

WWW – Grundlagen und Technologie

Hypertext Markup Language (HTML) und Dynamic HTML (DHTML)

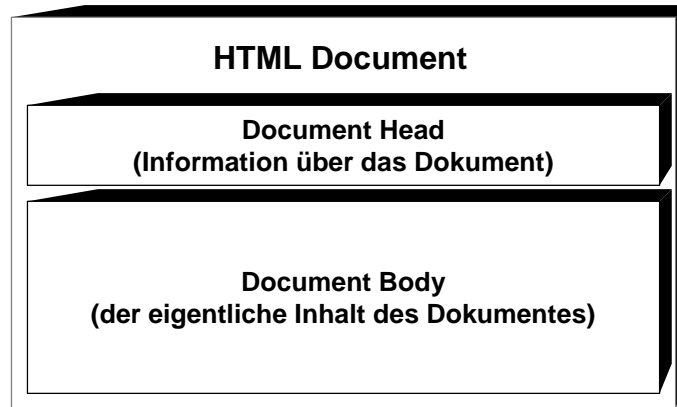


Erik Wilde
TIK – ETH Zürich
Sommersemester 2001

Übersicht

- Struktur einer HTML-Seite
- Tabellen als Mittel zur Formatierung
 - neue Features (HTML 4.01) von Tabellen
- Forms als Dateneingabefelder in HTML
 - Interface-Elemente und ihre Verwendung
 - neue Features (HTML 4.01) von Forms
- Framesets und Frames
- Image Maps (Clickable Images)
- Dynamic HTML (DHTML)

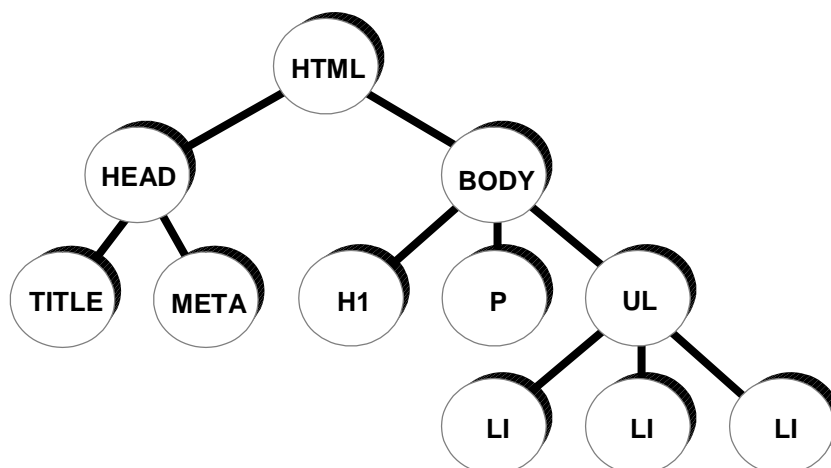
Aufbau einer HTML-Seite (I)



WWW (SS2001) - HTML

3

Aufbau einer HTML-Seite (II)



WWW (SS2001) - HTML

4

Elemente des Document Head

- Dokumententitel: <TITLE>
 - üblicherweise als Fenstertitel und in Listen
- Basisadresse: <BASE>
 - wichtig für die Interpretation relativer URLs
- Links zu anderen Ressourcen: <LINK>
 - für Beziehungen zwischen Ressourcen
- Style Sheets: <STYLE>
- Scripte für Scripting Languages: <SCRIPT>
- Meta-Information: <META>

Elemente des Document Body

- Überschriften (auf verschiedenen Ebenen)
- Absätze (normaler Fliesstext)
- verschieden formatierter Text
- Listen (numeriert und unnumeriert)
- Verweise auf andere Dokumente (Hyperlinks)
- horizontale Begrenzungslinien
- Tabellen
- Graphiken (in verschiedenen Formaten)
- allgemeine externe Objekte (z.B. Applets)
- Formulare (im Browser auszufüllen)

Tabellen

- Folge von Zeilen, Folge von Zellen
 - <TABLE>, <CAPTION>, <TR>, <TH>, <TD>
- Layoutbeeinflussung durch Attribute
 - ALIGN, ersetzbar durch <DIV>
 - Umrandung von Tabelle und Zellen mit BORDER
 - Breite in Pixel oder Prozent (WIDTH)
 - Abstand der Tabellenzellen (CELLSPACING)
 - Abstand Zelleninhalt und Border (CELLPADDING)
- Positionierung des Inhaltes der Zellen
 - ALIGN und VALIGN für <TR>, <TH> oder <TD>
- COLSPAN und ROWSPAN (<TH>, <TD>)

Tabellen in HTML 4.01 vs. 3.2

- Ausrichtung an Zeichen (z.B. Dezimalpunkte)
 - ermöglicht die Formatierung mit weniger Spalten
- bessere Ausrichtung innerhalb der Cells
- mehr Varianten für Borders
- Strukturierung in *Head, Body, Foot*
 - bessere Formatierung beim Scrollen oder Drucken
- Strukturierung in *Column Groups*
 - bessere Lesbarkeit und nicht-graphische Darstellung
- keine Unterstützung in *CSS1*, aber in *CSS2*

HTML Forms

- ermöglichen die Eingabe von Daten
- brauchen Datenübermittlung (HTTP/CGI)
- mit dem <FORM> Element spezifiziert
 - ACTION="URL" bestimmt die Verarbeitung
 - METHOD="GET"/"POST" bestimmt HTTP Methode
- für Eingaben werden Elemente verwendet
 - <INPUT> definiert viele Eingabemöglichkeiten
 - <TEXTAREA> für mehrzeiligen Text
 - <SELECT> definiert Optionslisten
- andere HTML-Elemente sind ebenso erlaubt

Verwendung von HTML Forms

File Edit View Go Communicator Help

Herr Frau

Name:

Vorname:

100%

`/cgi-bin/form.pl?A=H&N=Wilde&V=Erik`

Forms in HTML

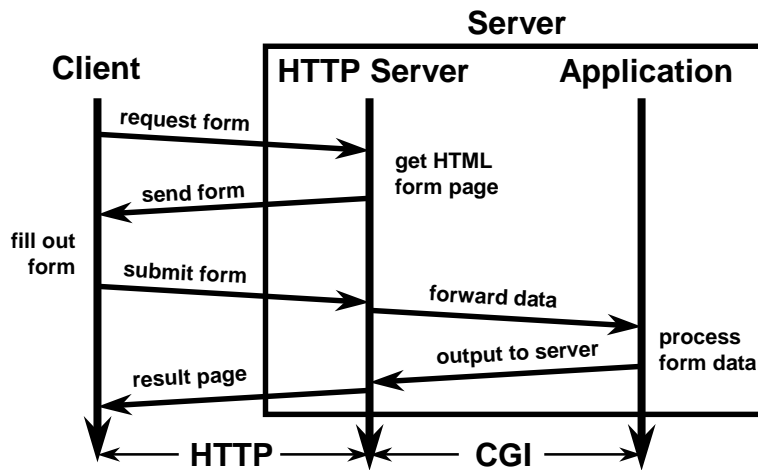
```
<FORM ACTION="/cgi-bin/form.pl" METHOD="GET">
<TABLE>
  <TR>
    <TD COLSPAN=2>
      <INPUT TYPE="RADIO" NAME="A" VALUE="H"> Herr
      <INPUT TYPE="RADIO" NAME="A" VALUE="F"> Frau
    <TR>
      <TD>Name:
      <TD><INPUT TYPE="TEXT" NAME="N" SIZE="20">
    <TR>
      <TD>Vorname:
      <TD><INPUT TYPE="TEXT" NAME="V" SIZE="20">
    </TABLE>
<INPUT TYPE="SUBMIT" VALUE="Absenden">
</FORM>
```

Checkboxes

- Erdbeer
- Zitrone
- Schoko
- Vanille
- Straciatella

```
<INPUT TYPE="checkbox" NAME="Geschmack"
VALUE="Erdbeer">Erdbeer<BR>
<INPUT TYPE="checkbox" NAME="Geschmack"
VALUE="Zitrone">Zitrone<BR>
<INPUT TYPE="checkbox" NAME="Geschmack"
VALUE="Schoko"
CHECKED="checked">Schoko<BR>
<INPUT TYPE="checkbox" NAME="Geschmack"
VALUE="Vanille"
CHECKED="checked">Vanille<BR>
<INPUT TYPE="checkbox" NAME="Geschmack"
VALUE="Straciatella">Straciatella
```

Form Handling



WWW (SS2001) - HTML

13

HTML 4.01 Features von Forms (I)

- TABINDEX wählt Reihenfolge der Felder
 - für tastaturgesteuerte Eingabe in Forms
- ACCESSKEY wählt Tastatur-Abkürzungen
 - werden vom Browser hervorgehoben angezeigt
- DISABLED macht Elemente insensitiv
 - Verwendung für konsistentes Interface-Design
- TITLE enthält Hilfs-Texte und Erläuterungen
 - wird vom Browser angezeigt
- ONCHANGE für Skript zur Eingabepfung
 - Felder können schon beim Client getestet werden
- ACCEPT für Limitierung erlaubter File-Typen
 - nur für `<INPUT FILE>` definiert

WWW (SS2001) - HTML

14

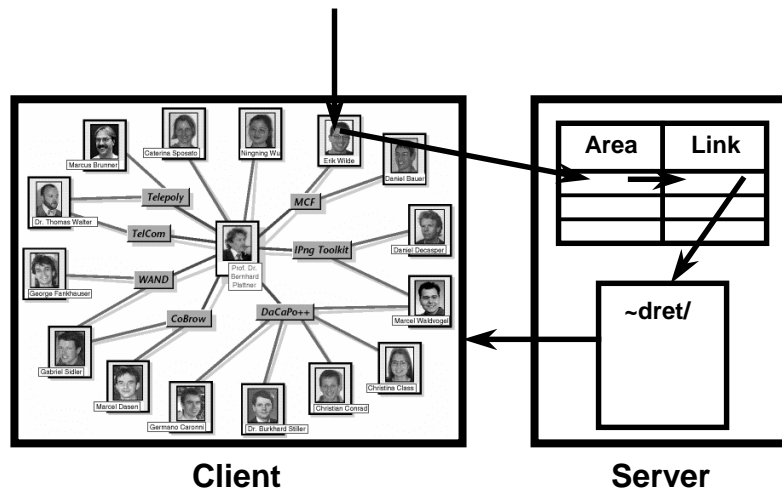
HTML 4.01 Features von Forms (II)

- **<FIELDSET>** für strukturierte Forms
 - noch nicht allgemein unterstützt (neu in HTML 4.0)
 - Gruppierung nach Themengruppen
 - besonders nützlich für nicht-visuelle Browser
- **<LEGEND>** für Benennung der Bereiche
 - Text oder andere Inline Elemente
 - sollte auch ohne **<FIELDSET>** sinnvoll sein
 - z.B. "Angaben zur Person" und "Bestellung"
- Ausnutzung der HTML-Konvention
 - Ignorieren unbekannter Elemente und Attribute
 - trotzdem Darstellung des Inhaltes

Clickable Images (Image Maps)

- Zuordnung von Links zu Bildbereichen
- ermöglicht "graphische" Interfaces
- verbleibende Limitierungen
 - keine dynamischen Menüs (pop-up, scroll-down)
- Design der Oberfläche als Bild
- Zuordnung sensibler Bereiche zu Bildteilen
 - einfache geometrische Bereiche
 - damit assoziierte URIs

Server-Side Image Map



WWW (SS2001) - HTML

17

Server-Side Image Map - HTML

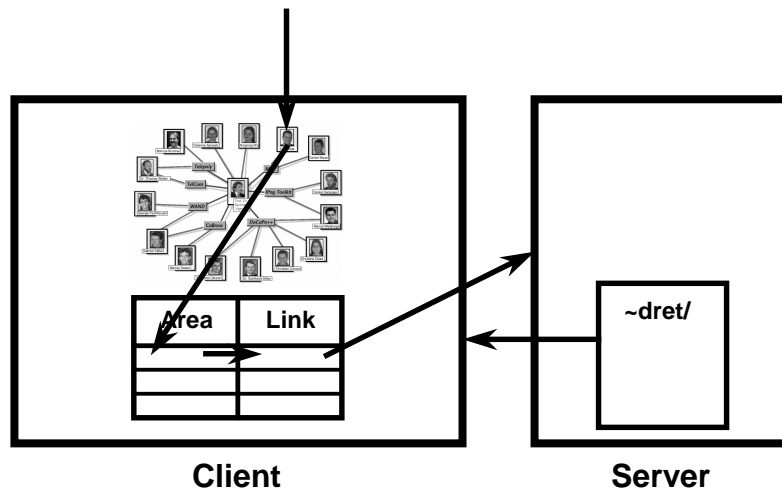
```
<A HREF="http://www.ethz.ch/cgi-bin/index">
<IMG ALT="Auswahl verschiedener ETH-Dienste"
    SRC="dienste.gif" WIDTH="200" HEIGHT="30" ISMAP>
```

- Script zur Verarbeitung der Koordinaten
 - Client sendet *query URL* "...bin/index?153,21"
- Map-Verarbeitung in den Server integriert
 - typischerweise `.map` Referenzen
 - Map-Files enthalten Bereiche und URLs
- Scripte sind langsam und fehleranfällig, wenn vorhanden, Servermechanismus verwenden

WWW (SS2001) - HTML

18

Client-Side Image Map



WWW (SS2001) - HTML

19

Client-Side Image Map - HTML

```
<IMG SRC="img003.gif" usemap="#Objmap"  
WIDTH=500 HEIGHT=370 BORDER=0>
```

```
<MAP NAME="Objmap">
```

```
<AREA SHAPE="POLYGON"
```

```
COORDS="87, 261, 405, 261, 405, 281, 87, 281"
```

```
href="http://www.tik.ee.ethz.ch/~dret/WWW-VL/" >
```

```
</MAP>
```

- Beispiel aus generierten Frames (PowerPoint)
- Map-Verarbeitung in den Client integriert
- ermöglicht neue Features (ändernde Pointer)

WWW (SS2001) - HTML

20

Verwendung mehrerer Fenster

- Zuordnung eines Links zu einem Fenster
- ermöglichen das Erzeugen neuer Fenster
 - `Link`
 - `<BASE TARGET="Fenster2">`
 - `<FORM TARGET="Fenster3">`
 - reserviert: `_blank`, `_self`, `_parent`, `_top`
- Frames ermöglichen das Unterteilen eines Fensters ("Fenster im Fenster")

```
<FRAMESET COLS="25%,75%">
  <FRAME SRC="Inhalt.html">
  <FRAME SRC="Pre.html" NAME="Text">
</FRAMESET>
```

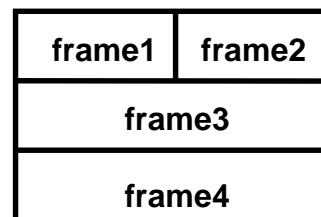
Fenster im Fenster

The screenshot shows a web browser window displaying a glossary page. The page is divided into two main sections by a horizontal line. The top section, which is a frame, contains a list of abbreviations and their full names, starting with 'AAC' and ending with 'AOL'. The bottom section contains a welcome message: 'Welcome to the online version of Wilde's WWW glossary. The abbreviation index (shown below) provides a concise list of all abbreviations, while the individual letter indices (always accessible via the letter bar) list the full names as well. In addition to this online version, a PDF version is available. Please note that the PDF version contains less topics than the online version, and also less structural information. Basically, the PDF version contains all topics which have descriptive text, and with this text in hyperlinked form, but without the structural information for these topics. If you have any comments, please send me an email. Thanks.' Below the welcome message is another list of abbreviations, starting with 'AAC' and ending with 'AOL'.

Darstellung von Framesets

- Schachtelung möglich

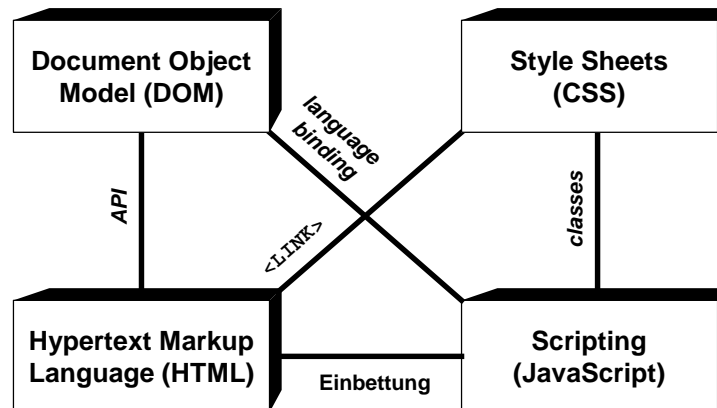
```
<FRAMESET ROWS="33%,33%,33%">
  <FRAMESET COLS="50%,50%">
    <FRAME SRC="" NAME="fr1">
    <FRAME SRC="" NAME="fr2">
  </FRAMESET>
  <FRAME SRC="" NAME="fr3">
  <FRAME SRC="" NAME="fr4">
</FRAMESET>
<NOFRAMES><BODY>
...contents to display in
non-frame-capable user agent...
</BODY></NOFRAMES>
```



Probleme mit Frames

- lange Zeit kein offizielles HTML
 - Aufnahme erst in HTML 4.0
 - nicht in allen Browsern implementiert
- Darstellung zum Teil problematisch
 - unübersichtlich durch Scrollmöglichkeiten
 - Darstellung auf kleineren Bildschirmen
- Probleme mit Bookmarks
 - dynamische Seitenstruktur, kein fester Status
- Probleme mit Search Engines
 - Framesets werden meistens ignoriert
 - Seiten auch ohne Frames zugänglich machen
- Fazit: Frames mit Bedacht verwenden

Dynamic HTML (DHTML)



WWW (SS2001) - HTML

25

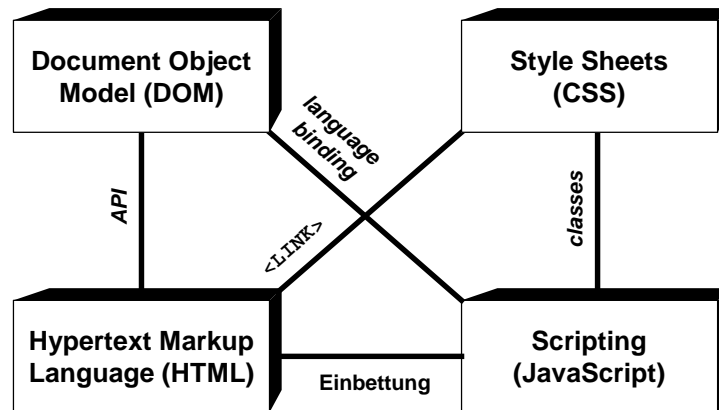
Hypertext Markup Language

- Beschreibung des Inhalts einer Seite
- deklarativer Markup (Gliederung)
- nur statische Konzepte
 - keine Konzepte für wechselnden Inhalt
 - keine Konzepte für Interaktionen (ausser Links)
 - Attribute für Aktionen bei Ereignissen
 - Einbeziehung eines externen Mechanismus'
- Präsentation in Style Sheets spezifiziert
 - Verbindung über Links (<LINK> Element)
 - Einbettung in die Seite (<STYLE> oder STYLE)
- dynamische Komponenten ebenfalls extern

WWW (SS2001) - HTML

26

Dynamic HTML (DHTML)



WWW (SS2001) - HTML

27

Cascading Style Sheets

- bestimmen Präsentation einer Seite
- Entkopplung von Inhalt und Präsentation
 - Inhalt auf der HTML-Seite
 - Präsentation durch ein Style Sheet gesteuert
- Verwendung anwendungsabhängig
 - medienabhängige und alternative Style Sheets
- unterstützen Gruppierung von Elementen
 - Selektoren für die Anwendung von Regeln
- Selektion nach verschiedenen Kriterien
 - Elementname (u.U. kontextuell)
 - Elementklasse (CLASS Attribut)

WWW (SS2001) - HTML

28

Verwendung von CSS Styles

- drei Varianten der CSS Verwendung
 1. als individueller Style eines Elementes
 - Spezifikation mit dem STYLE Attribute
 - `<P STYLE="font-size: small">Text`
 - keine Wiederverwendung von Styles
 2. als eingebettetes Style Sheet
 - Einbettung mit dem `<STYLE>` Element
 - `<HTML><HEAD><STYLE TYPE="text/css">`
 - keine Wiederverwendung von Style Sheets
 3. als referenziertes Style Sheet
 - Referenzierung mit dem `<LINK>` Element
 - `<HTML><HEAD><LINK REL="stylesheet">`
 - beste Trennung zwischen Inhalt und Style Sheet

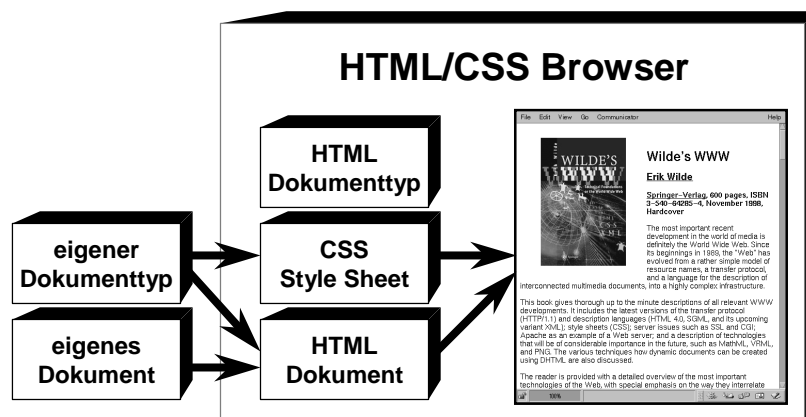
Verbreitung von Style Sheets

- Unterstützung in den grossen Browsern
 - unvollständige Implementierungen
 - keinerlei Unterstützung in älteren Versionen
- Generierung durch neue Applikationen
 - Erzeugung von HTML Seiten
 - Erzeugung von HTML und CSS Dateien

Einfaches CSS Beispiel

```
<HTML><HEAD>
<STYLE TYPE="text/css">
P { font-size: small }
.IMPORTANT { color: red }
</STYLE>
<BODY>
<H1>Beispiel-Seite
<P>normaler Text...
<P CLASS="IMPORTANT">Wichtig!
</BODY></HTML>
```

Publishing mit HTML/CSS



Selektoren (I)

- zur Selektion von Elementen
- Kennzeichnung durch den Namen
 - `H1, H2 { color: red }`
- Kennzeichnung durch Attribute
 - `CLASS` klassifiziert Elemente
 - `.BOLD { font-weight: bold }`
 - `ID` kennzeichnet genau ein Element
 - `#WARN { text-decoration: underline }`
- direkt im Element durch das `STYLE` Attribut
 - `<P STYLE="text-transform: capitalize">caPITAL`
- Kombinationen von Selektoren sind möglich
 - `P.BOLD { font-weight: bold }`

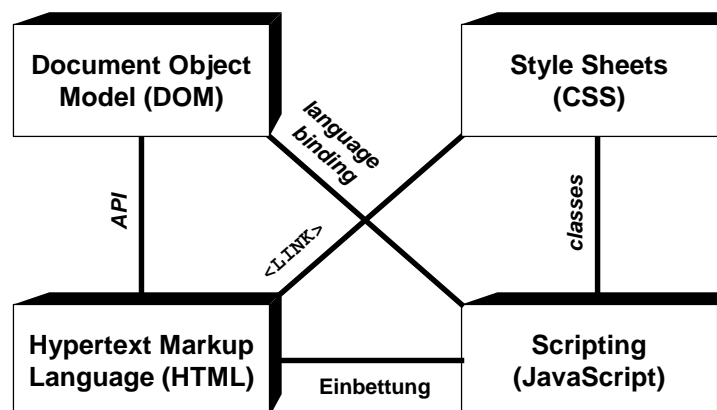
Selektoren (II)

- Contextual Selectors selektieren Kontext
 - `H1 EM { color: red }`
 - `OL { list-style: decimal }`
 - `OL OL { list-style: upper-alpha }`
 - `OL OL OL { list-style : lower-alpha }`
- Pseudo Classes definieren "Klassen"
 - `A:LINK { color: red }`
 - `A:VISITED { color: green }`
- Pseudo Elements definieren "Elemente"
 - `P:FIRST-LINE { text-transform: uppercase }`
 - `P.INITIAL:FIRST-LETTER { font-size: 200% }`

CSS, Level 2 (CSS2)

- unterstützt Medientypen in Style Sheets
 - CSS1 kennt keine Medientypen
- Paged Media als spezieller Medientyp
- Unterstützung für Tabellenformatierung
 - CSS1 kann nur den Inhalt von Tabellen formatieren
- neue Selektions-Mechanismen
 - mehr "Navigation" im Document Tree
- Generierung von Inhalten
- automatische Numerierung
 - Formatierung von Kapitelüberschriften
- Positioning (CSS-P) meistens implementiert
 - ansonsten Unterstützung bisher eher problematisch

Dynamic HTML (DHTML)



Document Object Model (DOM)

- entstanden für portables JavaScript
- Programmierinterface für Dokumente
- unabhängig vom Dokumentenmodell
 - *core* definiert einen strukturellen Kern
 - *HTML* definiert HTML-spezifische Aspekte
 - *XML* erlaubt den Zugriff auf XML-Dokumente
- unabhängig von der Programmiersprache
 - Definition in *Interface Definition Language (IDL)*
 - *language bindings* für diverse Sprachen
 - bisher *ECMAScript* und *Java*

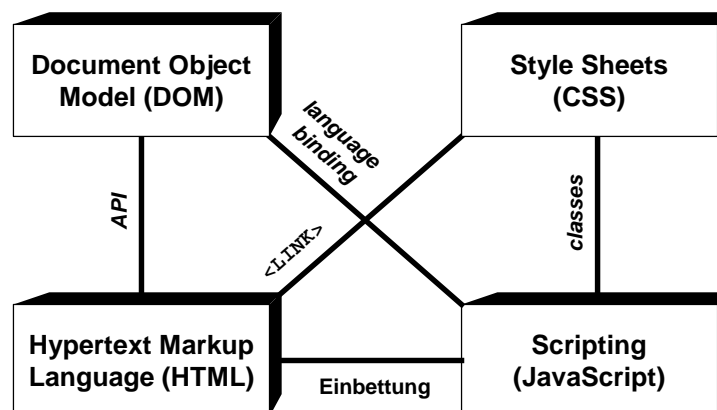
DOM Dokumentenmodell

- das DOM Modell beschreibt
 - logische Struktur eines Dokumentes
 - Zugriff auf Teile eines Dokumentes
 - Manipulation eines Dokumentes
- zur Beschreibung werden verwendet
 - Schnittstellen und Objekte
 - deren Semantik (Verhalten und Attribute)
 - Verbindung von Schnittstellen und Objekten
- *core* definiert allgemeine Eigenschaften
 - 12 Objekttypen (u.U. abgeleitete Typen)
- HTML und XML bilden separate Einheiten
 - Abbildung von HTML Elementen auf Objekte
 - u.U. Methoden (z.B. *InsertCell* in Tabellenzeilen)

DOM ist nicht alles...

- DOM definiert einen Standard für Dokumente
 - Schnittstellen zu Dokumenten
 - Methoden für Abfragen und Manipulationen
- Netscape und Microsoft implementieren mehr
 - Zugriff auf Datenstrukturen des Browsers
 - Zugriff auf die History-Liste
 - Zugriff auf Cookies
 - Dokumentation bei Microsoft
 - Dokumentation bei Netscape
- DHTML Entwicklung in heterogenem Umfeld
 - Berücksichtigung verschiedener Produkte
 - Berücksichtigung verschiedener Versionen

Dynamic HTML (DHTML)



Scripting Languages

- Einbindung in HTML
 - Ausführung beim Laden der Seite (<SCRIPT>)
 - Ausführung beim Auftreten von Ereignissen
- dynamische Seiten sind Browser-abhängig
 - Inhalte nur mit Scripting zugreifbar
 - alternative Seiten anbieten
- *JavaScript* von Netscape
 - eingeführt von Netscape als *LiveScript*
 - Umbenennung in *JavaScript* nach Javas Erfolg
 - Standardisierung als *ECMAScript*
 - *JScript* als Microsoft's ECMAScript Produkt

Einbettung in HTML

- Einbindung in HTML über *Events*
 - beim Laden oder Verlassen einer Seite
 - für bestimmte Benutzeraktionen
 - <SCRIPT> Element an beliebiger Stelle
 - spezielle Attribute: ONMOUSEOVER, ...
- Einbettung direkt als Skript-Element

```
<P>last modification on
<SCRIPT TYPE="text/javascript">
  document.write(document.lastModified);
</SCRIPT>
```

Varianten der Einbettung

- Einbettung mit `<SCRIPT>`-Tag im `<HEAD>`:
`<SCRIPT TYPE="text/javascript"> ... </SCRIPT>`
- Einbettung in HTML-Tags:
 - Event-Aufrufe
`<INPUT TYPE=BUTTON VALUE="press" onClick="if (top.alreadyRegistered('doAlert')) { top.callFunction('string 1', 'string 2'); }">`
 - *javascript*-URLs
`window.close()">The End`
- Einbettung mit dem SRC-Attribut:
`<SCRIPT TYPE="text/javascript" src="MyCode.js">`

JavaScript - Anwendungen

- Unterscheidung in *Client-Side-JavaScript* und *Server-Side-JavaScript*
- *Client-Side-JavaScript*, im Browser ausgeführt
 - aktive Web-Seiten, mittlere Web-Anwendungen
 - Client-basierte Erweiterung von Web-Seiten
 - Verlagerung von Aktivität hin zum Client
 - weniger Serverbelastung, weniger Kommunikations-Overhead
- *Server-Side-JavaScript*, vom Server ausgeführt
 - Ersatz für CGI-Scripts durch in den Server integrierte Script-Sprache (*Microsoft ASP*)

JavaScript - Die Sprache

- Gemeinsamkeiten mit Java
 - ähnliche Syntax
 - ähnliche Kontrollstrukturen
 - automatische Speicherverwaltung
- Unterschiede zu Java
 - interpretiert (auf Server-Seite u.U. vorkompiliert)
 - textuell in Web-Seiten eingebettet (Ggs. Applets)
 - keine (statische) Typenprüfung
 - schwächere Kapselung von Daten/Funktionen
 - kein Multi-Threading

Zusammenfassung

- HTML als Seitenbeschreibungssprache
 - Tabellen für die Layout-Steuerung
 - Clickable Images als Interface-Design
 - HTML Forms als Dateneingabemöglichkeit
- DHTML als Kombination mehrerer Techniken
 - HTML und Einbettungsmechanismen
 - CSS für die Präsentationssteuerung
 - DOM als Schnittstelle für HTML-Dokumente
 - JavaScript als Programmiersprache