Multimedia- & Webtechnologien

# XML + CSS 2/3
# Grundlagen – Möglichkeiten – Grenzen – Beispiele

eingereicht als

## Ausarbeitung

an der

**Fachhochschule Lippe und Höxter**
**University of Applied Sciences**

---

Fachbereich Produktion und Wirtschaft
im Studiengang Wirtschaft
Wintersemester 2005/06

von

Waldemar Herber, Christian Melcher, Malte Wattenberg

Lemgo, im Januar 2006

**Inhaltsverzeichnis**

## 1. Was ist XML?

### 1.1 Einführung

Das Kürzel XML steht für Extensible Markup Language; welches ins deutsche übersetzt soviel heißt wie erweiterbare Auszeichnungssprache[1].
XML wurde von einer Arbeitsgruppe des World Wide Web Consortiums (W3C) in dem Zeitraum vom Mitte 1996 bis Anfang 1998 entworfen, mit dem Ziel Daten bzw. Dokumente jeglicher Art strukturiert abzubilden. Das universelle Format von XML stellt sicher, dass Dokumente plattform- und anwendungsübergreifend verwendet werden können.
Obwohl der Vorgänger SGML bereits weitaus umfangreicher war, kam es nie zu einer breiten Akzeptanz in der Öffentlichkeit. Der Grund dafür liegt in der Komplexität SGMLs, die die Softwareentwicklung stark erschwert. Die Komplexität von SGML und XML kann mit der Pareto-Verteilung beschrieben werden: obwohl XML nur ca. 20% der Komplexität von SGML hat, können damit ca. 80% der Anwendungsfälle abgedeckt werden. Der Bedarf nach einem unbeschränkten weltweiten Informationsaustausch und die Popularität von HTML brachten das deutlich einfachere XML hervor.
Es ist ganz klar gesagt dass es sich bei XML um keine Programmiersprache bzw. eine Seitenbeschreibungssprache wie etwa Postscript[2] handelt, sondern um eine Metasprache. Eine Metasprache ist eine Sprache mit der eine andere Sprache erklärte oder definiert wird, mit der also die Regeln für eine Auszeichnungssprache festgelegt werden.

### 1.2 XML - HTML

Im Unterschied zu HTML gibt es bei XML keine festgelegte Liste von Tags. Der Entwickler kann seine eigenen Tags für die Strukturierung seiner Daten erfinden

[1] Eine **Auszeichnungssprache** (engl. *Markup Language*, Abk. *ML*) dient zur Beschreibung von Daten oder des Verfahrens, das zur Darstellung nötig ist. Bei einer Auszeichnungssprache werden die Eigenschaften, Zugehörigkeiten und Verfahren von bestimmten Wörtern, Sätzen und Abschnitten eines Textes beschrieben bzw. zugeteilt, meist in dem sie mit Tags markiert werden. Wie jede Programmiersprache besitzen auch die Auszeichnungssprachen eine Syntax, eine Grammatik und eine Semantik.
[2] **Postscript** ist optimiert für die Druckerausgabe und eignet sich nur bedingt für die Anzeige am Bildschirm. Postscript-Dateien sind außerdem nicht für die Bearbeitung gedacht, sondern ein endgültiges Ausgabeformat.

oder kann sich auf öffentliche verfügbare Vokabulare stützen, die für immer mehr Gegenstandsbereiche entwickelt werden. Die große Stärke von XML ist dabei insbesondere, dass inhaltliche Strukturen in einer beliebigen Tiefe verschachtelt werden können, so dass auch hochkomplexe Hierarchien jeder Art repräsentiert werden können.

Mit Hilfe von XML ist es möglich, die Struktur, den Inhalt und die Darstellung eines Dokumentes streng zu trennen und entsprechend dann auch unabhängig von einander zu be- und verarbeiten. Während die Tags in HTML in erster Linie festlegen, in welcher Form Inhalte in einem entsprechenden Medium ausgegeben werden sollen, wird mit XML versucht, die Bedeutung von Daten so festzuhalten, dass nicht nur Menschen, sondern auch Maschinen damit etwas anfangen können. XML ist somit ein Standard zur Erstellung von maschinen- und menschenlesbarer Dokumente in Form einer Baumstruktur. Das erlaubt zum einen eine Prüfung der Gültigkeit von Dokumenten, ist zugleich aber auch die Basis für erweiterte Formen der Gestaltung und der Verknüpfung von Dokumenten.

Der wesentliche Unterschied zwischen einer XML- und einer HTML-Datei ist, dass das XML-Dokument eine reine Datensammlung ist, die zunächst noch keinerlei Hinweise darüber enthält, wie die Daten etwa in einem Browser zu präsentieren sind. Um die Daten in einer brauchbaren Form auszugeben, kann und muss dem XML-Dokument ein Stylesheet[3] beigegeben werden, entweder in Form von CSS-Stylesheets, wie sie auch für die Formatierung von HTML-Seiten verwendet werden, oder mit Stylesheets, die in der XML-Sprache XSLT oder XSL geschrieben sind.

Anstelle der Anzeige von XML-Daten durch ein Stylesheet kann die Ausgabe auch über ein JavaSript oder eine andere Programmiersprache gesteuert werden. Das ist vor allem dann sinnvoll, wenn gezielt auf bestimmte Informationen aus einem XML-Dokument zugegriffen werden soll.

---

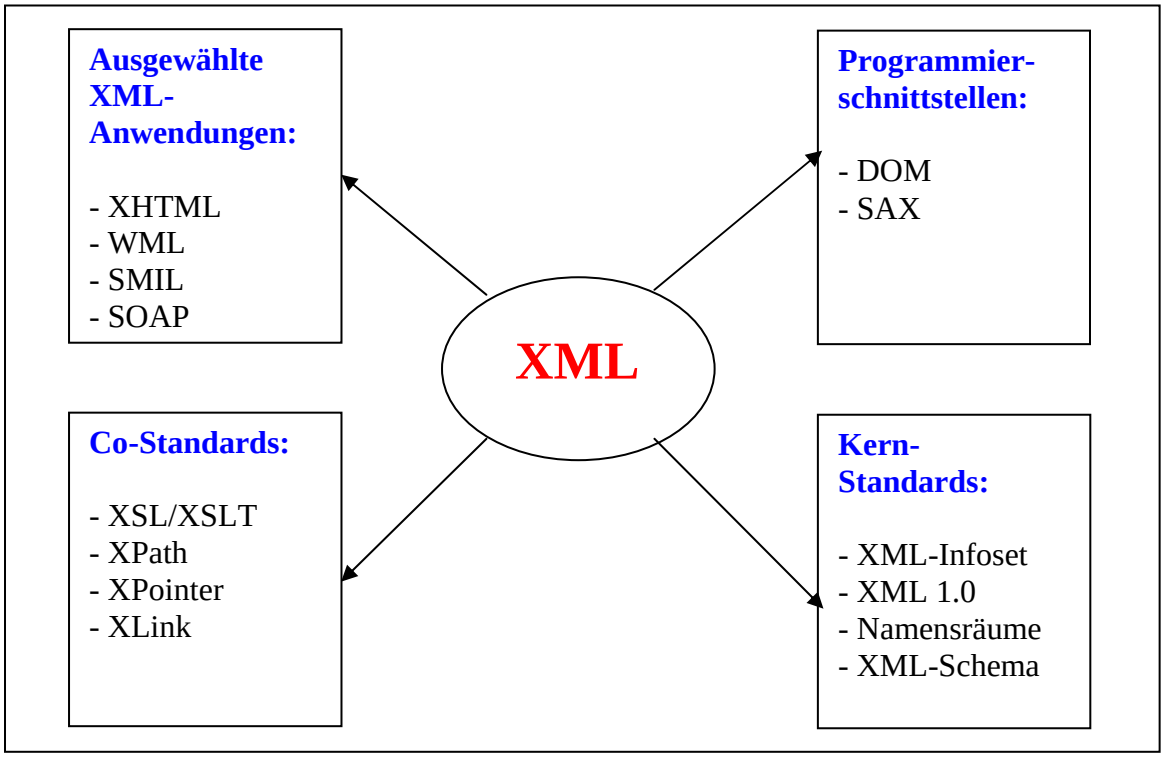
[3] **Stylesheet** ist in der Informationstechnik eine Beschreibungssprache. Ein Stylesheet ist am ehesten mit einer Formatvorlage zu vergleichen. Grundidee hierbei ist die Trennung von Information (Daten) und Darstellung. Das Stylesheet interpretiert die zugewiesenen Daten (Text, Tabellen, Grafiken etc.) und formatiert sie (z.B. für die Bildschirmausgabe) entsprechend der vorgegebenen Regeln. Mit Stylesheets ist in höherem Maße eine Arbeitsteilung möglich, als das früher z. B. bei HTML und eingebetteten Formatierungsbefehlen möglich war.

Beispiel:

Ein JavaSript wird in eine HTML-Seite eingefügt, um auf Knopfdruck eine bestimmte Datenauswahl anzuzeigen. Dabei wertet das Skript die XML-Datei nicht direkt aus, sondern vermittelt über ein Baum-Modell, dass der XML-Prozessor[4], der im Internet Browser integriert ist, aus den XML-Daten zusammengebaut. Mit Hilfe dieses Modells stehen dann bestimmte Schnittstellen zur Verfügung, um auf einzelne Elemente oder Attributwerte des XML-Dokuments zugreifen.

### 1.3 Übersicht über die Sprachfamilie XML

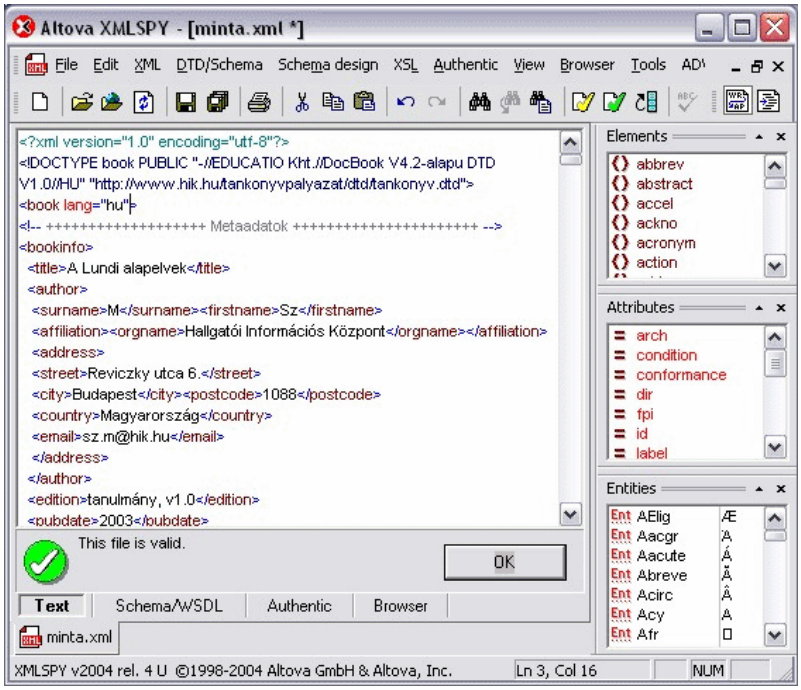
Der Grad der Standardisierung ist bei den verschiedenen Familienmitgliedern von XML noch etwa uneinheitlich, zudem kommen immer wieder Erweiterungen hinzu. Den schnellsten Zugang zum aktuellen Stand aller Standardisierungsprojekte des W3C finden Sie über die folgende Adresse: www.w3.org/TR/#Recommendations. Die Abbildung bietet einen Überblick über die Sprachfamilie XML.




[4] Ein Software-Modul, genannt **XML-Prozessor**, dient dazu, XML-Dokumente zu lesen und den Zugriff auf ihren Inhalt und ihre Struktur zu erlauben.

## 1.4 XML-Editoren

Aufgrund der Transformation lassen sich XML-Dokumente, DTD´s und Schemas oder XSL- und XSLT-Dateien im Prinzip mit jedem Texteditor erstellen und bearbeiten. Das verspricht allerdings eine Menge <und> und wieder </und>. Und der Name in jedem End-Tag muss haargenau mit dem im Start-Tag übereinstimmen. Inzwischen gibt eine Reihe von komfortablen XML- und Stylesheet-Editoren, die den Anwendern vieles abnehmen können, schon dadurch, dass End-Tags immer automatisch erzeugt werden, wenn der Start-Tag abgeschlossen ist.




Komplette XML-Entwicklungswerkzeuge wie XML Spy von Altova (siehe Bild oben) oder XMetal von SoftQuad, um nur zwei zu nennen, unterstützen den User bei der Arbeit, zum Beispiel kontextsensitive Elementenlisten oder Attribut-Inspektoren, und sorgen durch integrierte XML-Prozessoren dafür, dass die Wohlgeformtheit und auch Gültigkeit der Dokumente schon während der Entwicklung jederzeit geprüft werden kann. XML Spy unterstützt zudem auch eine Generierung einfacher Eingabemasken für XML-Dokumente, die ihnen die manuelle Eingabe von Tags gleich ganz abnehmen.

## 2. Was ist XSLT?

### 2.1 Einführung

Das XSLT ist die Abkürzung für E**x**tensible **S**tylesheet **L**anguage for **T**ransformation. XSLT ist eine in XML beschriebene Sprache, die ihrerseits den Prozess der Umwandlung eines XML Dokumentes in ein anderes XML-Dokument oder ein anderes Ausgabeformat (HTML, PDF, SVG…) beschreibt. Diesen Vorgang nennt man Transformation. Das Quelldokument und das Ergebnisdokument, das mit Hilfe des Stylesheets erzeugt wird, können dabei beliebig weit von einander abweichen. Im Extrem könnte ein Stylesheet sogar ganz auf eine Quelldatei verzichten und einfach selbstherrlich festlegen, was in das Ergebnisdokument geschrieben werden soll.

Das Stylesheet mag beispielsweise nur einen Teil der Informationen aus dem Quelldokument beachten und das Ergebnisdokument um beliebig viele Informationen erweitern, die nicht im Quelldokument vorhanden sind. Dabei arbeitet XSLT immer nur mit dem in einem XML-Dokument vorhandene Markup, DTDs[5] und Schemas[6] spielen für XSLT keine Rolle. Die Quelldatei muss wohlgeformt[7] sein, mehr nicht.

XSLT-Stylesheets können deshalb nicht HTML-Dateien angewendet werden, aber auf XHTML-Dateien, weil diese ja dem XML-Format entsprechen. Es kann also sinnvoll sein, entsprechende Konvertierungen (Änderung eines Datenformates) nach XHTML vorzunehmen, um die Möglichkeiten von XSLT zu nutzen. Die Transformation wird kodiert in Form von XSLT-Stylesheets, die selbst wohlgeformte XML-Dokumente sind.

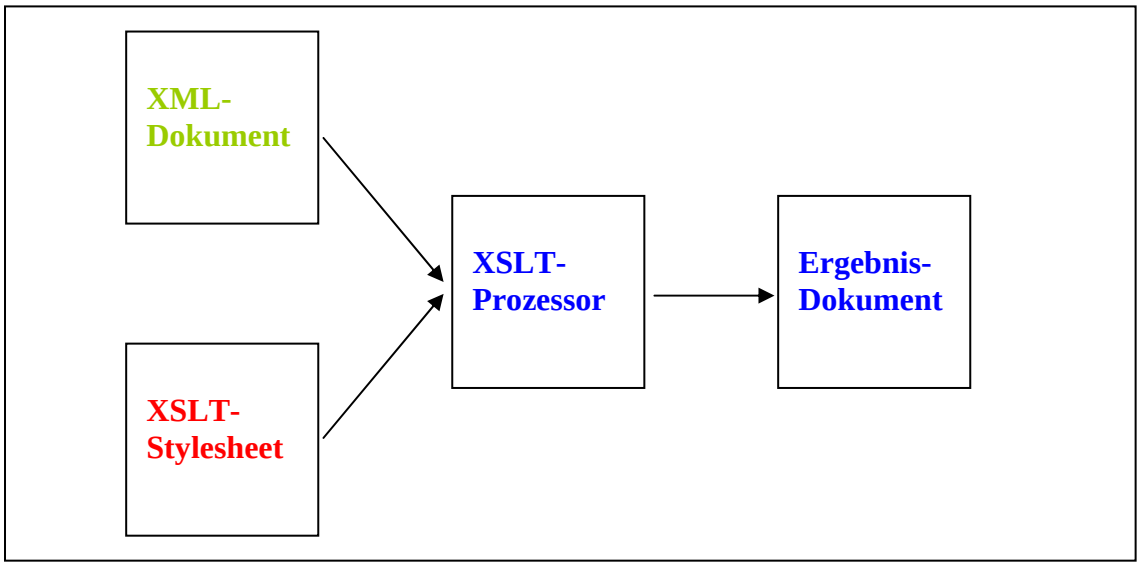
---

[5] Mit **DTD** (Dokument Type Definition) wird ein XML-Dokument auf seine Gültigkeit überprüft. Genauer gesagt mit Hilfe von DTDs können Regeln für ein XML-Dokument bestimmt werden. Diese Regeln besagen, welche Daten enthalten sein dürfen und welche nicht.

[6] **XML-Schema** ist ein neuer Ansatz, ein XML-Dokument auf seine Gültigkeit zu überprüfen. Es stellt wie ein DTD eine Beschreibung der Struktur eines XML-Dokumentes dar, kann aber auch den Inhalt von Elementen und Attributen einschränken

[7] **Wohlgeformtheit** beschreibt eine Reihe von Bedingungen und Regeln, die als Mindestanforderungen erfüllt sein müssen um ein XML-Dokument mit XML-Werkzeugen verarbeiten zu können.

## 2.2 Generelles Schema der Transformation durch XSLT




Damit die Transformation stattfindet, muss ein spezieller XSLT-Prozessor vorhanden sein, der das Stylesheet und die Quelldaten einliest, die im Stylesheet enthaltenen Anweisungen ausführt und das dementsprechende Ergebnisdokument erzeugt. Einige der in der Praxis gängigsten Prozessoren sind Xalan, Saxon, TransforMiix, Sablotron etc. Diese Prozessoren setzten die Anweisungen des Stylesheets selbständig in die dafür notwendigen Bearbeitungsschritte um. XSLT ist in diesem Sinne also ein „High-level Language“, bei der sich der Entwickler nicht um die groben Details zu kümmern braucht.
Dabei sind mehrere Wege möglich. Die Anweisungen des Stylesheets auf das XML-Quelldokument kann schon auf einem Webserver ausgeführt werden, der dann das fertige Ergebnisprodukt für die Clientseite bereitstellt. Das Stylesheet kann aber auch direkt auf der Clientseite mit Hilfe eines geeigneten Browsers oder eines separaten Programms angewendet werden. Ein solches Programm ist zum Beispiel das bekannte Java-Programm XT.

## 2.3 Verknüpfung zwischen XML(Dokument) und XSLT(Stylesheet)

Obwohl XSLT eine Reihe von Elementen enthält, die sich auch in allgemeinen Programmiersprachen finden, etwa Elemente für die Bildung von Verzweigungen oder Schleifen, handelt es sich doch im wesentlichen um eine beschreibende Sprache, die dem XSLT-Prozessor gewissermaßen in Form von Beispielen sagt,

was er tun soll. „Wenn Du auf ein Element mit dem Namen <Thema> triffst, dann schreibe es in das Ergebnisdokument“. Das wie bleibt dabei den eingebauten Funktionen des XSLT-Prozessors überlassen, derer sich das Stylesheet einfach bedient.

Damit ein Browser wie der Internet Explorer die XML-Quelldaten so ausgibt wie im Stylesheet vorgesehen, muss in die Quelldatei eine entsprechende Verknüpfung eingebaut werden. Dies geschieht mit Hilfe einer entsprechenden Verarbeitungsanweisung, die so aussieht:

```
<?xml-stylesheet type="text/xsl" href="kursliste.xslt"?>
```

Beispiel:

***kursliste.xml:***

```
<?xml version="1.0" encoding="iso-8859-1"?>
<?xml-stylesheet type="text/xsl" href="kursliste.xslt"?>
<kursprogramm>
    <kurs name="XML-Grundlagen" dauer="2 Tage">
        <referent>Stefan Goessner</referent>  //alles was grün ist =
        Anzeige im Browser
        <termin>01.01.2006</termin>
        <beschreibung>Einführungsseminar für Programmierer und IT-
Manager        </beschreibung>
    </kurs>
    <kurs name="XSL-Praxis" dauer="5 Tage">
        <referent>Bodo Klare</referent>
        <termin>02.01.2006</termin>
        <beschreibung>Praktische Übungen für Programmierer
        </beschreibung>
    </kurs>
    <kurs name="XSLT-Einstieg" dauer="2 Tage">
        <referent>Hanna Horn</referent>
        <termin>03.01.2006</termin>
        <beschreibung>Einführung in die Transformation von XML-
        Dokumenten für Programmierer</beschreibung>
    </kurs> </kursprogramm>
```

***kurslite.xslt:***

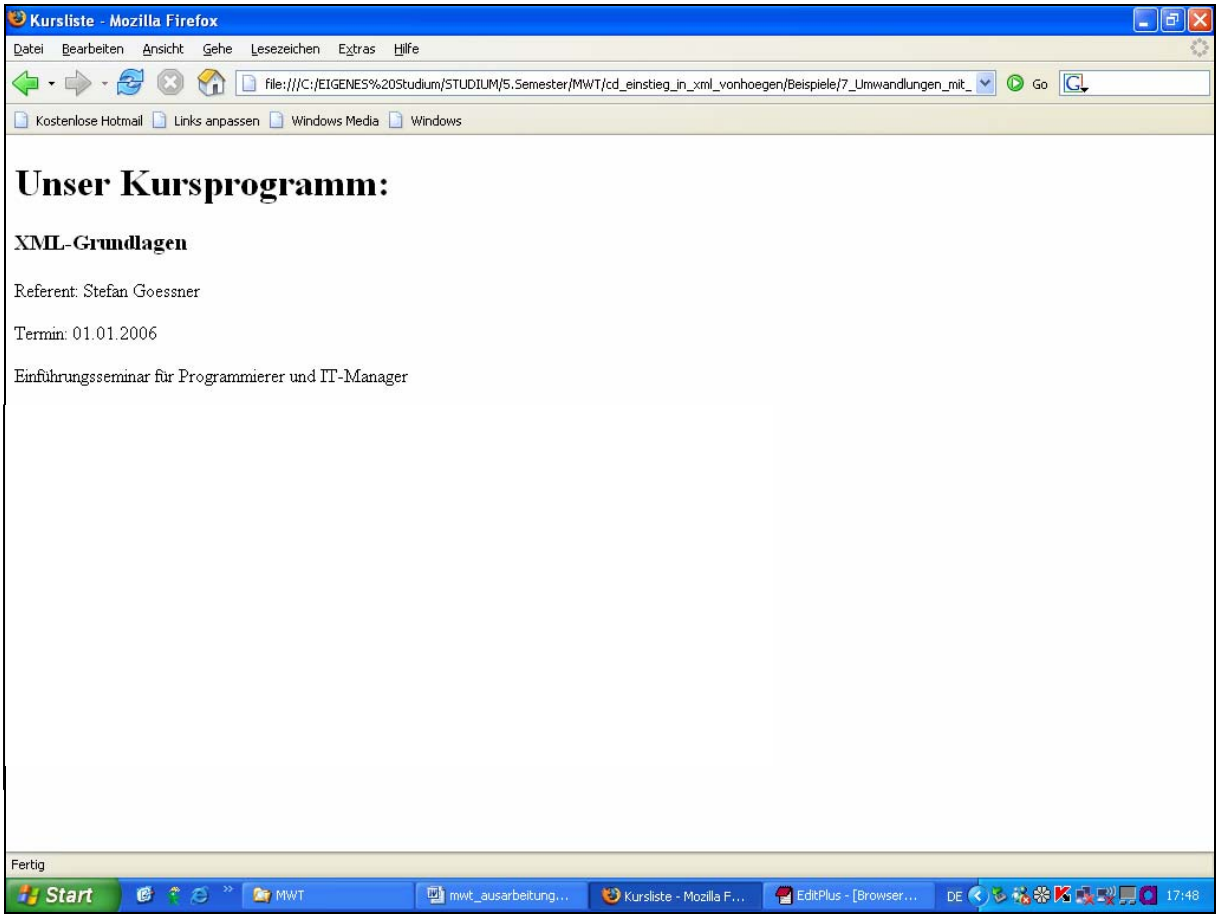
```
<?xml version="1.0"  // XML-Deklaration
      encoding="ISO-8859-1"?>  //um die Ausgabe der Umlaute zu gewährleisten
<xsl:stylesheet  //Wurzelelement (Synonym auch <xsl:transform>)
      version="1.0"  //Version 1.0
      xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  //URI des XSLT-
                                                          Namensraumes
      <xsl:output method="html"/>  //Ausgabeart im HTML-Format

      <xsl:template match="/">  //Template Start-Tag
            <html>
                  <head>
                        <title>Kursliste</title>
                  </head>
                  <body>
                  <h1>Unser Kursprogramm:</h1>
                  <xsl:apply-templates />  //XSLT-Anweisung
                  </body>
            </html>
      </xsl:template> //beendet das erste Template
      <xsl:template match="kurs"> //Beginn zweite Template-Regel
            <h3>
                  <xsl:value-of  //XSLT-Element → Name d. jew. Kurses
            select="@name"/> //greift auf den String-Wert d. Attributs name
            </h3>
            <p>Referent:  <xsl:value-of select="referent"/></p>
            <p>Termin:  <xsl:value-of select="termin"/></p>
            <p><xsl:value-of select="beschreibung"/></p>
      </xsl:template>  //beendet das zweite Template </xsl:stylesheet>
```

Template[8]

[8] Template-Regel wird als Suchmuster (*match="/"*) verwendet, d.h., der Prozessor soll diese Regel anwenden, wenn er den Wurzelknoten der Quelldatei vor sich hat. Die Zeilen zwischen dem Start- und End-Tag sind das eigentliche **Template**, das dem Prozessor beschreibt, was zu tun ist, wenn beim Durchwandern des Quellbaums ein Knoten gefunden worden ist, der exakt dem Suchmuster entspricht

***Der Browser gibt das formatierte XML-Dokument als Liste aus:***



# 3. CSS 2.1

## 3.1 Intention

Um Stil und Inhalt einer Website zu trennen wurde CSS (Cascading Style Sheets) entwickelt. Hierbei handelt es sich um eine Skriptsprache, die die begrenzten Gestaltungsmöglichkeiten von HTML erweitert und ergänzt. Die Trennung von Stil und Inhalt ermöglicht außerdem die Erstellung von Formatvorlagen, ähnlich wie es in einem Textverarbeitungsprogramm möglich ist. Somit eignet sich CSS hervorragend um beispielsweise ein einheitliches Layout für den Internetauftritt, z.B. für das Corporate Designs eines Unternehmens oder eines Projektes, zu schaffen.
Die gängigsten Elemente sind die Textgestaltung, die Gestaltung von Hintergründen, Tabellen und Formularen sowie die Platzierung von Bildern oder Texten. CSS eignet sich außerdem noch…

- zur Steuerung des Drucklayouts. Die HTML-Seite wird hierbei in einem angemessenen Format an den Drucker übermittelt, sodass alle Zeichen gut sichtbar sind und nicht Teile der Seite abgeschnitten werden.
- zur akustischen Wiedergabe von HTML-Seiten mit Hilfe einer künstlichen Sprachausgabe.
- zur Änderung des Aussehens des Cursors.
- zur Änderung des Seitenlayouts. Wenn die HTML-Seite auf einem anderen Anzeigemedium mit anderer Auslösung (z.B. einem PDA) dargestellt wird, wird die Seite dementsprechend angepasst.

## 3.2 Syntax

Die Syntax von CSS besteht im Wesentlichen aus Regeln, wobei eine oder mehrere Eigenschaften (properties) mit den dazu gehörenden Eigenschaftswerten (values) Selektoren zugewiesen werden. Diese Selektoren sind dabei HTML – Elemente.

Bsp.:

```
Allgemeiner Aufbau
 selector {
     property1: value1;
     ...;
     propertyN: valueN;
     }

Beispiel
 h1 {
    font-size: 14pt;
    ...;
    background-color: blue;
    }
```

**Selector:** Auswahl der gewünschten Elemente (p, h1-h7, table, div, body...)

**Property:** festzulegende Eigenschaft (color, border, margin, padding...)

**Value:** Eigenschaftswert (red, solid, 12px...)

Selektoren können nun, mit variabler Zahl von Eigenschaften, in vielfältiger Form angewandt und kombiniert werden:

Gruppierung:

Um Wiederholungen von Stilregeln für verschiedene Befehle zu vermeiden können mehrere, durch ein Komma getrennt, gleichzeitig angeführt werden. Das folgende Beispiel definiert somit für die Überschriften <h2>, <h3> und <h4> eine blaue, kursive Schrift:

```
 h2, h3, h4     { color: blue; font-style: italic }
```

Descendant Selector:

Kontextabhängige Selektoren (descendant selectors) werden nur angewandt, wenn bestimmte HTML Befehle in einander geschachtelt sind. Soll z.B. in einer Überschrift <h2> ein tiefgestellter Index in blauer Schrift hinterlegt werden, ist folgende Stilregel geeignet:

```
 h2 sub {color: blue}
```

Diese Stilregel wird nur wirksam, wenn <sub> in <h2> eingeschachtelt ist:

```
<h2> Überschrift mit einem tiefgestellten <sub>blauen Index</sub><h2>
```

Ausgabe: Überschrift mit einem tiefgestellten blauen Index

Child Selector:

Eine weitere Einschränkung der Wirkung von Stilregeln ist durch die „Vater- Kind-Stilregel“ (child selector) möglich. So gilt

```
Ul>li {color: blue}
```

nur für die li-Listenelemente (Kinder) von ul-Listen. Ausgedrückt wird die Vater-Kind-Beziehung durch ein Größer-Zeichen (>).

Adjacent Selector:

Benachbarte Stilregeln (adjacent selectors) legen eine bestimmte Reihenfolge fest, in der die Befehle aufeinander folgen müssen. Soll z.B. ein Absatz der direkt auf einen anderen Absatz folgt eine blaue Schrift haben, so gilt:

```
p+p {color: blue}
```

Alle folgenden Absätze haben nun natürlich auch eine blaue Schrift. Um dies zu umgehen können adjacent selectors natürlich auch gruppiert werden. Folgende Stilregel würde z.B. dem dritten und fünften Absatz eine schwarze Schrift zuweisen:

```
p+p+p, p+p+p+p+p {color: black}
```

### 3.4 Klassen

Mit den bisherigen Mitteln können alle Elemente (bspw. <p> oder <h1>) nur einmal definiert werden. Es ist aber wünschenswert, dass z.B. Absätze unterschiedlich gestaltet werden sollen. Klassen werden nun in genau dieser Situation eingesetzt. An dem zu definierenden Element werden dazu einfach ein Punkt und ein beliebiger Klassenname angehängt:

```
p.beispiel {color:blue}
```

An den Stellen im Dokument, an denen diese Stilregel angewandt werden soll, schreibt man:

```
<p class=“beispiel“> Hallo Welt</p>
```

Sinnvollerweise benennt man Klassen nach ihrer Funktion und nicht nach Erscheinung. So muss der Klassenname nicht abgeändert werden weil sich beispielsweise die Schriftfarbe ändert. Es ist aber erlaubt für mehrere Befehle den gleichen Klassennamen zu vergeben.
Ebenso ist es möglich sog. „*generische Klassen*“ zubilden. Hierbei werden Klassen unabhängig von anderen Elementen definiert und können somit in verschiedenen Kombinationen mit diesen eingesetzt werden:

Bsp.:

```
.zentriert {text-align: center}
```

Diese generische Klasse zentriert den Text und eignet sich bspw. für Absätze oder Überschriften.

```
<p class=“zentriert“> Dieser Absatz ist zentriert</p>
<h1 class=“zentriert“> Diese Überschrift ist ebenfalls zentriert</h1>
```

### 3.5 Einbindung von CSS

Insgesamt gibt es drei verschiedene Möglichkeiten CSS in Dokumente einzubinden.

Definition im jeweiligen Tag:
In dieser Variante wird CSS direkt im jeweiligen Tag definiert. Zunächst muss das Tag geöffnet werden (also z.B. <p) und um style ergänzt werden. Hierdurch wird erkannt, dass es sich um eine Stilvorlage handelt. Nach dem Gleichheitszeichen folgen die Eigenschaften und ihre Werte, eingerahmt durch Anführungsstriche (“ “). Eigenschaften werden von ihren Werten werden durch einen Doppelpunkt getrennt, ein Semikolon nach den Werten erlaubt eine beliebige Anzahl weiterer Festlegungen. Den Abschluss der Definition bildet nun eine schließende Klammer (>). Um ein gültiges Dokument zu erhalten fehlt nur noch das schließende Tag (</p>).

Bsp.:

```
<p style="background-color: yellow; color: red; border: dashed; font-
weight: bolder">
 CSS Definition direkt im Tag </p>
```

Anzeige im Browser:

**CSS Definition direkt im Tag**

Definition im Head einer HTML-Datei

Bei dieser Einbindungsmöglichkeit werden alle zu gestaltenden Tags bereits im Head des Dokumentes definiert. Der Aufbau von Eigenschaften und Eigenschaftswerten ist dabei nur minimal abweichend von der ersten Variante. Nachdem der <head> geöffnet wurde werden durch den Tag <style> die Stilvorlagen eingeleitet und durch den type “text/css“ als CSS gekennzeichnet. Die Eigenschaften werden nun von einer geschweiften Klammer eingeschlossen und anschließend der geöffnete style Tag wieder geschlossen.

Bsp.:

```
<html>
 <head>
 <title>Definition im Head</title>

 <style type="text/css">
 p      {
        background-color: white;        // weißer Hintergrund
        font-weight: bold;              // Schriftstärke: fett
        font-size: 14pt;                // Schriftgröße: 14 Punkte
        color: red;                     // Schriftfarbe: rot
        }
 </style>
 </head>
 <body>
 <p> Eine schöne rote Textzeile </p>
 </body>
 </html>
```

Anzeige im Browser:

**Eine schöne rote Textzeile**

Zu beachten ist in diesem Beispiel jedoch dass Kommentare zwischen Eigenschaften im Microsoft Internet Explorer zur Nichtbeachtung aller weiteren Eigenschaften führen.
Der Vorteil dieser Methode gegenüber der ersten erscheint klar. Da die Stilvorlage nur im head – Bereich definiert wird ist das Dokument insgesamt schlanker und vor allem übersichtlicher.

Erstellung einer separaten Datei

Diese Variante ist in der Praxis am weitesten verbreitet. Zunächst wird eine Datei erstellt und mit der Endung .css gespeichert. In dieser Datei werden nun alle Eigenschaften und ihre Werte ausgelagert. Jeder beliebige Texteditor kann diese Modifikationen vornehmen.
Nach Erstellung der .css Datei wird diese nun in das HTML – Dokument durch einen Verweis (<link>) im Head- Bereich eingebunden. Als nächstes wird Bezug (rel=relation) zu einem Stylesheet, also einer Stilvorlage, vom Type “text/css“ genommen. Schließlich muss durch eine Referenz (href=hyper reference) der Speicherort der .css Datei angegeben werden. Diese kann im selben Ordner wie das Dokument befinden, in einem anderen Ordner oder auch bei einer beliebigen Adresse des Webs.

Bsp.:

```
<html>
<head>
<title> Separate CSS-Datei </title>

<link REL=stylesheet TYPE="text/css" HREF="stilvorlage.css">
</head>

<body>
</body>
</html>
```

Alternativ zu dieser Variante kann die .css Datei auch mit dem CSS – spezifischem Befehl @import ausgewählt werden:

```
 <style>
@import url("stilvorlage.css");
</style>
```

Vorteil dieser Möglichkeit CSS einzubinden ist ganz klar dass nun beliebig viele Dokumente auf die .css Dateien zugreifen können. Somit ist es besonders effizient möglich durch wenige Änderungen das komplette Design eines Webauftritts zu verändern. Ein weiterer Vorteil ist ebenfalls, dass Inhalt und Stil nun am Deutlichsten getrennt wurden. Dies resultiert in zusätzlicher Übersicht in den Dokumenten.

## 4. CSS 3

CSS 3 nimmt immer mehr Konturen an. Und auch wenn CSS 3, das im Gegensatz zu CSS 1 und 2 aus einzelnen "Modulen" besteht, in seiner Gesamtheit auch 2008 noch keine offizielle W3C-Spezifikation sein wird, sind einige Module wie das Syntax- oder Farben-Modul relativ weit fortgeschritten. Dieses Kapitel beschreibt ein paar Neuerungen, die mit unterschiedlicher Wahrscheinlichkeit mit CSS 3 kommen werden.

Neues Boxmodel mit „box-sizing“

Mit der box-sizing-Eigenschaft wird CSS 3 eine alternative Form des CSS-Boxmodells bieten. Nach CSS 1 und 2 definieren die Werte der width- und height-Eigenschaften die Breite und Höhe des Inhalts eines Elements, nicht seiner gesamten Box (mitsamt Inhalt, Innenabstand, Rahmen und Außenabstand). Der Standardwert von box-sizing, content-box, entspricht diesem Modell. Der alternative Wert, border-box, bezieht in die Breite und Höhe neben dem Elementinhalt auch Innenabstand und Rahmen ein.

Bsp.:

```
div#beispiel {
            padding: 50px;
            width: 300px;}
```

Gemäß CSS 1 und 2 ergibt dies eine Gesamtbreite von 400 Pixel, während dieselbe Regel mit der Deklaration box-sizing: border-box; in einer Breite von 300 Pixel resultiert, wobei dem Elementinhalt aufgrund des horizontalen Gesamtinnenabstands von 100 Pixel nur noch 200 Pixel zustehen würden.

Werte berechnen mit „calc“

Wenn die Funktion es in angedachter Form in die Spezifikation schafft, wird sie es ermöglichen, Werte einer Eigenschaft wie folgt zu berechnen (hier bei drei angenommenen spalte-Containern):

```
div.spalte {
  float: left;
  margin: 1em;
  width: calc(100%/3 – 2*1em); }
```

Mehrere Hintergrundbilder mit „background-image“

Eine nützliche Veränderung erfährt wahrscheinlich die bereits bekannte background-image-Eigenschaft, die mit der neuen Spezifikation mehrere Hintergrundbilder zulässt. In der einfachsten Form sieht diese vor, dass bei Angabe von zwei Hintergrundbildern das erste über dem zweiten angezeigt wird:

```
html { background-image: url(bild-1.gif), url(bild-2.gif); }
```

In Kombination mit anderen background-Eigenschaften (wie background-position, background-repeat, aber auch dem neuen background-size, bei dem Hintergrundbilder neu skaliert werden können) ergeben sich einige neue Möglichkeiten für Designer und Entwickler.

Transparenz mit „opacity“

Die opacity-Eigenschaft macht offiziell, was abseits der Spezifikation bereits in proprietärer Form vom Internet Explorer (via filter) und Gecko-User-Agents (via -moz-opacity) angeboten wird: Die Änderung der Deckkraft eines Elements. Bil-

der, die nur mit halber Deckkraft dargestellt werden sollen, würden mit der neuen Eigenschaft wie folgt bedacht werden:

```
img {opacity: .5;}
```

Abgerundete Ecken mit „border-radius“

Eine oft gewünschte Formatierungsmöglichkeit stellen abgerundete Ecken dar. Ein zwei Pixel breiter schwarzer Rahmen, der in allen Ecken mit fünf Pixeln Radius abgerundet wird, könnte dann einfach so erzielt werden:

```
div {
  border: 2px solid black;
  border-radius: 5px;  }
```

Neue Tabs mit „target“

Neue Browser bieten heute oft die Möglichkeit des sog. Tabbed-Browsing. CSS 3 wird nun die Gelegenheit bieten Links in neuen Tabs zu öffnen.

```
a {target: tab}
```

Neue Selektoren

Fest steht bereits, dass CSS3 einige neue Selektoren mitbringt. Aus Gründen der Übersichtlichkeit wird an dieser Stelle auf die vollständige Referenz des W3C verwiesen: http://www.w3.org/TR/css3-selectors/#selectors

## 5. Darstellung von XML mit CSS

Zu den Aufgaben von XML gehört es nicht HTML zu ersetzen.[9] Es ist aber in bestimmten Fällen sinnvoll, eine graphische Fassung einer XML-Datei zu haben. In der folgenden Abbildung werden die Möglichkeiten zur Ausgabe von XML dargestellt:

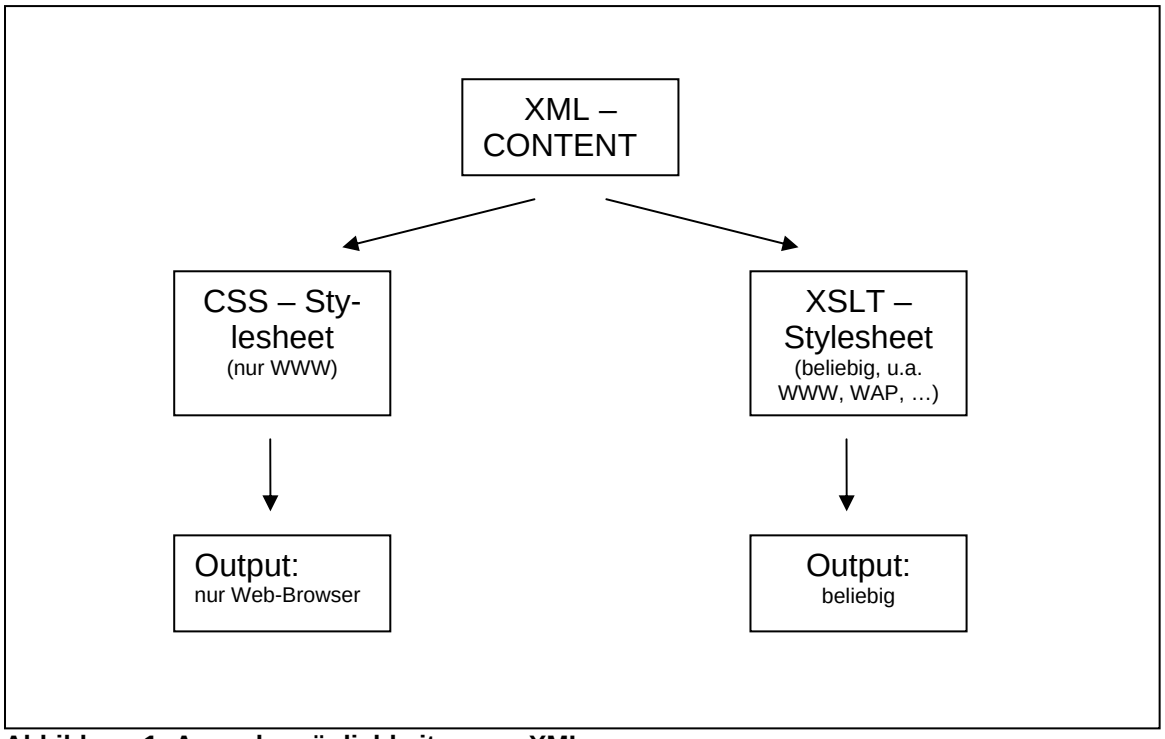



**Abbildung 1: Ausgabemöglichkeiten von XML**
**Quelle: Seeboerger-Weichselbaum, M; Das Einsteigerseminar, XML; Bonn; 2004; S. 197**

Prinzipiell wird für die XML-Ausgabe XSLT benutzt. XSLT ist ideal zur Ausgabe von XML-Dokumenten in verschiedenen Formaten, wie z.B. in HTML, Textformaten und anderen XML-Dialekten. Die einfachste Art und Weise XML-Dokumente anzuzeigen, sind jedoch die Cascading Style Sheets. Im Unterschied zu XSLT beschränkt sich die Darstellung bei CSS nur auf den Web-Browser. Daher sind die CSS auch wesentlich einfacher zu handhaben. XSLT bietet hingegen weitaus mehr Möglichkeiten als Dokumente anzuzeigen.[10]

Im folgenden Beispiel wird gezeigt, wie eine Stildefinition in ein XML-Dokument eingebunden wird:

[9] Seeboerger-Weichselbaum, M; Das Einsteigerseminar, XML; Bonn; 2004; S. 197
[10] Seeboerger-Weichselbaumm M; Das Einsteigerseminar, XML; a.a.O.; S. 198 f.

### 5.1 Beispiel 1

```
<?xml version="1.0"?>
<?xml-stylesheet href="test.css" type="text/css"?>
<Homepage>
        <Ueberschrift>XML mit CSS ausgeben...</Ueberschrift>
        <Text>Toller Test.</Text>
        <Ueberschrift>Die Standardbrowser stellen dies alles ohne Probleme dar!</Ueberschrift>
</Homepage>
```

<*Homepage*> ist das Wurzelelement und enthält die Tags <*Ueberschrift*> und <*Text*>. Die zweite Zeile enthält die Einbindung der CSS-Datei. Eingeleitet wird dies durch <*?xml-stylesheet*>. Das Attribut *href* gibt den Dateinamen an und über das Attribut *type* wird genau festgelegt, um welches Style Sheet es sich handelt. Die dazugehörige CSS-Datei sieht folgendermaßen aus:[11]

```
Ueberschrift
{
  font-family: Arial;
  font-size: 35pt;
  color: red;
}

Text
{
  font-family: Times;
  font-size: 20pt;
  color: blue;
}
```

Die CSS-Datei sieht wie eine herkömmliche CSS-Datei, wie sie auch in HTML verwendet werden könnte, aus. Es besteht aber ein deutlicher Unterschied, wenn CSS in HTML eingesetzt wird. Dort müssen sog. Stilklassen verwendet werden. In der CSS-Datei für HTML wird ein Name für diese Klasse vergeben und über dass Attribut *class* dem HTML-Tag dieser Klasse zugewiesen. Mit XML ist es einfacher. Für jeden Tag wird eine eigene Stilklasse geschrieben. Der Browser erkennt dann automatisch, dass die Stilklasse *Überschrift* für den Tag <*Überschrift*> verwendet werden soll. Genauso funktioniert das bei <*Text*>. Hierfür findet sich in der CSS-Datei die definierte Stilklasse *Text*. Bei der Definition der Stilklassen ist jedoch zu beachten, dass die Schreibweise mit der Schreibweise im XML-Dokument übereinstimmen muss. Auch die Groß- und Kleinschreibung ist hiervon nicht ausgeschlossen. Darüber hinaus dürfen keine Sonderzeichen wie

[11] Seeboerger-Weichselbaum, M; Das Einsteigerseminar, XML; a.a.O.; S. 199

z.B. Ä ,* , & , : , oder # verwendet werden. Ist für untergeordnete Tags keine eigene Stilklasse definiert, erben sie die Stildefinition ihres Wurzelelementes. Die konkreten Eigenschaften einer CSS-Stilklasse werden in geschweiften Klammern geschrieben. Dann werden die Formatierungseigenschaften aufgelistet und voneinander durch Semikolon getrennt. Es können beliebig viele Eigenschaften untereinander aufgelistet werden, auch die Reihenfolge kann beliebig gewählt werden.[12]

Die folgende Abbildung zeigt das Ergebnis des ersten Beispiels zur Darstellung von XML mit CSS:

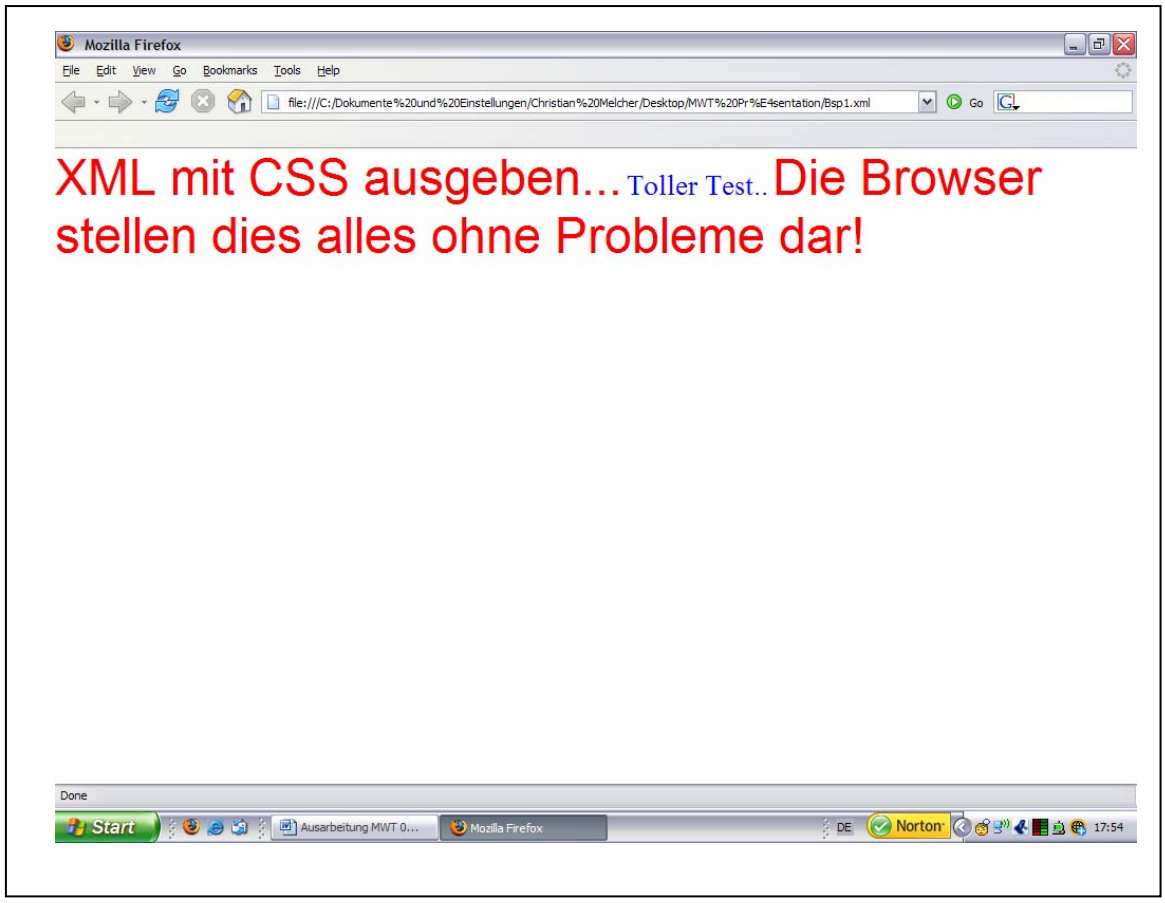



**Abbildung 2: Ausgabe des ersten Beispiels**

Durch die Möglichkeit von CSS direkt auf die in XML verwendeten Tags zuzugreifen, ergeben sich auch flexiblere Möglichkeiten beim Einsatz von Hintergrundfarben und –grafiken. In XML kann man über CSS für best. Bereiche eine andere Hintergrundfarbe definieren. Hierzu weist man dem jeweiligen Tag über die Eigenschaft *background-color* die gewünschte Farbe zu. Das funktioniert aufgrund

[12] Seeboerger-Weichselbaum, M; Das Einsteigerseminar, XML; a.a.O.; S. 200

der möglichen Vererbung der Stildefinitionen. Bei HTML in Verbindung mit CSS ist dies für einzelne Bereiche oder Textpassagen nur möglich, wenn diese mit den vorgegebenen Inline-Elementen wie z.B. span, markiert werden.
Das Einbinden von Hintergrundgrafiken funktioniert genauso. Auch hier kann für jeden Tag, in einer eigenen Stilklasse, ein Hintergrundbild eingefügt werden. Hierfür wird die CSS-Eigenschaft *background-image* verwendet. Allerdings können wie in HTML nur die Grafikformate GIF, JPG und PNG verwendet werden.[13]
Als Beispiel für die Einbindung von Grafiken und Hintergrundfarben dient folgendes XML-Dokument:

### 5.2 Beispiel 2

```
<?xml version="1.0"?>
<?xml-stylesheet href="xmldoc02.css" type="Text/css"?>
<Homepage>
        <Ueberschrift>Multimedia- und Webtechnologien </Ueberschrift>
        <Thema>XML und CSS 2; 3 </Thema>
        <Von>Waldemar Herber; Christian Melcher; Malte Wattenberg</Von>
        <Wunsch>Wir bedanken uns für die Aufmerksamkeit!</Wunsch>
        <Wunsch2>Viel Glück bei den anstehenden Prüfungen!</Wunsch2>
</Homepage>
```

Das dazugehörige CSS-File:

```
Homepage
{
background-image: url(strand.jpg);
}

Ueberschrift
{
font-family: Arial;
font-size: 40pt;
color: blue;
margin-left: 110px;
background-color: papayawhip;
}


Thema
{
font-family: Arial;
font-size: 30pt;
color: red;
display: block;
margin-left:300px;
margin-top:100px;
font-weight: bold
}
```

[13] Seeboerger-Weichselbaum, M; Das Einsteigerseminar, XML; a.a.O.; S. 205 ff.

```
Von
{
font-family: Arial;
font-size: 20pt;
color: silver;
display: block;
margin-left:150px;
margin-top:100px;
}

Wunsch
{
font-family: Arial;
font-size: 14pt;
color: yellow;
display: block;
margin-left: 300px;
margin-top: 100px;
}

Wunsch2
{
font-family: Arial;
font-size: 14pt;
color: black;
margin-left: 300px;
background-color: white;
}
```

Diese Abbildung zeigt das Ergebnis der CSS-Formatierung des XML-Dokuments:


**Abbildung 3: Ausgabe des zweiten Beispiels**

## 6. RSS

### 6.1 Definition

RSS 2.0 (Really Simple Syndication) ist für den plattformunabhängigen Austausch von Informationen entwickelt worden. RSS 2.0 wurde 2002 entwickelt, ist aber umstritten, da es zu seinen Vorgängerversionen keine vollständige Abwärtskompatibilität besitzt. Es wird häufig genutzt um Inhalte einer Webseite komplett oder teilweise zu abonnieren. Die aktuellen Inhalte werden automatisch auf den Computer (oder auf andere Endgeräte) des Abonnenten geladen, sobald sie veröffentlicht werden. So bekommt der Abonnent immer die neuesten Informationen automatisch und bequem geliefert. Das Abonnement ist aber nicht nur auf Texte beschränkt. Es besteht auch die Möglichkeit Audio- und Videoinhalte per RSS zu

abonnieren. Die Bereitstellung solcher RSS-Formate wird auch RSS-Feed genannt. Die RSS-Technologie hat sich hauptsächlich durch die Anwendung in Blogs durchgesetzt, da viele Blogger relativ früh RSS-Feeds für ihre Artikel angeboten haben. Dadurch dass RSS ein standardisiertes Format hat, lassen sich auch fremde RSS-Feeds in die eigene Webseite integrieren. Oftmals werden RSS-Dateien von Webseiten, die regelmäßig Artikel publizieren automatisch generiert. Mit einem RSS-Reader können dann aktuelle Informationen auf dem PC angezeigt werden. Von der technischen Seite betrachtet gehört RSS zu den auf XML basierenden Dateiformaten.[14]

## 6.2 Aufbau

Am folgenden XML-Quelltext, soll der Aufbau einer RSS-Datei gezeigt werden:

```
<?xml version="1.0" encoding="utf-8"?>

<rss version="2.0">
  <channel>

    <title>Titel des Feeds</title>
    <link>Adresse der Webpräsenz</link>
    <description>Kurze Beschreibung des Feeds</description>
    <language>de-de</language>
    <copyright>urheberrechtliche Informationen</copyright>
    <pubDate>Datum der Erstellung</pubDate>

     <image>
      <url>Soll eine Grafik, z.B. ein Logo, eingebunden werden,
          hier dessen Adresse eintragen</url>
      <title>Titel des Bildes</title>
      <link>Adresse, mit der das Bild verknüpft werden soll, z. B.
           die Adresse der Webpräsenz des Herausgebers</link>
    </image>

    <item>
      <title>Titel des ersten Artikels</title>
      <description>Eine kurze Zusammenfassung des Artikels</description>
      <link>Adresse zur Gesamtansicht des Artikels</link>
      <author>Autor des Artikels</author>
    </item>
```

[14] URL: http://www.wikipedia.org/wiki/RSS; 09.01.2006

```
<item>
    <title>Titel des zweiten Artikels</title>
    <description>
     <![CDATA[
     <h1>Hier kann auch der vollständige HTML-Inhalt
       des Artikels stehen</h1>
     <p>…</p>
     ]]>
    </description>
    <author>Autor des Artikels</author>
  </item>

</channel>
</rss>
```

Der Kopf des RSS-Feeds ist immer gleich aufgebaut. Zu ihm gehören die Pflichtelemente *<channel>, <title>*, *<link>* und *<description>*, die im obigen Beispiel schon erklärt wurden. Danach folgen die verschiedenen Beiträge, die durch das Element *<item>* gekennzeichnet werden. Dort müssen mindestens Überschrift, Beschreibung und Link angegeben werden. Es können aber z.B. auch HTML-Elemente und damit Links, Listen, Textauszeichnungen oder Grafiken enthalten sein. Beendet wird die Datei mit den Tags </channel> und </rss>.

### 6.3 Darstellung mit CSS

Die Darstellung von RSS mit CSS unterscheidet sich nicht von der Darstellung bei XML. Es muss lediglich wieder das Stylesheet auf dieselbe Art und Weise mit dem XML-Datei verknüpft werden, z.B.:

```
<?xml-stylesheet href="rssfeed.css" type="Text/css"?>
```

In der folgenden Abbildung wird eine einfache Darstellung eines RSS-Feeds gezeigt:

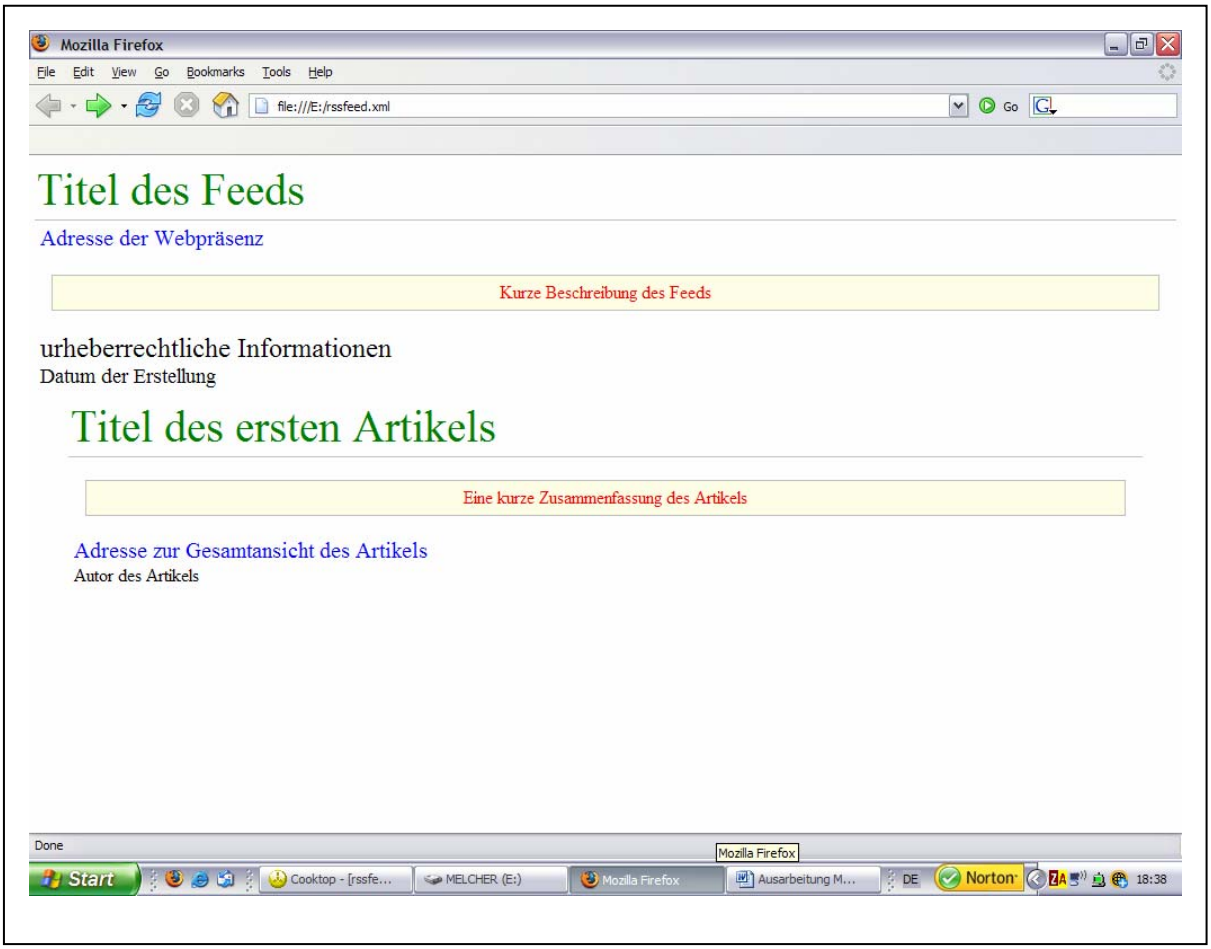



**Abbildung 4: Darstellung RSS mit CSS**

## 7. ATOM

### 7.1 Definition

ATOM ist ein zu RSS konkurrierendes Format, dass ebenfalls auf XML beruht. ATOM und RSS sind daher nicht kompatibel, aber eine beidseitige Konvertierung ist möglich. ATOM entstand aus dem Problem, dass verschiedene RSS-Formate existieren. Die Vorteile dieser Formate wurden in ATOM zusammengefasst. Die Entwickler, von denen die Mehrzahl Blogger sind, haben ATOM so gestaltet, dass es den speziellen Bedürfnissen von Blogs und Nachrichtenseiten besser gerecht wird. ATOM 1.0 ist seit Dezember 2005 offizieller Internetstandard.[15]

### 7.2 Aufbau

Anhand des folgenden Beispiels für Atom, kann man die Unterschiede zum RSS-Format erkennen.

```
<?xml version="1.0" encoding="utf-8"?>

<?xml-stylesheet href="test4.css" type="Text/css"?>

<feed xmlns="http://www.w3.org/2005/Atom" xml:lang="de">

  <title>Name des Weblogs</title>
  <link href="http://example.org/"/>
  <updated>2003-12-13T18:30:02Z</updated>
  <author>
    <name>Autor des Weblogs</name>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

  <entry>
    <title>Titel des Weblog-Eintrags</title>
    <link href="http://example.org/2003/12/13/atom-beispiel"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2003-12-13T18:30:02Z</updated>
    <summary>Text des Weblog-Eintrags</summary>
  </entry>

</feed>
```

[15] URL: http://www.wikipedia.org/wiki/Atom_%28XML-Format%29; 09.01.2006

## Literaturverzeichnis


Ammelburger, Dirk;
XML - Grundlagen der Sprache und Anwendung in der Praxis;
München/ Wien; 2004

Balzert, Helmut;
HTML, XHTML & CSS für Einsteiger; Dortmund; 2003

Michel, Thomas;
XML kompakt; München/ Wien; 1999

Kesper, Björn;
HTML, XHTML & CSS; Düsseldorf; 2004

Schmitt, Chr.;
CSS Kochbuch; Köln; 2005

Seeboerger-Weichselbaum, M.;
Das Einsteigerseminar, XML; 4. Auflage; Bonn; 2004

Vonhoegen, Helmut;
Einstieg in XML; 3. Auflage; Bonn; 2005

Wikipedia;
URL: http://www.wikipedia.org/wiki/Atom_%28XML-Format%29;
09.01.2006
URL: http://www.wikipedia.org/wiki/RSS; 09.01.2006