

Web-Technologien: Eine Einführung

Stefan Gössner

Webtechnologien

Als Webtechnologien bezeichnen wir die Summe aller Technologien, mittels derer das WWW betrieben wird.

- XML
 - HTML
 - HTTP
 - CSS
 - Javascript
 - DOM
 - DTD
-

Basis-Dienste?

TCP-IP Protokoll

Zerlegung von Daten in kleine Datenpakete – Senden jener Päckchen an IP-Adressen – Dort wieder Zusammensetzen.

IP-Adressen

Vier Zahlen – durch Punkte getrennt (149.77.211.198) . Die Vier Zahlen teilen sich in die *Netzwerknummer* und die *Hostnummer* auf (Class-A,B,C Netze).

DNS

Wir Menschen können mit IP-Adressen nicht gut umgehen. Daher wird alternativ ein Domain-Name **fh-luh.de** zur Verfügung gestellt.

Client-Server Prinzip

Ein Server ist ein *Hostrechner*, auf dem eine Webserver-Software unter ständiger Verbindung mit dem Internet läuft. Ein *Client* ist eine Software (Browser, Email, ...), die Dienste des Servers mittels HTTP anfordert.

Internet Dienste

- E-Mail
- Telnet und SSH
- File Transfer (FTP)
- Chat (IRC)
- Newsgroups (News)
- World Wide Web (WWW)

Entstehung des Web



- **Tim Berners-Lee** erzeugte als Informatiker am CERN (Genf) ein *Hypertext* System zum Eigenbedarf. 1990 formulierte er schliesslich die 3 Säulen des WWW:
 - HTTP
 - URI Uniform Resource Identifier
 - HTML Hypertext Markup Language



- **Marc Andreessen** programmierte als Student den ersten Browser **Mosaic**. Seine spätere Firma *Netscape* erreichte dann einen Marktanteil von 90% mit dem *Navigator*.



- Das WWW-Konsortium als Zusammenschluss interessierter Personen/Firmen erarbeitet freie, offene (Web)standards.

offene Standards für Dokumentenformate

von einem Standard sprechen wir hier, wenn ...

- die Dokumentstruktur festgelegten Regel gehorcht.
- eine gewisse Verbreitung vorliegt.

ein Standard ist offen, wenn ...

- die Dokumentenstruktur vollständig offengelegt ist.
- die Festlegung der Dokumentstruktur durch mehrere Personen und/oder Firmen erfolgt.
- eine Erweiterung der Dokumentstruktur möglich ist.
- die Verarbeitung eines Dokuments nicht an ein einzelnes Programm gebunden ist.

... ansonsten nennen wir den Standard proprietär.

HTML - (1)

HTML ist ...

- ... ein reines *Textformat* und damit plattform- und softwareunabhängig.
- ... eine *Auszeichnungssprache* (dies ist `fett`).
- ... imstande, auf andere Dokumente verweisen zu können (Hypertext).
- ... ist nicht mehr im Entwicklungsplan des **W3C**.

HTML - (2)

HTML genügt nach der Meinung des W3C nicht mehr den künftigen Herausforderungen.

- neue Geräte: Digital TV, Mobiltelefon, Automobil, Kühlschrank erfordern ...
 - einen erweiterten Sprachumfang für moderne, leistungsfähige Browser, oder ...
 - einen reduzierten Sprachumfang für einfache Geräte.
- die Internationalisierung (asiatische Schriften) wird nicht ausreichend unterstützt.
- die heutigen interaktiven Fähigkeiten reichen nicht aus.
- die geforderten Erweiterungen in Hinblick auf Grafik, Audi, Video, math. Formeln, etc. würden die Sprache überfrachten.

HTML - (3)

HTML Grundgerüst

```
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
<h1>Überschrift</h1>
</body>
</html>
```

CSS - (1)

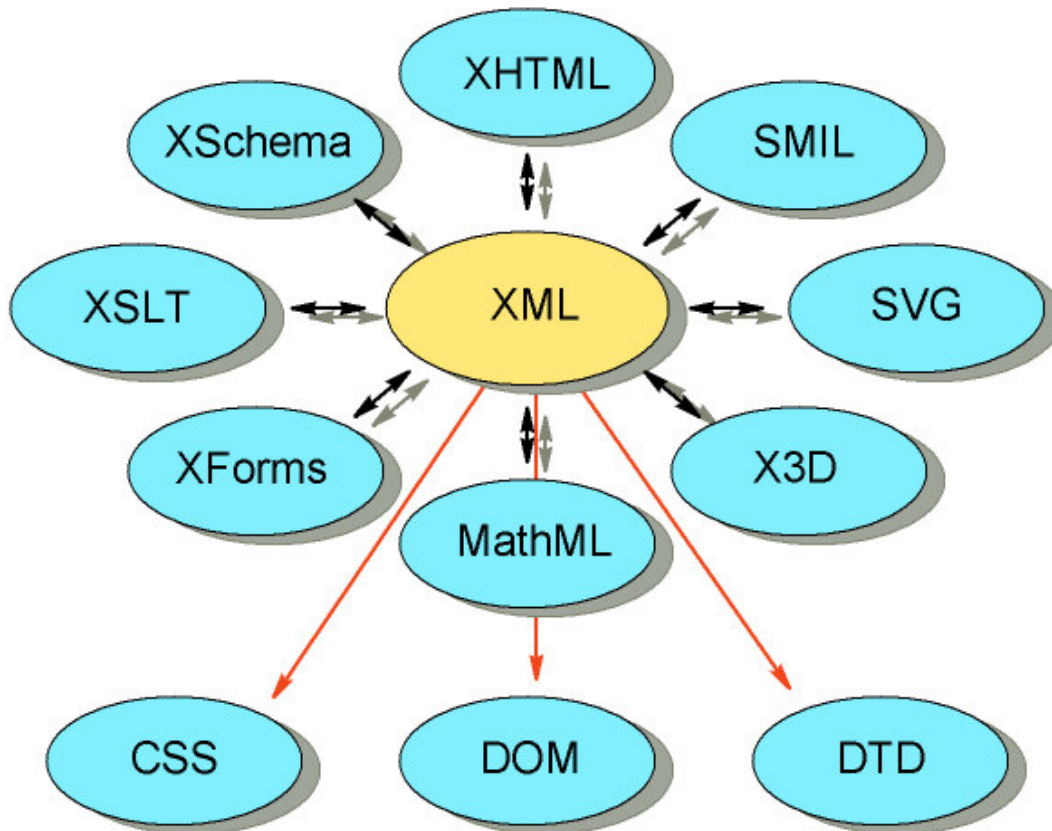
CSS Cascading Style Sheets erlauben die Trennung von Inhalt und Aussehen. Es spielt in etwa dieselbe Rolle, wie Formatvorlagen einer Textverarbeitung.

Ohne CSS wären das einheitliche Aussehen eines Webauftritts (corporate Identity) grosser Firmen schwer zu realisieren.

[CSS in der Praxis](#)

XML-Familie

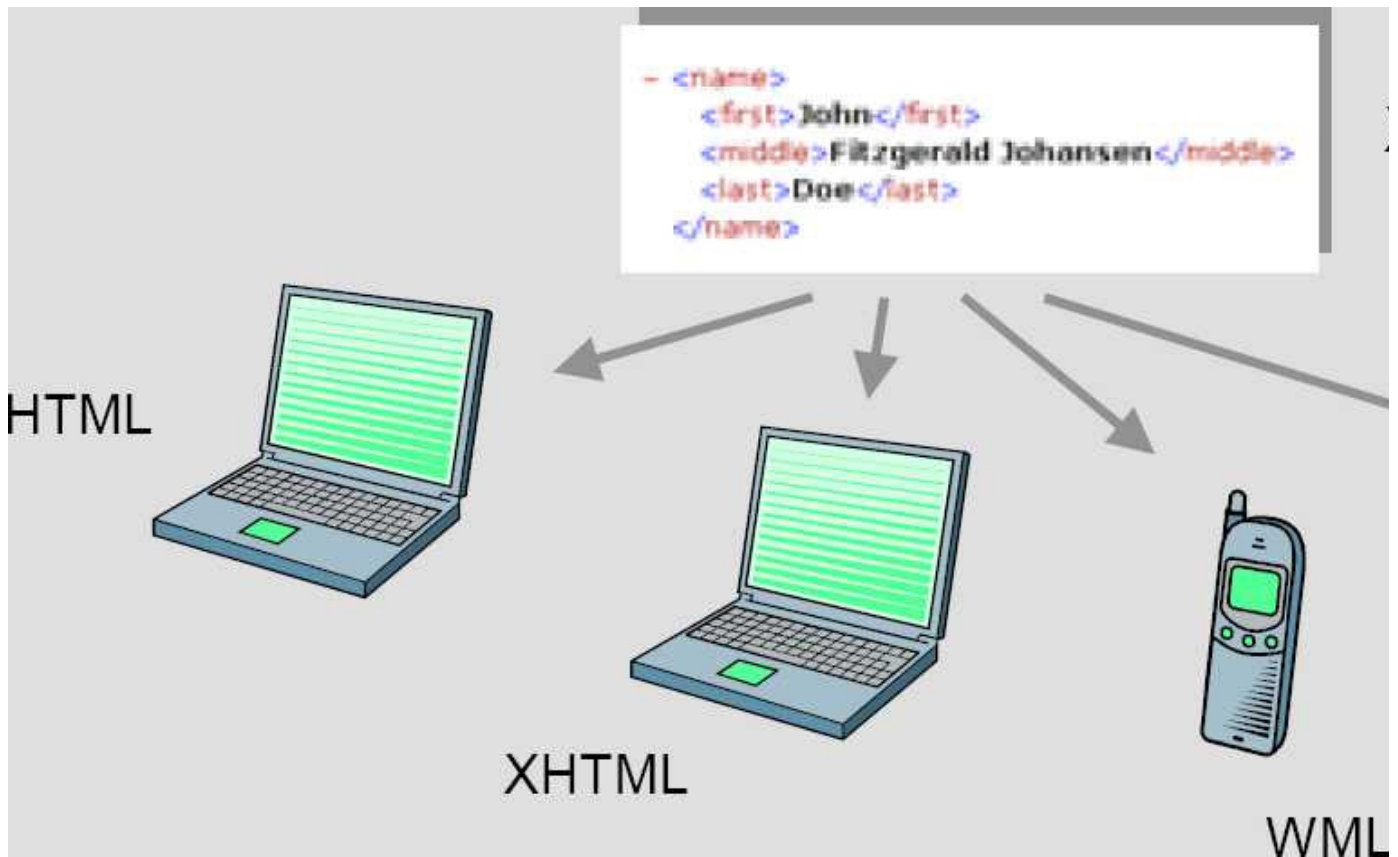
XML ist die Mutter einer Familie standardisierter Markup Sprachen.



XML

- XML ist die zentrale Basis der neuen Webtechnologien.
- XML ist eine *Metasprache*, d.h. ..
- .. XML ist nicht gebrauchsfertig, sondern stellt ein flexibles Regelwerk dar, mittels dessen Dokumentformate, Datenstrukturen und -protokolle formuliert werden.
- XML erlaubt die Definition eines problemspezifischen Vokabulars (Metasprache).
- XML eignet sich besonders zur Beschreibung von Datenstrukturen (Datenformat der Zukunft).
- XML unterstützt die Trennung von Inhalt und Darstellung (XSLT).
- XML ist "*einfach*" ?
- XML ist "*in*" ?

Trennung von Inhalt und Präsentation



Nochmal, Was ist XML?

- XML ist eine Methode, um strukturierte Daten in einer Textdatei darzustellen.
- XML sieht fast aus wie HTML, ist aber kein HTML.
- XML ist Text, aber nicht zum Lesen.
- XML ist eine Familie von Techniken.
- XML ist neu, aber nicht so neu.
- XML ist lizenzfrei, plattformunabhängig und gut unterstützt.

Baumstruktur von XML

- Jedes XML-Dokument hat genau ein Wurzelement.
- Dieses Wurzelement entspricht dem Dokument-Typ.
- Kind-Elemente sind immer geordnet.

Grundbausteine von XML

- Elemente: beschreiben die Struktur eines XMLDokumentes
- Attribute: Zusatzinformationen zu einzelnen Elementen
- Die XML-Deklaration: Informationen für Parser
- Namensräume: geben Elemente eine bestimmte Bedeutung

```
<?xml version="1.0" encoding="UTF-8"?>
<name id="1232345">
<first>John</first>
<middle>Fitzgerald Johansen</middle>
<last>Doe</last>
```

</name>

- Namensräume: geben Elemente eine bestimmte Bedeutung

Elemente

- Elemente haben einen Namen und bestehen aus:
 - einem Anfangs-Tag (engl. start tag)
 - einem dazugehörigen Ende-Tag (engl. end tag)
 - einem Inhalt.
- Beispiel: <first>John</first>
 - „first“ ist der Name des Elementes.
 - <first> ist ein Anfangs-Tag.
 - </first> ist das dazugehörige Ende-Tag.
 - „John“ ist der Inhalt des Elementes.
 - <first>John</first> wird Element genannt.

Elementinhalt

Der Inhalt von Elementen kann:

- aus einfachem Text bestehen (unstrukturierter Inhalt)
- weitere Elemente enthalten (strukturierter Inhalt)
- gleichzeitig Elemente und Text enthalten (gemischter Inhalt)
- leer sein

Attribute

- Ein Element kann eine beliebige Anzahl von Attributen haben.
- Attribute sind Name-Wert-Paare der Form name="wert" oder name='wert'.
- Der Wert eines Attributes ist immer vom Typ String (unstrukturierter Inhalt).
- Beachte: Die Reihenfolge der Attribute ist belanglos.

Attribute oder Element

Wann also Attribute und wann Elemente verwenden?

- Ein Attribut kann nur einen String als Wert haben, ein Element kann beliebig strukturiert werden.
- Die Reihenfolge der Attribute ist belanglos, diejenige von Elementen nicht.
- Einheitliche Darstellung mit Elementen ist eleganter, Darstellung mit Attributen kompakter.
- Fazit: Attribute eignen sich besonders für einfache, unstrukturierte Zusatzinformationen (Metadaten).

XML Deklaration

<?xml version="1.0" encoding="UTF-8"?>

- enthält Informationen für Parser, insbesondere die verwendete XML-Version und Kodierung
- muss immer am Anfang der Datei erscheinen
- Ist zwar optional, sollte aber dennoch immer vorhanden sein!
- version (obligatorisch): verwendete XML-Version, aktuelle Version ist "1.0".
- standalone (optional): Gibt an, ob es eine zugehörige Dokument-Typ-Definition gibt ("no") oder nicht ("yes").
- encoding (optional): verwendete Kodierung des XMLDokumentes
- Die drei Attribute müssen immer in dieser Reihenfolge erscheinen.

Regeln für wohlgeformte XML-Dokumente

1. Jedes Anfangs-Tag muss ein zugehöriges Ende-Tag haben.
2. Elemente dürfen sich nicht überlappen.
3. XML-Dokumente haben genau ein Wurzel-Element.
4. Elementnamen müssen bestimmten Namenskonventionen entsprechen.
5. XML beachtet grundsätzlich Groß- und Kleinschreibung.
6. XML belässt Formatierungen (white space) im Text.
7. Ein Element darf niemals zwei Attribute mit dem selben Namen haben.

Regel 4: Namenskonventionen

- Namen beginnen entweder mit einem Buchstaben oder „_“: z.B. first, First oder _First
- Nach dem ersten Zeichen sind zusätzlich Zahlen sowie- und „_“ erlaubt: z.B. _1st- name oder _1st.name
- Namen enthalten keine Leerzeichen.
- Namen enthalten kein „:“.
- Namen beginnen nicht mit „xml“, unabhängig davon, ob die einzelnen Buchstaben groß- oder kleingeschrieben sind.
- Diese Konventionen gelten für alle Bezeichner in XML, nicht nur für Element-Namen.

Dokument-Typen

... beschreiben den prinzipiellen Aufbau von Dokumenten eines bestimmten Typs, z.B.:

1. Das Wurzelement ist <name>.
2. <first>, <middle> und <last> sind Kind-Elemente von <name>.
3. <first> und <last> sind obligatorisch, optional.
4. <first>, <middle> und <last> erscheinen in dieser Reihenfolge.

Dokument-Typen (2)

- Dokument-Typen können entweder mit DTDs (Document Typ Definitions) oder XML-Schemata spezifiziert werden.
- DTDs für Spezifikation von Text-Dokumenten ausreichend, XML-Schemata zur Spezifikation von Daten besser geeignet.

XML - Beispiel

```

<?xml version="1.0"?>
<rezept typ="hauptgericht" personen="4" z-dauer="30" brennwert="600">
  <titel>Zucchini-Mozzarella-Auflauf</titel>
  <zutatenliste>
    <zutat>1 Zwiebel</zutat>                <zutat>2 Knoblauchzehen</zutat>
    <zutat>800 g Tomaten, geschält</zutat>   <zutat>600 g Zucchini</zutat>
    <zutat>Salz </zutat>                    <zutat>Pfeffer </zutat>
    <zutat>1 Bund Basilikum</zutat>        <zutat>1 TL Gemüseextrakt </zutat>
    <zutat>300 g Mozzarella</zutat>        <zutat>100 g Parmesan, gerieben</zutat>
  </zutatenliste>
  <zubereitung>
    <schritt>
      Zwiebel und Knoblauch in wenig Öl glasig dünsten. Tomaten und Gemüsebrüheextrakt dazugeben
      und einige Minuten köcheln.
    </schritt>
    <schritt>
      Zucchini waschen, schneiden, dünsten, mit Salz und Pfeffer kräftig würzen. In eine Auflaufform die
      Tomatensauce geben, darauf die Hälfte der Zucchinischeiben, abschliessend die Hälfte der Mozzarella
      verteilen. Mit Basilikum bestreuen. In der gleichen Reihenfolge die restlichen Zutaten einschicht
    </schritt>
    <schritt notwendig="nein"> Die letzte Schicht dick mit Parmesan abdecken. </schritt>
    <schritt> Bei 180 - 200 Grad etwa 30 Minuten backen. </schritt>
  </zubereitung>
</rezept>

```

XML-Beispiel (Quelle: Ulrike Kirchgesser)