

# PPS-Veranstaltung

## 1. Praxisveranstaltung

# Praxisstunden Ablauf

- Ziel:
  - Besprechung von ausgewählten Themen
  - Beantwortung von praktischen Fragen
- Praxisstunde jeweils Freitags 13.15 und zwar (SIEHE AUCH Webseite!)
  - 15. November 2002, HTML Navigation Elemente
  - 29. November 2002, Style Sheets
  - 06. Dezember 2002, Server-Side Skripts
  - 13. Dezember 2002, Forms/Animationen (Client-Side Skripts)
  - 20. Dezember 2002, Fragestunde vor Abgabe
  - 24. Januar 2003, Fragestunde II vor Abgabe (Reserve)
- Abgabe PPS:
  - 07. Februar 2003, 13.15 - 16.00, Schlusspraesentation-Bewertung

*Fuer weitere Informationen: siehe [webseite des Praktikums](#)*

# PPS-Web Aufgabenstellung:

## 4 Anforderungen an Webseite

1. Navigationsmechanismen
  - Mehrere HTML Seiten sollen erstellt werden
  - Frames, Tabellen, Skripts (kurz)
2. Einheitliche Gestaltung
  - Style-Sheets (CSS)
3. Dynamische Elemente
  - Seite oder Teile von HTML werden dynamisch erstellt (DHTML)
  - Skripts (CGI, Perl, PHP, JavaScript, ...)
  - Bsp: Gaestebuch
4. Animation
  - JavaScript, Java-Applet
  - evtl. Macromedia, Flash, ...

# Beispiel fuer Navigationsmechanismus: Frames

*Frames → "Informationsverteilung auf mehrere, voneinander unabhängige Fenster"*

(Bsp.)

# Frames Bezogene HTML Elemente (1)

- Frameset Element → Layout von einzelnen Frame  
*Beachte:* Ein Dokument, dass Frames enthaelt ("Frameset Document") hat ein Head und ein Frameset Element (anstatt von einem Body Element, wie ein normales ohne Frames HTML Dokument)

Wichtigere Attributen: cols (Spalten), rows (Reihen)

```
<html>
<head><title>...</title></head>
<frameset cols="20%,*">
...Inhalt erstens Frames...
...Inhalt zweitens Frames...
</frameset> ...
</html>
```

Bsp: ["Leere" Frameset Seite](#)

# Frames Bezogene HTML Elemente (2)

- Frame Element → Inhalt und Aussehen eines einzelnen Frames

Wichtigere Attributen: src, name

```
<html>
<head><title>...</title></head>
<frameset cols="20%,*">
  <frame name="TOC" src="toc.html">
  <frame name="CONTENT" src="doc1.html">
</frameset> ...
</html>
```

Bsp: Einfache Frameset Seite

# Frames Bezogene HTML Elemente (3)

- NoFrames Element → Alternativer Inhalt, der von Browsers angezeigt wird, die keine Frames unterstützen

```
<html>
<head><title>...</title></head>
<frameset cols="20%,*">
  <frame name="TOC" src="toc.html">
  <frame name="CONTENT" src="doc1.html">
  <noframes> <p>sorry, but this page is for frame-
capable browsers only...</p>
</noframes>
</frameset> ...
</html>
```

Bsp: [Einfache Frameset Seite](#) mit noframes im Lynx

# Frameset mit 3 Frames

Case Study: Beispielseite fuer ZVV Fahrplan

The screenshot shows a Netscape browser window titled "Netscape: Fahrplan". The browser interface includes a menu bar (File, Edit, View, Go, Communicator, Help) and a toolbar with various icons. The main content area is divided into three frames:

- Frame "TRAMLINES"**: Located on the left, it contains a list of tram lines under "Tramlinien ETH Zentrum:" and "Voltastrasse:". It also lists "ETH/Unispital:" lines. A link "Linie 5" is highlighted with a dashed arrow pointing to the "SCHEDULES" frame.
- Frame "CURRLINE"**: Located at the bottom left, it displays "Linie 5 ab Voltastrasse" with links for "Richtung Enge", "Richtung Kirche", and "Fluntern". A solid arrow points from this frame to the "SCHEDULES" frame.
- Frame "SCHEDULES"**: The main content area, displaying the "Haltestellen-Fahrpläne Linie 5" for the year 2000/2001. It includes a navigation menu with links like "Die Fahrkarten", "Die Spassmacher", "Die Stadtrundfahrt", "Die VBZ", "Der Fahrplan", "Die Angebote", "Der Service", and "Ihr Wegweiser". It also features a "Willkommen ZVV-Fahrplan" section with various service links and a "Haltestellen-Fahrpläne Linie 5" section with directions to "Bahnhof Enge" and "Zoo".

Beim Klicken auf Link sollen beide Frames geaendert werden

# JavaScript Integration

Um durch ein Event, mehrere Frames gleichzeitig zu aendern, gebraucht man DHTML bzw. Javascript

(TRAMLINES Frame - main.html)

```
<html> <head>...</head>
<script type="text/javascript" language="javascript">
  function choose(line) {
    top.frames["CURRLINE"].location="line"+line+".html"; return
    true; }
</script>
<p><b>Tramlinien ETH Zentrum:</b>
<p>Voltastrasse:
<ul>
  <li> <a
    href="http://www.zvv.ch/abfahrtsfahrplan.asp?linie=005"
    onClick="choose(5); return true;" target="SCHEDULES">Linie
    5</a>
  </li>
</ul> <p>ETH/Unispital: ... </body> </html>
```

# Gebrauch von Frames

- Zum ständigen Einblenden projektglobaler Steuerverweise
- Zum gleichzeitigen Anzeigen von zu vergleichenden Informationen
- Bei besonders kunstvoller Seitengestaltung !

# Nachteile von Frames

- Probleme bei nicht Frame-fähigen Web-Browsern
- Frames und Bildschirmauflösung
- Ladezeiten
- Problematisches Direktansteuern von untergeordneten Seiten (juristisch bedenklich)
- Rekursive Framesets (Bsp. 1)

Mögliche Lösung zu den zwei letzten Problemen:

```
<a href="http://..." target="_new">Verweistext</a>
```

(Bsp. 2)

# Tabellen als Navigationsmechanismus

- "Blinde Tabellen": Tabellen mit keinen sichtbaren Gitternetzlinien. Man gebraucht zur Sitenavigation eine die ganze Seite ueberspannende "blinde" Tabelle.

Bsp: [Einfache Seite mit blinder Tabelle](#)

- Eine "klassische" Design-Struktur:
  - Meistens gebraucht man alle Spalten der obersten Reihe fuer die Seitenuberschrift und andere Elemente (Logos, Banners usw.)
  - Eine Spalte am links (schmaler Flaeche) als Navigation Bar
  - Eine groessere Flaeche rechts vom Navigation Bar fuer den Inhalt der Seite

Bsp: [Einfache Seite mit blinder Tabelle gezeigt mit Gitternetzlinien](#)

# Gestaltung von blinden Tabellen

- Gitternetzlinien und Rahmen sollen nicht angezeigt werden (border)
- Zwischen Benachbarten Zellen, sowie zwischen Zellen und Rahmen keine sichtbare Luecken (cellpadding, cellspacing)
- Die gesamte Tabelle soll sich ueber die volle verfügbare Breite der Seite erstrecken (width)

```
<table border="0" cellpadding="0" cellspacing="0"  
width="100%" ...>  
...  
</table>
```

Bsp: [Einfache Seite mit blinder Tabelle im ganzen Site](#)

# Accessibility/Usability Links

- [Dive into Accessibility](#) (Mark Pilgrim)
- [Building an Accessible Website](#) (CNET)

## Und zur Beachtung:

- Rechtliche Regelungen: [Internet-Richtlinien der ETHZ](#)

# Einfuehrung in XHTML

XHTML: "XML-izing" HTML !

Aber Warum genuegt HTML nicht?  
Bisschen Geschichte hilft immer!

# Evolution von Markup Sprachen

**SGML** : Eine Sprache, mit der man Markup Sprachen beschreiben kann

- Flexibel
- Umfangreich
- Aber dazu: sehr kompliziert, um fuer diversen Bereichen zu gebrauchen (z.B. WWW)

**HTML** : Anwendung von SGML

- Einfacher als SGML
- Wurde stark fuer die Darstellung von wissenschaftlichen Dokumenten und von Dokumenten allgemeiner Art gebraucht.
- Darstellung von Multimedia Komponenten
- **ABER:** Die Anforderungen neuer Applikationen hat dazu gefuehrt, dass immer neue HTML-Elemente definiert wurden. Das hat Kompatibilitaet Problemen zwischen verschiedenen Plattformen verursacht.

## Evolution von Markup Sprachen (2)

XML : Auch Anwendung von SGML, die umfangreich und flexibel wie SGML ist, aber gleichzeitig nicht extrem kompliziert

XHTML : Untermenge von XML, eine "strikttere" HTML

- XHTML Dokumente sind XML conforming
- XHTML Dokumente sind (unter dem Gebrauch von bestimmten Richtlinien) kompatibel mit HTML 4 Browsers.
- XHTML ist flexibel: die Herstellung von neuen Elementen ist einfach

# Gestaltung von XHTML

- Wurzel-Element muss das `<html>` Element sein.
- Bestimmung vom XML namespace

```
<html xmlns=http://www.w3.org/1999/xhtml  
xml:lang="en" lang="en">
```

- Bestimmung vom DTD (Document Type Definition).  
Das XHTML Dokument wird laut den "Regeln", die in dem angegebenen DTD vorhanden sind, nach Richtigkeit validiert.
- Das Dokument muss sich richtig (laut angegebenem DTD) validieren lassen.

# Beispiel: Minimales XHTML Dokument

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
  Strict//EN" "DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xml:lang="en" lang="en">
  <head>
    <title>Minimal XHTML Doc</title>
  </head>
  <body>
    <p> Minimal saves always time!
    </p>
  </body>
</html>
```

[\(Xhtml minimal.html\)](#)

# XHTML Regeln

- Um XHTML-compliant Dokumente zu erstellen, braucht man einfach sich die Unterschiede zur HTML 4 zu merken.
- Anschliessend kann man das hergestellte Dokument von dem W3C Validator validieren lassen.

# Backup: Framesets und Frames

