihr Browser kann das Objekt leider nicht anzeigen!
    </object></p>

    <p>Ein historischer Versuch bei der Apollo 15-Mission</p>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>Praktische Informatik 1</title>
  </head>
  <body>
    <h1>Kino mit Plugin</h1>

    <p><a>
<video src="../../multimedia/apollo15.ogg"
        width="768" height="576" controls>
    </video>
    </a></p>

    <p>Ein historischer Versuch bei der Apollo 15-Mission</p>
  </body>
</html>
```

CSS-Beispiele



- [austen.css \(cssdemo.html\)](#)
- [ohne Stylesheet \(cssdemo01.html\)](#)
- [austen02.css \(cssdemo02.html\)](#)
- [austen03.css \(cssdemo03.html\)](#)
- [austen04.css \(cssdemo04.html\)](#)
- [austen05.css \(cssdemo05.html\)](#)
- [austen06.css \(cssdemo06.html\)](#)
- [austen07.css \(cssdemo07.html\)](#)

```
body {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  color: black;
  background: white;
  margin: 0px;
}
table {
  width: 100%;
  margin: 0px;
}
table td {
  padding: 0px;
  border-width: 0;
  vertical-align: top;
}
a:link {
  color: #660000;
}
a:visited {
  color: #990000;
}
h1 {
  font-size: 180%;
  font-weight: bold;
  color: #602020;
  border-bottom: 3px solid #804040;
  padding-bottom: 2px;
}

td#title {
  vertical-align: middle;
  background: url(../images/bgpaper.jpg);
  color: #501000;
  font: bold 600% "Comic Sans MS", Verdana, sans-serif;
  padding-left: 5px;
}
td#portrait {
  width: 253px;
}
td#portrait img {
  vertical-align: bottom; /* Unterlänge des "Buchstabens" muss weg */
}

#content-top td {
  vertical-align: middle;
  color: white;
  font-weight: bold;
  padding: 0.1em 0.2em 0;
}
td#content-left {
  background: #501000;
  font: bold 115% Arial, Helvetica, Verdana, sans-serif;
}
td#content-links {
  background: #997753;
  font-size: 85%;
}
```

```

td#content-links a:link {
  color: white;
  padding: 10px;
}
td#content-links a:visited {
  color: gray;
  padding: 10px;
}

td#leftside {
  width: 120px;
  background: #f0e0d0;
}
td#leftside ul {
  list-style-type: none;
  list-style-position: inside;
  margin-top: 3px;
  padding: 0px;
}
td#leftside li {
  font-size: 85%;
  border-bottom: 1px solid #A98763;
  padding: 0 0 1px 4px;
}

td#content {
  padding: 17px 42px;
}
td#content p {
  font: 10pt Arial, Helvetica, sans-serif;
}

td#rightside {
  width: 150px;
}
td#rightside h3 {
  text-align: center;
  font-size: 85%;
  background: #774411;
  color: white;
  padding: 2px;
  margin: 0px;
}
td#rightside td {
  font-size: 66%;
  padding: 1px 3px 1px 1px;
}

tr#footer td {
  text-align: center;
  vertical-align: middle;
  font-size: 66%;
  border-top: 3px solid #d6c3ab;
  padding: 0.2em;
}
td#feedback {
  background: #EFE1D1;
}
tr#footer td#author {
  font-size: 85%;

```

```

}
tr#footer td#date {
    text-align: right;
    font-style: italic;
    color: #999;
}

div#cite {
    border: 3px solid #804040;
    background: #EBDAC6;
    text-align: center;
    margin-top: 1.5em;
    margin-right: 3px;
    padding: 8px;
    font-size: 66%;
}

div#cite h4 {
    margin: 0px;
}

div.infobox {
    float: right;
    width: 140px;
    color: #A09080;
    border: solid #908070;
    border-width: 7px 0;
    font: bold 1em Arial, Helvetica, Verdana, sans-serif;
    padding: 3px 2px;
    margin: 1px 70px 1px 7px;
}

h4.rightside-head {
    background: #D6B58C;
    text-align: center;
    font-weight: bold;
    font-size: 66%;
    margin: 0px;
    padding: 1px 0 1px 0;
}

tr.even td {
    background: #F7F0E7;
}

tr.odd td {
    background: white;
}

td.preis {
    color: #660;
    width: 5em;
    text-align: right;
}

```

austen02.css



```
table {border: 2px solid red; margin: 3px;}
td {border: 1px dotted red; padding: 2px;}
ul {border: 2px solid blue;}
h1, h2, h3, h4, h5, h6, h7 {border: 1px solid green;}
div {border: 1px solid black;}
span {border: 1px dotted black;}
```

austen03.css



```
body {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  color: black;
  background: white;
  margin: 0px;
}
table {
  width: 100%;
  margin: 0px;
}
table td {
  padding: 0px;
  border-width: 0;
  vertical-align: top;
}
a:link {
  color: #660000;
}
a:visited {
  color: #990000;
}
h1 {
  font-size: 180%;
  font-weight: bold;
  color: #602020;
  border-bottom: 3px solid #804040;
  padding-bottom: 2px;
}
```

austen04.css



```
body {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  color: black;
  background: white;
  margin: 0px;
}
table {
  width: 100%;
  margin: 0px;
}
table td {
  padding: 0px;
  border-width: 0;
  vertical-align: top;
}
a:link {
  color: #660000;
}
a:visited {
  color: #990000;
}
h1 {
  font-size: 180%;
  font-weight: bold;
  color: #602020;
  border-bottom: 3px solid #804040;
  padding-bottom: 2px;
}

td#title {
  vertical-align: middle;
  background: url(../images/bgpaper.jpg);
  color: #501000;
  font: bold 600% "Comic Sans MS", Verdana, sans-serif;
  padding-left: 5px;
}
td#portrait {
  width: 253px;
}
td#portrait img {
  vertical-align: bottom; /* Unterlänge des "Buchstabens" muss weg */
}
```

austen05.css



```
body {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  color: black;
  background: white;
  margin: 0px;
}
table {
  width: 100%;
  margin: 0px;
}
table td {
  padding: 0px;
  border-width: 0;
  vertical-align: top;
}
a:link {
  color: #660000;
}
a:visited {
  color: #990000;
}
h1 {
  font-size: 180%;
  font-weight: bold;
  color: #602020;
  border-bottom: 3px solid #804040;
  padding-bottom: 2px;
}

td#title {
  vertical-align: middle;
  background: url(../images/bgpaper.jpg);
  color: #501000;
  font: bold 600% "Comic Sans MS", Verdana, sans-serif;
  padding-left: 5px;
}
td#portrait {
  width: 253px;
}
td#portrait img {
  vertical-align: bottom; /* Unterlänge des "Buchstabens" muss weg */
}

#content-top td {
  vertical-align: middle;
  color: white;
  font-weight: bold;
  padding: 0.1em 0.2em 0;
}
td#content-left {
  background: #501000;
  font: bold 115% Arial, Helvetica, Verdana, sans-serif;
}
td#content-links {
  background: #997753;
  font-size: 85%;
}
```

```
td#content-links a:link {
  color: white;
  padding: 10px;
}
td#content-links a:visited {
  color: gray;
  padding: 10px;
}
```

austen06.css



```
body {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  color: black;
  background: white;
  margin: 0px;
}
table {
  width: 100%;
  margin: 0px;
}
table td {
  padding: 0px;
  border-width: 0;
  vertical-align: top;
}
a:link {
  color: #660000;
}
a:visited {
  color: #990000;
}
h1 {
  font-size: 180%;
  font-weight: bold;
  color: #602020;
  border-bottom: 3px solid #804040;
  padding-bottom: 2px;
}

td#title {
  vertical-align: middle;
  background: url(../images/bgpaper.jpg);
  color: #501000;
  font: bold 600% "Comic Sans MS", Verdana, sans-serif;
  padding-left: 5px;
}
td#portrait {
  width: 253px;
}
td#portrait img {
  vertical-align: bottom; /* Unterlänge des "Buchstabens" muss weg */
}

#content-top td {
  vertical-align: middle;
  color: white;
  font-weight: bold;
  padding: 0.1em 0.2em 0;
}
td#content-left {
  background: #501000;
  font: bold 115% Arial, Helvetica, Verdana, sans-serif;
}
td#content-links {
  background: #997753;
  font-size: 85%;
}
```

```

td#content-links a:link {
  color: white;
  padding: 10px;
}
td#content-links a:visited {
  color: gray;
  padding: 10px;
}

td#leftside {
  width: 120px;
  background: #f0e0d0;
}
td#leftside ul {
  list-style-type: none;
  list-style-position: inside;
  margin-top: 3px;
  padding: 0px;
}
td#leftside li {
  font-size: 85%;
  border-bottom: 1px solid #A98763;
  padding: 0 0 1px 4px;
}

td#content {
  padding: 17px 42px;
}
td#content p {
  font: 10pt Arial, Helvetica, sans-serif;
}

td#rightside {
  width: 150px;
}
td#rightside h3 {
  text-align: center;
  font-size: 85%;
  background: #774411;
  color: white;
  padding: 2px;
  margin: 0px;
}
td#rightside td {
  font-size: 66%;
  padding: 1px 3px 1px 1px;
}

tr#footer td {
  text-align: center;
  vertical-align: middle;
  font-size: 66%;
  border-top: 3px solid #d6c3ab;
  padding: 0.2em;
}
td#feedback {
  background: #EFE1D1;
}
tr#footer td#author {
  font-size: 85%;

```

```
}  
tr#footer td#date {  
  text-align: right;  
  font-style: italic;  
  color: #999;  
}
```

austen07.css



```
body {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  color: black;
  background: white;
  margin: 0px;
}
table {
  width: 100%;
  margin: 0px;
}
table td {
  padding: 0px;
  border-width: 0;
  vertical-align: top;
}
a:link {
  color: #660000;
}
a:visited {
  color: #990000;
}
h1 {
  font-size: 180%;
  font-weight: bold;
  color: #602020;
  border-bottom: 3px solid #804040;
  padding-bottom: 2px;
}

td#title {
  vertical-align: middle;
  background: url(../images/bgpaper.jpg);
  color: #501000;
  font: bold 600% "Comic Sans MS", Verdana, sans-serif;
  padding-left: 5px;
}
td#portrait {
  width: 253px;
}
td#portrait img {
  vertical-align: bottom; /* Unterlänge des "Buchstabens" muss weg */
}

#content-top td {
  vertical-align: middle;
  color: white;
  font-weight: bold;
  padding: 0.1em 0.2em 0;
}
td#content-left {
  background: #501000;
  font: bold 115% Arial, Helvetica, Verdana, sans-serif;
}
td#content-links {
  background: #997753;
  font-size: 85%;
}
```

```

td#content-links a:link {
  color: white;
  padding: 10px;
}
td#content-links a:visited {
  color: gray;
  padding: 10px;
}

td#leftside {
  width: 120px;
  background: #f0e0d0;
}
td#leftside ul {
  list-style-type: none;
  list-style-position: inside;
  margin-top: 3px;
  padding: 0px;
}
td#leftside li {
  font-size: 85%;
  border-bottom: 1px solid #A98763;
  padding: 0 0 1px 4px;
}

td#content {
  padding: 17px 42px;
}
td#content p {
  font: 10pt Arial, Helvetica, sans-serif;
}

td#rightside {
  width: 150px;
}
td#rightside h3 {
  text-align: center;
  font-size: 85%;
  background: #774411;
  color: white;
  padding: 2px;
  margin: 0px;
}
td#rightside td {
  font-size: 66%;
  padding: 1px 3px 1px 1px;
}

tr#footer td {
  text-align: center;
  vertical-align: middle;
  font-size: 66%;
  border-top: 3px solid #d6c3ab;
  padding: 0.2em;
}
td#feedback {
  background: #EFE1D1;
}
tr#footer td#author {
  font-size: 85%;

```

```

}
tr#footer td#date {
    text-align: right;
    font-style: italic;
    color: #999;
}

div#cite {
    border: 3px solid #804040;
    background: #EBDAC6;
    text-align: center;
    margin-top: 1.5em;
    margin-right: 3px;
    padding: 8px;
    font-size: 66%;
}

div#cite h4 {
    margin: 0px;
}

div.infobox {
    float: right;
    width: 140px;
    color: #A09080;
    border: solid #908070;
    border-width: 7px 0;
    font: bold 1em Arial, Helvetica, Verdana, sans-serif;
    padding: 3px 2px;
    margin: 1px 70px 1px 7px;
}

h4.rightside-head {
    background: #D6B58C;
    text-align: center;
    font-weight: bold;
    font-size: 66%;
    margin: 0px;
    padding: 1px 0 1px 0;
}

tr.even td {
    background: #F7F0E7;
}

tr.odd td {
    background: white;
}

td.preis {
    color: #660;
    width: 5em;
    text-align: right;
}

```

Multimedia-Beispiele



- [Video Apollo 15 \(MPG\)](#)
- [Video Apollo 15 \(Theora\)](#)
- [Applet "Additive Farbmischung"](#)
- [Applet "Subtraktive Farbmischung"](#)

SQL-Beispiele



- Beispiel-Datenbank
- Ergebnisse der SQL-Abfragen für die Beispiel-Datenbank

Beispiel-Datenbank



- Tabelle Buch:

titel	seiten	jahr	auflage	preis	isbn	verlag_id
Physik für Ingenieure	744	1999	7	79.00	3540661352	3
Taschenbuch der Physik	1092	2000	4	68.00	3817116284	1
Fundamentals of Physics Extended	1198	1997	5	293.37	471105597	2
Physik - Der Grundkurs	426	1999	1	38.00	3817115768	1
Einführung in die Thermodynamik	468	1999	12	59.80	3446211101	4
Thermodynamik für Ingenieure	NULL	1999	2	54.00	3528147857	5
Relationale Datenbanken und SQL	352	2000	2	59.90	3827315581	6
MySQL & mSQL	512	2000	1	79.00	3897211637	7
Thermodynamics and Statistical Mechanics	463	2000	3	114.66	3540942998	3
Grundlagen der Gastechnik	515	NULL	5	79.80	3446211098	4

- Tabelle Autor:

vorname	name	autor_id
Horst	Stöcker	1
David	Halliday	2
Robert	Resnick	3
Jearl	Walker	4
Günter	Cerbe	5
Walter	Greiner	6
Ludwig	Neise	7
Klaus	Langeheinecke	8
Peter	Jany	9
Eugen	Sapper	10
Rudolf	Pitka	11
Steffen	Bohrmann	12
Günther	Terlecki	13
Hans-Joachim	Hoffmann	14
Günter	Matthiesen	15
Michael	Unterstein	16
Randy	Yarger	17
George	Reese	18
Tim	King	19
Ekbert	Hering	20
Rolf	Martin	21
Martin	Stohrer	22

- Tabelle Verlag:

name	ort	verlag_id
Harri Deutsch	Frankfurt	1
Wiley	New York	2
Springer	Heidelberg	3
Hanser	München	4
Vieweg	Wiesbaden	5
Addison-Wesley	Boston	6
O'Reilly	Cambridge	7

- Tabelle Schlagwort:

wort	wort_id
Physik	1
Thermodynamik	2
Java	3
Quantenmechanik	4
SQL	5
Datenbank	6
Lehrbuch	7
Handbuch	8
Ingenieurwesen	9
Statistische Physik	10

- Tabelle BuchAutor:

autor_id	isbn
2	471105597
3	471105597
4	471105597
5	3446211098
5	3446211101
14	3446211101
8	3528147857
9	3528147857
10	3528147857
20	3540661352
21	3540661352
22	3540661352
1	3540942998
6	3540942998
7	3540942998
1	3817115768
11	3817115768
12	3817115768
13	3817115768
1	3817116284
15	3827315581
16	3827315581
17	3897211637
18	3897211637
19	3897211637

- Tabelle BuchSchlagwort:

wort_id	isbn
1	471105597
2	471105597
4	471105597
7	471105597
2	3446211101
7	3446211101
2	3528147857
7	3528147857
9	3528147857
1	3540661352
2	3540661352
4	3540661352
7	3540661352
1	3540942998
2	3540942998
10	3540942998
1	3817115768
2	3817115768
7	3817115768
1	3817116284
2	3817116284
4	3817116284
8	3817116284
5	3827315581
6	3827315581
7	3827315581
5	3897211637
6	3897211637
8	3897211637

Ergebnisse der SQL-Abfragen für die Beispiel-Datenbank



- Beispiel 1

Abfrage

```
SELECT titel, seiten, jahr FROM Buch
ORDER BY jahr, seiten DESC;
```

Ergebnis

titel	seiten	jahr
Grundlagen der Gastechnik	515	NULL
Fundamentals of Physics Extended	1198	1997
Physik für Ingenieure	744	1999
Einführung in die Thermodynamik	468	1999
Physik - Der Grundkurs	426	1999
Thermodynamik für Ingenieure	NULL	1999
Taschenbuch der Physik	1092	2000
MySQL & mSQL	512	2000
Thermodynamics and Statistical Mechanics	463	2000
Relationale Datenbanken und SQL	352	2000

- Beispiel 2

Abfrage

```
SELECT titel, seiten, jahr FROM Buch
WHERE jahr >= 2000;
```

Ergebnis

titel	seiten	jahr
Taschenbuch der Physik	1092	2000
Relationale Datenbanken und SQL	352	2000
MySQL & mSQL	512	2000
Thermodynamics and Statistical Mechanics	463	2000

- Beispiel 3

Abfrage

```
SELECT titel, seiten, jahr FROM Buch
WHERE titel LIKE "%Physi%";
```

Ergebnis

titel	seiten	jahr
Physik für Ingenieure	744	1999
Taschenbuch der Physik	1092	2000
Fundamentals of Physics Extended	1198	1997
Physik - Der Grundkurs	426	1999

- Beispiel 4

Abfrage

```
SELECT titel, preis/2.04 FROM Buch
WHERE titel LIKE "%Physics%";
```

Ergebnis

titel	preis/2.04
Fundamentals of Physics Extended	143.8088

- Beispiel 5

Abfrage

```
SELECT titel, name FROM Buch NATURAL JOIN Verlag;
```

Ergebnis

titel	name
Physik für Ingenieure	Springer
Taschenbuch der Physik	Harri Deutsch
Fundamentals of Physics Extended	Wiley
Physik - Der Grundkurs	Harri Deutsch
Einführung in die Thermodynamik	Hanser
Thermodynamik für Ingenieure	Vieweg
Relationale Datenbanken und SQL	Addison-Wesley
MySQL & mSQL	O'Reilly
Thermodynamics and Statistical Mechanics	Springer
Grundlagen der Gastechnik	Hanser

- Beispiel 6

Abfrage

```
SELECT titel, name AS verlag
FROM Buch NATURAL JOIN Verlag
ORDER BY verlag;
```

Ergebnis

titel	verlag
Relationale Datenbanken und SQL	Addison-Wesley
Grundlagen der Gastechnik	Hanser
Einführung in die Thermodynamik	Hanser
Taschenbuch der Physik	Harri Deutsch
Physik - Der Grundkurs	Harri Deutsch
MySQL & mSQL	O'Reilly
Physik für Ingenieure	Springer
Thermodynamics and Statistical Mechanics	Springer
Thermodynamik für Ingenieure	Vieweg
Fundamentals of Physics Extended	Wiley

- Beispiel 7

Abfrage

```
SELECT COUNT(*), SUM(preis), jahr FROM Buch
GROUP BY jahr;
```

Ergebnis

COUNT(*)	SUM(preis)	jahr
1	79.80	NULL
1	293.37	1997
4	230.80	1999
4	321.56	2000

- Beispiel 8

Abfrage

```
SELECT name as verlag, COUNT(*)
FROM Buch NATURAL JOIN Verlag
GROUP BY name;
```

Ergebnis

verlag	COUNT(*)
Addison-Wesley	1
Hanser	2
Harri Deutsch	2
O'Reilly	1
Springer	2
Vieweg	1
Wiley	1

- Beispiel 9

Abfrage

```
SELECT name AS verlag , SUM(preis) AS gesamtpreis
FROM Buch NATURAL JOIN Verlag
GROUP BY verlag
ORDER BY gesamtpreis DESC;
```

Ergebnis

verlag	gesamtpreis
Wiley	293.37
Springer	193.66
Hanser	139.60
Harri Deutsch	106.00
O'Reilly	79.00
Addison-Wesley	59.90
Vieweg	54.00

- 📄 zeichneFunktion.m
- 📄 demo1.dat
- 📄 trapez1.m
- 📄 trapez2.m
- 📄 trapez.m
- 📄 glocke.m
- 📄 radioaktiv.m
- 📄 radiokette.m
- 📄 erzwungen.m
- 📄 erzwungenp.m
- 📄 exercise7.m
- 📄 ex8_1.dat
- 📄 exercise8.m
- 📄 exercise9.m
- 📄 exercise10.m
- 📄 exercise10a.m
- 📄 kreisprozess.dat
- 📄 exercise11.m
- 📄 exercise12.m
- 📄 laminar.m
- 📄 turbulent.m
- 📄 freifall.m

zeichneFunktion.m



```
% Beispiel: Plot von Funktionen
t = 0:0.1:4*pi;
y = sin(t);
y2 = 0.5*log(t);
y3 = exp(-0.1*t.*t);

plot(t, y, 'bo', t, y2, 'g.', t, y3, 'r-.', [0 4*pi], [0 0], 'k-')
title('Meine Lieblingsfunktionen', ...
      'FontSize', 14, 'FontWeight', 'bold')
xlabel('Zeit');
ylabel('Amplitude');
legend('Sinus', 'Logarithmus', 'Glocke', 'Location', 'Best');
xlim([-0.1, 4*pi + 0.1])

% Plot als Datei abspeichern
F = getframe(gcf);
imwrite(F.cdata, 'bild25.png');
```

demo1.dat



0.0000000e+00	3.8941834e-01	7.1735609e-01	9.3203909e-01	9.9957360e-01	9.0929743e-01	6.7546318e-01
3.8941834e-01	5.3599459e-01	7.7985065e-01	9.5358074e-01	9.9692470e-01	8.9210592e-01	6.5068611e-01
7.1735609e-01	7.7985065e-01	9.0499623e-01	9.9174551e-01	9.7631940e-01	8.3466637e-01	5.7431777e-01
9.3203909e-01	9.5358074e-01	9.9174551e-01	9.9203980e-01	9.0929743e-01	7.2374355e-01	4.4243411e-01
9.9957360e-01	9.9692470e-01	9.7631940e-01	9.0929743e-01	7.7000625e-01	5.4831078e-01	2.5432688e-01
9.0929743e-01	8.9210592e-01	8.3466637e-01	7.2374355e-01	5.4831078e-01	3.0807174e-01	1.7491891e-01
6.7546318e-01	6.5068611e-01	5.7431777e-01	4.4243411e-01	2.5432688e-01	1.7491891e-02	-2.4984473e-01
3.3498815e-01	3.0807174e-01	2.2753808e-01	9.5139298e-02	-8.3214108e-02	-2.9488719e-01	-5.1946534e-01
-5.8374143e-02	-8.3214108e-02	-1.5624899e-01	-2.7251774e-01	-4.2242231e-01	-5.9075947e-01	-7.5680250e-01
-4.4252044e-01	-4.6227706e-01	-5.1946534e-01	-6.0768355e-01	-7.1592666e-01	-8.2863195e-01	-9.2652499e-01
-7.5680250e-01	-7.6969138e-01	-8.0615387e-01	-8.5962229e-01	-9.1939485e-01	-9.7127780e-01	-9.9886603e-01

trapez1.m



```
% Integration mit der Trapezregel
% anpassen:
%     Grenzen a, b
%     Funktion (statt sin)
%     Zahl der Intervalle N

N = 100;
a = 0;
b = pi/2;
h = (b-a)/N;
x = a:h:b;
y = sin(x);
I = h*(0.5*y(1) + sum(y(2:N)) + 0.5*y(N+1))
```

trapez2.m



```
function I = trapez2(a, b, N)
% Integration mit der Trapezregel
%   Argumente
%       Grenzen a, b
%       Zahl der Intervalle N
%   Ergebnis
%       Integral des Sinus von a bis b,
%       genaehert mit Trapezregel mit N Intervallen

h = (b-a)/N;
x = a:h:b;
y = sin(x);
I = h*(0.5*y(1) + sum(y(2:N)) + 0.5*y(N+1));
```

trapez.m



```
function I = trapez(func, a, b, N)
% Integration mit der Trapezregel
%   Argumente
%       zu integrierende Funktion func
%       Grenzen a, b
%       Zahl der Intervalle N
%   Ergebnis
%       Integral von func von a bis b,
%       genaehert mit Trapezregel mit N Intervallen

h = (b-a)/N;
x = a:h:b;
y = func(x);
I = h*(0.5*y(1) + sum(y(2:N)) + 0.5*y(N+1));
```

glocke.m



```
function y = glocke(x)
% Glockenfunktion e^(-x^2)
y = exp(-x.^2);
```

radioaktiv.m



```
function dydt = radioaktiv(t, y)
% rechte Seite der DGL beim radioaktiven Zerfall
% Parameter lambda fest
lambda = 0.1;
dydt = -lambda*y;
```

radiokette.m



```
function dydt = radiokette(t, y)
% rechte Seite der DGL bei radioaktiver Zerfallskette

% Parameter lambda1, lambda2 fest
lambda1 = 0.1;
lambda2 = 0.03;

dy1 = -lambda1*y(1);
dy2 = lambda1*y(1)-lambda2*y(2);
dydt = [dy1 dy2]';          % Spaltenvektor
```

erzwungen.m



```
function dydt = erzwungen(t, y)
% rechte Seite der DGL bei erzwungener Schwingung

% Parameter
m = 1.0;
b = 0.3;
c = 1.0;
A = 1.0;
omega = 1.0;

dy1 = y(2);
dy2 = -b/m*y(2) - c/m*y(1) + A/m*cos(omega*t);
dydt = [dy1 dy2]';
```

erzwungenp.m



```
function dydt = erzwungenp(t, y, m, b, c, A, omega)
% rechte Seite der DGL bei erzwungener Schwingung
% Parameter:
% Masse: m
% Reibungskoeffizient: b
% Federkonstante: c
% Amplitude der aeusseren Kraft: A
% Kreisfrequenz der aeusseren Kraft: omega

dy1 = y(2);
dy2 = -b/m*y(2) - c/m*y(1) + A/m*cos(omega*t);
dydt = [dy1 dy2]';
```

exercise7.m



```
% Loesung von Aufgabe 7
```

```
t = 0:0.1:20*pi;
```

```
omega1 = 1.0;
```

```
omega2 = 1.2;
```

```
y1 = sin(omega1*t);
```

```
y2 = sin(omega2*t);
```

```
y = y1 + y2;
```

```
plot(t, y1, 'm-', t, y2, 'g-', t, y, 'k-')
```

6.99
4.202
10.25
4.184
25.00
4.170
29.03
4.173
41.40
4.173
49.73
4.173
53.23
4.177
64.52
4.187
70.97
4.189
95.70
4.209

exercise8.m



```
% Loesung von Aufgabe 8

% Wert der Temperatur
t1 = 23.1; % in Grad Celsius

% Einlesen der Messwerte
load -ascii ex8_1.dat
t_mess = ex8_1(1:2:end);
cp_mess = ex8_1(2:2:end);

% Fitfunktionen
lin = polyfit(t_mess, cp_mess, 1);
para2 = polyfit(t_mess, cp_mess, 2);
para4 = polyfit(t_mess, cp_mess, 4);

% Bestimmen des Schaetzwertes
cp1 = polyval(lin, t1)
cp2 = polyval(para2, t1)
cp4 = polyval(para4, t1)

% Plotten der Messwerte und Interpolationsfunktionen
t = 0:100;
cp_lin = polyval(lin, t);
cp_para2 = polyval(para2, t);
cp_para4 = polyval(para4, t);

% schoene Ausgabe
plot(t_mess, cp_mess, 'r*', t, cp_lin, 'b:', t, cp_para2, 'b-.', ...
      t, cp_para4, 'b-');

xmin = 0;
xmax = 100;
ymin = 4.16;
ymax = 4.24;
axis([xmin xmax ymin ymax]);

title('Spezifische Waermekapazitaet des Wassers')
xlabel('T [°C]');
ylabel('c_p [kJ/(kg K)]');
legend('Messwerte', 'Fit 1.Ordnung', 'Fit 2. Ordnung', ...
      'Fit 4. Ordnung', 'Location', 'Best')

% senkrechte Linie am gesuchten Wert dazumalen
hold on; % kein neuer Plot
plot([t1 t1], [ymin ymax], 'g-');
hold off;
```

```
% Bild abspeichern  
F = getframe(gcf);  
imwrite(F.cdata, 'bild27.png');
```

exercise9.m



```
% Loesung von Aufgabe 9 mit schoenem Plot
set(0,'defaultfigurecolor','white'); % weißer Hintergrund

% Definition der Funktion
func = @(x) tan(x) - x .* log(0.1*x);

% plotten, um Anfangswerte zu finden
x = [0:0.01:10];
y = func(x);

% riesige Werte durch NaNs ersetzen -> keine Sprunglinie im Plot
max = 40;
indizes = find(abs(y) > max);
y(indizes) = NaN;

plot(x, y, [0 10], [0 0]);

% Bild abspeichern
F = getframe(gcf);
imwrite(F.cdata, 'bild42.png');

% Nullstellen ablesen liefert
x0 = [1.9 5.0 8.5];

% Genaue Werte mit fzero
x1 = fzero(func, x0(1))
x2 = fzero(func, x0(2))
x3 = fzero(func, x0(3))
```

exercise10.m



```
% Loesung von Aufgabe 10

% gegebene Werte
p1 = [2.7 2.5];
p2 = [36.1 5.5];
p3 = [28.4 31.9];
p4 = [4.0 24.6];

% Winkel gleich im Bogenmass!
alpha1 = -64.5*pi/180;
alpha2 = 62.0*pi/180;
alpha3 = 147.6*pi/180;
alpha4 = -122.7*pi/180;

% Berechnung der Steigungen
m1 = tan(0.5*pi + alpha1);
m2 = tan(0.5*pi + alpha2);
m3 = tan(0.5*pi + alpha3);
m4 = tan(0.5*pi + alpha4);

% Geraden sind gegeben durch
% -m*x + y = -m*x0 + y0
% also

A = [-m1 1; -m2 1; -m3 1; -m4 1];
r = [-m1*p1(1)+p1(2); -m2*p2(1)+p2(2); -m3*p3(1)+p3(2); -m4*p4(1)+p4(2)];

% Damit lautet die Loesung
x = A \ r
```

exercise10a.m



```
% Loesung von Aufgabe 10, Matrixversion
% funktioniert auch fuer beliebige Zahl von Messwerten

% gegebene Werte
alpha = [-64.5 62.0 147.6 -122.7]'*pi/180; % Spaltenvektor
punkte = [2.7 2.5; 36.1 5.5; 28.4 31.9; 4.0 24.6];

% Berechnung der Steigungen
m = tan(0.5*pi + alpha);

% Geraden sind gegeben durch
% -m*x + y = b = -m*x0 + y0
% also
n = size(alpha, 1);
A = [-m, ones(n,1)];
r = -m.*punkte(:,1) + punkte(:,2);

% Damit lautet die Loesung
x = A \ r
```

kreisprozess.dat



14.00	15.0	15.0
16.25	11.0	47.0
18.50	9.5	42.0
20.75	8.0	37.0
23.00	7.0	33.0
32.00	4.5	21.0
41.00	3.0	14.5
50.00	2.0	11.0
59.00	1.5	8.5
68.00	1.0	7.0
77.00	1.5	5.0
81.50	1.5	4.5
86.00	3.5	3.5

exercise11.m



```
% Loesung von Aufgabe 11

% Daten einlesen
load -ascii 'kreisprozess.dat';
V = kreisprozess(:,1);
pu = kreisprozess(:,2);
po = kreisprozess(:,3);

% Kreisprozess zeichnen
plot(V, po, 'k', V, pu, 'k');
title('Kreisprozess eines Verbrennungsmotors');
xlabel('V [cm^3]');
ylabel('p [bar]');

% Umrechnung auf SI-Einheiten
V = 1e-6*V;      % in m^3
pu = 1e5*pu;    % in Pa
po = 1e5*po;    % in Pa

% Bestimmung der Kreisprozess-Arbeit
intu = trapz(V, pu);
into = trapz(V, po);
W = into - intu

% Bestimmung der Leistung
n = 3000/60;    % in 1/s
P = W * n      % in W
```

exercise12.m



```
% Loesung von Aufgabe 12
x0 = [5000 0];      % Anfangswerte
tmax = 70;

% Loesung der drei DGLs
[tf, xf] = ode45(@freifall, [0 tmax], x0);
[tl, xl] = ode45(@laminar, [0 tmax], x0);
[tt, xt] = ode45(@turbulent, [0 tmax], x0);

plot(tf, xf(:,1), tl, xl(:,1), tt, xt(:,1));
legend('frei', 'laminar', 'turbulent', 'Location', 'Best');
title('Bahn bei freiem Fall mit Reibung');
xlabel('t [s]');
ylabel('x(t) [m]');

figure
plot(tf, xf(:,2), tl, xl(:,2), tt, xt(:,2));
legend('frei', 'laminar', 'turbulent', 'Location', 'Best');
title('Geschwindigkeit bei freiem Fall mit Reibung');
xlabel('t [s]');
ylabel('v(t) [m/s]');
```

laminar.m



```
function dy = laminar(t, y)
% rechte Seite der DGL beim freien Fall mit laminarer Reibung
g = 9.81;      % in m/s^2
m = 75;       % in kg
b1 = 5;       % in kg/s

dy1 = y(2);
dy2 = -g - b1/m*y(2);
dy = [dy1 dy2]';
```

turbulent.m



```
function dy = turbulent(t, y)
% rechte Seite der DGL beim freien Fall mit turbulenter Reibung
g = 9.81;      % in m/s^2
m = 75;       % in kg
b2 = 0.1;     % in kg/m

dy1 = y(2);
dy2 = -g - b2/m*y(2).^2.*sign(y(2));
dy = [dy1 dy2]';
```

freifall.m



```
function dy = freifall(t, y)
% rechte Seite der DGL beim freien Fall
g = 9.81;    % in m/s^2

dy1 = y(2);
dy2 = -g;
dy = [dy1 dy2]';
```

Lösung von Aufgabe 6



a. SQL-Kommando:

```
SELECT wort, COUNT(*) AS anzahl
FROM Schlagwort NATURAL JOIN BuchSchlagwort
GROUP BY wort
ORDER BY anzahl DESC;
```

Ergebnis:

wort	anzahl
Thermodynamik	7
Lehrbuch	6
Physik	5
Quantenmechanik	3
Datenbank	2
Handbuch	2
SQL	2
Ingenieurwesen	1
Statistische Physik	1

b. SQL-Kommando:

```
SELECT name AS autor, titel
FROM Buch NATURAL JOIN BuchAutor NATURAL JOIN Autor
ORDER BY autor, titel;
```

Ergebnis:

autor	titel
Bohrmann	Physik - Der Grundkurs
Cerbe	Einführung in die Thermodynamik
Cerbe	Grundlagen der Gastechnik
Greiner	Thermodynamics and Statistical Mechanics
Halliday	Fundamentals of Physics Extended
Hering	Physik für Ingenieure
Hoffmann	Einführung in die Thermodynamik
Jany	Thermodynamik für Ingenieure
King	MySQL & mSQL
Langeheinecke	Thermodynamik für Ingenieure
Martin	Physik für Ingenieure
Matthiesen	Relationale Datenbanken und SQL
Neise	Thermodynamics and Statistical Mechanics

Pitka	Physik - Der Grundkurs
Reese	MySQL & mSQL
Resnick	Fundamentals of Physics Extended
Sapper	Thermodynamik für Ingenieure
Stöcker	Physik - Der Grundkurs
Stöcker	Taschenbuch der Physik
Stöcker	Thermodynamics and Statistical Mechanics
Stohrer	Physik für Ingenieure
Terlecki	Physik - Der Grundkurs
Unterstein	Relationale Datenbanken und SQL
Walker	Fundamentals of Physics Extended
Yarger	MySQL & mSQL

c. SQL-Kommando:

```
SELECT Autor.name AS autor, Verlag.name AS verlag
FROM Autor NATURAL JOIN BuchAutor NATURAL JOIN Buch
INNER JOIN Verlag ON Buch.verlag_id = Verlag.verlag_id
GROUP BY autor, verlag
ORDER BY autor;
```

Ergebnis:

autor	verlag
Bohrmann	Harri Deutsch
Cerbe	Hanser
Greiner	Springer
Halliday	Wiley
Hering	Springer
Hoffmann	Hanser
Jany	Vieweg
King	O'Reilly
Langeheinecke	Vieweg
Martin	Springer
Matthiesen	Addison-Wesley
Neise	Springer
Pitka	Harri Deutsch
Reese	O'Reilly
Resnick	Wiley
Sapper	Vieweg
Stohrer	Springer
Stöcker	Harri Deutsch
Stöcker	Springer
Terlecki	Harri Deutsch
Unterstein	Addison-Wesley
Walker	Wiley

```
| Yarger          | O'Reilly        |
+-----+-----+
```

d. SQL-Kommando (mit Buchtitel):

```
SELECT Autor.name AS autor, titel
FROM Autor NATURAL JOIN BuchAutor NATURAL JOIN Buch
      NATURAL JOIN BuchSchlagwort NATURAL JOIN Schlagwort
WHERE wort = "Thermodynamik"
ORDER BY autor;
```

Ergebnis:

```
+-----+-----+
| autor          | titel           |
+-----+-----+
| Bohrmann       | Physik - Der Grundkurs |
| Cerbe          | Einführung in die Thermodynamik |
| Greiner        | Thermodynamics and Statistical Mechanics |
| Halliday       | Fundamentals of Physics Extended |
| Hering         | Physik für Ingenieure |
| Hoffmann       | Einführung in die Thermodynamik |
| Jany           | Thermodynamik für Ingenieure |
| Langeheinecke | Thermodynamik für Ingenieure |
| Martin         | Physik für Ingenieure |
| Neise          | Thermodynamics and Statistical Mechanics |
| Pitka          | Physik - Der Grundkurs |
| Resnick        | Fundamentals of Physics Extended |
| Sapper         | Thermodynamik für Ingenieure |
| Stohrer        | Physik für Ingenieure |
| Stöcker        | Physik - Der Grundkurs |
| Stöcker        | Taschenbuch der Physik |
| Stöcker        | Thermodynamics and Statistical Mechanics |
| Terlecki       | Physik - Der Grundkurs |
| Walker         | Fundamentals of Physics Extended |
+-----+-----+
```

SQL-Kommando (ohne Buchtitel):

```
SELECT Autor.name AS autor
FROM Autor NATURAL JOIN BuchAutor
      NATURAL JOIN BuchSchlagwort NATURAL JOIN Schlagwort
WHERE wort = "Thermodynamik"
GROUP BY autor
ORDER BY autor;
```

Ergebnis:

```
+-----+
| autor          |
```

Bohrmann
Cerbe
Greiner
Halliday
Hering
Hoffmann
Jany
Langeheinecke
Martin
Neise
Pitka
Resnick
Sapper
Stohrer
Stöcker
Terlecki
Walker

e. SQL-Kommando:

```
SELECT A2.vorname, A2.name
      FROM Autor AS A1 NATURAL JOIN BuchAutor AS BA1
      INNER JOIN BuchAutor AS BA2 ON BA1.isbn = BA2.isbn
      INNER JOIN Autor AS A2 ON BA2.autor_id = A2.autor_id
      WHERE A1.name = "Stöcker"
      AND BA1.autor_id != BA2.autor_id;
```

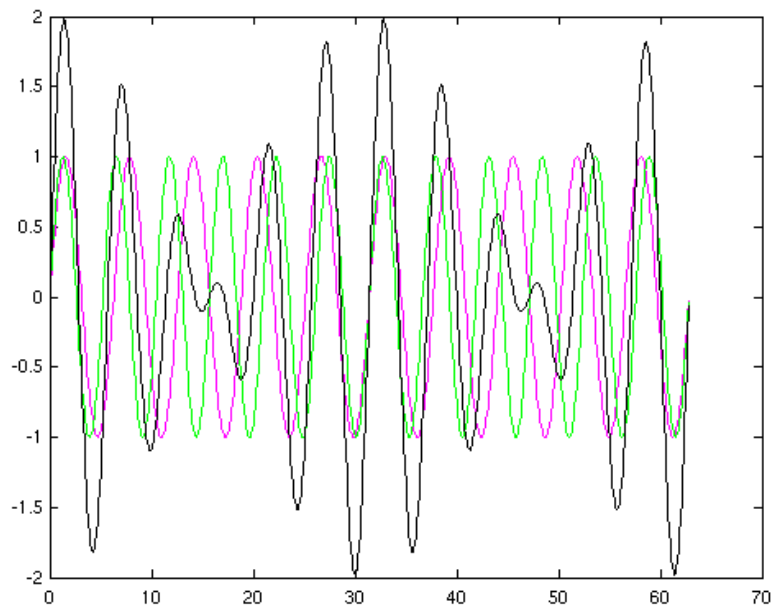
Ergebnis:

vorname	name
Walter	Greiner
Ludwig	Neise
Rudolf	Pitka
Steffen	Bohrmann
Günther	Terlecki

Lösung von Aufgabe 7



- komplette Lösung in Matlab-Datei [exercise7.m](#)
- Plot:



Lösung von Aufgabe 8



- Komplette Lösung in Matlab-Datei [exercise8.m](#)
- Zunächst wird die Tabelle mit `load` eingelesen und in `t`- und `cp`-Werte aufgetrennt.
- Mit `poly = polyfit(x, y, n)` wird ein Fitpolynom n-ten Grades zu den Punkten $(x(i), y(i))$ berechnet. Der Rückgabewert `poly` enthält die Koeffizienten des Polynoms, beginnend mit dem höchsten. Z. B. für die 2. Ordnung lautet das entsprechende Polynom:

$$P(x) = \text{poly}(1) x^2 + \text{poly}(2) x + \text{poly}(3)$$

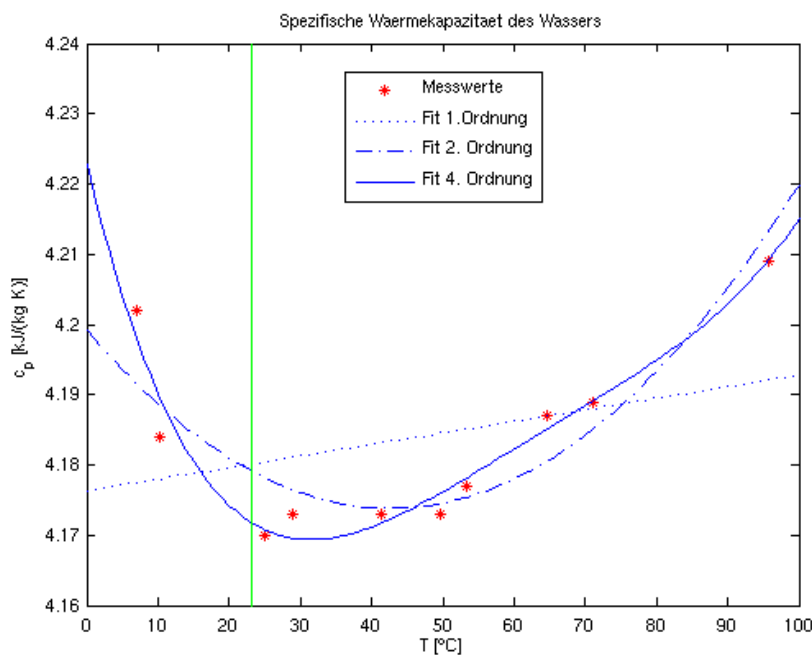
- Mit Hilfe von `poly` kann man nun den Wert des Polynoms zum gesuchten `t1`-Wert berechnen (oder für viele Werte zum Plotten). Das geht entweder direkt mit

$$\text{cp1} = \text{poly}(1) * \text{t1}^2 + \text{poly}(2) * \text{t1} + \text{poly}(3)$$

oder eleganter und einfacher mit der Funktion `polyval`

$$\text{cp1} = \text{polyval}(\text{poly}, \text{t1})$$

- Plot:



Lösung von Aufgabe 9



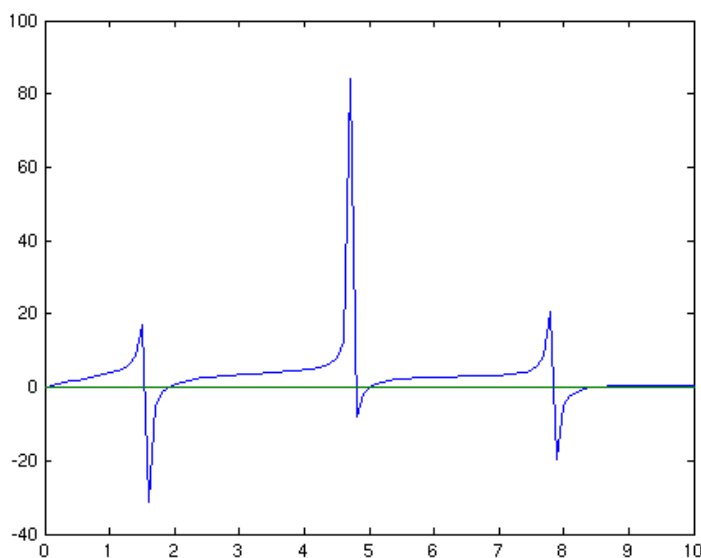
- Die Gleichung wird in der Form $\text{func}(x) = 0$ geschrieben und die Funktion wie üblich mit $@(x)$ definiert:

```
func = @(x) tan(x) - x .* log(0.1*x);
```

- Um Anfangswerte für `fzero` zu finden, wird sie nun geplottet:

```
x = [0:0.1:10];  
y = func(x);  
plot(x, y, [0 10], [0 0]);
```

Dabei stellt man fest, dass an den Stellen $\pi/2$, $3\pi/2$ und $5\pi/2$ große Sprünge auftreten, die von den Polen des Tangens an diesen Stellen herrühren:



Die nahezu senkrechten Linien gehören nicht zur Funktion, sondern sie werden durch die Vorzeichenwechsel an den Polstellen verursacht: Matlabs Plot-Funktion verbindet einfach die beiden an einen Pol angrenzenden Werte. Insbesondere liegen hier keine Nullstellen!

- Durch die ungünstige Skala sind die wahren Nullstellen etwas schwierig zu erkennen, aber ggf. mit Hilfe der Lupe liest man folgende Werte ab:

1.9, 5.0, 8.5

Mit `fzero(func, 1.9)` etc. bestimmt man mit diesen Startwerten die genauen Nullstellen zu

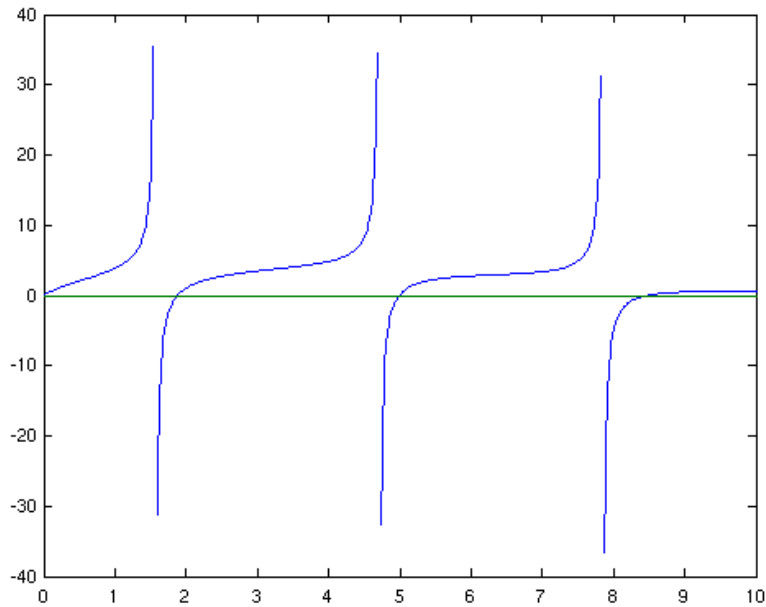
1.8790, 4.9931, 8.4728

- Der Plot legt nahe, dass bei 0 ebenfalls eine Nullstelle vorliegt. Dort ist die Funktion zwar nicht definiert, lässt sich aber tatsächlich stetig zu 0 fortsetzen, wie man z.B. über den Satz von l'Hospital beweisen kann. Natürlich lässt sich hier mit `fzero` nichts machen, weil die Funktion auf der linken Seite der Nullstelle gar nicht definiert ist.
- Übrigens lassen sich die hässlichen Sprünge im Plot mit einem kleinen Trick

vermeiden: Man ersetzt dazu in der Variablen y die betragsmäßig sehr großen Werte an den Polen durch den Wert NaN.

```
max = 40;  
indizes = find(abs(y) > max);  
y(indizes) = NaN;  
plot(x, y, [0 10], [0 0]);
```

Die Plot-Funktion lässt diese Werte dann aus, insbesondere verbindet sie sie nicht. Das Ergebnis ist deutlich hübscher:



Die Details zeigt das Skript [exercise9.m](#) .

Lösung von Aufgabe 10



- Die Steigung m der Geraden von einer Station zum Störsender ist gegeben durch
$$m = \tan(90^\circ + \alpha)$$

Die Gerade geht außerdem durch den Ort (x_0, y_0) der Peilstation, hat also die Form

$$y = m(x - x_0) + y_0$$

oder

$$-m x + y = -m x_0 + y_0$$

- Für die Position des Störsenders ergeben sich also 4 Gleichungen mit zwei Unbekannten, die in einfacher Weise in Matlab gelöst werden können (im Sinne der kleinsten quadratischen Abweichung).
- komplette Lösung in Matlab-Datei [exercise10.m](#)
- kompakter gelöst mit Hilfe von Matrixfunktionen in [exercise10a.m](#)

Lösung von Aufgabe 11



- Die Kreisprozess-Arbeit ist gegeben durch die umschlossene Fläche

$$W = \oint p dV$$

Die Leistung ist dann

$$P = W n$$

- Alles weitere in [exercise11.m](#)

Lösung von Aufgabe 12



- Zunächst muss die Differentialgleichung für Matlab aufbereitet werden. Dazu wird sie zunächst nach \ddot{x} aufgelöst:

$$\ddot{x} = -g - (b_1/m) \dot{x}$$

- Dies wird in ein System von zwei Gleichungen 1. Ordnung umgeformt:

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= -g - \frac{b_1}{m} y_2 \end{aligned}$$

und als Matlab-Funktion in [laminar.m](#) abgespeichert.

Ganz analog definiert man [turbulent.m](#) und [freifall.m](#). Das Lösen und Plotten geschieht dann in wenigen Zeilen, etwa wie in [exercise12.m](#)