Internettechnologie 1

5 ECTS
mittwochs 10 – 12 Uhr
Raum S308
Dipl.-Phys. T. Severiens

Terminplan#

Datum	Inhalt
4. März	Organisatorisches, HTML
11. März	HTML (Vertiefung)
18. März	HTML (Formulare)
25. März	JavaScript
1. April	**Ideenpräsentation**
8. April	CSS
15. April	CSS
22. April	SSI (Server)

Datum	Inhalt
29. April	PHP vs. CGI
6. Mai	Multimedia
13. Mai	Krypto, XML
20. Mai	XML (DTD)
27. Mai	XML XSL(T)
3. Juni	**Präsentation***
10. Juni	**Präsentation***

#) vorbehaltlich Abweichungen
*) Reihenfolge wird am 1.4. ausgelost

Aufteilung der Übungsgruppen

Datum	Gruppe A	Gruppe B
4.3.	12 Uhr	14 Uhr
11.3.	14 Uhr	12 Uhr
18.3.	12 Uhr	14 Uhr
25.3.	14 Uhr	12 Uhr
1.4.	12 Uhr	14 Uhr
8.4.	14 Uhr	12 Uhr
15.4.	12 Uhr	14 Uhr
22.4.	14 Uhr	12 Uhr
29.4.	12 Uhr	14 Uhr
6.5.	14 Uhr	12 Uhr
13.5.	12 Uhr	14 Uhr
20.5.	14 Uhr	12 Uhr
27.5.	12 Uhr	14 Uhr
3.6.	14 Uhr	12 Uhr
10.6.	12 Uhr	14 Uhr

TeilnehmerInnen Gruppe A

Bellan
Borchardt
Böse
Dierkes
Drees
Dühnen
Fiola
Frers
Gawal
Gerdes
Gerhards
Gesken
Grosser

Hettwer
Heuer
Ivanov
Jacobs
Janßen
Kaiser
Kapp
Lotz
Lübben
Möller
Osmanoska
Plate
Puls

Rog
Rohlmann
Saalfeld
Schirrmann
Schröder
Schultze
Snater
Stasch
Striepens
Tietze
Trautenberg
Vogt
Wedekind
Zentner

TeilnehmerInnen Gruppe B

Brassat
Döscher
Eilers
Eilts
Eitzert
Franke
Groß
Gülker
Herrling
Heuer
Heykes
Hölzen
Israel

Klapetz
Klus
Köhler
Lang
Lange
Liggig
Liss
Lühn
Niemann
Reinert
Riede
Schwarzer

Siehsmeger
Strasser
Stuhaan

Prüfungsleistung

- Es gibt keine Klausur
- aber…

Prüfungsleistung: Generelles

- Die Zulassung zur Prüfung erfolgt durch die **Vorstellung der Ideen** und die **Präsentation.**
- Die Prüfungsleistung wird als **Kursarbeit** erbracht. Das bedeutet, Sie schreiben keine Klausur, sondern Kodieren eine Website in HTML, CSS und Javascript. Zusätzlich schreiben Sie eine XML-Datei sowie eine dazugehörende DTD. Die Website sowie die XML und DTD - Datei sind von Ihnen zu **dokumentieren.**
- Die erstellte Website und die XML- und DTD-Dateien, sowie die Dokumentation geben Sie bitte in elektronischer Form ab.
  In elektronischer Form heißt, Sie senden eine E-Mail. Bitte verwenden Sie ein komprimiertes Format wie ZIP oder RAR.
  Der Eingang der elektronischen Form wird von mir bestätigt.
  Einer aus der Gruppe sendet die E-Mail, alle anderen stehen im CC dieser E-Mail.

Prüfungsleistung: Inhalt

- Das Programm soll eine Internetdarstellung eines Unternehmens oder einer vergleichbaren Institution sein:
  - mindestens drei Seiten, z.B. Hauptseite, Produktkatalog, Anfahrtsbeschreibung, Wir über uns, Kontaktformular etc. Dabei berücksichtigen Sie die im Bewertungsplan angegebenen Kriterien.
  - In die Seiten ist mindestens
    - eine Grafik und ein anderes nicht textbezogenes Element (Java-Applet oder Sound oder Video etc.)

    einzubinden.
  - Eine der HTML-Seiten ist ein Formular, welches zur Bestellung eines Produktes dient. Darin sind mindestens die im Bewertungsplan angegebenen Elemente enthalten.
  - Der Absendebutton muss keine echte Funktionalität aufweisen!
  - Eine Überprüfung der Formularinhalte ist hingegen notwendig. Hier sollen Sie mindestens zwei Formularinhalte auf die eingegebenen Werte überprüfen. Dies können z.B. sein:
    - PLZ fünfstellig,
    - Telefonnummervorwahl mit einer führenden Null,
    - E-Mail-Adresse über @-Symbol verifizierbar etc.
  - Hinzu kommt eine Seite, auf der Sie ein Impressum unterbringen, damit Ihre Seite auch rechtskonform ist.

Inhalt der XML- und der DTD-Datei

- Sie kodieren eine XML- Struktur für eine CD*-Sammlung. Es reichen drei CDs. Dabei ist zu beachten, dass sowohl das Album selbst, als auch die darauf enthaltenen Titel mit beschreibenden Daten erfasst werden. Diese sind mindestens:
  - die Titel der Alben
  - das Genre
  - das Erscheinungsjahr des Albums
  - die einzelnen Titel
  - der Interpret des Titels
  - der Autor des Titels
- Die XML-Datei muss jederzeit um weitere Alben erweiterbar sein.
- Die DTD beschreibt die von Ihnen gewählte XML - Struktur, so dass Ihre XML-Datei gültig ist.

*) Natürlich dürfen Sie auch eine MP3-Sammlung etc. beschreiben.

Inhalt der Dokumentation

- Die Dokumentation liefert eine Erläuterung zu der von Ihnen erstellten Website und der XML/DTD-Struktur. Darin ist enthalten,
  - welche Anweisungen des HTML-Codes welche Aktionen bewirken.
  - welche Anweisungen in anderen Programmteilen (Java-Script, CSS, Java etc.) welche Aktionen bewirken
  - ein Schaubild, das anschaulich erläutert, wie die einzelnen Programmteile (HTML-Dokumente, CSS-Dateien, Java-Skript-Dateien, Java-Applets etc.) zusammenspielen.

  - Welche XML Elemente Sie für welche Informationen gewählt haben.
  - die Definitionen in der DTD (Vorkommen, Beziehung zu anderen Elementen etc.)
- Es ist Ihnen frei gestellt, die Dokumentation innerhalb des Programms durch Kommentare zu verwirklichen, oder als beigelegte Programmdokumentation abzugeben. Für ein Schaubild, wie die einzelnen Programme untereinander verknüpft sind, ist eine separate Dokumentation unabdingbar.

Präsentation und Abgabe der Arbeit

- Die Arbeit ist bis spätestens 17. Juni 2015, 23:59 Uhr CEST abzugeben.
- Sollten sich aus der Abgabe Fragen ergeben, behalte ich mir vor, die Ersteller zu einem Präsentationskolloquium einzuladen. Dann werden Fragen über von Ihnen erstellten Dateien und Dokumentation gestellt.

Bewertungsplan für die Prüfungsleistung Internettechnologie I (SoSe 2015)

Bewertungselemente HTML	Punkte
Grundlegende Struktur *	
Seite 1 (z.B. Startseite)	
Seite 2 (z.B. weiterführende Seite)	
Formularseite	
Seite Impressum / Disclaimer	
Fehlerfreiheit	
Inhalte Seiten*	
Liste	
Verweise (intern, extern)	
Bilder	
Element (z.B. Java, Multi-Media,…)	
Tabelle min. 2xS,3xZ	
Formularinhalte*	
Grundsätzliche <form> Funktionalität	
Checkbox	
Radiobutton	
Auswahlliste	
Textfeld	
Textarea	
Button Absenden / Rücksetzen	
Cascading Stylesheets*	
Externe CSS-Datei eingebunden	
Min. 2 (div-)Tags oder HTML5 für Navigation/Content	
Farb-und Hintergrundformatierung	
Schrift- und Textformatierung	
Überprüfung der Formularinhalte*	
Überprüfung 1	
Überprüfung 2	
Rechtliche Aspekte **	
Disclaimer / Impressum	
Summe:	

Bewertungselemente Doku	Punkte
Grundlegende Struktur *	
Zusammenhang d. Elemente/Dateien	
Dokumentation Seiteninhalte	
Grundsätzliche Seitenstruktur	
Liste	
Bilder	
Element	
Tabelle	
Dokumentation Formularinhalte	
Grundsätzliche <form> Funktionalität	
Checkbox	
Radiobutton	
Auswahlliste	
Textfeld	
Textarea	
Button Absenden / Rücksetzen	
Cascading Stylesheets	
Externe CSS-Datei eingebunden	
Min. 2 (div-)Tags für Navigation/Content	
Farb-, Schrift, Text- und Hintergrund-formatierung	
Dok. zur Überpr. der Formularinhalte	
Überprüfung 1	
Überprüfung 2	
Verständlichkeit der Dokumentation	
Nachvollziehbarkeit	
Vollständigkeit	
Summe:	

Bewertungselemente XML/DTD	Punkte
Grundlegende Struktur *	
XML-Header	
Aufbau XML-Datei	
Funktion DTD	
Dokumentation XML und DTD	
Grundsätzliche Struktur XML	
Elemente XML	
Elemente DTD	
Element / Attribut / Entity - Eigenschaften	
Summe:	

Bewertungselement Präsentation	Punkte
Website	
Grundstruktur der Site	
Aufbau CSS	
Aufbau Formular	
Aufbau Javascript	
Funktionalität	
Präsentation selbst	
Summe:	

Extrapunkte* **

Gesamtsumme:

* Die max. Punktezahl wird bei funktionsfähiger Verwendung erreicht

** selbst erarbeiten / recherchieren

*** Bonus, können bei Bedarf zu den regulären Punkten hinzu kommen

Die maximalen Punktzahlen werden nur vergeben, wenn innerhalb der Präsentation erfolgreich nachgewiesen werden konnte, dass die Erstellung des Programms und die Dokumentation von Ihnen eigenständig erbracht wurde.

Bewertung der Arbeit, Bewertungskriterien

- Einfluss auf die Note haben ausschließlich die unten genannten Punkte. Es wird nicht bewertet, ob die Layouts gefällig oder ergonomisch gut sind. Trotzdem sollten Sie sich auch bei der Anordnung der Elemente auf den Seiten Mühe geben, da sich die Funktionalität auch über die Anordnung erschließt. Sie sollten darauf achten, dass ein Bestellvorgang mit dem von Ihnen erstellten Formular auch eindeutig abgeschlossen werden könnte. So sind beispielsweise die Lieferadresse und die Zahlungsart abzufragen. Die Inhalte der Seiten können Sie selbst definieren, soweit sie ethisch und moralisch einer Prüfungsleistung angemessen sind. Die Art und Aussage der Inhalte wird nicht bewertet.
- Bewertet werden:
  - Die Vollständigkeit im Sinne der Aufgabenstellung
  - Die Funktion der Seite und der XML/DTD-Struktur
  - Die Vollständigkeit der Dokumentation
  - Die Präsentation Ihrer Arbeit, im Sinne eines Nachweises, das Sie wirklich verstanden haben, was Sie programmiert und dokumentiert haben.
- Wird einer der Teile Programmerstellung, Dokumentation oder Präsentation nicht erbracht, gilt die Gesamtleistung als nicht erbracht und wird mit „nicht bestanden“ gewertet.
- Gruppengröße: 2 Personen

Internettechnologie für Medienwirte und Journalisten – überflüssig?

- Social media
- Online als Vertriebsweg
- Online als Recherche-Weg
- Beispiel Wissenschaft: Repositorien, OA
- Beispiel Online als Thema:
  - Investigativ
  - Schnell
  - Innovativ

Markup-Languages

zunächst am Beispiel: HTML

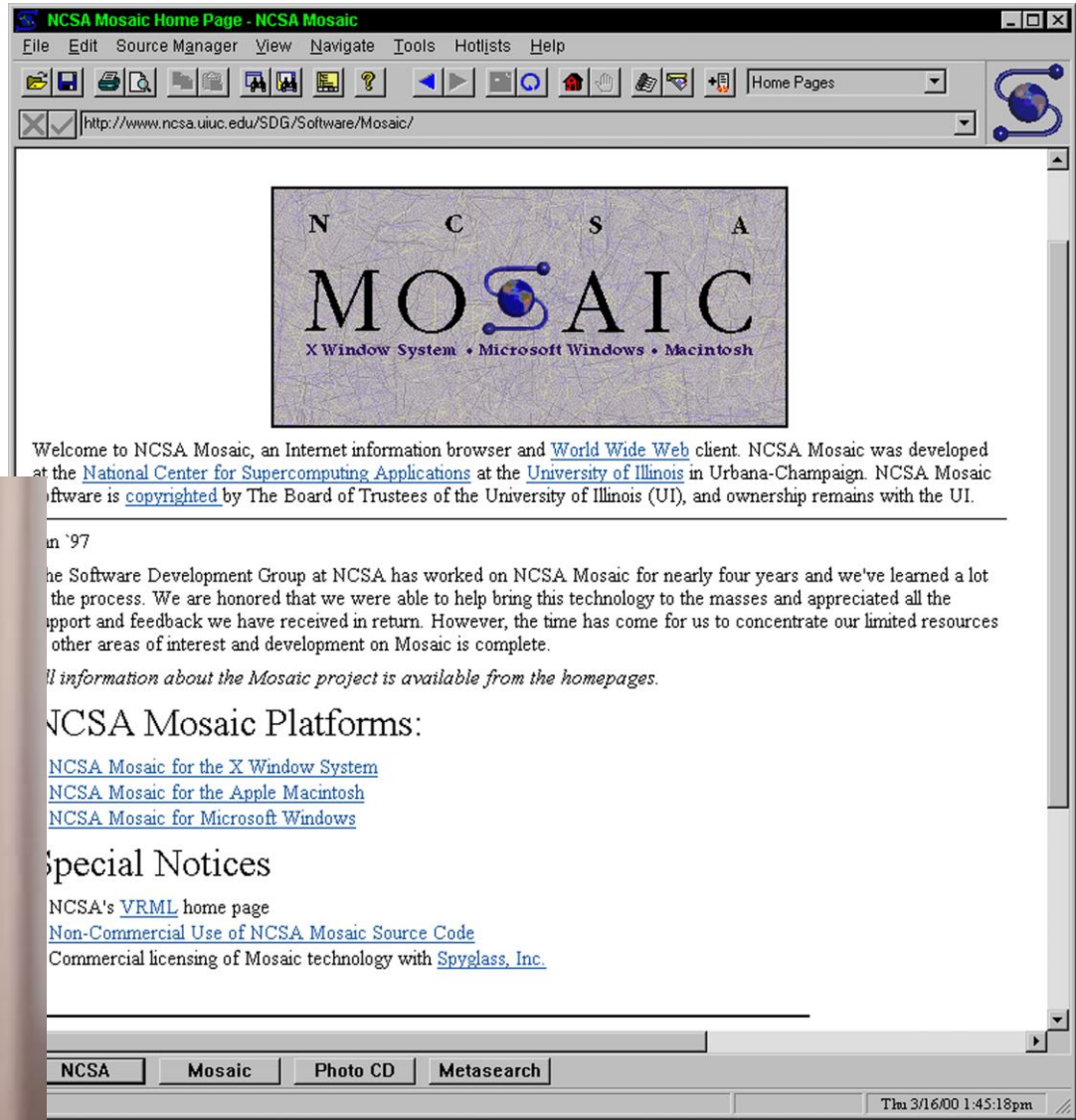
Hypertext Markup Language

Literatur und URLs

- Für Bücher gilt:
  Das Internet entwickelt sich permanent weiter. Nun sind die grundlegenden Funktionen, die in dieser Veranstaltung behandelt werden, über die Jahre relativ unverändert geblieben, so dass auch ältere Bücher *in dieser Hinsicht* nicht unbedingt veraltet sind.
  Es gibt aber auch hervorragende Online-Dokumentationen. Insbesondere *SELFHTML* bietet den Vorteil, dass diese Seiten im Lauf der Zeit immer wieder an die neuesten Entwicklungen angepasst wurden und somit hoch aktuell sind (Wiki).
- Stefan Münz: http://aktuell.de.selfhtml.org/
  Aus der von Stefan Münz entwickelte HTML-Dokumentation wurde inzwischen ein Wiki mit den aktuellen und umfassenden Inhalten.
- Hubert Partl:
  https://www.dropbox.com/l/hNO8F2GYrPBSfCKMEUgo5t
  Dokumentation von HTML mit Stand 2001, aber sehr umfassend.

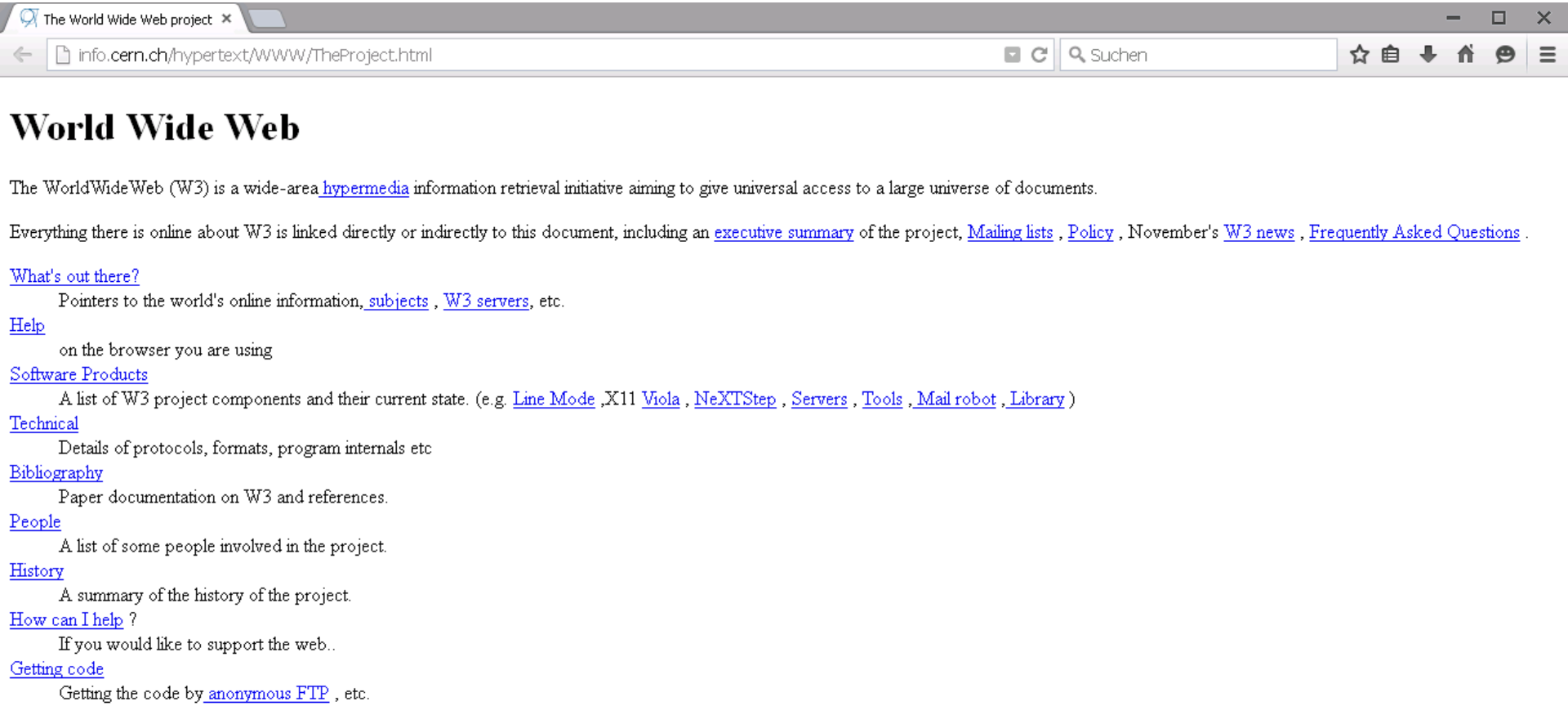
Historische Einführung

Tim Berners-Lee
(gilt als „Erfinder“ des WWW)



Die erste Homepage

So ging sie am 20. Dezember 1990 online und informierte über das Projekt (1. Proposal März 1989) eines systemübergreifenden Datenaustausch-Systems von CERN (so noch im Antrag)… der Begriff
„World Wide Web“ taucht hier auf der Website erstmals auf und verdeutlicht die Potentiale.



Die Essentials

- Server
- Client (Browser)
- hypertextfähige Markup-Sprache
  - hierzu gleich mehr
- Leitungen
  - hierzu mehr in Internettechnologie 2

SGML

- Wir beginnen bei **SGML** (Standard Generalized Markup Language). Diese Sprache wurde von der International Standard Organization (ISO) als ISO/IEC 8879-Standard **1986** veröffentlicht. Die Motivation war es ein Format zu schaffen, welches nicht (wie beispielsweise bei MS-WORD) das Layout des Dokumentes mit dem Inhalt vereint. (Problem damals: Versionswechsel von WORD 2 auf WORD6).

SGML

- Ziel von SGML war es, eine hersteller-unabhängige Beschreibungssprache für beliebig komplexe Dokumente darzustellen. Auf Grund der Mächtigkeit der SGML-Konstrukte und der weitreichenden Unterstützung seitens der Softwarehersteller passen sich SGML-definierte Dokumente an heutige heterogene EDV-Strukturen an.

SGML

- SGML unterscheidet sich im Gegensatz zum prozeduralen Markup dadurch, dass bei ihm einer Textdokumentenpassage eine bestimmte Rolle zugewiesen wird. Wo beim prozeduralen Markup einer Passage z.B. ein bestimmter Schriftsatz in einer bestimmter Größe usw. zugewiesen wird, wird sie beim deskriptiven Markup einfach als Überschrift 2. Ordnung identifiziert.
- Das kennen Sie aus modernen Textverarbeitungen, wo Sie dann entsprechend Inhaltsverzeichnisse etc. automatisiert erzeugen können.
- Vor SGML bzw. seiner Weiterentwicklung SGML-DTD (ISO 12083) war dies so nicht verbreitet.

Von SGML zu HTML

- Als Tim Berners Lee sein Konzept für das World Wide Web entwickelte, benötigte er eine Sprache, die Inhalte elektronisch über Datennetze senden konnte und gleichzeitig von Mensch und Maschine „verstanden“ werden konnte. SGML war für seine Zwecke ideal. So entwickelte er eine DTD (Document Type Definition), die wesentlich weniger komplex war und für seine Zwecke ausreichte: HTML.
- Ursprünglich war HTML als reine Dokumentbeschreibungssprache (keine Seitenlayout-Sprache!) ausgelegt und zunächst auf die Belange elektronisch übertragener Dokumente, später auf die des Internet (Hyperlinks, Grafiken, später Multimedia) ausgelegt. Historisch bedingt ist HTML immer mehr um Formatierungs-Tags gewachsen.
- Erst mit CSS und HTML5 wird wieder ein Weg geöffnet, der eine konsequente Trennung von Content und Layout ermöglicht (dazu in ein paar Wochen mehr), aber doch viele „Altlasten“ enthält.

xterm

#<<<>>> Lynx (web browser) - Wikipedia, the free encyclopedia (p1 of 5)

#copyright

Your continued donations keep Wikipedia running!

Lynx (web browser)

From Wikipedia, the free encyclopedia

Jump to: navigation, search

CAPTION: Lynx

Wikipedia Main Page displayed in Lynx
Wikipedia Main Page displayed in Lynx

Maintainer:	Thomas Dickey
Stable release:	2.8.5 (February 4, 2004) [[+/-]]
Preview release:	2.8.6 (?) [[+/-]]
OS:	Cross-platform
Use:	web browser
License:	GPL
Website:	lynx.isc.org

Lynx is a text-only Web browser and Internet Gopher client for use on cursor-addressable, character cell terminals.

Browsing in Lynx consists of highlighting the chosen link using cursor keys, or having all links on a page numbered and entering the chosen link's number. Current versions support SSL and many HTML features. Tables are linearized (scrunched together one cell after another without tabular structure), while frames are identified by name and can be explored as if they were separate pages.

Lynx is a product of the Distributed Computing Group within Academic Computing Services of the University of Kansas, and was initially developed in 1992 by a team of students at the university (Lou Montulli, Michael Grobe and Charles Rezac) as a hypertext browser used solely to distribute campus information as part of a Campus-Wide Information Server. In 1993 Montulli added an Internet interface and released a new version (2.0) of the browser [1] [2] [3].

-more- http://en.wikipedia.org/wiki/Image:Lynx_%28web_browser%29.png

Denken Sie dran, dass Anfangs der Inhalt wichtiger war als Layout… die Leitungen waren schmalbandig. Hier ein Shell-basierter Browser: Lynx

HTML: Keine absolute Layout-Kontrolle

- Ausdrücklich muss darauf hingewiesen werden, dass HTML keine absolute Kontrolle des Layouts beim Nutzer ermöglicht, auch wenn viele Webseitenersteller und Kunden dies noch immer nicht akzeptieren wollen.
- Der Vorteil von SGML ist übernommen worden: Nicht die Maschine, sondern Nutzer bestimmt das Layout des Dokumentes. Z.B. können sehschwache Menschen die Schriftart im Browser auf die für Sie lesbare Größe einstellen.
- Ein weiterer Vorteil ist, dass HTML-Seiten ohne „harte“ grafische Vorgaben auch auf kleineren Ausgabegeräten sinnvoll abgerufen werden kann, indem Formatierungsanweisungen anders interpretiert oder unterdrückt werden. (Mobile-Views)

Aufbau eines HTML-Dokuments

```
<html>

    <head>
    <title>Titel der HTML-Datei (das, was ganz oben in der
    'blauen' Leiste steht)</title>
    </head>

    <body>
    Inhalt der HTML-Datei (das, was im Hauptfenster steht)
    </body>

</html>
```

HTML-Source (Homepage der Jade-Hochschule – Code in Auswahl)

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="keywords" content="Jade, Hochschule, Fachhochschule, Studium,
    Studieren, …" />
    <title>Jade Hochschule</title>
    <meta name="DC.title" content="Jade Hochschule" />
    <meta name="keywords" content="Jade, Hochschule, Fachhochschule, Studium,
    Studieren, …" />
    <meta name="DC.Subject" content="Jade, Hochschule, Fachhochschule, Studium,
    Studieren, …" />
    <meta name="DC.Language" scheme="NISOZ39.50" content="de" />
    <meta name="date" content="2015-02-23T15:04:15+01:00" />
    <meta name="DC.date" content="2015-02-23T15:04:15+01:00" />
    <meta name="robots" content="index,follow" />
</head>
<body>
…
</body>
</html>
```

Meta-Tags beschreiben den Inhalt der Seite (wichtig für Suchmaschinen), weil sie die die Möglichkeit sind, Semantik in die Syntax zu geben.
DC = Dublin Core (eine Sammlung von semantischen Definitionen = Standard – NISO)

Character-Sets

- Haben Sie die Zeile
  `<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />`
  auf der vorigen Folie bemerkt?
- Ein Browser kann ohne näheren Angaben nicht wissen, welchen Zeichensatz er darstellen soll (wieder Historie – denn 1989 konnten die CERN-IBM-Systeme nur ASCII und ASCII enthält keine Umlaute), er wird dann auf einen im Programm eingestellten Standard zurückgreifen. Somit werden Umlaute auf deutschen Systemen dann (zufällig) richtig dargestellt - aber nicht zwangsläufig auch auf einem tschechischen (und schon gar nicht auf einem russischen). Daher kann man in dem Quelltext eine geeignete Kodierungstabelle angeben.

Vordefinierte Zeichen – HTML Entities

- Alternativ kann man auch HTML-Entities verwenden - das sind quasi HTML-Befehle für die direkte Erzeugung von Sonderzeichen, unabhängig von der Kodierung.
  Beispiele:
  - Umlaute: ä ö ü (ergibt ä ö ü)
  - ß: ß
  - Kaufmännisches Und: &
  - kleiner / größer: < / >

Suchmaschinen

```
<html>

    <head>
    <title>Meine erste HTML-Datei</title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <meta name="author" content="Philip J. Fry">
    <meta name="keywords" content="HTML, Anfänger, Übung, neu">
    <meta name="robots" content="index, nofollow">
    </head>

    <body>
    Jetzt gehts los...
    </body>
</html>
```

Suchmaschinen

- Metadaten sind für Suchmaschinen wichtig!
- Robots.txt Datei und Robots-Metadaten werden von (fast) allen Suchmaschinen beachtet (alles andere wäre übrigens rechtlich abmahnbar)
- Wegen SEO über Metadaten, genießen diese inzwischen nicht mehr den Stellenwert beim Ranking, wie noch um 2005.

Semantische Standards

- Metadaten sind Name-Wert Paare
  - <meta name="dc.creator" value="Du">
  - Aber was ist mit dem „Name“ wirklich gemeint? Die Frage nach der semantischen Bedeutung stellt sich hier!
  - Dafür gibt es das „dc“ vorab, das den Link auf http://dublincore.org/ legt

DCMI

URI:	http://purl.org/dc/terms/creator
Label:	Creator
Definition:	An entity primarily responsible for making the resource.
Comment:	Examples of a Creator include a person, an organization, or a service.

Übersicht Header-Informationen

Header Information	Erklärung
<meta http-equiv="content-type" content="text/html; charset=UTF-8">	Angabe eines Content-Types, wichtig für z.B. Umlaute UTF-8 kann mehr als 100000 Zeichen aus verschiedenen Schriften abbilden
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">	Alternativer Content-Type ISO 8859-1 ist Standard in den meisten Editoren und in Windows
<meta name="author" content="Name">	Angabe des Autors
<meta name="keywords" content="Wort, Wort, Wort">	Angabe von Stichwörtern
<meta name="robots" content="index | noindex , follow | nofollow">	Angabe für Suchmaschinen der senkrechte Strich | steht dabei für 'oder' (gilt auch nachfolgend)
<meta http-equiv="refresh" content="0; URL=http://www.xy.de/">	Angabe des Gültigkeitszeitraums für Caching
<title>HTML - Eine kurze Einführung</title>	Titel der HTML-Seite
<link rel=stylesheet type="text/css" href="kb.css">	Link zu Inhalten, die einzubinden sind. Hier ein Link zu einem Stylesheet.

Anmerkung zu CSS

<meta name="robots" content="index | noindex , follow | nofollow">	der senkrechte Strich | steht dabei für 'oder' (gilt auch nachfolgend)
<meta http-equiv="refresh" content="0; URL=http://www.xy.de/">	Angabe des Gültigkeitszeitraums für Caching
<title>HTML - Eine kurze Einführung</title>	Titel der HTML-Seite
<link rel=stylesheet type="text/css" href="kb.css">	Link zu Inhalten, die einzubinden sind. Hier ein Link zu einem Stylesheet.

Die an dieser Stelle und in den nachfolgenden Abschnitten vorgestellten Optionen zur Darstellung des Textes sind die grundlegenden, ‚traditionellen‘ Verfahren.
Mittlerweile erfolgt die Variation der Text-Darstellung in der Regel mit *Cascading Style Sheets (CSS)*, die sehr viel mehr Möglichkeiten bieten und letztlich auch einfacher in der Handhabung sind (wenn man sie denn verstanden hat). CSS werden im Lauf der Veranstaltung noch ausführlich behandelt und sind letztlich den hier aufgezeigten Methoden vorzuziehen.
Dennoch sind die 'traditionellen' Optionen nicht obsolet, darüber hinaus lässt sich mit ihnen das Prinzip der Textformatierung besser und einfacher erklären.

Ein wenig auf in den HTML-Body…

- Die Hintergrundfarbe wird hier als hexadezimale Zahl eingegeben.
  `<body bgcolor="#CCFFAA">`
- Was hexadezimale Zahlen sind, haben Sie im 1. Semester gelernt.
- Die ersten zwei Ziffern bestimmen den Rot-Wert, gefolgt vom Grün-Wert und vom Blau-Wert (RGB).

Farb-Werte

- "#000000" ergibt **Schwarz** (kein Anteil irgendeiner Farbe)
- "#FFFFFF" ergibt **Weiß** (alle Anteile maximal)
- "#0000ff" ergibt ein starkes **Blau** (Blau maximal, alle anderen minimal)
- "#aaaaaa" ergibt ein mittleres **Grau** (zwischen Schwarz und Weiß)
- "#ff00ff" ergibt **Violett** (Mischung aus Rot und Blau)
- Moderne Browser kennen viele vordefinierte Farben, so dass „background-color:white“ im CSS einem „background-color:#ffffff“ entspricht.

Optionen des Body-Tags

- `body` kennt noch weitere Optionen, mit denen Sie die Farben des Textes und der Hyperlinks definieren können.
  - Mit `text=` definieren Sie eine Farbe für den Text.
  - Mit `link=` definieren Sie eine Farbe für Verweise zu noch nicht besuchten Dateien (link = Verweis).
  - Mit `vlink=` definieren Sie eine Farbe für Verweise zu bereits besuchten Dateien (vlink = visited link = besuchter Verweis).
  - Mit `alink=` definieren Sie eine Farbe für Verweise, die der Anwender gerade anklickt (alink = activated link = aktivierter Verweis).

- Die Optionen werden dabei in dem body - Tag einfach hintereinander weg aufgelistet, also z.B.:

```
<body bgcolor="#CCFFAA" text="#EECCFF"
link="#AACCEE" vlink="#BBCCAA" alink="#00EECC">
```

Ein wenig Multimedia

- Wenn Sie als Hintergrund ein Bild benutzen wollen (Wallpaper) geben Sie die URL (wird weiter unten noch erklärt) des Bilder ein.

```
<body background="URL">
```

- Auch Sound ist als Hintergrund erlaubt, allerdings ist dies kein offizieller HTML Standard und hängt daher vom Browser ab! (siehe Beispiel)
  Sound / Musik nervt auch recht schnell.

  Zusätzlich sollten Sie auf das Dateiformat achten. Damit ein Browser die Datei wiedergeben kann, muss der Rechner diese interpretieren können. Verwenden Sie z.B. MP3, muss der entsprechende Codec auf dem Rechner (Betriebssystem) installiert sein. MP3 sollte mittlerweile auf Windows-Rechnern kein Problem mehr sein. Allerdings ist MP3 kein freies Format, weswegen ein Decoder z.B. bei Linux-Installationen meistens nicht installiert ist. Auch aus diesem Grund verwendet z.B. Wikipedia das freie OGG-Format, das wiederum auf Windows nachinstalliert werden muss. Und dann gibt es noch RAM, MOV, WAV, …

```
<html>
<head>
<title>Hintergrundmusik</title>
<!-- Internet Explorer: -->
<bgsound src="sound-datei.mid" loop="infinite">
</head>
<body>
<!-- Firefox: -->
<embed src="sound-datei.mid" autostart="true" loop="true" hidden="true" height="0"
width="0">
<h1>Inhalt der Seite</h1>
</body>
</html>
```

Sie kennen bereits

- *Sie kennen die Teilung zwischen Header und Body.*
- *Sie kennen verschiedene Metadateninhalte.*
- *Sie wissen, dass dem Dokument grundsätzliche Formatierungen mitgegeben werden können.*

Textauszeichnung: Strukturierung und Überschriften

Die Strukturierung von Dokumenten wird in der Regel über Überschriften vorgenommen. In HTML stehen sechs Ebenen (h1 - h6) von Überschriften zur Verfügung. Beachten Sie, dass Sie sauberes HTML schreiben. Grundsätzlich haben alle Tags einen Ende-Tag.

```
<h1 align="left"> Überschrift erster Ordnung </h1>
```

Die Angabe H1 bis H6 ist dabei als Hierarchie - dies verdeutlicht wieder den Gedanken hinter dem 'reinen' HTML: die Auszeichnungen dienen primär nicht der Text-Darstellung, sondern der Text-Strukturierung. Das unterschiedliche optische Erscheinungsbild ist nur ein 'Nebeneffekt'.

Die Angabe der Option

`align="left"`

sagt aus, dass die Überschrift links am Dokumentenrand ausgerichtet sein soll. Weitere Optionen sind

`align="right"` rechtsbündig

`align="center"` zentriert

`align="justify"` Blocksatz

Textauszeichnung: Absätze

Absätze werden mit dem Tag `<p>` erreicht. Denken Sie an das Ende-Tag `</p>`.

Auch Absätze können eine Ausrichtung zugewiesen bekommen.

Die Umbrüche eines Textes werden in HTML automatisch vom Browser vorgenommen und hängen von der Fenstergröße ab.

Sie können aber mit `<br>` auch Umbrüche erzwingen.

Textauszeichnung: Leerzeichen

Mehrere Leerzeichen hintereinander im Quelltext werden vom Browser bei der Darstellung ignoriert.

Es wird immer ein Leerzeichen angezeigt.
`"Wie geht es Dir?"`
wird zu „Wie geht es Dir?“

Wenn Sie mehr Leerzeichen benötigen, können Sie ein Sonderzeichen ` ` einfügen, einen „non breaking space“. Dabei entspricht ` ` einem Leerzeichen.

Textauszeichnung: Kommentare

Sie können Ihren HTML-Quelltext mit Kommentaren versehen, die dann im Browser nicht angezeigt werden. Dies ist z.B. hilfreich, bei längeren Quellcodes die Übersicht zu behalten.

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

Das funktioniert auch über mehrere Zeilen; solange `<!--` nicht mit einem `-->` geschlossen wird, wird alles innerhalb als Kommentar behandelt.

```
<!--
Dies ist ein Kommentar.
Dies auch.
Und dies immer noch
-->
```

Aber Achtung: Der Inhalt eines Kommentars darf nie mit einer Raute beginnen.

`<!--#` hat eine besondere Bedeutung, die wir in einigen Wochen noch schätzen lernen werden.

Beispiele

Beispiele	Erklärung
<!-- Dies ist ein Kommentar -->	Kommentar in HTML
Text	Überschrift, hier 'Typ 1' (oberste Ebene; die nachfolgenden wären dann entsprechend h2, h3, ..., h6) Groß- / Kleinschreibung ist hier egal: h1 = H1, etc.
Text	Überschrift mit Positionsangabe; **der senkrechte Strich | steht hier (und nachfolgend immer) für 'oder'**
Text	Absatz
Das ist der Text, der in die nächste Zeile soll	Zeilenumbruch
Das ist der Text, der in eine Zeile soll	Zeilenumbruch verhindern
Das gibt je drei Leerzeichen	Das gibt je drei Leerzeichen
verschiedene HTML-Elemente	Zenrieren

Textauszeichnung: Listen - Beispiele

Beispiele	Ansicht	Erklärung
<ol type="a | A | i | I"> - nummerierter Listeneintrag - nummerierter Listeneintrag 	A. nummerierter Listeneintrag B. nummerierter Listeneintrag i. nummerierter Listeneintrag ii. nummerierter Listeneintrag	Numerierte Listeneinträge
<ul type="square | circle | disc"> - Listeneintrag - Listeneintrag 	▪ Listeneintrag ▪ Listeneintrag ○ Listeneintrag ○ Listeneintrag	Unnumerierte Listeneinträge (verschiedene Symbole)

Textauszeichnung: Text-Erscheinung

Beispiele	Ansicht	Erklärung
<blockquote>Text</blockquote>	Text	Geblocktes (eingerücktes) Zitat
<address>Text</address>	*Text*	Ist gedacht, um Adressen von Zitaten zu kennzeichnen. Wird allerding kaum verwendet.
<pre>Text</pre>	Text	Vorformatierter Text, erscheint wie im Editor angelegt mit Tabs und Spaces
betont	*betont*	Betonter Text (kursiv)
italic	*italic*	italic (alternativ kursiv)
stark betont	**stark betont**	Stark betonter Text (Fettdruck)
bold	**bold**	bold (alternativer Fettdruck)
<code>Quellcode</code>	Quellcode	Quellcode (ähnlich wie vorformatiert)
<samp>Beispiel</samp>	Beispiel	Beispiel
<kbd>Tastatureingabe</kbd>	Tastatureingabe	Tastatureingabe
<var>Variable</var>	*Variable*	Variable
<cite>Zitat</cite>	Zitat	Zitat
~~Gelöschter Text~~	~~Gelöschter Text~~	Zur Kennzeichnung von Text, der nicht mehr erscheinen soll, aber ursprüglich im Text enthalten war
<ins>Eingefügter Text</ins>	Eingefügter Text	Hinzugefügter Text
<u>underline</u>	underline	underline (alternativ unterstrichen)

Textauszeichnung: Text-Erscheinung

Annotation: Unterschied Struktur / Format

Dies kann man sich gut an dem letzten Beispiel verdeutlichen:
<ins>Eingefügter Text</ins> liefert 'visuell' einen unterstrichenen Text: Eingefügter Text.
Eine Unterstreichung lässt sich aber auch anders realisieren:
<u>Eingefügter Text</u> liefert: <u>Eingefügter Text</u>.
'Optisch' ist das Resultat gleich. Es verbirgt sich aber jeweils eine andere Information dahinter. Dies könnte z.B. bei einer automatischen Sortierung von Texten durch Computerprogramme oder Skripte ausgenutzt werden.

Das Wichtigste in HTML: Hyperlinks (sagt der Name ja schon)

Ein wesentliches Merkmal von HTML ist die Idee bzw. Möglichkeit, Dokumente miteinander zu verknüpfen und eine Navigation innerhalb des Dokumentes zu ermöglichen.

Zu einem Link gehört der Tag `<a> </a>`

Das ganze sieht typischerweise so aus:

```
<a href="Adresse-des-Ziels"> hier das was anklickbar ist
</a>
```

also z.B.

```
... bla <a href="http://www./whv/">hier gehts zu den VL-
Infos</a> bla ...
```

liefert

... bla hier gehts zu den VL-Infos bla ...

Hyperlinks

Verweise auf weitere Dokumente werden unter Angabe der **URL** (Uniform Resource Locator; "einheitlicher Quellenanzeiger") vorgenommen.

Dabei sollte beachtet werden, dass die Verweise innerhalb einer Präsentation über **relative Verweise** vorgenommen werden. Vorteil: Die Dokumente können auf andere Server umziehen, wobei sich die Domain ändert. Der Auftritt muss nicht nachgearbeitet werden.

Soll auf externe (nicht zur Präsentation gehörende) Dokumente verwiesen werden, muss die vollständige URL angegeben werden

Interne Verlinkung

Innerhalb eines Dokumentes (der aktuellen Seite) kann ebenfalls verlinkt werden.

Dazu werden Sprunganker definiert, um beispielsweise von einem Inhaltsverzeichnis direkt auf ein Kapitel zu springen (dabei liegt die gesamte Präsentation in einer Datei. Vorteil: Die Datei wird nicht neu geladen. Langfristig ist die Verlinkungsstruktur stabil.)

Bsp: im Quelltext wird ein Sprunganker gesetzt

> ... hier ist der Absatz zuende und es beginnt ein neues Kapitel.
>
> `<a name="kap_2"> <h2>Kapitel 2</h2> </a>`
>
> Hier geht es weiter mit dem Text.

Dann ist `kap_2` die Bezeichnung für eine unsichtbare (es wird nichts besonderes an der Stelle angezeigt) Marke, die ein Ziel für einen Link darstellt.

Dieser kann dann mit

```
<a href="#kap_2">Hier geht es zum Kapitel 2</a>
```

angesprungen werden. Da es eine interne Verlinkung ist, muss ein # am Anfang stehen.

Anklickbar ist also der Text „Hier geht es zum Kapitel 2“, ein Klick führt zur Sprungmarke namens kap_2 (der name ist beliebig.

Die aktuelle Seite verschiebt sich dann, so dass die Sprungmarke im Browserfenster oben steht (immer vorausgesetzt, dass die Seite 'lang' genug ist). Das ist in diesem Fall gerade eine Überschrift 'Typ 2' mit dem Titel 'Kapitel 2'.

Aufgabenblatt zur ersten Übung

Aufgabe 1

Erstellen Sie eine HTML-Seite, die lediglich den Text: "HALLO WELT" anzeigt. Der Text soll in einer Größe der Überschrift 1 erscheinen.

Aufgabe 2

Erstellen Sie eine weitere HTML-Seite, die einen strukturieren Text anzeigt.
Erzeugen Sie verschiedene Überschriftenarten, Absätze und erzwungene Umbrüche, Aufzählungslisten, unterschiedliche Textfarben etc.
Zur Erzeugung von Fülltext können Sie z.B. das Angebot von http://www.blindtextgenerator.de/ nutzen.

Aufgabe 3

Versehen Sie den Beispieltext der vorhergehenden Aufgabe mit internen Links: Unterteilen Sie den Text in verschiedene Abschnitte mit jeweils einer eigenen Überschrift und legen Sie oben auf der Seite ein entsprechendes Inhaltsverzeichnis an. Die Punkte des Inhaltsverzeichnis sollen dann auf das jeweilige Kapitel (auf der gleichen Seite) verlinken.
Fügen Sie einen externen Link auf den angegebenen Blindtext-Generator ein.

Aufgabe 4

Probieren Sie aus, was passiert, wenn Sie Ihren eigenen Namen als „Color-Code“ verwenden. Was lernen wir aus dem Ergebnis?

*** Aufgabe 5**

Erzeugen Sie eine weitere HTML-Seite und verlinken Sie von der ersten zur zweiten - und zurück. Probieren Sie eine absolute Verlinkung und eine relative. Was ist der Unterschied und welche Vor- bzw. Nachteile bringt dies mit sich?