



Internet-Technologien

Sommersemester 2002


3. Mai

Ingo Meents
IBM Deutschland Speichersysteme GmbH
Mainz
meents@de.ibm.com

Administratives

- Vortragende
 - Martin Kramer
 - Accenture GmbH
 - e-mail: martin.kramer@accenture.com
 - Ingo Meents
 - IBM Deutschland Speichersysteme GmbH
 - e-mail: meents@de.ibm.com
- Termine
 - Freitags, 14-tägig
 - 13.00 - 17.00 Uhr
- Ort
 - Institut für Informatik
 - Tannenhöhe T1


Übersicht



Termin	Inhalt	Vortragender
3. Mai	Einführung XHTML, JavaScript	Meents
10. Mai	WebDesign Java	Kramer
29. Mai	Web/Application Server Middleware (XML)	Meents
7. Juni	Java 2 Enterprise Edition	Kramer
21. Juni	Workgroup Computing Datenbanken	Meents
5. Juli	e-commerce Ausblick	Kramer

3

Motivation

- 
- Internet ist in vielen Bereichen integraler Bestandteil des täglichen Lebens geworden
 - Entwicklung von Homepages zu Mehrwertdiensten
 - e-Commerce
 - B2C: web shops, Auktionen, Infodienste
 - B2B: Supply chain management
 - Präsentation und Kommunikation
 - Ausnutzen offener Standards in Unternehmensnetzwerken
 - Ersetzen von proprietäre Protokolle und Dokumentformate
 - Plattformunabhängigkeit

4

Ziele

- Kennenlernen der grundlegenden Technologien des Internet
- Entwicklung: von einfachen HTML Seiten zum e-Commerce
- Gestaltung von Web-Seiten
- Statische und dynamische Seiten
- Architektur von Web-Applikationen
 - Client / Server und Multi-Tier Konzepte
 - Realisierungsmöglichkeiten: Scripting, J2EE
- Datenbanken und das Web
- Workgroup Computing
- Anregungen für eigene Experimente

5

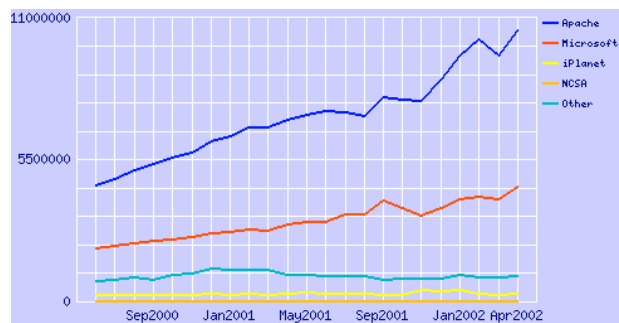
Definitionen

- **Internet**
Öffentliches, dezentral verwaltetes weltweites Netzwerk von Netzwerken auf Basis des Internet Protokolls (IP)
- **Intranet**
Private Anwendung des Internet-Protokolls, von Diensten und Software innerhalb eines Unternehmens, einer Organisation
- **Extranet**
Erweiterte, private Anwendung des Internet-Protokolls auf ein Netzwerk mit geschlossener Benutzergruppe, das z.B. Handelspartnern, Dienstleistern, Produzenten und Kunden erlaubt, auf gemeinsame Geschäftsdaten sowie Anwendungen zuzugreifen

6

Größe des Internets

- Studie April 2002: 37.585.233 Sites
- Aktive Server



Quelle: Netcraft: <http://www.netcraft.com/survey>

7

Zielsetzungen / Erwartungen von Unternehmen

- Internet
 - Präsentation
 - E-Commerce: Verkauf / Handel / Dienstleistung
- Intranet
 - Nutzen von Standards, freier und fertiger Software, wie z.B. Middleware, Browser, etc.
 - Einsatz gleicher Technik für alle Arten der Kommunikation, Sachbearbeiter, Aussendienst, Call-Center, Automaten, Internet
 - gezielte, aktuelle, schnelle, einfache Verteilung von Information an Mitarbeiter
 - Kostensenkung durch Vereinheitlichung, z.B. durch Trennung von Applikationslogik und GUIs (Multi-Tier-Applikationen)
- Extranet
 - vereinfachte Zusammenarbeit mit Handelspartnern
 - "Outsourcing" von Aufgaben an spezialisierte Dienstleister


8

Anwendungen im WWW

- 
- Homepage
 - Portale
 - e-Business
 - B2C (Shops, Auktionen)
 - B2B (Marktplätze, Ausschreibungen, eProcurement)
 - e-Government
 - z.B. <http://www.vorteil.niedersachsen.de/e-government.htm>
 - Mobile
 - WAP
 - Location Based Services
 - Legacy Applications
 - EAI (Enterprise Application Integration)
 - Web GUIs

9

Ressourcen

- 
- *Internet & World Wide Web - How to program*, H.M. Deitel, P.J. Deitel, T.R. Nieto, 2002, Prentice Hall
 - *Internet Intern - Technik und Programmierung*, M. Tischer, B. Jennrich, 1997, Data Becker
 - *TCP/IP - Aufbau und Betrieb eines TCP/IP Netzes*, K. Washburn, J. Evans, 1994, Addison Wesley
 - W3C Consortium: www.w3.org
 - *Wilde's WWW - Technical Foundations of the WWW*, Erik Wilde, Springer, 1998, 1999, 2002
 - Das Internet selbst: www.google.com :-)

10

Geschichte des Internet

- Ende der 60'er: ARPAnet (Advanced Research Projects Agency of the Department of defense)
 - Netzwerk von Rechnern mit 56 Kbits zum Nutzen von Rechner-Ressourcen
 - zusätzlich: e-mail zur schnellen Kommunikation
 - Ziele
 - bidirektionale, asynchrone Kommunikation, mehrer Nutzer
 - dezentrale Verwaltung
 - Ausfallsicherheit dynamisches Routing
 - Technologie: Paket-Switching TCP (Transmission Control Protocol)
- Gleichzeitig entstanden viele Intranets mit unterschiedlicher Hard- und Software
- Verbindung durch IP (Intranet Protocol, ARPA): Netzwerk von Netzwerken -> TCP/IP
- erst Forschungseinrichtungen, dann Militär, dann Freigabe für kommerzielle Zwecke

11

Einordnung ins Schichtenmodell

ISO/OSI

TCP/IP

7: Anwendung	Hypertext Transfer	File-Transfer	eMail	Terminal	DN Service	NetzwerkMM	NFS
6: Darstellung	HTTP	FTP	SMTP	rlogin	DNS	SNMP	XDR...
5: Steuerung	SSL / TLS						
4: Transport	TCP			UDP			
3: Vermittlung	IP, ICMP, ARP, RARP						
2: Sicherung	Ethernet, ATM, DECnet, TR						
1: Übertragung	Netzwerkhardware						

12

Weitere Protokolle / Anwendungen

- HTTP Hypertext
- FTP Dateittransfer
- SMTP, POP3, IMAP4 e-Mail
- NNTP Diskussionforen (News)
- IRC Chatting
- Netzwerk-Dateisysteme
 - Network File System NFS
 - Server Message Block SMB
 - Common Internet File System CIFS (Microsoft)
 - CODA (Carnegie Mellon University, <http://www.coda.cs.cmu.edu>)
- ODBC, JDBC, LDAP, X.500 Datenbankzugriff
- RPC, CORBA, DCOM, RMI verteilte Anwendungen

13

Grundelemente des WWW

- Referenzen
 - URI / URL Universal Ressource Identifier/Locator
 - Adressierung
- Protokoll zwischen Client und Server
 - HTTP Hypertext Transfer Protocol
 - Transport
- Auszeichnungssprache für Web-Seiten
 - XHTML Hypertext Markup Language
 - Darstellung

14

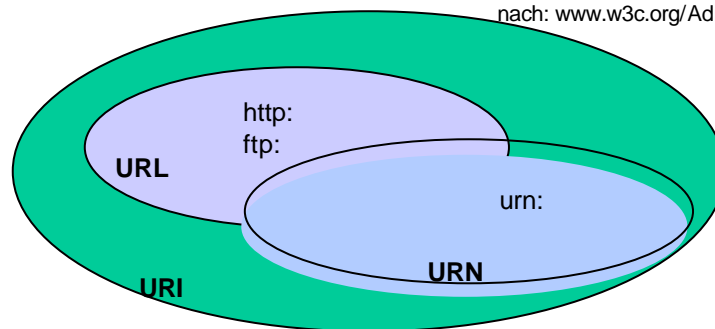
Universal Resource Identifier

- einfaches, erweiterbares Mittel, zur Identifikation von Ressourcen
- uniform: Nutzen verschiedener Identifier-Typen im selben Kontext
- resource: elektronisches Dokument, Bild, Script, etc
- identifier: Referenz, kurze Zeichenkette
- definiert in RFC 2396
- uri = <scheme> : <scheme-specific-part>
- <scheme>
 - legt Schema fest (z.B. **http, https, mailto, ftp, news**)
- <scheme-specific-part>
 - allgemeine Syntax
 - Scheme-spezifische Bedeutung

15

URI, URL und URN

nach: www.w3c.org/Adressing



- URI (uniform resource identifier)
generische Strings, die Ressourcen bezeichnen
- URL (uniform resource locator)
informell: bekannte Web-Adressen zu Schemes wie http, ftp, mailto
- URN (uniform resource name)
persistente, lokations-unabhängige Identifier

16

Definitions

- URI: Uniform Resource Identifier
The generic set of all names/addresses that are short strings that refer to resources.
- URL: Uniform Resource Locator
An informal term (no longer used in technical specifications) associated with popular URI schemes: http, ftp, mailto.
- URN: Uniform Resource Name
 1. An URI has an institutional commitment to persistence, availability, etc. Note that this sort of URI may also be a URL.
 2. A particular name, urn; specified by RFC 2141 and related documents, intended to serve as persistent, location independent, resource identifiers.

(www.w3c.org/Adressing)

17

Syntax von URI (1)

- Hierarchie
 - Trennung durch /
 - z.B. `http://www.math.tu-clausthal.de/intsim`
- Relative URI
 - Basis
 - URL der aktuellen HTML Seite
 - explizite Angabe in aktueller HTML Seite `<BASE>`
 - Angabe eines Teil-Pfades
 - z.B. `graphics/logo.gif`
 - übergeordnetes Verzeichnis durch „..“
 - z.B. `../graphics/banner.html`

18

Syntax von URI (2)

- Sonderzeichen (RFC 1738):
 - Grundlage: ANSI 8859-1 Zeichensatz
 - reservierte Zeichen für URI sind :
? % ; / <space> <CR> <LF> <TAB> Umlaute
 - Escape-Zeichen: % <hex-ziffer> <hex-ziffer>
 - z.B. ? ? %3F = ? %3D
 % ? %25 ; ? %3B etc.
- Fragmente:
 - Verweise auf Anker in der HTML-Seite
 - Definition:
 - <p>Hobbies
 - <p id="Hobbies">Hobbies:
 - Referenz
 - <http://www.my-server.de/index.html#hobbies>
 - beliebige Elemente können identifiziert werden
 - Identifizierung durch Client (Übermittlung kompletter Resource)

19

Syntax von URI (3)

- Parameter (*Query strings*)
 - Parameterübergabe vor allem für (CGI) Scripte
 - method = „get“
 - alternativ: method=„post“ Übergabe über Umgebungsvariable
 - Einleitung durch Fragezeichen
 - Verkettung durch &
 - Leerzeichen durch +
 - URL: ...?name1=wert1&name2=wert2...
 - z.B. <http://localhost/search.pl?Name=Hans+Meier>
 - Vorteil: Parameter Teil der URL, Bookmark
 - Nachteil:
 - Übergebene Parameter sichtbar, Passwörter in History
 - beschränkte Länge

20

URL

- Zweck: Adressierung einer Ressource im Internet
- Verwendung in HTML-Seiten u. a. für
 - Links
 - Ziel für das Versenden von Formular-Daten
 - Einbinden von Bildern
- Syntax:
`// [<user> [: <password>] @] <host> [: <port>] / <path>`
 - obligatorisch: Hostname, Pfad
 - optional: User, Passwort, Port
- z.B. `http://meents:mypassword@www.server.de/index.html`
- größtes Problem: ungültige URL durch Änderungen

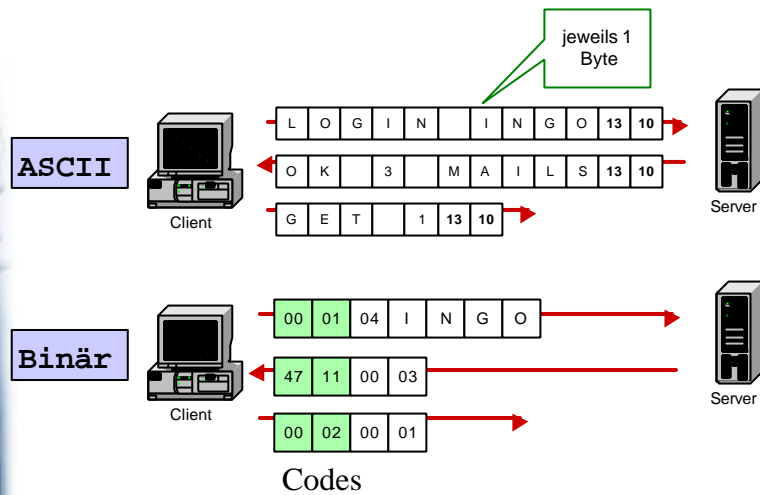
21

HTTP zwischen Client und Server

- Server
 - Verteilen der Informationen
 - Problem: Performance
 - Produkte:
 - Apache Web Server
 - Microsoft IIS
- Client
 - Anzeige der Daten Text und Grafik
 - Problem: korrekte Darstellung bei sich rasch änderndem HTML und proprietären Erweiterungen
 - Produkte:
 - Netscape Navigator
 - Microsoft Internet Explorer
 - Opera
 - Lynx

22

ASCII vs. Binär-Protokolle (1)



23

ASCII vs. Binär Protokolle (2)

- Binär
 - feste Strukturen, Kommando-Codes
 - schwieriger zu implementieren
 - kürzer
 - Problem: Byteorder
- ASCII
 - variable Strukturen, Klartext-Kommandos
 - leichter zu implementieren und zu testen
 - länger
 - Zeichensätze
- Binärprotokolle vorwiegend auf unteren Ebenen (Paket- und Transportschicht), z.B. TCP, UDP
- ASCII Protokolle auf Anwendungsschicht, z.B. FTP, Telnet, HTTP

24

HTTP Protokoll

- Übertragungsprotokoll der Anwendungsschicht für das Web
- ASCII Protokoll auf gesicherter TCP/IP Verbindung
- Standard TCP Port 80
- Statuslos, Erweiterung Cookies: RFC 2109
- nachrichtenorientiert, zwei Nachrichtentypen:
 - Request
 - Response
- Versionen: erste Versionen seit anfang der 90'er Jahre
 - 1996: HTTP 0.9, RFC 1945
 - auch HTTP 1.0
 - 1997: HTTP 1.1, RFC 2068
 - 1999: update RFC 2616
 - HTTP Authentication RFC 2617
- <http://www.w3.org/Protocols/>

25

Ablauf einer Kommunikation über HTTP

- Client baut Verbindung auf
 - Ermittlung der IP Adresse mittels DNS
 - zum HTTP Server normalerweise über TCP port 80
- Client schickt Request
 - Auswahl einer Methode
 - Parameter-Übergabe
- Server erhält Request, bearbeitet ihn
- Server schickt Response (sendet Dokument)
 - Statuscode, z.B. HTTP/1.1 200 OK, HTTP/1.1 404 Not found
 - Resultat
- Server schliesst die Verbindung wieder
 - neuere Versionen (ab HTTP/1.1) können die Verbindung offen lassen

26

Request Nachricht

- request = <methode> <uri> **HTTP**/**<major>**.<minor>
<header>
- <entity>
- Methoden:
 - GET: Anfordern eines Dokuments
 - HEAD: Anfordern von Meta-Daten
 - POST: Senden von Daten
 - weitere: PUT, DELETE, LINK, UNLINK
- <uri> Ziel-URL
- <header> Informationen (z. B. Absender, akzeptierte Inhalte)
- <entity> ggf. Nachrichtenrumpf mit Daten
- Komptibilität: gleiche major Nummern

27

Response Nachricht

- **HTTP**/**<major>**.<minor> <state> <description>
<header>
- <entity>
- <state> Statusinformationen
 - z.B. 200 OK, 201 OK nach POST, 400 Syntaxfehler, 404 Not found
- header: Informationen (z. B. MIME-Type, Länge und Gültigkeitsdauer der Daten)
- entity: ggf. Nachrichtenrumpf mit Daten

28

Kommunikation: Beispiel 1 - OK

Client:

```
> telnet localhost 8080
GET /index.html HTTP/1.1
Host: localhost
Accept-Language: en, de
```

Server:

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 6827
Date: Thu, 02 May 2002 09:07:05 GMT
Server: Apache Tomcat/4.0.3 (HTTP/1.1 Connector)
Last-Modified: Fri, 01 Mar 2002 12:48:10 GMT
ETag: "6827-1014986890000"

<!doctype html public ...
```

29

Kommunikation: Beispiel 2 - Fehler 404

Client:

```
> telnet localhost 8080
GET /hallo.html HTTP/1.1
Host: localhost
Accept-Language: en, de
```

Server:

```
HTTP/1.1 404 /hallo.html
Date: Thu, 02 May 2002 09:16:50 GMT
Server: Apache Tomcat/4.0.3 (HTTP/1.1 Connector)
Connection: close

<html><head><title>Apache Tomcat/4.0.3 - Error
report</title> ...
```

30

Auszeichnungssprachen

- Auszeichnung (Markup)
 - Metainformationen, die die Struktur eines Textes beschreiben
 - z.B:

```
<Lehrveranstaltung Dauer="2V">  
    <Titel>Internet Technologie</Titel>  
    <Vortragender>Ingo Meents</Vortragender>  
</Lehrveranstaltung>
```
- Sprachen
 - SGML Standard Generalized Markup Language
 - XML Extensible Markup Language
 - HTML Hypertext Markup Language
 - XHTML Extensible Hypertext Markup Language
 - HTML Hypertext Markup Language
- Ziele:
 - Maschinelle Verarbeitung von Dokumenten
 - Trennung von Struktur und Inhalt

31

Standard Generalized Markup Language SGML

- Entwicklung frühe 80'er Jahre
- seit 1986 ISO Standard für Markup in elektronischen Dokumenten
- Markup-Language spezifiziert:
 - welches Markup ist erlaubt
 - welches Markup ist erforderlich
 - wie ist das Markup vom Text zu unterscheiden
- Beschreibung korrekter Dokumente durch eine Grammatik, die document type definition (DTD)
- vornehmlich für technische Dokumentation
- zu komplex, daher kaum kommerzieller Erfolg

32

Extensible Markup Language XML

- Methode, um strukturierte Informationen in einer (Text)-Datei darzustellen
 - Nachteil: hoher Platzbedarf
 - Vorteile beim Debugging
 - große Platten verfügbar
 - Komprimierung
- Untermenge von SGML
 - geregelter, einfacher, genauso mächtig
- Freidefinierbare Tags und Attribute
- XML ist lizenzfrei (Entwicklung durch W3C), plattformunabhängig und gut unterstützt.
- mehr dazu später ...

33

Hypertext Markup Language HTML

- **die** Sprache des WWW
- Verbreitung für alle Arten von elektronischen Dokumenten, z.B. on-line Hilfen, e-mails, etc.
- Probleme
 - proprietäre Erweiterungen
 - keine durchgängige Trennung von Inhalt und Struktur
 - fest definierte Bedeutung von Tags
 - viele fehlerhafte (not well-formed) Dokumente, da Browser sehr fehlertolerant
- Versionen
 - HTML 2.0, 3.2, 4.0, 4.1
 - alle nun missbilligt (deprecated)

34

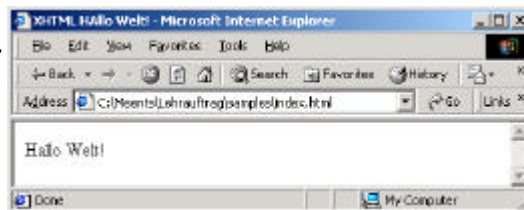
Extensible Hypertext Markup Language XHTML

- Referenz: www.w3.org/markup
- Reformulierung von HTML in XML 1.0
 - 1.0 Strict
 - keine Layout Tags mehr
 - reines XML
 - Transitional
 - Übergangslösung für Rückwärtskompatibilität für Browser, die keine Stylesheets verarbeiten können
 - Frameset
 - für Web Seiten mit Frames
- Layout über Style Sheets
- W3C XHTML validation service: validator.w3.org
- Referenz: www.webreference.com/xml/reference/xhtml.html

35

Aufbau einer XHTML Seite

- Dokument `<html>`
 - Kopf `<Head>`
 - Body `<Body>`
- Beispiel:



```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>XHTML Hallo Welt!</title>
    <meta name="keywords" content="Test Welt Hallo" />
  </head>
  <body>
    <p>Hallo Welt!</p>
  </body>
</html>
```

36

XHTML - Elemente im Dokument-Kopf

- Dokumententitel <title>
- Basispfad für relative URL <base>
- Links <link>
- Style Sheets <style>
- Skriptsprachen <script>
- Meta-Information <meta> mit Attributen name und content
 - Schlüsselworte
 - Beschreibung
 - für Suchmaschinen

37

XHTML - Elemente im Dokument-Rumpf

- Überschriften <h1> <h2> .. <h6>
- Absätze (normaler Fliesstext) <p>
- Formatierungen
- Listen
- Hyperlinks <a>
- leere Elemente
 - horizontale Linien <hr />
 - Absatz

 - Bilder (jpg, gif, png,...)
- Tabellen
- Applets <applet>
- Formulare <form>
- Kommentare <!-- -->
- Image-Maps
- Framesets

38

XHTML - Beispiel: Formulare

- Formular definiert durch <FORM> Element
 - Attribut action="<url>" bestimmt verarbeitendes Script
 - Attribut method="GET" oder "POST" bestimmt Parameterübergabe
- Auswertung mittels CGI (Common Gateway Interface)
- Eingabe-Elemente
 - Eingabefelder <INPUT>
 - Attribut type = text | radio | checkbox | hidden | password | image | reset | submit
 - Drop-Down-Menüs <SELECT>
 - Mehrzeiliges Eingabefeld <TEXTAREA>

39

XHTML - Beispiel: Formular / Tabelle

The screenshot shows a web browser window with the title "XHTML Formular Beispiel - Microsoft Internet Explorer". The address bar shows "http://localhost:8080/Form.html". The page content includes a heading "Anmeldung" and a sub-heading "Bitte füllen Sie das Formular aus:". Below this is a table with two columns: "Frage" and "Antwort". The table contains three rows: "Name" with the value "Ingo Meents", "Alter" with radio buttons for "kleiner als 10", "11-20", and "größer als 20", and "Fahrzeug" with a dropdown menu showing "Opel". At the bottom of the form are two buttons: "Abschicken" and "Löschen".

```
<body>
<h1>Anmeldung</h1>
<p>Bitte füllen
  Sie das
  Formular aus: </p>
...
</body>
```

http://localhost:8080/evalform?name=Ingo+Meents&alter=le_10&auto=Opel

.... zur Verarbeitung später mehr

40

XHTML - Beispiel: Quelltext Formular

```
<form method = "GET" action = "evalform">
<table border=1>
<caption> Formular-&Uuml;berschrift </caption>
<thead> <tr> <th> Frage </th> <th> Antwort </th>
</thead>
<tbody>
<tr>
<td> Name </td>
<td> <input name = "name" type = "text" size = "20" /> </td> </tr>
<tr>
<td> Alter </td>
<td>
<label> kleiner als 10 </label>
<input name = "alter" type = "radio" value = "le_10" checked = "checked" />
<label> 11-20 </label>
<input name = "alter" type = "radio" value = "11to20" />
<label> gr&ouml;Ber als 20 </label>
<input name = "alter" type = "radio" value = "ge_20" /> </td>
<tr>
<td> Fahrzeug </td>
<td>
<select name = "auto">
<option selected = "selected">Opel</option>
<option>VW</option>
<option>Porsche</option>
<option>Audi</option> </select> </td>
</tbody>
</table>
<input type = "submit" value = "Abschicken" />
<input type = "reset" value = "R&uuml;ckschicken" />
</form>
```

41

Style Sheets

- DSSSL (Document Style Semantics and Specification Language)
 - Layout-Sprache für SGML
 - Umformung des Dokuments durch Regeln
 - internationaler Standard seit 1996
- XSL (Extensible Style Language)
 - "Kreuzung" zwischen CSS und vereinfachten DSSSL
 - seit 8/98 W3C "Working Draft"
- CSS 1+2 (Cascading Stylesheets)
 - Festlegung des Layouts von bestimmten Tags oder von durch Attribute gekennzeichneten Tags
 - W3C Empfehlung
 - CSS 1 seit 12/96
 - CSS2 seit 5/98

42

Cascading Stylesheets

- Zentrale Definition des Layouts einer Webseite
 - Schriftart und -Größe
 - Farben für Schrift und Hintergrund
 - Ränder, Padding
 - Positionierung von Seitenelementen
 - Größe von Seitenelementen
 - Textfluss
- Einheitliches Aussehen aller Seiten einer Site
- Trennung von Darstellung und Inhalt
 - Inhalt in XHTML/XML Datei
 - Layout-Definitionen in Style Sheet
- Cascading: verschiedene Ebenen der Definition
 - Autoren Styles überschreiben User-Styles, diese überschreiben Web-Browser Styles
 - Styles von Elementen werden vererbt
- Medienabhängig: Browser, Druck, Mobile

43

Cascading Stylesheets - Typen

- Inline Styles
 - definiert direkt im Element
 - z.B. `<p style = "font-size: 20pt; color: #0000ff">`
 - keine Wiederverwendung
 - überschreiben alle anderen Definitionen
- Embedded Style Sheets
 - Einbettung im Kopf eines XHTML Dokumentes
 - gilt für ganzes Dokument
 - partielle Wiederverwendung
- External Style Sheets
 - Wiederverwendung für ganze Site möglich
 - Einbinden des externen style sheets mittels:

```
<head>
  <link rel="stylesheet" type="text/css"
        href="mystyles.css" />
  ...
</head>
```

44

Syntax

- `<property-name>: <value>; <property-name>: <value>; ...`
- Beispiele:
 - `color, background-color: white, green, ...`
 - `font-family:`
 - Special Font: arial, verdana,
 - Generischer Font: sans-serif, serif, cursive, fantasy, monospace,...
 - `font-size:`
 - absolut: 12pt
 - relativ: xx-small, x-small, small, smaller, medium, large, x-large, ..
- Definition von Klassen: `.<classname>`
- Validation service: jigsaw.w3.org/css-validator

45

Selektion von Elementen

- Kennzeichnung durch Element-Namen
 - `h1, h2, p { color: red }`
- Definition von Klassen über Attribut
 - `.myclass {color green}`
 - `<p class = "myclass" >`
- style Attribut im Element
 - `<p style="text-decoration: underline">`

46

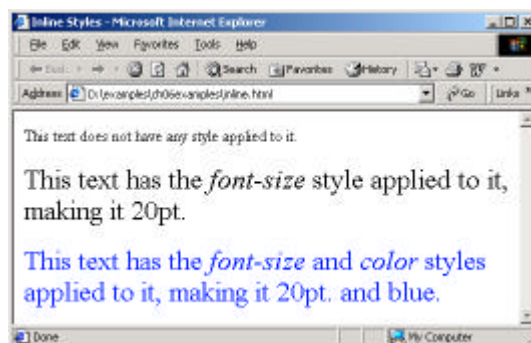
Selektion von Elementen (Fortsetzung)

- Kontextabhängige Selektion
 - li em { color: red; font-weight: bold}
 - ul { margin-left: 75px}
 - ul ul { text-decoration: underline; margin-left: 15px }
- Pseudoklassen
 - a:hover { text-decoration: underline; color:red}
 - a:link { color: green }
 - a:visited { color: red }
- Pseudoelemente
 - p:first-line { text-transform: uppercase }

47

Beispiel: Inline Style Sheet

```
<p>This text does not have any style applied to it.</p>  
  
<p style = "font-size: 20pt">This text has the  
<em>font-size</em> style applied to it, making it 20pt.  
</p>  
  
<p style = "font-size: 20pt; color: #0000ff">  
This text has the <em>font-size</em> and  
<em>color</em> styles applied to it, making it  
20pt. and blue.</p>
```



48

Beispiel: Embedded Style Sheet

```
<head>
<title>Style Sheets</title>
<style type = "text/css">

    em      { background-color: #8000ff;
              color: white }
    h1      { font-family: arial, sans-serif }
    p       { font-size: 14pt }
    .special { color: green; font-size: 20pt }
</style>
</head>
<body>
<h1 class = "special">Dies ist eine Überschrift</h1>
<p>Dieses ist der erste Abschnitt.</p>

<h1>Noch eine Überschrift</h1>
<p class = "special"> Dieses ist der zweite Abschnitt.
<em>Dies hier ist ganz wichtig!</em> Hier ist es wieder
weniger wichtig.</p>
</body>
```



49

Scripting

- 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
- Prozedurale Programmiersprache
 - interpretiert
 - übliche Daten- und Kontrollstrukturen (vgl. Java, C)
 - schwächere Kapselung von Daten/Funktionen als bei Java
 - automatische Speicherverwaltung
 - Einbettung in XHTML-Seiten
 - keine (statische) Typenprüfung
 - kein Multi-Threading

50

Ursprünge / Ressourcen

- JavaScript von Netscape
- eingeführt von Netscape als LiveScript
- Umbenennung in JavaScript nach Javas Erfolg
- Standardisierung als ECMAScript
 - European Computer Manufacturer's Association
 - Standard-Dokument
www.ecma.ch/ecma1/stand/ecma-262.htm
- JScript als Microsoft's ECMAScript Produkt
 - msdn.microsoft.com/scripting/default.htm
- Tutorials
 - www.webteacher.com/javascript
 - www.webdeveloper.com/javascript

51

Einbettung in XHTML

- Einbindung in HTML über *Events*
 - Dokument-Events (onload, onunload, onscroll, onbeforeprint, ...)
 - Mouse-Events (ondrag, ondblclick, onmouseover, ...)
 - Clipboard-Events (onbeforecut, oncopy, oncut, ...)
 - Keyboard-events (onhelp, onkeydown, onkeyup, ...)
- Einbettung direkt als Skript-Element

```
<p>Datum der letzten Änderung
<script type="text/javascript">
  document.write(document.lastModified);
</script>
</p>
```

52

Einbettung in XHTML (2)

- Einbettung als `<script>` Element im `<head>`
 - `<script type="text/javascript"> ... </script>`
- Einbettung in XHTML Tags:
 - Event-Aufrufe
`<input type="button" value="Click me" onClick="window.alert('button clicked')"`
- JavaScript-URLs
`Ende`
- Einbettung mit dem SRC-Attribut
 - `<script type="text/javascript" src="MyCode.js">`

53

Beispiel: JavaScript mit Rekursion

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
<title>Recursive Factorial Function</title>

<script type = "text/javascript">
<!--
document.writeln( "<h1>Factorials of 1 to 10</h1>" );
document.writeln(
" <table border = '1' width = '100%'>" );

for ( var i = 0; i <= 10; i++ )
document.writeln( "<tr><td>" + i + "!</td><td>" +
factorial( i ) + "</td></tr>" );

document.writeln( "</table>" );

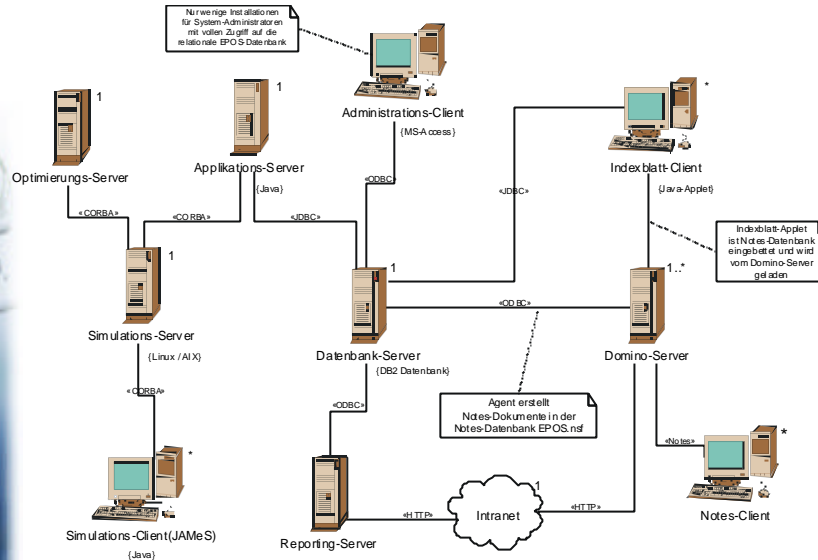
// Recursive definition of function factorial
function factorial( number )
{
if ( number <= 1 ) // base case
return 1;
else
return number * factorial( number - 1 );
}
// -->
</script>
</head><body></body>
</html>
```

0!	1
1!	1
2!	2
3!	6
4!	24
5!	120
6!	720
7!	5040
8!	40320
9!	362880
10!	3628800

aus: Deitel et al.

54

Anwendungsbeispiel: EPOS



55

Ende - Aber es geht weiter ...

- am 10.5.2002 mit Martin Kramer und
 - Web Design
 - Java
- am 24.5.2002 mit Ingo Meents und
 - Web Servern
 - Middleware



56