Grundgerüst einer HTML-Datei

SGML-konformer Dokumenttyp

```
<!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-8859-1">
    <title>
      Text des Titels
    </title>
  </head>
  <body>
    <p>Eigentlicher Inhalt</p>
  </body>
</html>
```

Zeichensatz

Titel für Browserfenster

THIS PAGE IS VALID HTML 4.01 STRICT!

Schreibregeln

- Leerzeichen, Tabulator und Zeilenvorschub sind Trenner
  - ⇨ Anzahl spielt keine Rolle, außer in Attributwerten
  - ⇨ Ausnahme: in `<pre>` Abschnitten (=preformatted)
- Einrückung dient nur der Lesbarkeit
  - ⇨ wird vom Browser ignoriert
  - ⇨ wird von manchen WYSIWYG Tools ruiniert
- Kommentare

```
<!-- das ist ein Kommentar -->
```

- Sonderzeichen und Umlaute können kodiert werden

<	`<`	>	`>`	&	`&`	"	`&quot;`
ä	`&auml;`	Ä	`&Auml;`	ö	`&ouml;`	Ö	`&Ouml;`
ü	`&uuml;`	Ü	`&Uuml;`	ß	`&szlig;`	€	`&euro;`

Tags und Attribute

- Tags (Marken) markieren Abschnitte im Text
  - ⇨ Name in spitzen Klammern
  - ⇨ gleicher Name für öffnendes und schliessendes Tag
  - ⇨ schliessendes Tag kenntlich an zusätzlichem /
  - ⇨ XHTML ist case sensitive (Kleinschreibung), HTML nicht

```
<h2>Willkommen in unserem Hotel</h2>
```

- öffnende Tags können zusätzliche Attribute enthalten
  - ⇨ Attribute haben einen Namen und einen Wert
  - ⇨ Attributwerte werden in Anführungszeichen geschrieben (XHTML)

```
<h2 id="hallo">Willkommen in unserem Hotel</h2>
```

- mit Tags markierte Abschnitte können verschachtelt sein

```
<h2><em>Willkommen</em> in unserem Hotel</h2>
```

Standalone-Tags

- es gibt einige wenige „Standalone-Tags“
  - ⇨ leere Elemente = Abschnitte ohne Inhalt

```
Willkommen <br> auf unserer Homepage
```

- Standalone-Tags passen nicht in das DOM
  - ⇨ dieses verlangt eine strenge Baumstruktur
- in XML und XHTML ganz verboten
  - ⇨ ersatzweise

    `<br />` oder `<br></br>`

Document Object Model (kommt später)

Universalattribute

- **können zu jedem Tag hinzugefügt werden**
  - ⇨ `id` dateiweit eindeutiger Bezeichner für Scripte
  - ⇨ `class` Name der zugehörigen Style Sheet Klasse
  - ⇨ `title` Erläuterung zum Element, erscheint als Tooltip
  - ⇨ `style` eingebettete Style Sheet Attribute
  - ⇨ `lang, dir` Landessprache und Textlaufrichtung

```
<h2
      id="JB007" class="mycssstyleclass" title="mytooltip"
      style="color:red" lang="de" dir="ltr">
Hallo</h2>
```

Logische Formatierung

- markiert Bedeutung von Textabschnitten
- macht keine Aussage über visuelles Erscheinungsbild
  - ⇨ das wird erst per CSS definiert
  - ⇨ für Sprachausgabe muss stattdessen das akustische Erscheinungsbild definiert werden...
- ein paar Beispiele:

`<em>`	emphatisch (betont)
`<strong>`	stark betont
`<samp>`	Beispiel
`<dfn>`	Definition
`<cite>`	inline-Zitat (z.B. für Eigennamen; oft kursic)
`<q cite="URL">`	Zitat mit Quellenangabe (oft in Anführungszeichen)
`<blockquote>`	Zitat als Block gesetzt

von IE nicht umgesetzt

Block

Strukturierung von Text

alle außer und <br> erzeugen einen Block

- Überschriften
  - `<h1>` Überschrift der höchsten Gliederungsebene
  - `<h6>` Überschrift der niedrigsten Gliederungsebene

  header1,…, header6
- Abschnitte
  - `<p>` Textabsatz (immer paarweise mit `</p>`)
  - `<div>` allgemeiner Block

  div = division = Bereich
  - `<span>` Inline-Element

  kein Block

  "Aufhänger" für CSS
- Aufzählungen (nummeriert oder auch nicht)
  - `<ol>, <ul>, <li>`

  ordered list, unordered list, list item
- Zeilenumbruch erzwingen und verhindern
  - `<br>` expliziter Zeilenumbruch (standalone tag)

  kein Block
  - ` ` geschütztes Leerzeichen - verhindert hässlichen Zeilenumbruch

  z.B. 3. Kapitel

Verpönte Attribute und Tags (deprecated)

abgesetzt, unerwünscht

- Farbangaben
  - ⇨ `background, bgcolor, text`
  - ⇨ `link, alink, vlink`
- Schrift
  - ⇨ `<font>, <basefont>`
  - ⇨ `compact, strike, s, u`
- Ausrichtung
  - ⇨ `align, nowrap, <center>`
- Größe
  - ⇨ `size, width, height`
- Rand
  - ⇨ `hspace, vspace, border`

active, visited

s=strike, u=underline

Angaben dieser Art dienen der physischen Formatierung, sind in der **Strict**-Variante verboten und sollen durch CSS-Angaben ersetzt werden

Zulässig in **Transitional**-Variante, damit Altlasten weiterleben

Grenzfall: physische Formatierung

- definiert das visuelle Erscheinungsbild
- nicht verpönt, aber besser zu vermeiden
- ein paar Beispiele:

`<b>`	**fette Schrift (bold)**
`<i>`	***kursive* Schrift (italic)**
`<tt>`	**`dicktengleiche` Schrift (monospaced, Teletype)**
`<big>`	**Schrift größer als normal**
`<small>`	**Schrift kleiner als normal**
`<sup>`	**Schrift hochgestellt**
`<sub>`	**Schrift tiefgestellt**

```
<pre>    präformatierter Text (z.B. für Quellcode)
```

Block

Einbindung von Pixelbildern (Grafiken)

- für Bilddateien der Formate GIF, PNG und JPG
- notwendige Attribute:

`src`	Quelle
`width, height`	Breite und Höhe
`alt`	Ersatztext

- Beispiel

```
<img src="bilder/datei.gif" width="250" height="450"
  alt="Beschreibung des Bildes">
```

standalone tag

- besser mittels CSS festlegen:

`border`	Rahmen
`hspace, vspace`	Abstand zur Umgebung
`align`	Ausrichtung und Textumfluss

Bilddateien

- GIF für Grafiken (z.B. Screenshots, ClipArts)
  - ⇨ 256 Farben Palette, LZW komprimiert
  - ⇨ Freistellen (transparenter Hintergrund) möglich
  - ⇨ Nachfolger: PNG

Lempel-Ziv-Welch

- JPEG für Photos
  - ⇨ Echtfarben, DCT komprimiert, verlustbehaftet
  - ⇨ kein Freistellen

Discrete Cosinus Transformation

bisher Exoten: Fraktale und Wavelet-Kompression

- Häufig unvollständiges oder springendes Gesamtbild während Seitenaufbau
  - ⇨ Gängiger Trick: Platzierung mit unsichtbarem GIF erzwingen

- Möglichkeiten zur Speicherplatzersparnis
  - ⇨ Hintergrund mit kleiner Grafik "kacheln"
  - ⇨ Größe und Farbtiefe reduzieren
  - ⇨ Beschränkung auf wenige Grafiken

Bandbreite!

Audio und Video

<embed> ist nicht mehr standardkonform – aber die Alternative <object> funktioniert nur in neuen Browsern.

- Prinzip: Einbettung von Media-Clips
  - ⇨ Media-Clip als eigenständige Datei; HTML-Datei enthält Verweis

```
<embed src="datei.ext" width="150" height="60">
```

  - ⇨ Platzierung wie Grafik im Fließtext
- Abspielen
  - ⇨ interaktiv per Mausklick
  - ⇨ automatisch beim Öffnen der Seite
- Implementation uneinheitlich und plattformabhängig
  - ⇨ manche Formate vom Browser direkt unterstützt
  - ⇨ andere über installierbare Plug-Ins
- vorzugsweise "streaming mode" wegen Übertragungszeit

Meta-Angaben

- Anweisungen für WWW-Server, WWW-Browser und automatische Suchprogramme ("Robots")
- eine kleine Auswahl von Meta-Angaben:

```
<meta name="description" content="Autovermietung">
<meta name="author" content="B. Kreling">
<meta name="keywords" content="Hotel,Urlaub,Meer">
<meta name="robots" content="noindex">
<meta name="date" content="2001-02-06">
<meta name="language" content="de">

<meta http-equiv="Content-Script-Type"
                        content="text/javascript">
<meta http-equiv="Content-Style-Type"
                        content="text/css">
```

Anwendungsfälle für Hyperlinks

- Beispiele für Einsatzmöglichkeiten
  - ⇨ Querverweis (vgl. Lexikon, Literaturstelle)
  - ⇨ Blättern (nächste Seite / vorige Seite)
  - ⇨ Inhaltsverzeichnis (Unterkapitel / Oberthema)
  - ⇨ Stichwortverzeichnis
  - ⇨ freie Navigation, neue Dokumentstrukturen ⇨ Hypermedia
  - ⇨ Download einer Datei
  - ⇨ sonstiger Dienst
- Einsatzgebiet klären und gestalterisch unterscheiden
- "Hyperlink" ist lediglich eine technische Realisierung !

Gestaltungstipps für Verweise

- ein Verweis ist ein Blickfang
  - ⇨ nur bedeutungstragende Begriffe mit Hyperlink hinterlegen
- Verweistext soll das Verweisziel deutlich machen
  - ⇨ vorzugsweise immer derselbe Text für dasselbe Ziel
- Verweis sollte unmittelbar erkennbar sein
  - ⇨ nicht erst nach "Abtasten" mit der Maus
- nicht zu viele Verweise auf dieselbe Stelle
  - ⇨ Surfer folgen Verweisen mitunter auch um die Website vollständig zu besuchen
- alle Seiten vollständig verlinken
  - ⇨ "Zurück"-Button des Browsers sollte innerhalb einer Website überflüssig sein

Ziele von Verweisen

- eine Datei, die der Browser als Seite darstellen kann
  - ⇨ meistens HTML, aber auch andere
  - ⇨ im Internet oder lokal
- bestimmte Position ("Anker") innerhalb einer darstellbaren Datei
- eine Datei, die der Browser selbst nicht darstellen kann
  - ⇨ diese wird zum Download angeboten oder
    mit einer Hilfsanwendung geöffnet
- andere Dienste neben WWW
  - ⇨ mailto, gopher, ftp, telnet, news

```
<a href="mailto:r.hahn@fbi.h-da.de">R. Hahn</a>
<a href="ftp://www.xyz.de/setup.zip">Download</a>
<a href="file:///c:/lokal.htm">lokale Datei</a>
```

Verweise

Der Verweistext sollte eine klare Information über das Ziel des Verweises geben !

- Allgemeine Form

```
<a href="Dienst://Server:Port/Verz/Datei#Anker"> Text</a>
```

  Teile davon können weggelassen werden

- Datei im selben / unter- / übergeordneten Verzeichnis

```
<a href="start.htm">Text</a>
<a href="sub/Datei.html">Text</a>
<a href="../inhalt.htm">Text</a>
```

relativ

- Datei auf anderem Server

auch: localhost

```
<a href="http://www.xyz.de/datei.htm">Text</a>
```

absolut

- Groß-/Kleinschreibung beachten
  - ⇨ Server laufen meist unter Unix und Unix ist case sensitive bezüglich Datei- und Verzeichnisnamen

beliebter Fehler unter Windows

Absolute und relative Verweise

- relative Verweise innerhalb der eigenen Website (projekt-intern) sind vorteilhaft für
  - ⇨ Migration auf anderen Server oder in anderes Verzeichnis
  - ⇨ Entwicklung auf lokaler Festplatte mit späterem Upload
  - ⇨ Download als ZIP und lokale Installation (vgl. SELFHTML)

- absolute Verweise sind vorteilhaft für
  - ⇨ versenden von Seiten per eMail (z.B. Werbung, Stundenplan; sofern der Leser online ist wird er direkt auf den Webserver weitergeleitet)
  - ⇨ Verweise auf fremde Websites (projekt-extern)

Verweise innerhalb einer Datei ("Anker")

- wird häufig eingesetzt für "Inhaltsverzeichnis" am Anfang einer Datei
  - ⇨ z.B. bei FAQ
- Verweisziel definieren

```
<a name="Erl"><h2>Erläuterung</h2></a>
```

- Verweis definieren

```
siehe die <a href="#Erl">Erläuterung</a> unten
```

- der Verweis kann auch zu einer bestimmten Position in einer anderen Datei zeigen

```
<a href="datei.htm#Erl">Erläuterung</a>
<a href="http://www.xyz.de/datei.htm#Anker">...</a>
```

- der Browser scrollt die Seite so, dass der Anker an der Oberkante des Fensters erscheint

Zusammenfassung

- Grundgerüst: DOCTYPE, <html>, <head>, <body>, <title>, charset...
- Schreibregeln: Zeilenumbruch, Kommentare und Sonderzeichen
- Tags und Attribute
- Logische Formatierung und verpönte Formatierung
- Einbinden von Grafiken, Audio, Video...
- Meta-Angaben
- Verwendung von Hyperlinks <a href="... >
- Verweise innerhalb einer Seite (Anker)

Jetzt können Sie eine einfache HTML-Seite schreiben!

Problematik des Layouts

- Fließtext statt fester Platzierung (ggfs. mit autom. Zoom)
  - ⇨ Informationsdarstellung auf verschiedensten Monitoren
  - ⇨ Gestaltungsmöglichkeiten sehr beschränkt

    (eigentlich möchte man einen Screen als Ganzes gestalten ...)

- ursprünglich keine Überdeckung von Objekten
  - ⇨ ist in anderen Medien und Tools eine Grundfunktion

- die meisten Seitengestalter denken in statischen Layouts
  - ⇨ dynamische Layouts sind wesentlich schwieriger zu entwerfen
  - ⇨ zwei "Layoutmanager" verfügbar: `<table>` und `<frameset>`
  - ⇨ nach tricky programming nun tricky designing

Tabelle als Layoutmanager – Beispiel

Tabelle als Layoutmanager

Rahmen verbergen

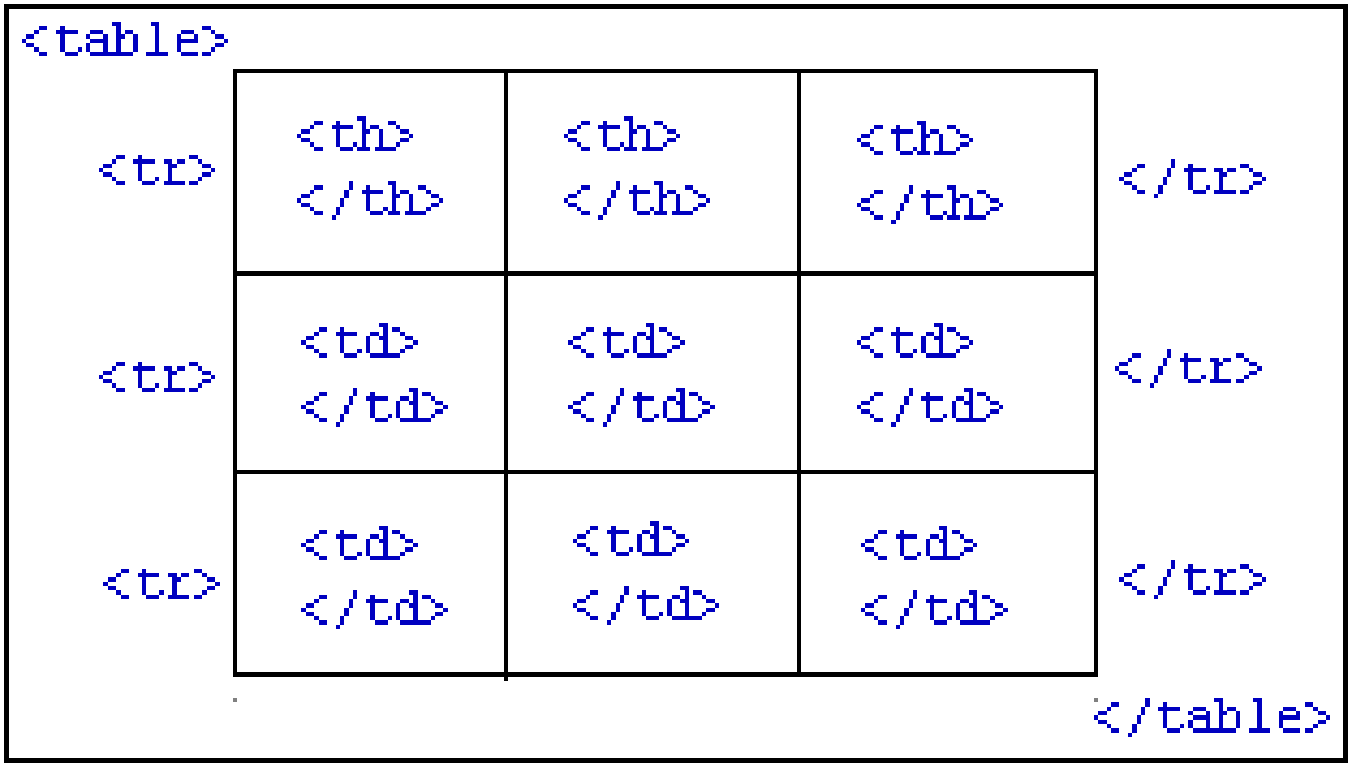
h_da HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES	FB Informatik	Prof. Dr.-Ing. B. Kreling
Multimedia II		Entwicklung von Anwendungssystemen
Inhalte		Beschreibung der Lehrinhalte Digitales Skript
Hinweis		Multimedia I ist **nicht** Voraussetzung für Multimedia II; die Veranstaltungen ergänzen einander, können aber in beliebiger Reihenfolge belegt werden.
Status		Wahlpflichtfach für alle Schwerpunkte
Art		Vorlesung + Praktikum, 2+2 SWS
Belegnummer		I WA.M2 I
Ort und Zeit	Vorlesung Praktikum Beginn u. Anmeldung	mittwochs 4. Block, I104 montags 1.+2. Block oder mittwochs 1.+2. Block, B035 Mittwoch 8.10.97, 14:15 Uhr, I104

Praktikumstermine	Gruppe 1	Gruppe 2	Gruppe 3
	20.10.97	22.10.97	27.10.97
	03.11.97	05.11.97	10.11.97
	17.11.97	19.11.97	24.11.97
	01.12.97	03.12.97	08.12.97
	15.12.97	17.12.97	05.01.98
	12.01.98	14.01.98	19.01.98

Klausur	Mittwoch 28.1.1998, 14:15 Uhr, I104 Keine Hilfsmittel zugelassen. Erfolgreiche Teilnahme am Praktikum ist Voraussetzung.

Struktur einer Tabelle (1)

Grundidee:



aus S. Münz: SELFHTML

Struktur einer Tabelle (2)

ursprünglich zur Darstellung tabellarischer Daten

```
<table summary="Tabelle zur...">
     <tr>
        <th>Kopfzelle: 1. Zeile, 1. Spalte</th>
        <th>Kopfzelle: 1. Zeile, 2. Spalte</th>
     </tr>
     <tr>
        <td>Datenzelle: 2. Zeile, 1. Spalte</td>
        <td>Datenzelle: 2. Zeile, 2. Spalte</td>
     </tr>
     <tr>
        <td>Datenzelle: 3. Zeile, 1. Spalte</td>
        <td>Datenzelle: 3. Zeile, 2. Spalte</td>
     </tr>
  </table>
```

zeilenweise (tr = table row)

beliebig viele Zeilen und Spalten

Struktur einer Tabelle (3)

- Gesamtbreite definieren

```
<table width="80%">...</table>
```

  - ⇨ Gesamthöhe können die Browser zwar, ist auch nützlich, aber nicht standard-konform

- Spaltenbreite vordefinieren
  - ⇨ ermöglicht schnelleren Tabellenaufbau im Browser

```
<colgroup>
   <col width="80"> <col width="120">
</colgroup>
```

- Zellen verbinden
  - ⇨ Spalten verbinden `<td colspan="3">...</td>`
  - ⇨ Zeilen verbinden `<td rowspan="2">...</td>`

in Pixel oder Prozent

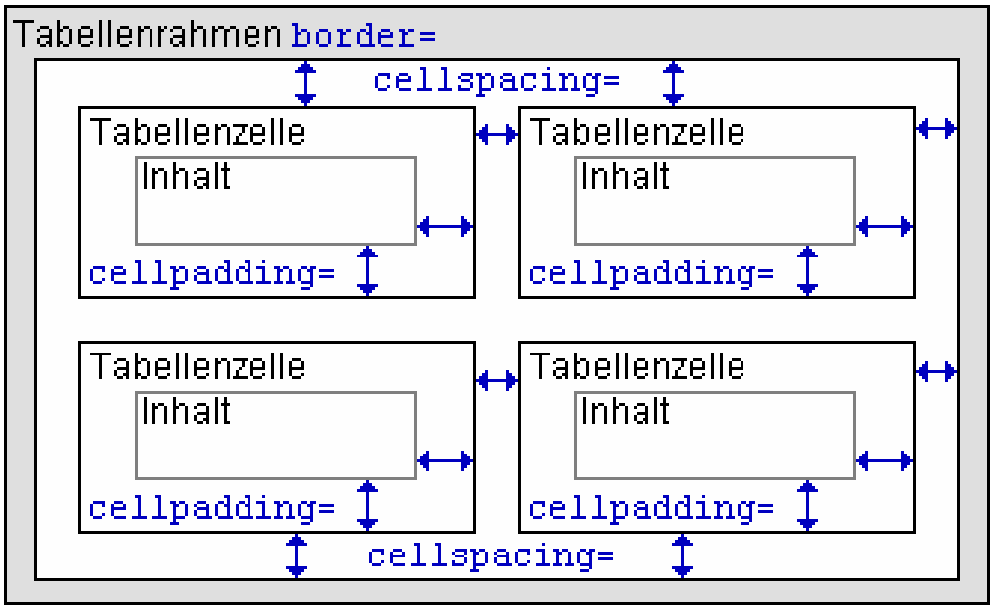
auch kombinierbar

Erscheinungsbild einer Tabelle

- Rahmen gestalten

```
<table border="8"
   cellspacing="10"
   cellpadding="20">
```

⇨ verfeinerte Rahmenformatierung besser mit CSS



aus S. Münz: SELFHTML

- Ausrichtung der Zelleninhalte, Verhinderung von Zeilenumbruch, Farben etc. besser mit CSS

Tabelle als Layoutmanager

absolut verpönt im
Hinblick auf Barrierefreiheit!

- Tabelle ist (immer noch) häufig Basis des Seitenlayouts
  - ⇨ statisches Layoutraster durch Bemaßung in Pixel
  - ⇨ dynamischer Layoutmanager durch Bemaßung in Prozent
    (vergleichbar mit GridBagLayout in Java)
- normalerweise „blinde“ Tabelle, d.h. ohne Rand
- Freiformen, Rundbögen etc. als eingebettete Grafik

- Entwurfsmethodik sinngemäß übertragen
  - ⇨ siehe "Dynamisches Layout" in
    "Entwurf und Realisierung grafischer Oberflächen"

- künftig ersetzen durch CSS-Layouthilfsmittel

Blinde Tabelle als Layoutmanager – Beispiel

Tabelle als Layoutmanager

Rahmen zeigen

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

FB Informatik — Prof. Dr.-Ing. B. Kreling

Multimedia II — Entwicklung von Anwendungssystemen

Inhalte	Beschreibung der Lehrinhalte Digitales Skript		
Hinweis	Multimedia I ist **nicht** Voraussetzung für Multimedia II; die Veranstaltungen ergänzen einander, können aber in beliebiger Reihenfolge belegt werden.		
Status	Wahlpflichtfach für alle Schwerpunkte		
Art	Vorlesung + Praktikum, 2+2 SWS		
Belegnummer	I WA.M2 I		
Ort und Zeit	Vorlesung	mittwochs 4. Block, I104	
	Praktikum	montags 1.+2. Block oder mittwochs 1.+2. Block, B035	
	Beginn u. Anmeldung	Mittwoch 8.10.97, 14:15 Uhr, I104	
Praktikumstermine	Gruppe 1	Gruppe 2	Gruppe 3
	20.10.97	22.10.97	27.10.97
	03.11.97	05.11.97	10.11.97
	17.11.97	19.11.97	24.11.97
	01.12.97	03.12.97	08.12.97
	15.12.97	17.12.97	05.01.98
	12.01.98	14.01.98	19.01.98
Klausur	Mittwoch 28.1.1998, 14:15 Uhr, I104 Keine Hilfsmittel zugelassen. Erfolgreiche Teilnahme am Praktikum ist Voraussetzung.		

FB I, Kreling, 06.01.98