

A world map is visible in the background, rendered in a light yellow color against a light blue background. The map shows the continents of North America, South America, Europe, Africa, Asia, and Australia.

HTML - Die Sprache des WEB

Alfred Beringer

HTBLVA für Textilindustrie und Datenverarbeitung

Überblick

Kurze Einführung
Begriffsdefinitionen
Übersicht HTML
HTML-Befehle
Beispiele

WWW - World Wide Web

- ◆ Eine Komponente des Internet
- ◆ Aufrufen und ansehen von Informationen auf beliebigen Servern
- ◆ Informationen sind in
"SEITEN"
abgelegt (abgespeichert)

Seite ("Page")

- ◆ Datei bzw. Dokument auf einem beliebigen Server
- ◆ Inhalt einer Seite (= Information):
 - Texte
 - Bilder und Zeichnungen
 - Animationen (bewegte Bilder)
 - Verweise auf andere Seiten ("**Links**")
- ◆ "**HOMEPAGE**" = Startseite

URL

Uniform Resource Locator

- ◆ Adresse einer Seite
- ◆ URL's können in Seiten enthalten sein und bilden so **Querverweise** ("**LINKS**") zu anderen Seiten
- ◆ **HYPertext** (vgl. Lexikon)

URL - Beispiele (1)

- ◆ <http://www.beringer.at/>
- ◆ <http://www.spengergasse.at/index.html>
- ◆ Ein abschließender Schrägstrich definiert ein Verzeichnis (Ordner bzw. "directory"), meist sucht der Browser dann eine Seite mit dem Namen **index.htm** bzw. **index.html**

URL - Beispiele (2)

- ◆ HTML-Einführung (Dr. Hubert Partl):
<http://www.boku.ac.at/html Einf/>
- ◆ W3-Consortium:
<http://www.w3.org/>
- ◆ Diese PowerPoint-Folien:
<http://www.beringer.at/downloads/htmlKurs.pdf>

Browser

- ◆ Programm zum Ansehen einer Seite
- ◆ Beispiele:
 - Mozilla Firefox
 - Microsoft Internet Explorer
 - Opera
 - Safari
 - Chrome

HTML

Hypertext Markup Language

- ◆ Sprache (Werkzeug) zur Beschreibung des Aufbaus und der optischen Darstellung einer Seite
- ◆ Befehle als **"Markups"** bzw. **"Tags"**
 - werden an beliebigen (sinnvollen) Stellen einer Seite eingefügt
 - kennzeichnen (markieren) Bereiche einer Seite

HTML - Allgemeines

- ◆ HTML-Befehle können groß- oder kleingeschrieben werden
- ◆ Viele Befehle funktionieren wie Schalter, die ein- und ausgeschaltet werden - durch sog. **Container-Tags** (**Anfang-** und **Ende-Tags**)
- ◆ Einige Befehle bewirken nur 1 bestimmte Aktion (sog. **leere Tags**)
- ◆ Bei vielen Befehlen können noch **Attribute** festgelegt werden

HTML-Befehle (Syntax)

- ◆ Eingeschlossen von 2 spitzen Klammern:
 - <tag>** ... leeres Tag oder Anfangs-Tag
 - </tag>** ... Ende-Tag
- ◆ Attribute:
 - <tag attribut1 attribut2 ... >**
- ◆ "Tags" werden in eine Seite eingefügt
- ◆ Schachtelungen möglich

XHTML-Befehle (Syntax)

- ◆ Strenge (exaktere) Syntaxregeln
- ◆ Befehle entsprechen jenen von HTML:

<tag /> ... leeres Tag

<tag> Anfangs-Tag

</tag> Ende-Tag

- ◆ Attribute:

<tag attr1="wert1" attr2="wert2" ... >

<tag attr1='wert1' attr2='wert2' ... >

XHTML - HTML

- ◆ Leeres Tag wird mit `
` abgeschlossen

`
`

`
`

- ◆ Ende-Tag darf **nicht** weggelassen werden

`<p> ... </p>`

`<p> ...`

`<p> ... </p>`

`<p> ...`

- ◆ Schachtelungen dürfen sich **nicht** überschneiden

`<big><i> ...`

`</i></big>`

`<i><big> ...`

`</i></big>`

XHTML - HTML

- ◆ Attribute immer in Form einer **Zuweisung**
- ◆ Attributwerte immer entweder unter **Hochkommata** (' ... ') oder unter **Anführungszeichen** (" ... ")

◆ Bsp.:

XHTML: `<input type='text' readonly="readonly" />`

HTML: `<input type="text" readonly>`

HTML-Befehle (Beispiel)

- ◆ Kennzeichnen einer Überschrift
- ◆ Inhalt der Seite:

... Ende Kapitel Sieben. </p>

<h2 align="center">Kapitel Acht</h2>

<p>Beginn Kapitel Acht ...

- ◆ Problem: Sonderzeichen (z.B. Umlaute)

Text - Sonderzeichen

^ <
v >
& &
" "
€ €
ß ß

Ä Ä
ä ä
Ö Ö
ö ö
Ü Ü
ü ü

Sonderzeichen - Beispiele

... Ende Kapitel Sieben.

`</p>`

`<h2 align="center" >`

`Ü`berschrift fuer Kapitel Acht

`</h2>`

`<p>`

Beginn Kapitel Acht ...

HTML-Befehle ("Tags")

◆ Leere Tags

- kennzeichnen 1 Stelle in einer Seite
z.B. Zeilenumbruch

◆ Container-Tags (Anfang- und Ende-Tag)

- kennzeichnen 1 Bereich in einer Seite
z.B. Fettdrucken eines Wortes

HTML-Seite (1)

- ◆ Jede HTML-Seite beginnt mit `<html>` und endet mit `</html>`
- ◆ Kennzeichnet ein Dokument als HTML-Seite und wird vom Browser als solche dargestellt

HTML-Seite (2)

- ◆ Jede Seite besteht aus 2 Teilen:
- ◆ **Kopf ("Header"):**
`<head> ... </head>`
- ◆ **Körper ("Body"):**
`<body> ... </body>`

HTML-Seite (Kopf)

- ◆ Beinhaltet allgemeine Angaben, die die Seite näher beschreiben:

- ◆ **Titel:**

<title> ... </title>

Der markierte Text wird vom Browser als
Überschrift verwendet

Suchdienste verwenden diesen Text

HTML-Seite (Körper)

- ◆ Beinhaltet die eigentliche Information, die mit Hilfe von HTML-Befehlen entsprechend aufbereitet werden kann
- ◆ Am Ende sollte immer eine Art **"Impressum"** (mit E-Mail-Adresse) bekanntgegeben werden, sowie das **Erstellungsdatum** bzw. das **letzte Änderungsdatum** der Seite.

Eine einfache HTML-Seite

<html>

<head>

<title>Erste HTML-Seite**</title>**

</head>

<body>

Hier steht nun die eigentliche Information, die aus Texten, Bildern, Animationen und Verweisen ("Links") besteht.**
**

Impressum:**
**

..., Erstelldatum, letztes Änderungsdatum

</body>

</html>

HTML-Befehle (Übersicht)

- ◆ Hypertext-Verweise (Links) **<a>**
- ◆ Überschriften **<h1>** bis **<h6>**
- ◆ Absätze **<p>**
- ◆ Abschnitte **<div>**, ****
- ◆ Text-Hervorhebungen
- ◆ Listen ****, ****, **<dl>**
- ◆ Tabellen **<table>**
- ◆ Rahmen **<frameset>**, **<frame>**
- ◆ Formulare **<form>**

Kommentar

- ◆ `<!-- beliebiger Text -->`
- ◆ Ein solcherart geklammerter Text wird vom Browser ignoriert, d.h. nicht angezeigt

Universalattribut `id`

- ◆ Kann in fast jedem HTML-Befehl verwendet werden (ausgenommen `head`, `html`, `title`, `style`)
- ◆ Ordnet dem HTML-Befehl einen dateiweit eindeutigen Bezeichnernamen zu (dabei wird Groß- und Kleinschreibung unterschieden!)
- ◆ Der Bezeichnername sollte innerhalb einer HTML-Datei nur **ein einziges** Mal vorkommen
- ◆ Dient u.a. als **Sprunganker** bei Verweisen ("Links")
- ◆ Weitere Verwendung bei CSS-Stylesheets und bei JavaScript

Hypertext-Verweise ("Links")

- ◆ Realisierung mit `<a> ... `
- ◆ Dieses Tag definiert einen **"ANKER"**
- ◆ Durch einfaches Anklicken des markierten Bereichs wird an eine andere Stelle innerhalb einer Seite oder in eine neue Seite verzweigt (in diesem Fall wird diese neue Seite automatisch auch geladen)

LINK (auf eine neue Seite)

◆ Quelldokument:

...

klickbarer Bereich

...

◆ Zieldokument:

hat die Adresse
XURL

Sprung an den
Anfang der neuen
Seite

LINK (innerhalb einer Seite)

- ◆ Quelldokument:

...

klickbarer Bereich

...

- ◆ Zieldokument =
Quelldokument mit
Sprungziel:

<tag id="xxx" ...>

oder

bedeutungslos

LINK (in eine neue Seite)

◆ Quelldokument:

...

klickbarer Bereich

...

◆ Zieldokument:

hat die Adresse
XURL und ein
Sprungziel:

<tag id="xxx" ...>

oder

bedeutungslos

LINK (zu `<a href>`-Adressen)

◆ absolute Adresse

- vollständige URL (z.B. `http://www...`)

◆ relative Adresse

- Pfadangabe ausgehend vom Server und vom Verzeichnis der Seite mit der href-Angabe

z.B. `file.html` (im selben Verzeichnis)

`subdir/file.html` (Unterverzeichnis)

`../parallel/file.html` (Verz. auf gleicher Stufe)

`/dir/file.html` (absolut am Server)

<a href>-Attribut TARGET (1)

- ◆ Definiert das Fenster (bzw. den Rahmen), **in dem** der Browser die (durch **href**) referenzierte HTML-Seite darstellt
- ◆ Das Fenster wird entweder durch einen **<frame>**-Tag-Namen oder durch einen **vordefinierten** Namen ausgewählt.

<a href>-Attribut TARGET (2)

- ◆ **Vordefinierte Namen:**

_blank neues Fenster

_self gerade aktives Fenster

_top Hauptfenster des Browsers

_parent bei geschachtelten Rahmen
übergeordnetes Fenster

- ◆ Seite wird in das entsprechende Fenster geladen

mailto-LINK

- ◆ Als Referenzadresse kann auch eine E-Mail-Adresse angegeben werden:

```
<a href="mailto:name@adresse">
```

```
e-mail-adresse
```

```
</a>
```

- ◆ Browser ermöglicht automatisch das Absenden einer Nachricht

LINK - Beispiele

- ◆ schlecht ("Hier"-Syndrom):

Wenn Sie zusätzliche Informationen über Computer erhalten wollen, dann klicken Sie `hier`.

- ◆ besser:

Zusätzliche Informationen gibt es über `Computer`.

Text - Formatierung

- ◆ Definition von Schriftart, Schriftgröße, Farbe usw. durch HTML-Befehle möglich
- ◆ Aufeinanderfolgende Leerzeichen, Leerzeilen, Zeilen- und Seitenumbrüche und sonstige Steuerzeichen werden ignoriert!
- ◆ Ausnahme: **<pre> ... </pre>**

Überschriften

- ◆ Kennzeichnung durch `<h1>` bis `<h6>`
- ◆ Zahl gibt die Schriftgröße an (1 ist die größte Überschrift)
- ◆ Ende-Tag **muss** vorhanden sein
- ◆ Überschriften sind Absätze und müssen nicht mehr als solche gekennzeichnet werden

Absatz <p>

- ◆ Kennzeichnung durch

`<p [align="..."] > ... </p>`

- ◆ Automatischer **Zeilenvorschub** vor und nach dem Absatz
- ◆ **Abstand** (Leerzeile) vor und nach dem Absatz
- ◆ Da im Allg. Absätze immer unmittelbar aufeinanderfolgen, kann das Ende-Tag `</p>` entfallen (nur in HTML)

Abschnitt `<div>`

- ◆ Kennzeichnung durch
`<div align="..." > ... </div>`
- ◆ Entspricht `<p>` (Zeilenvorschub vor und nach dem Absatz), jedoch **ohne Abstand** vor und nach dem Abschnitt
- ◆ Abschnitte fassen mehrere Seiten-Einheiten (z.B. mehrere Absätze und Überschriften) zusammen, die gemeinsam formatiert (z.B. zentriert) werden.

`<center>` entspricht `<div align="center">`

Achtung!

Ich sagte: **<center>nichts</center>**
würldiger Herr Direktor.

Ich sagte:
nichts
würldiger Herr Direktor.

Ich sagte: nichtswürldiger Herr Direktor.

Attribut **ALIGN**

- ◆ Gibt es bei mehreren HTML-Befehlen - u.a. bei Absätzen, Überschriften und Tabellen
- ◆ Ermöglicht die **Ausrichtung** eines markierten Bereichs (z.B. Absatz)
- ◆ Mögliche Werte:

align="left"

(Standardannahme)

align="center"

align="right"

Attribut ALIGN - Beispiel

<h2 align="center" >

Zentrierte Überschrift

</h2>

<p>

Linksbündiger Absatz

<p align="right" >

Rechtsbündiger Absatz

Abschnitt ``

- ◆ Entspricht `<div>`, jedoch **ohne Zeilenumbruch** vor und nach dem Abschnitt
- ◆ Kennzeichnung durch ` ... `
- ◆ Zur Formatierung von einzelnen Sätzen oder Wörtern innerhalb eines Absatzes
- ◆ Ebenso wie `<div>` sehr gut geeignet für die Verwendung von CSS ("Stylesheets")

Listen

◆ Ungeordnete Listen `` ... ``

- Listelemente (``) werden durch sog. "Bullets" markiert

◆ Geordnete Listen `` ... ``

- Listelemente (``) werden automatisch nummeriert

◆ Definitionslisten `<dl>` ... `</dl>`

- Listelemente bestehen aus einem Begriff (`<dt>`) und einer Erklärung (`<dd>`)

Ungeordnete Listen (nicht nummeriert)

- ◆ Kennzeichnung durch ` ... `
- ◆ Kennzeichnung jedes>Listelementes durch ``
 - Ende-Tag `` kann weggelassen werden (nur in HTML)
- ◆ Vor jedem>Listelement steht ein **Markierungszeichen** ("Bullet")

Ungeordnete Listen

Attribut TYPE

- ◆ Legt das Aussehen des Markierungszeichens fest
- ◆ Mögliche Werte:

type="disc" (Punkt, Standardannahme)

type="circle" (Kreis)

type="square" (Quadrat)

Ungeordnete Listen - Beispiel

```
<ul type="circle" >  
  <li>Erd&auml;pfel  
  <li>Paradeiser  
  <li>Paprika  
  <li>Fisolen  
</ul>
```

Geordnete Listen (nummeriert)

- ◆ Kennzeichnung durch ` ... `
- ◆ Kennzeichnung jedes>Listelementes durch ``
 - Ende-Tag `` kann weggelassen werden
(nur in HTML)
- ◆ Vor jedem>Listelement steht eine **automatische Nummerierung**

Geordnete Listen - Attribute

◆ TYPE

– legt die **Art** der Nummerierung fest

◆ START

– legt die **Startnummer** fest

Attribut TYPE

Mögliche Werte:

- type="1"** (arab. Ziffern, Standardannahme)
- type="I"** (röm. Ziffern mit Großbuchstaben)
- type="i"** (röm. Ziffern mit Kleinbuchstaben)
- type="A"** (Aufzählung mit Großbuchstaben)
- type="a"** (Aufzählung mit Kleinbuchstaben)

Geordnete Listen - Beispiel

```
<h3>Inhaltsverzeichnis</h3>
```

```
<ol type="a" start="4" >
```

```
  <li><a href="#d">Kapitel f&uuml;r d-Text</a>
```

```
  <li><a href="#e">Kapitel f&uuml;r e-Text</a>
```

```
  <li><a href="#f">Kapitel f&uuml;r f-Text</a>
```

```
  <li><a href="#g">Kapitel f&uuml;r g-Text</a>
```

```
</ol>
```

Definitionslisten

- ◆ Kennzeichnung durch `<dl> ... </dl>`
- ◆ Jedes Listelement wird gekennzeichnet durch `<dt>` und `<dd>`
 - `<dt>` markiert den zu erklärenden Begriff (**Definitionsterm**)
 - `<dd>` markiert die Erklärung (**Definition**)
 - Ende-Tags `</dt>`, `</dd>` können fehlen (nur in HTML)

Definitionslisten - Beispiel

<dl>

<dt> Begriff1

<dd> Erklärung zu Begriff1

<dt> Begriff2

<dd> Erklärung zu Begriff2

<dt> Begriff3

<dd> Erklärung zu Begriff3

</dl>

Text-Hervorhebungen (1)

- ◆ Alle nachfolgenden Befehle benötigen ein abschließendes **Ende-Tag!**
- ◆ **<big>** größere Schrift
- ◆ **<small>** ... kleinere Schrift
- ◆ **<sub>** tiefgesetzte Schrift (Index)
- ◆ **<sup>** hochgesetzte Schrift (Exponent)

Text-Hervorhebungen (2)

- ◆ Alle nachfolgenden Befehle benötigen ein abschließendes **Ende-Tag!**
- ◆ **** fettgedruckte Schrift
- ◆ **<i>** Kursivschrift
- ◆ **<tt>** nicht proportionale Schrift
(meist Courier)

Text-Hervorhebungen (3)

- ◆ Alle nachfolgenden Befehle benötigen ein abschließendes **Ende-Tag!**
- ◆ **** hervorgehobener Text
(meist kursiv)
- ◆ **** ... stark hervorgehobener Text
(meist fett)

Text-Hervorhebungen (4)

- ◆ Alle nachfolgenden Befehle benötigen ein abschließendes **Ende-Tag**!
- ◆ **<address>** für Impressum
(E-Mail-Adresse, Autor)
(meist kursiv)
- ◆ **<blockquote>** ... für Zitate
(meist eingerückt)
- ◆ **<cite>** für Quellenangaben
(meist kursiv)

Text-Hervorhebungen (5)

- ◆ Alle nachfolgenden Befehle benötigen ein abschließendes **Ende-Tag!**
- ◆ **<dfn>** für Definitionen (meist fett)
- ◆ **<samp>** ... für Beispiele
(meist eingerückt)
- ◆ **<code>** für Programmcode
(meist nichtproportional)

Text-Trennung

- ◆ **<hr />** ... Einfügen einer horizontalen Linie
(leeres Tag)

Textumbrüche erfolgen **automatisch** durch den Browser (je nach Fenstergröße)

- ◆ **
** ... erzwingt Zeilenumbruch
(leeres Tag)

- ◆ **<nobr> ... </nobr>**
verhindert automatischen Zeilenumbruch

Bilder

- ◆ Einfügen in eine Seite durch das leere Tag

```

```

- ◆ XURL kann eine absolute oder relative Adresse einer Bilddatei sein
- ◆ Alternativer Text möglich, wenn Browser das Bild nicht darstellen kann
- ◆ Format des Bildes:

GIF oder **JPG (JPEG)**

Tabellen

- ◆ Kennzeichnung durch `<table> ... </table>`
- ◆ Tabellenüberschrift (oder -unterschrift) durch `<caption> ... </caption>`
- ◆ **Zeilen** einer Tabelle werden durch `<tr>` voneinander getrennt
- ◆ **Spalten** innerhalb einer Zeile werden durch `<td>` oder `<th>` voneinander getrennt

Tabellenbeschriftung

- ◆ Kennzeichnung durch
<caption> ... </caption>
- ◆ Attribut **ALIGN** definiert den Text als
Über- oder Unterschrift
- ◆ Mögliche Werte:
align="top" (Standardannahme)
align="bottom"

Tabellenelemente

- ◆ Ein Tabellenelement wird auch **ZELLE** genannt
- ◆ Eine Zelle kann sich über mehrere Spalten und/oder mehrere Zeilen erstrecken
- ◆ Der Inhalt einer Zelle wird durch `<td> ... </td>` oder durch `<th> ... </th>` festgelegt

Tabellenelemente (Zellen)

- ◆ Kennzeichnung durch `<td>`
 - Normalfall
- ◆ Kennzeichnung durch `<th>`
 - Inhalt wird fett gedruckt
 - ein solches Tabellenelement kann daher als Spaltenüberschrift verwendet werden
- ◆ Die abschließenden Ende-Tags `</td>` und `</th>` können fehlen (nur in HTML)

Tabelle - Beispiel (1)

<pre>

Buchhaltung

```
-----  
| Name | Soll | Haben |  
|-----|-----|-----|  
| Alfred | 12,40 | 27,35 |  
|-----|-----|-----|  
| Franz | 2,70 | 32,55 |  
-----
```

</pre>

Tabelle - Beispiel (2)

```
<table border="1" >
  <caption>Buchhaltung</caption>
  <tr align="center" >
    <th>Name<th>Soll<th>Haben
  <tr align="right" >
    <td align="left" >Alfred<td>12,40<td>27,35
  <tr align="right" >
    <td align="left" >Franz<td>2,70<td>32,55
</table>
```

<table> - Attribute (1)

◆ ALIGN

- Ausrichtung
- **left**, **center**, **right**

◆ BORDER

- Umrahmung der Zellen mit Linien
- **0** keine Linien (Standardannahme)
- **1**, **2**, ... (Liniendicke)

<table> - Attribute (2)

◆ **COLS**

- Anzahl der Spalten
- beschleunigt den Tabellenaufbau

◆ **WIDTH**

- Breite der Tabelle in Pixel (absolut) oder in % der Fensterbreite

<tr> - Attribute

◆ **ALIGN**

- horizontale Ausrichtung
- **left, center, right**

◆ **VALIGN**

- vertikale Ausrichtung
- **top, middle, bottom, baseline**

<td> / <th> - Attribute (1)

◆ ALIGN

- Ausrichtung
- **left, center, right**

◆ VALIGN

- vertikale Ausrichtung
- **top, middle, bottom, baseline**

◆ WIDTH

- Breite der Tabelle in Pixel (absolut) oder in % der Fensterbreite

<td> / <th> - Attribute (2)

◆ NOWRAP

- Zelleninhalt wird nicht automatisch umbrochen

◆ COLSPAN

- Zelle belegt mehrere Spalten
- Anzahl der Spalten

◆ ROWSPAN

- Zelle belegt mehrere Zeilen
- Anzahl der Zeilen

Applets

- ◆ Kennzeichnung durch

<applet> ... </applet>

- ◆ Dazwischen kann ein alternativer Text stehen, wenn Browser das Applet nicht darstellen kann
- ◆ Eingefügt wird immer eine **Java-Applet-Klasse** (d.h. eine **.class**-Datei)
- ◆ Die Größe des Applet-Fensters muss ebenfalls angegeben werden

<applet> - Attribute

◆ **CODE**

- Name (ev. mit URL) der Applet-Klasse

◆ **WIDTH**

- Breite des Applet-Fensters in Pixel

◆ **HEIGHT**

- Höhe des Applet-Fensters in Pixel

Applet - Beispiel

```
<applet code="AppletXY.class"  
        width="300"  
        height="200">
```

Leider kann Ihr Browser das Applet nicht darstellen.

```
</applet>
```

Rahmen (Frames)

- ◆ **Unterteilung** einer Seite in Rahmen mittels **<frameset> ... </frameset>**
- ◆ Diese Befehle ersetzen die Befehle **<body> ... </body>**
- ◆ **Rahmen** innerhalb eines **<frameset>**-Bereichs wird definiert durch ein leeres **<frame>**-Tag
- ◆ Jeder Rahmen ist eine eigene HTML-Datei

Rahmen (Frames)

- ◆ Für Browser, die keine Rahmen kennen, kann innerhalb eines **<noframes>**-Bereichs ein kompletter **<body>**-Bereich definiert werden, der anstelle des **<frameset>**-Bereichs dargestellt wird.
- ◆ Dieser **<noframes>**-Bereich wird von Browsern, die Rahmen kennen, völlig ignoriert.

Rahmen (Frames)

- ◆ Ein **<frameset>**-Bereich wird in Spalten und/oder in Zeilen aufgeteilt
- ◆ **<frameset>**-Bereiche können ineinander geschachtelt werden
- ◆ Damit wird **horizontale** und **vertikale** Unterteilung ermöglicht

<frameset> - Attribute

◆ ROWS

- vertikale Unterteilung in **Zeilen**
- Rahmengröße (= Zeilenhöhe) in Pixel (absolut) oder in % der <frameset>-Bereichshöhe

◆ COLS

- horizontale Unterteilung in **Spalten**
- Rahmengröße (= Spaltenbreite) in Pixel (absolut) oder in % der <frameset>-Bereichsbreite

- ◆ * ... Größe für verbleibende Zeile/Spalte

<frame> - Attribute

◆ SRC

- URL des Rahmens

◆ NAME

- eindeutiger Name ("label") für Erreichbarkeit des Rahmens über Querverweise (Links)
(Attribut **TARGET** in einem `<a href>`-Befehl)

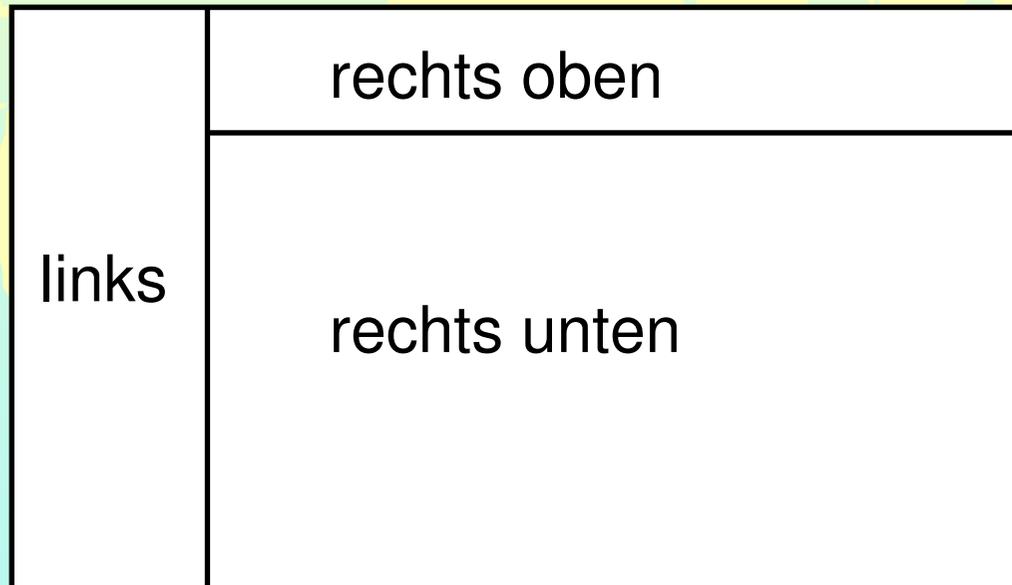
◆ SCROLLING

- **yes**, **no**, **auto**

◆ NORESIZE

Rahmen - Beispiel (1)

- ◆ Eine Seite soll folgendermaßen unterteilt werden:



Rahmen - Beispiel (2)

```
<html><head><title>Seite mit 3 Rahmen</title></head>
<frameset cols="20%,*" >
  <frame src="links.html" name="links" >
    <frameset rows="15%,*" >
      <frame src="reOben.html" name="reOb" >
      <frame src="reUnten.html" name="reUnt" >
    </frameset>
  </frame>
</frameset>
<noframes>
  <body>Ihr Browser kennt keine Frames</body>
</noframes>
</html>
```

<a href>-Attribut TARGET

- ◆ Verweise in einem Rahmen zu neuen Seiten sollten im Allg. die Zieldarstellung **"neues Fenster"** enthalten:

- ◆ ev. juristische Probleme (fremde Seiten)
- ◆ ev. Darstellungsprobleme (neue Seite ist selbst ein Rahmen)

Formulare

`<form action="URI" method="{ get } " >`
`post`

... (Eingabefelder,
Auswahllisten,
Schaltflächen)

`</form>`

URI ≈ **Programm** (z.B. PHP-Skript, Java-Servlet)
oder **Email-Adresse** (**mailto:...**)

Formulare - GET / POST

Es gibt 2 Methoden im HTTP-Protokoll zur Übertragung der **Formulardaten**:

◆ **GET-Methode:**

Daten werden mit URL in folgender Form übertragen:

URL?par1=arg1&par2=arg2&...

(Leerzeichen werden mit **+** dargestellt)

Datenlänge jedoch beschränkt (ca. 2000 Zeichen)

◆ **POST-Methode:**

Daten werden im HTTP-Körper übertragen

Datenlänge ist beliebig

Formulare - Textfelder

```
<input type=" { text } " name="Name"
```

```
[value="Vorgabetext"] [readonly]
```

```
[size="n"] [maxlength="m"] />
```

Über den Formularelementnamen "*Name*" erhält man den eingegebenen Text. "*readonly*" definiert ein Ausgabefeld mit dem Inhalt "*Vorgabetext*".

Formulare - Optionsfelder

```
<input type="{ radio  
checkbox }" name="Name"  
value="Option" [checked] />
```

Bei einer Auswahlkreisgruppe ("Radiobuttons") muss jedes input-Tag **denselben** Namen haben - über diesen Namen erhält man die (eine) ausgewählte Option!

Formulare - Schaltflächen

```
<input type="submit" name="Name"  
[value="Beschriftung"] />
```

"submit" löst die Übertragung der Formulardaten aus, "reset" setzt alle Formulardaten zurück.

Formulare - Verstecktes Feld

```
<input type="hidden" name="Name"  
value="versteckteInformation" />
```

Damit können Informationen über mehrere
Formulare hinweg übertragen werden.

Formulare - Textbox

```
<textarea name="Name" [readonly]  
  rows="zeilen" cols="spalten" >
```

... (beliebiger Vorgabetext)

```
</textarea>
```

Formulare - Listen

```
<select name="Name" [multiple] size="n" >  
  <option [value="text1"] [selected] >  
    optionstext1  
  </option>  
  ... (weitere Optionen)  
</select>
```

size="1" → Kombinationslistenfeld ("Combobox")

size > 1 → Liste

multiple → markieren mehrerer Optionen möglich

Weitere HTML-Befehle

Farben

Spezialeffekte

**Cascading Style Sheets
(CSS)**

VIEL ERFOLG

