

2.1 XML/Einführung

2.1.1 Wozu ein weiteres Web-Format?

HTML funktioniert gut und deckt viele Anwendungsbereiche ab. Wozu braucht man dann noch ein anderes Web-Format mit dem Namen XML?

HTML und XML haben völlig unterschiedliche Zwecke:

- HTML beschreibt ein Seiten-Layout im Web
- XML beschreibt die Struktur von Daten

HTML hat also genau einen Zweck; XML dagegen ist eine universell einsetzbare Sprache.

Andererseits sind HTML und XML verwandt: HTML ist ein mögliches Format innerhalb der XML-Familie.

Andere mögliche XML-Formate sind MathML und SVG. An diesen beiden Beispielen sieht man auch, dass man verschiedene XML-Formate in einer Datei zusammenfassen kann.

Außerdem erlaubt XML, dass man sich für einen eigenen Zweck ein eigenes Format innerhalb von XML zusammenbauen kann. Und eine Datei mit diesem Format kann man sich nun z. B. in einem Browser anzeigen lassen. Man kann sie aber auch benutzen, um damit ein Programm zu konfigurieren oder für jeden anderen Zweck.

2.1.2 Hello, XML-World!

Eine einfache XML-Datei sieht so zum Beispiel so aus (hallo1.xml):

```

1 <?xml version="1.0" standalone="yes"?>
2
3 <hallo>
4   Hello World!
5 </hallo>
```

Lädt man diese Datei in den Browser, erhält man die Ansicht wie in Abbildung 1. In der ersten

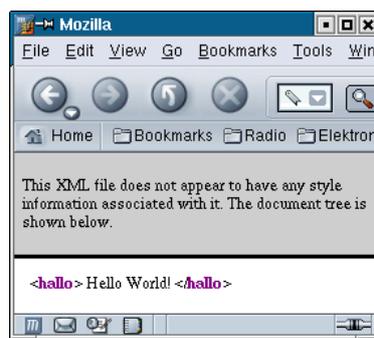


Abbildung 1: hallo1.xml

Zeile sieht man einen Verarbeitungs-Befehl in XML. Er hat das Format:

```
1 <? ... ?>
```

Tags dagegen sehen in XML aus wie in HTML:

```
1 <hallo>Saluto vos!</hallo>
```

Ein XML-Element besteht *immer* aus dem Start-Tag (eventuell mit Attributen), Inhalt (eventuell leer) und Ende-Tag. <hallo> ist das Start-Tag, </hallo> ist das Ende-Tag. Alles, was

zwischen `<hallo>` und `</hallo>` steht, ist der Inhalt des Elements. Dieser Inhalt kann — wie hier — aus beliebig viel normalem Text bestehen, kann aber auch leer sein. Für ein leeres Element gibt es noch die Abkürzung `<hallo />`.

Der Inhalt kann aber auch wiederum andere XML-Elemente enthalten, die wiederum Text oder weitere XML-Elemente enthalten können. Man erhält also eine Baumstruktur mit XML-Elementen als Zweigen und Textstücken als Blättern.

XML hat mit HTML also viele Gemeinsamkeiten:

- Es gibt ein Wurzel-Element. In HTML ist das `<html>`.
- Elemente können verschachtelt werden.
- Tags und Attribute funktionieren wie in HTML.

XML hat aber auch Besonderheiten gegenüber HTML:

- In XML darf man eigene Tags nach Belieben definieren ¹.
- XML ist streng *case-sensitive*, unterscheidet also Groß- und Kleinschreibung: `<hallo>` ist nicht `<Hallo>` und nicht `<HALLO>`!
- Zu jedem Start-Tag muss ein Ende-Tag vorhanden sein!
- Attribute sind erlaubt, müssen aber zwingend in Anführungszeichen stehen: `<hallo sprache="EN">hello!</hallo>`

Wenn HTML nur ein Anwendungsfall von XML ist, woher kommen dann die Unterschiede?

- HTML war ursprünglich ein Anwendungsfall von SGML, einem Vorläufer von XML. SGML ist in der Syntax etwas laxer und braucht dafür viel aufwändigere Werkzeuge
- Frühe HTML-Seiten wurden fast immer von Hand geschrieben, und dabei konnten schon einmal Syntax-Fehler passieren. Damit diese Seiten trotzdem dargestellt wurden, haben die Browser fast alles dargestellt, was auch nur teilweise nach XML, SGML oder HTML aussah

Wenn man alle Besonderheiten beachtet, erhält man ein *wohlgeformtes* XML-Dokument (*well-formed document*). XML-Werkzeuge (z.B. Browser) sind dazu verpflichtet, nur Dokumente zu bearbeiten, die wohlgeformt sind. Andernfalls müssen sie (eigentlich) die Bearbeitung abbrechen.

Wenn das wohlgeformte Dokument dann zusätzlich noch zu einem festgelegten Anwendungsfall (z. B. SVG, MathML oder HTML) passt, dann bekommt es zusätzlich die Eigenschaft *gültig* (*valid*).

Die Gültigkeit bezieht sich jedoch nur auf diesen einen Anwendungsfall. Dieser wurde vorher in einem speziellen Dokument (entweder einem XML-Schema oder einer DTD, siehe später) genauestens festgelegt.

2.1.3 Schöner brausen: XML und CSS

Cascading style sheets (CSS) kann man nicht nur zur Verschönerung der Standardformate von HTML, sondern auch zur grundlegenden Formatierung von XML benutzen. Als Beispiel dient hier die CSS-Datei `hallo2.css` für `hallo2.xml`:

```

1 hallo
2 {
3     background-color: yellow;
4     display:         block;
5     font-size:       32pt;
6     font-weight:     bold;
7     text-align:      center;
8 }
```

¹Die Definition dafür, welches Tag wann erlaubt ist (und mit welchen Attributen) gehört in einen eigenen Abschnitt oder eine eigene Datei, die entweder DTD (*document type definition*) oder XML-Schema heißt.

Nun muss `hallo1.xml` in `hallo2.xml` geändert werden, damit `hallo2.css` eingebunden wird:

```
1 <?xml version="1.0" ?>
2 <?xml-stylesheet type="text/css" href="hallo2.css" ?>
3
4 <hallo>
5     Hello World!
6 </hallo>
```

Hier wurde also ein weiterer Verarbeitungsbefehl `<? . . . ?>` eingefügt. Das Attribut `standalone="yes"` dagegen musste entfernt werden. Das Ergebnis sieht aus wie in Abbildung 2.



Abbildung 2: hallo2.xml mit CSS