

Id

ADOBE® INDESIGN® CS3

Version 5.0

<xml/>

Executing startup services...



© 1990-2007 Adobe Systems Incorporated. All rights reserved.
See the patent and legal notice in the about box.

Gregor Fellenz

Mediengestalter für Digital- und Printmedien
Studium Druck- und Medientechnologie, HdM Stuttgart
pagina GmbH, Tübingen

gregor.fellenz@indesignblog.com

Übersicht

1. Vorbereitung und Grundlagen

- XML Kurzübersicht
- XSLT (eXtensible Stylesheet Transformation Language)

2. InDesign und XML

- Wozu und für wen eigentlich?
- Systematisches Arbeiten mit InDesign
- Integrierte XML-Funktionalität nutzen
- XML in InDesign bearbeiten
- XML-Export

3. InDesign Programmierung

- Erweiterte Möglichkeiten durch Skripting
- XML Rules

XML Einführung

Warum überhaupt XML

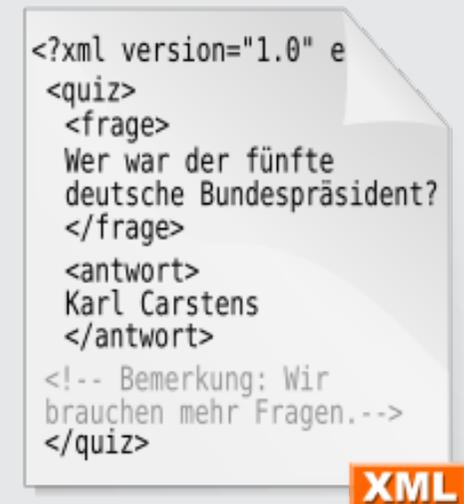
- **Datenchaos**
Formate, Datenprobleme, Konvertierungsprobleme
- Keine **Wiederverwertbarkeit**
Langzeitarchivierbarkeit, Kompatibilität
- **Mehrfachverwertung** der Daten
Satz, Internet, DVD Nachschlagewerke, Mobile Applications
- **Automatisierte Produktion**

Extensible Markup Language

Standardisiertes Dateiformat

- Austausch von Dokumenten, Definition von Datenstrukturen
- Sprache zur Beschreibung von **Dokumentstrukturen** bzw. eine Metasprache, die eine abstrakte Syntax für Auszeichnungssprachen definiert
- **Dokumentzentriert** vs. **Datenzentriert**
- XML ist **keine Datenbank**

```
<?xml version="1.0" e
<quiz>
  <frage>
    Wer war der fünfte
    deutsche Bundespräsident?
  </frage>
  <antwort>
    Karl Carstens
  </antwort>
  <!-- Bemerkung: Wir
  brauchen mehr Fragen.-->
</quiz>
```



Auch im Bereich Publishing?

Mit **Strukturierung** wurde schon immer das **Erfassen des Informationsgehaltes** erleichtert.

Umgesetzt durch **Formatierung** wird indirekt die Struktur vom Leser wahrgenommen.

Fachbegriffe

▪ Elemente

Im Dokument mittels

- passendem Paar aus **Start-Tag** (<Tag-Name>) und **End-Tag** (</Tag-Name>) oder
- mit einem **Empty-Element**-Tag (<Tag-Name />) ausgezeichnet.

▪ Attribut

Für Zusatz-Informationen zu Elementen (parameter="wert"), Meta-Informationen.

▪ Verarbeitungsanweisungen (<?Ziel-Name Parameter ?>, engl. Processing Instruction)

▪ Kommentare (<!-- Kommentar-Text -->)

▪ Namensräume (<html xmlns="http://www.w3.org/1999/xhtml" xmlns:svg="http://www.w3.org/2000/svg">) Elemente unterscheiden.

Ein XML-Dokument muss genau ein Element (**Root**) auf der obersten Ebene enthalten. Unterhalb von diesem Dokumentelement können weitere Elemente **verschachtelt** werden.

- Eine **Document Type Definition** (DTD) ist ein Regelwerk für XML-Dateien. Reihenfolge der Elemente und welche überhaupt vorkommen dürfen.
- XML-Dokumente sind idealerweise in **UTF-8** codiert.

- Ein **XML-Dokument** (DocBook Artikel)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.4//EN"
    "http://www.docbook.org/xml/4.4/docbookx.dtd">
<article>
  <title>Titel des Artikels</title>
  <sect1>
    <title>Erste Hirarchieebene</title>
    <para>Absätze sind Blocklevel-Elemente und beinhalten Text. Textinhalt kann mit
<emphasis role="bold">Inline-Elementen</emphasis> ausgezeichnet werden. Interessant
sind auch <link linkend="id22">Verweise</link> auf andere Textstellen.</para>
    <table id="id22"><!--Auf die Tabelle verweist der Link im vorigen Absatz-->
      <tgroup cols="2">
        <tbody>
          <row><entry>Spalte 1</entry><entry>Spalte 2</entry></row>
        </tbody>
      </tgroup>
    </table>
  </sect1>
</article>
```

XML Verstehen

- Die Dateien befinden sich im Ordner `01_xml`
 - Öffnen Sie die DocBook-Datei `unicode.xml` in einem XML-Editor
 - Öffnen Sie die Datei `unicode.html` in einem Web-Browser
- ! Vergleichen Sie Die DocBook-Struktur und die Darstellung als HTML
- ! Beachten Sie die Umsetzung der Hierarchie (`section/title` Elemente)

Quellen:

Unicode Wikipedia Artikel: http://de.wikipedia.org/wiki/Unicode#cite_note-0

Wiki2DocBook Converter: <http://toolserver.org/~magnus/wiki2xml/w2x.php>

Extensible Stylesheet Transformation Language

XSLT ist eine **Programmiersprache** zur Transformation von XML-Dokumenten, Teil der Extensible Stylesheet Language (XSL).

- **XSLT-Stylesheets** sind nach den Regeln des XML-Standards aufgebaut.
- XSLT baut auf der logischen **Baumstruktur** eines XML-Dokumentes auf.
- **XSLT-Prozessoren** (Saxon, MSXSL, ...) verarbeiten XML-Dokumente anhand der im XSLT-Stylesheet definierten Regeln zur gewünschten Ausgabestruktur.
- Missverständnis: **CSS als Alternative**. CSS ist lediglich eine Formatierungssprache.

Notwendigkeit von XML-Transformationen

- Vielfältige XML-Datenformate von verschiedensten Programmen verarbeiten.
- **Datenanalyse** und **Dateninterpretation**
- **Split/Join** von Strukturen zur Weiterverarbeitung
- Umstrukturierung
- ...

Einfache Datentransformationen

- **Aufbau eine Stylesheets:** Reines XML, XSL-Namespace, XSLT Versionen.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

- **Templates** für das Matching von Strukturen <xsl:template...
- **XPath** adressiert die Knoten im Quelldokument match="abs"
- Im Element werden Anweisungen zur Verarbeitung gegeben

```
<xsl:template match="abs">
  <xsl:apply-templates/>
  <xsl:text>&#x0a;</xsl:text>
</xsl:template>
```

- Mit <xsl:apply-templates/> werden die Templates für die **Kind-Elemente** aufgerufen (inkl. **Built-in-Templates** [Text kopieren, Rekursiv weitersuchen])

- **Whitespace Handling** Leerzeichen, Zeilenschaltungen und Tabulatoren zwischen Elementen entfernen: `<xsl:strip-space elements="*" />`

- Text und Elemente **hinzufügen**

```
<xsl:template match="telefon">
  <p><xsl:text>Telefonnummer:</xsl:text><xsl:apply-templates/></p>
</xsl:template>
```

- Aufzählung selber erstellen: XPath-Funktion `position()`

```
<xsl:template match="item">
  <xsl:element name="liststep">
    <xsl:value-of select="position()" />
    <xsl:text>&#x09;</xsl:text>
    <xsl:apply-templates/>
  </xsl:element>
  <xsl:text>&#x0a;</xsl:text>
</xsl:template>
</xsl:stylesheet>
```

Eine einfache XSL Transformation erstellen

- Die Dateien befinden sich im Ordner `02_xslt`
 - Öffnen Sie die DocBook-Datei `unicode.xml` in einem XML-Editor (Dies ist eine gekürzte Version des Wikipedia Artikels)
 - Öffnen Sie die Datei `convert.xslt` im XML-Editor
- ! Vereinfachen Sie die Struktur indem Sie die Daten auf einen Text-Container als Root-Element und Absätze, Bilder und verschiedene Überschriftgrößen als Block-Elemente reduzieren.
- ! Entfernen Sie mittels XSLT alle Elemente die nicht für die Druckausgabe notwendig sind (Links), entfernen sie weiterhin alle Fußnoten.
- ! Zusatz: Nummerieren Sie die Listen (`itemizedlist`) in der Datei `standard.xml`

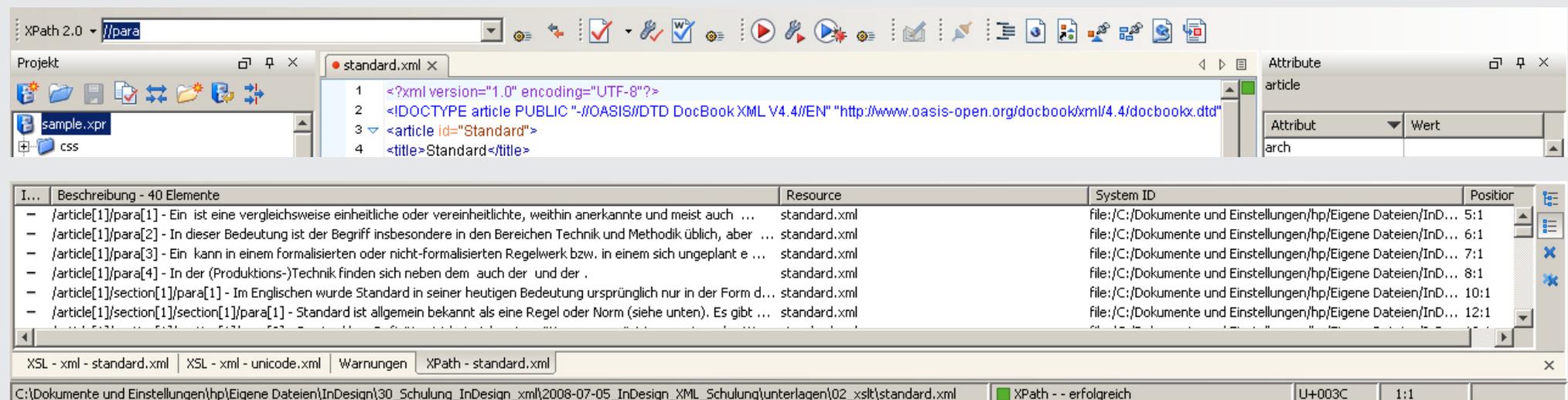
Exkurs XPath

XML Path Language (XPath) ist eine Abfragesprache um Teile eines XML-Dokumentes zu adressieren.

- Grundlage für **XSLT** – wird für das Template Matching verwendet
- Ein XPath-Ausdruck setzt sich zusammen aus:
 - einem oder mehreren **Lokalisierungsschritten** (mit dem /-Zeichen getrennt)
 - optional gefolgt von einem oder mehreren **Prädikaten**.
- Ein Lokalisierungsschritt besteht aus **Achse** und **Knotentest**
- Beispiele für XPath Ausdrücke
 - /dok/kap selektiert alle kap-Elemente innerhalb eines dok Elements
 - /dok/kap[1] selektiert das erste kap-Element innerhalb eines dok Elements
 - //pa selektiert alle pa-Elemente auf allen Ebenen
 - //kap[@freigabe="true"] selektiert alle kap-Elemente deren Attribut freigabe true ist
- Verwendung bei den **XML-Rules** von InDesign.

XPath verstehen und anwenden

- Die Dateien befinden sich im Ordner `02_xslt`
- Öffnen Sie die DocBook-Datei `standard.xml` in einem XML-Editor (Dies ist eine gekürzte Version des Wikipedia Artikels)
- ! Selektieren Sie im XML-Editor mit Hilfe eine XPath Ausdrucks alle para-Elemente.
- ! Selektieren Sie das erste Listenelement (`listitem`) aller Listen.
- ! Ermitteln Sie die Anzahl der Absätze im Dokument.



The screenshot shows an XML editor interface. The top toolbar contains various icons for editing and navigation. The main window displays the XML code for `standard.xml`:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.4//EN" "http://www.oasis-open.org/docbook/xml/4.4/docbookx.dtd"
3 <article id="Standard">
4 <title>Standard</title>
```

The right sidebar shows the 'Attribute' panel for the selected `article` element, with a table listing attributes:

Attribut	Wert
arch	

Below the XML editor is a table with 4 columns: 'I...', 'Beschreibung - 40 Elemente', 'Resource', 'System ID', and 'Positor'. It lists XML elements and their locations in the document:

I...	Beschreibung - 40 Elemente	Resource	System ID	Positor
-	/article[1]/para[1] - Ein ist eine vergleichsweise einheitliche oder vereinheitlichte, weithin anerkannte und meist auch ...	standard.xml	file:/C:/Dokumente und Einstellungen/hp/Eigene Dateien/InD...	5:1
-	/article[1]/para[2] - In dieser Bedeutung ist der Begriff insbesondere in den Bereichen Technik und Methodik üblich, aber ...	standard.xml	file:/C:/Dokumente und Einstellungen/hp/Eigene Dateien/InD...	6:1
-	/article[1]/para[3] - Ein kann in einem formalisierten oder nicht-formalisierten Regelwerk bzw. in einem sich ungeplant e ...	standard.xml	file:/C:/Dokumente und Einstellungen/hp/Eigene Dateien/InD...	7:1
-	/article[1]/para[4] - In der (Produktions-)Technik finden sich neben dem auch der und der .	standard.xml	file:/C:/Dokumente und Einstellungen/hp/Eigene Dateien/InD...	8:1
-	/article[1]/section[1]/para[1] - Im Englischen wurde Standard in seiner heutigen Bedeutung ursprünglich nur in der Form d...	standard.xml	file:/C:/Dokumente und Einstellungen/hp/Eigene Dateien/InD...	10:1
-	/article[1]/section[1]/section[1]/para[1] - Standard ist allgemein bekannt als eine Regel oder Norm (siehe unten). Es gibt ...	standard.xml	file:/C:/Dokumente und Einstellungen/hp/Eigene Dateien/InD...	12:1

The bottom status bar shows the current file path: `C:\Dokumente und Einstellungen\hp\Eigene Dateien\InDesign\30_Schulung_InDesign_xml\2008-07-05_InDesign_XML_Schulung\unterlagen\02_xslt\standard.xml` and the status: `XPath - - erfolgreich`.

XSLTs für InDesign

- Abbildbare **Struktur erstellen** (Text in Attributen ist nicht sichtbar).
- **Whitespace Handling: Zeilenwechsel** einfügen/entfernen.
- Text wegschalten/hinzufügen. Z.B. Adressbuch, Veranstaltungskalender ...
- Aufzählungen, Nummerierung oder Listen erstellen.
- **Tabellen** in darstellbares Tabellenformat umwandeln.

XSLT kann noch viel mehr ...

- Transformationen zwischen Datenformaten.
- Datenanalyse, Parsing, Berechnungen ...
- Erstellung von Strukturen.
- Semantische Prüfungen selber programmieren

InDesign und XML

*InDesign ist eine von **vielen Applikationen** die XML unterstützt und verarbeitet.*

- **Richtig erstellte** InDesign-Dokumente können als XML weiterverwendet werden.
- Nach **entsprechender Anpassung** kann InDesign beliebige XML-Dokumente formatieren.
- InDesign ist **nicht** als **XML-Editor** konzipiert.

Wozu und für wen eigentlich?

- Dokumente **liegen als XML** vor.
- Ähnliche Strukturen in **wiederkehrenden** Aufträgen/Projekten.
- Dokumente sollen **weiterverwendet** werden.
- Daten müssen **als XML abgeliefert** werden.
- Bereitschaft sich mit der **Technologie** auseinanderzusetzen.



Voraussetzungen

- XML muss **wohlgeformt, bzw. valide** sein.
- Struktur muss in InDesign **abbildbar** sein.
 - Attribute vs. Elemente.
 - Strukturen/Kapitel nur eingeschränkt abbildbar.
 - Whitespace-Handling.

Strategische Überlegungen

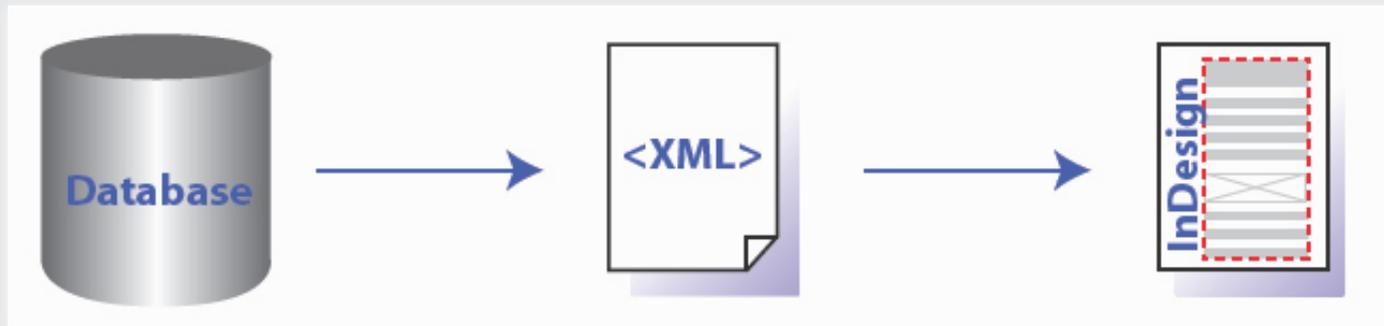
- Know-how im Unternehmen/Dienstleister vorhanden?
- Grad der **Automatisierung**.
- Anforderungen an den XML-Export.
 - Single-Source-Publishing
 - Roundtripping



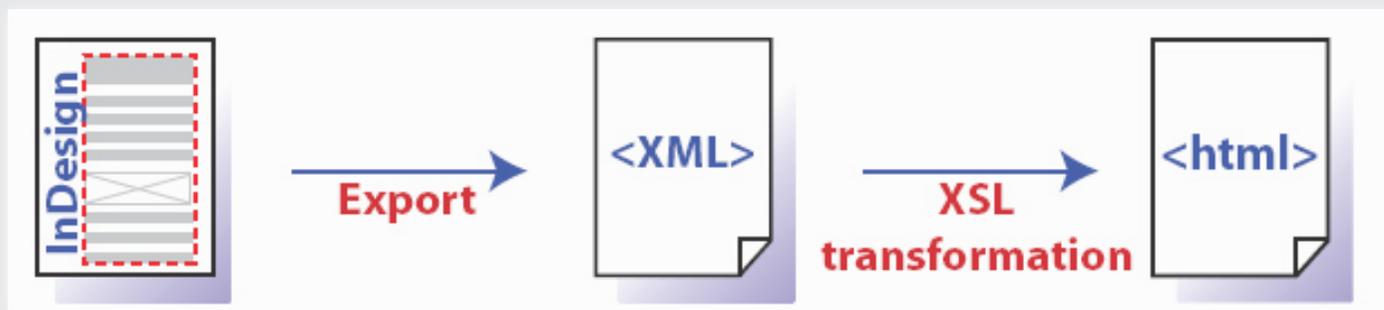
Prinzipielle Aspekte

Möglichkeiten für den Einsatz von XML in InDesign

- XML-Dokumente vorhanden → **Import**



- InDesign Dokumente vorhanden → **Export**



Systematisches Arbeiten mit InDesign

Bei der Bearbeitung von XML-Dokumenten ist der Einsatz von grundlegenden Arbeitstechniken (Absatzformate, Zeichenformate, Tabellenformate [CS3]) unabdingbar!

InDesign Dokumente richtig bearbeiten

▪ Absatzformate

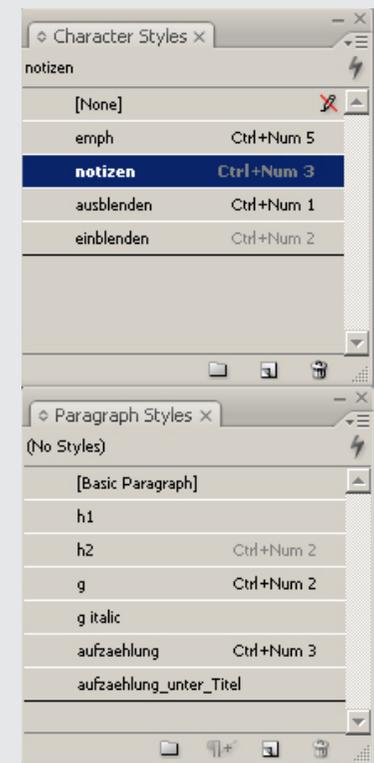
Mit Absatzformaten können Layoutoptionen auf gesamte Absätze angewendet werden.

XML-Mapping für **Blocklevel-Elemente**

▪ Zeichenformate

Textstellen oder einzelne Zeichen können durch Zeichenformate modifiziert werden.

XML-Mapping für **Inline-Elemente**



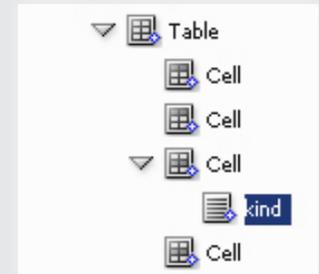
■ Tabellen- und Zellenformate (CS3)

Vollständige Tabellen werden durch Tabellenformate gestaltet.

Mit Zellenformaten können Zellen formatiert werden.

XML-Mapping möglich, aber gleiche Struktur benötigt.

Zellen können erst ab CS3 XML-Kind-Elemente enthalten (Formatierung).



The screenshot shows the 'Table Style Options' dialog box and the 'Table Styles' and 'Cell Styles' panels in Adobe InDesign.

Table Style Options Dialog:

- General:** Style Name: uebersicht
- Style Info:** Based On: [No Table Style]
- Style Settings:** [No Table Style] + Body Region Cell Style: body + Header Region Cell Style: header + Footer Region Cell Style: [None]
- Cell Styles:** Header Rows: header, Footer Rows: [None], Body Rows: body
- Left Column:** [Same as Body Rows]
- Right Column:** [Same as Body Rows]

Table Styles Panel: Shows the 'uebersicht' style selected under the '[Basic Table]' category.

Cell Styles Panel: Shows the '[None]' style selected, with a dropdown menu open showing options: [None], header, body, left.

Systematisch ohne XML

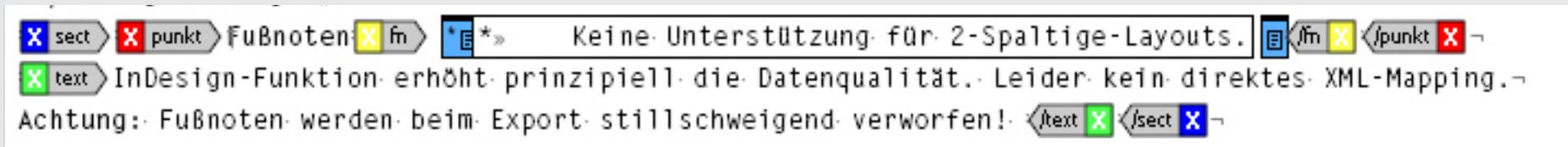
Fußnoten und Verweise können leider noch nicht out-of-the-box innerhalb von XML-Strukturen verwendet werden – hierfür wird eine Anpassung benötigt.

▪ Fußnoten*

InDesign-Funktion erhöht prinzipiell die Datenqualität. Leider **kein** direktes XML-Mapping.

Achtung: Fußnoten werden beim Export stillschweigend verworfen!

Konvertierungsskript möglich.



Wird zu:

```
<sect><punkt>Fußnoten<fn></fn></punkt> <text>InDesign-Funktion erhöht prinzipiell  
die Datenqualität. Leider kein direktes XML-Mapping. Achtung: Fußnoten werden  
beim Export stillschweigend verworfen! </text></sect>
```

* Keine Unterstützung für 2-Spaltige-Layouts.

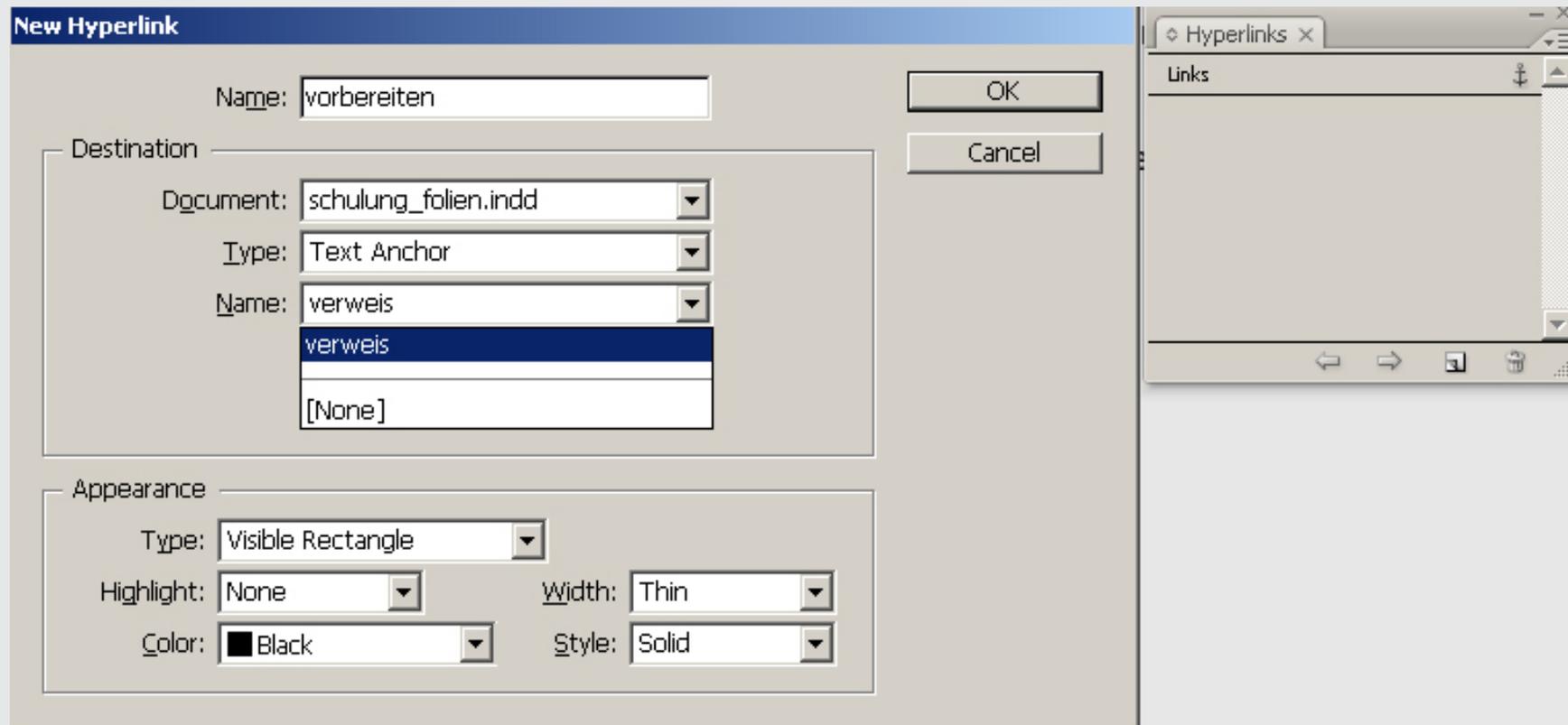
▪ Verweise

Können gesetzt werden, funktionieren aber nur beim PDF-Export.

Verweisziel/Verweis Beziehung – analog zu ID/IDREF

Auflösung muss selber programmiert werden – **Skript vorhanden.**

Kein direktes XML-Mapping, kein Export (aber kein Verlust von Inhalten).



Systematisches arbeiten mit InDesign

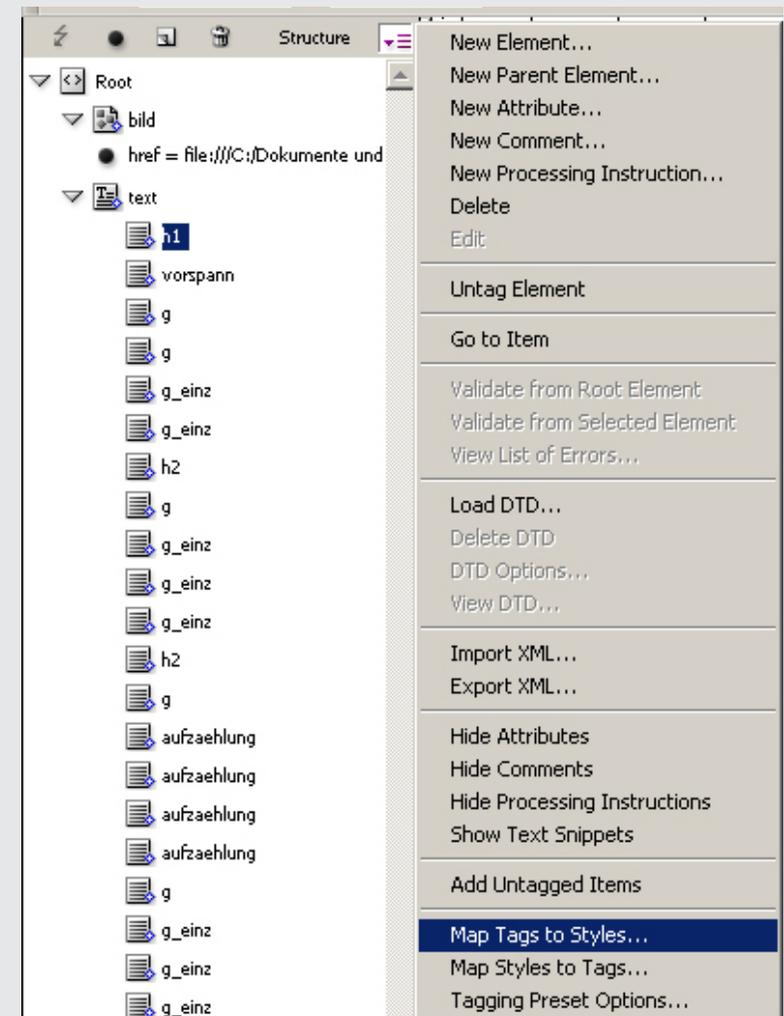
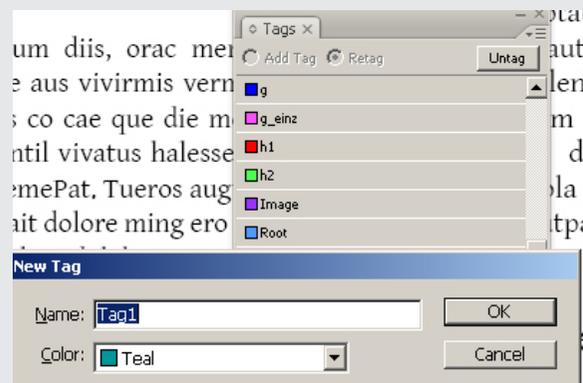
- ! Legen Sie ein neues Dokument in InDesign an.
- ! Erstellen Sie Absatz- und Zeichenformate und zeichnen Sie Text damit aus.
- ! Erstellen Sie einen Verweis auf eine andere Seite und testen Sie ihn (PDF-Export).
- ! Zusatz: Verwenden Sie das Skript `updatecrossrefs.jsx` um die Seitenzahlen der Verweise automatisch zu aktualisieren.
- ! **Demonstration InDesign CS3:** Möglichkeiten mit Tabellenformaten.

Integrierte XML-Funktionalität nutzen

Einfache Möglichkeiten zum Erstellen und Bearbeiten von XML-Dokumenten sind in InDesign vorhanden.

XML-Ansichtsoptionen

- Strukturansicht
- Tag-Marken/Rahmen mit Tags
- Textansicht/Story Editor
- **Tags**
 - **Elementnamen** können auch direkt erstellt und zugewiesen werden.
 - **Attribute** sind nicht so intuitiv.
 - **Unicode-Codepoint** aus der Infopalette



Die XML-Ansichtsoptionen verwenden

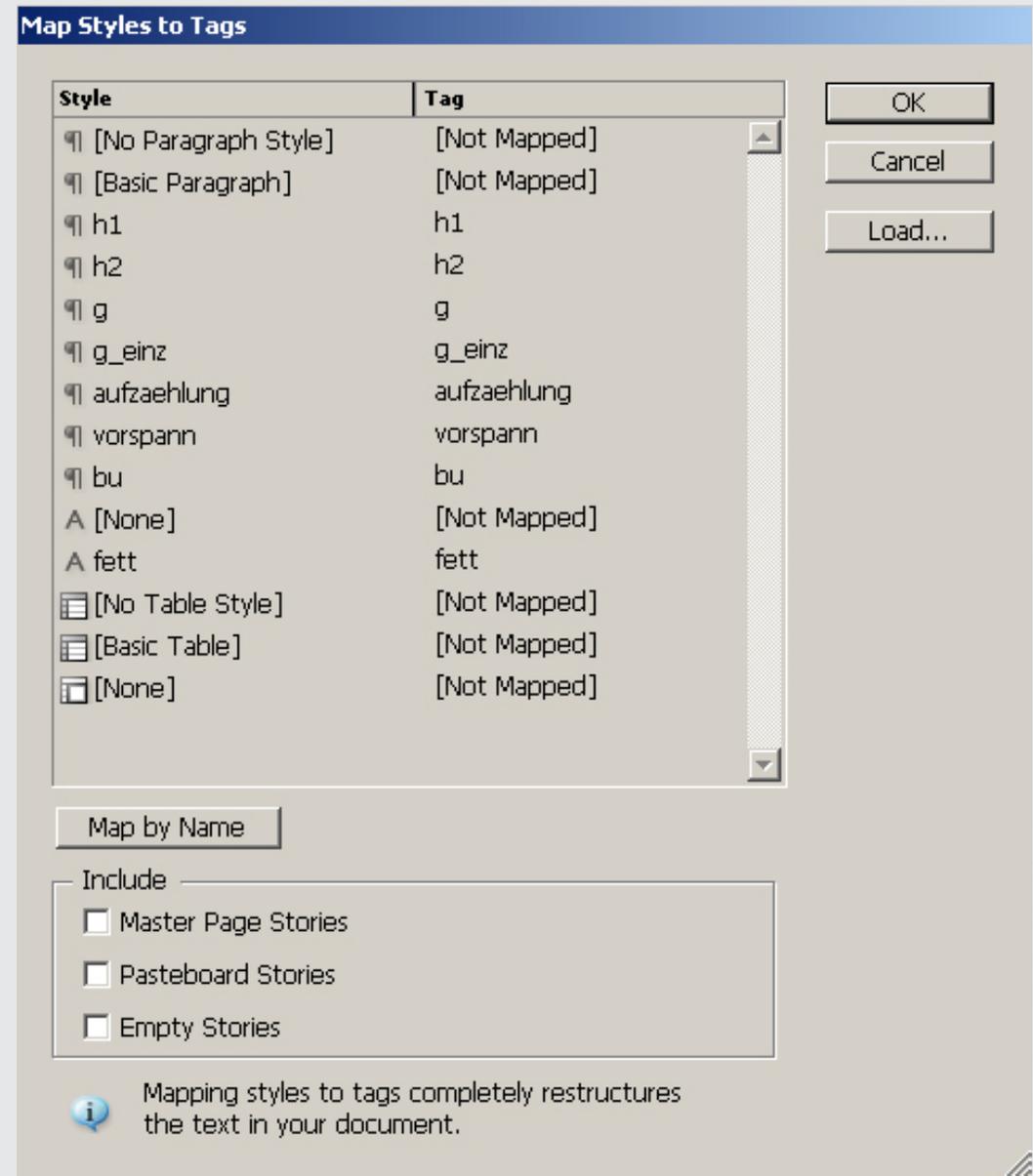
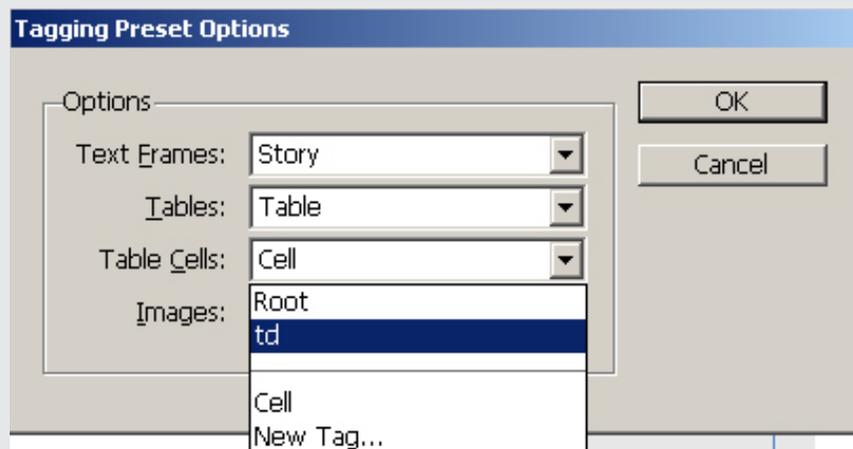
- Die Dateien befinden sich im Ordner `03_indd`
- Öffnen Sie die Datei `01_xml_bearbeiten.indd` in InDesign
- ! Machen Sie sich mit der Strukturansicht und den XML-Ansichten vertraut
`Ansicht ⇒ Struktur ⇒ Struktur einblenden [Alt + Strg + 1]`
- ! Erstellen Sie neue Tags mit Hilfe der Tags Paletten.
Zeichnen Sie Textstellen mit den Tags aus. Was fällt auf?
`Fenster ⇒ Tags`
- ! Wechseln Sie in den Story-Editor und zeichnen Sie hier Text aus.
`Bearbeiten Story Editor [STRG + Y]`
- ! Erstellen Sie Attribute für beliebige XML-Elemente.
`Kontextmenu der Strukturansicht`

Import von XML in InDesign

Vorbereitete InDesign-Dokumente
verarbeiten XML zu fertigen Layouts

Auszeichnung/Formatierung

- **Formate zu Tags** zuordnen
(oder vice versa)
Formaten werden Tags zugeordnet.
- **Zuweisungen** für Tabellen, Textrahmen
und (Bilder)
href="file://filename.eps"



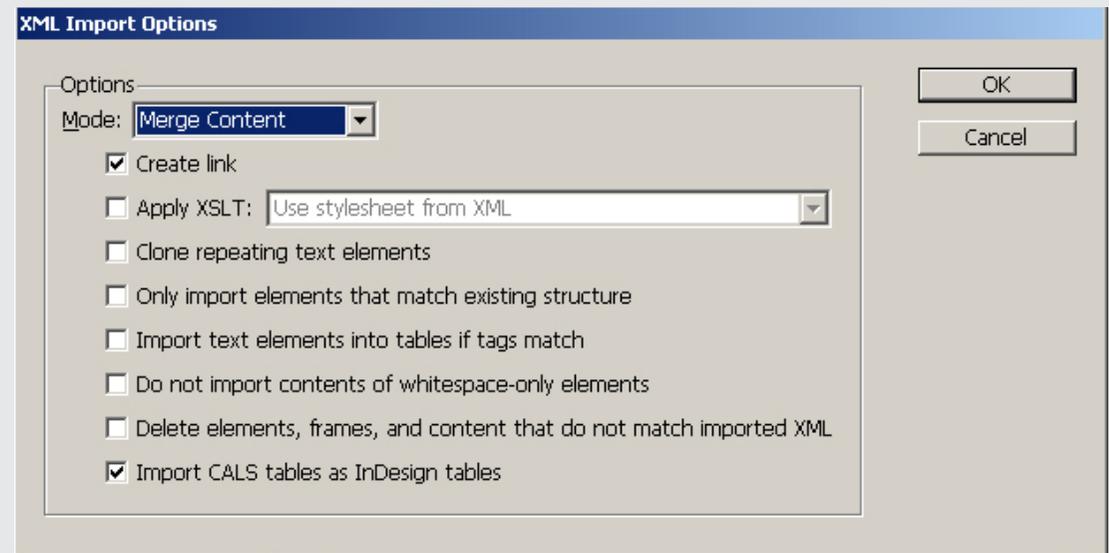
Formatierung über Attribute

- Elementen über **Attribute** Styles zuordnen.
InDesign Namespace `aid` im Quelldokument deklarieren
Absatzformat `aid:pstyle="name des formats"`
Zeichenformat `aid:cstyle="name des formats"`

```
<artikel xmlns:aid="http://ns.adobe.com/AdobeInDesign/4.0/" >
  <titel aid:pstyle="h1">Titel</titel>
  <teil id="d45">
    <abs aid:pstyle="g">Num diis, orac <fett aid:cstyle="fett">mendam</fett> iam non
    de aus vivirmis vermand enatimius es co cae que die mei sisus modiis. lintil vi-
    vatus halesse nos At inaturs remePat. Tueros auguer Tueros auguer Tueros auguer
    am duis augait dolore ming ero commy nisit ad vel ut dolobor aut nit ese tem iure
    praestrud duis nulputation henibh et, sum iriure ea facilis alit inciduis nostrud
    min hent lorem in ute.</abs>
  </teil>
</artikel>
```

XML Import

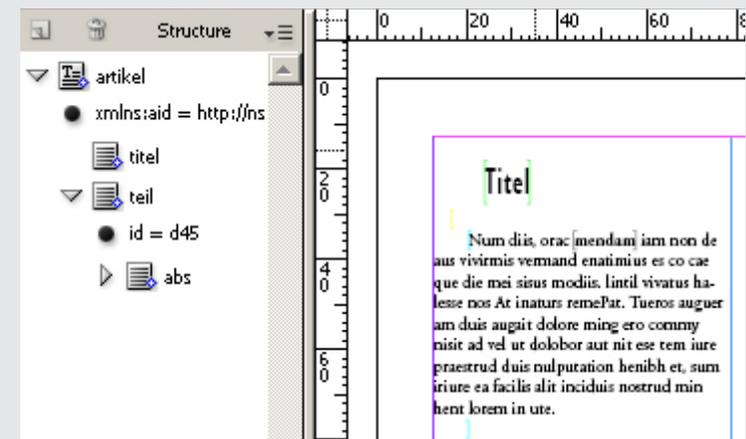
- **Einfache Strukturen** werden durch Formatzuweisungen gestaltet.
- Geschickte Lösungen möglich, aber wenig flexibel.
- **Vorteil:** Ohne Programmierung oder Plugins.



Whitespace Handling

- **Whitespace** wird vollständig **importiert** und dargestellt!
- Dokumente müssen entsprechend vorbereitet werden.
- **XSLT** für Absatzhandling.

```
<?xml version="1.0" encoding="UTF-8"?>
<artikel>
  <titel>Titel</titel>
  <teil id="d45">
    <abs>Num diis, orac <fett>mendam</fett>
```



XML Import und einfache Formatierungen

- Die Dateien befinden sich im Ordner `03_indd`.
- ! Importieren Sie die Datei `02_xml_importieren_attribs.xml` in das InDesign File `02_xml_importieren_attribs.indd`. Wie ist das Whitespace-Problem optimal zu lösen?
- Öffnen Sie die Datei `03_xml_import.indd` in InDesign
- Öffnen Sie die Datei `03_out_standard.xml` in einem XML-Editor
- ! Erstellen Sie die noch zusätzlich benötigten Formatvorlagen.
- ! Weisen Sie den Elementen mit Hilfe von **Formate zu Tags zuweisen** geeignete Formate zu.
Kontextmenu der Strukturansicht ⇒ Formate zu Tags zuweisen
- ! Importieren Sie die XML-Datei, optimieren Sie bei Bedarf das Ergebnis und wiederholen Sie den Vorgang.
- ! Zusatz: Erstellen Sie eine XML-Struktur mit mehreren Container-Elementen und verknüpfen Sie diese mit verschiedenen Textrahmen.

Tabellen

InDesign unterstützt das CALS-Tabellenformat und sein eigenes Tabellenformat.

- **InDesign Tabellenformat**

Eher Softwarenah? In anderen Kontexten aufwändig Konvertierung notwendig.

- **CALS Tabellen** (nur CS3)

OASIS Standard, kein xhtml aber Konvertierung vorhanden.

www.oasis-open.org/specs/tablemodels.php

- Formatierung über Attribute oder Formatzuweisungen (nur CS3).

- Struktur von InDesign Tabellen.

```
<Table xmlns:aid="http://ns.adobe.com/AdobeInDesign/4.0/" aid:table="table"
aid:trows="6" aid:tcols="2">
  <Cell aid:table="cell" aid:thead="1" aid:crows="1" aid:ccols="1"
aid:ccolwidth="67"><t_header>Verwendung</t_header></Cell>
  <Cell aid:table="cell" aid:thead="1" aid:crows="1" aid:ccols="1"
aid:ccolwidth="87"><t_header>Bauart</t_header></Cell>
  <Cell aid:table="cell" aid:crows="1" aid:ccols="1" aid:ccolwidth="67">
<t_abs>Industriesauger</t_abs></Cell>
  ...
</Table>
```

Entity Auflösung

Alle Entities werden beim Import aufgelöst.

- **Standard Entities**
& = & | < = < | > = >
- Character Entities für **UTF-8 Zeichen**
&#x####;
<http://www.decodeunicode.org/>
- Für eigene Entities muss eine **DTD verknüpft** sein
oder die Entity muss im **Dokument deklariert** sein.
<!ENTITY idcs „InDesign CS3“>
Aufruf im Dokument über &idcs;
- Bei nicht auflösbaren Entities **bricht der Import** ab!



< CHARACTER > BLOCK PROPERTIES
U+20B0 GERMAN PENNY SIGN

Bilder

Bilder werden automatisch gesucht wenn ein href Attribut vorhanden ist

- **Rudimentäres Handling** über href Attribut.
 - Nur Import
 - Keine Größen, Ausschnitte etc.
- Nur **verlinkte Bilder** möglich – Einbetten ist grundsätzlich nicht empfehlenswert.

```
<bild href="file://path/images/testfile.jpg" />
```

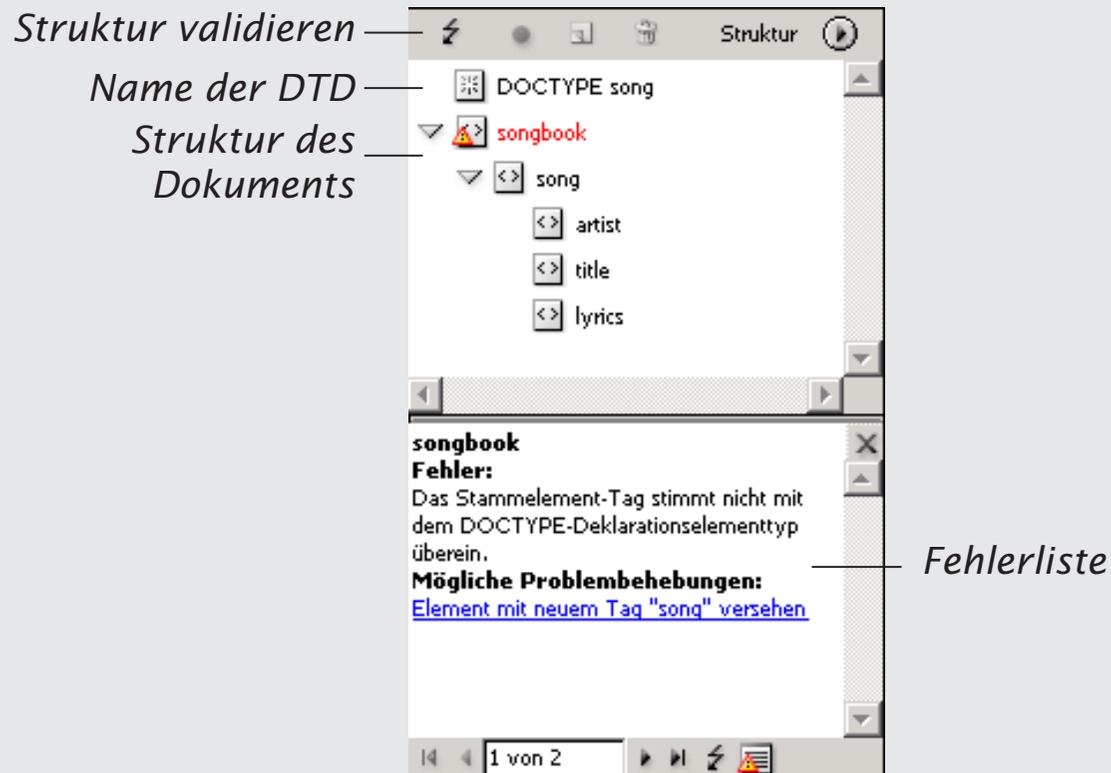
- Weitere Anpassung nur über Skripting möglich.

Bilder und Tabellen importieren

- Die Dateien befinden sich im Ordner `03_indd`.
- ! Importieren Sie die Datei `03_out_unicode.xml` in die bereits vorbereitete InDesign-Datei
achten Sie darauf dass die Bilder automatisch platziert werden.
- ! **Demonstration InDesign CS3:** Tabellenimport mit Tabellenvorlagen.
Grenzen dieser Formatierung erkennen (`02_tabelleimport.indd`).

Weitere XML-Funktionen

- DTD importieren
InDesign kann eine DTD (Document Type Definition) importieren und gegen diese validieren.
- DTD validieren
Ab dem Stammelement (Root-Element)
oder ab einem ausgewählten Element gegen die DTD validieren.



XML validieren

- Die Dateien befinden sich im Ordner `03_indd`.
 - Öffnen Sie die Datei `01_xml_bearbeiten.indd` erneut in InDesign
- ! Verknüpfen Sie die Datei mit der DTD `01_indesign.dtd` aus dem gleichem Ordner.
- ! Validieren Sie das Dokument gegen die DTD. Beheben Sie den Fehler!
- Kontextmenu der Strukturansicht

XML Export aus InDesign

▪ Remap Special Characters

Spezialzeichen die keine UTF-8 Entsprechung haben (z.B. Einzug hier) werden als Leerzeichen ins XML geschrieben.



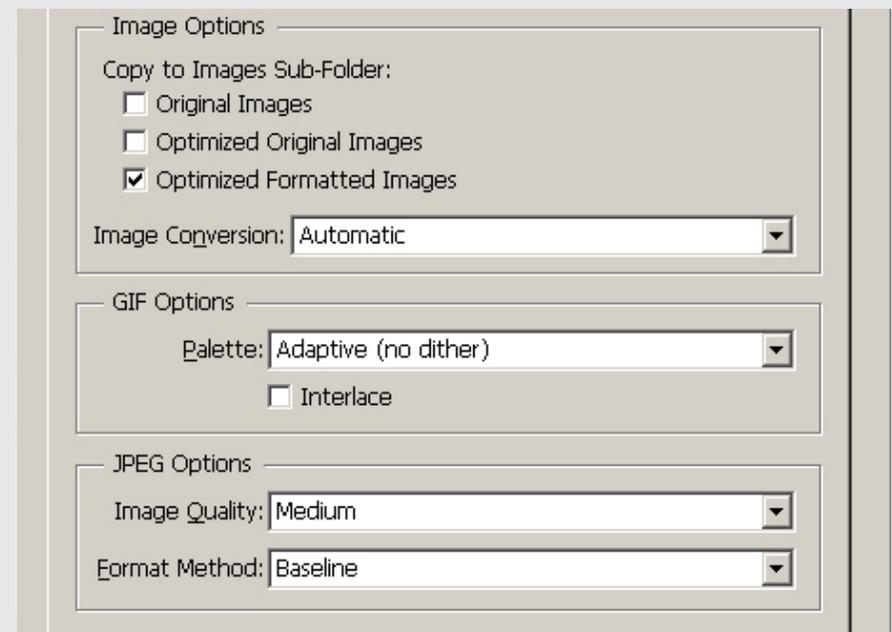
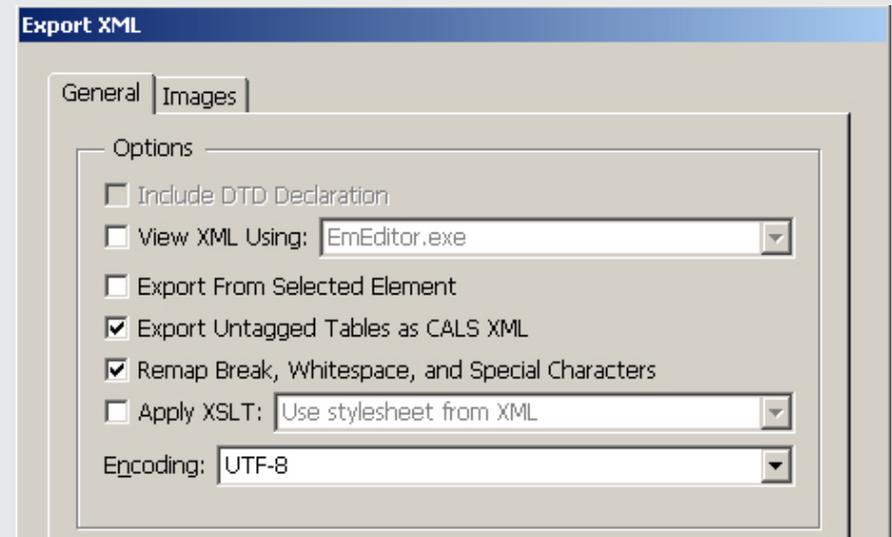
▪ Tabellen

Tabellen können als CALS-Tabellen exportiert werden

▪ Bilder

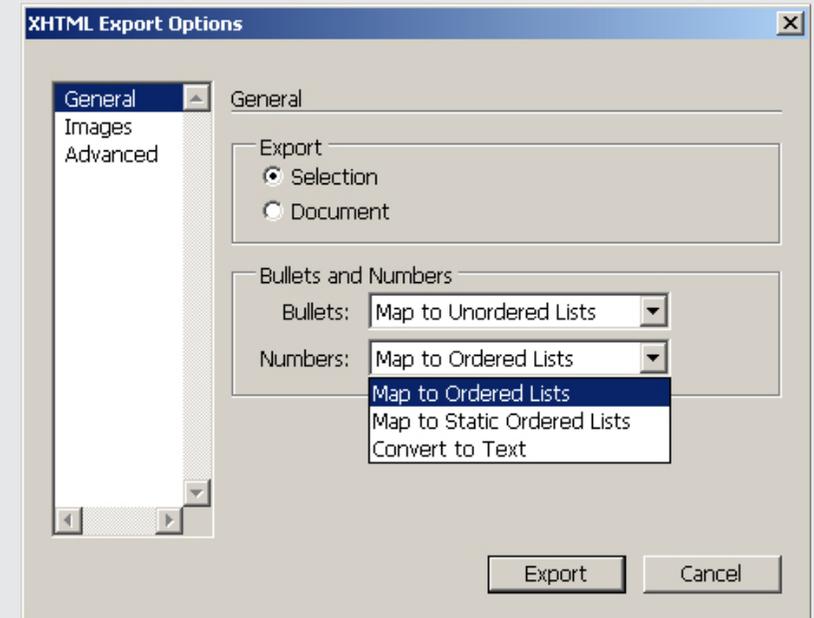
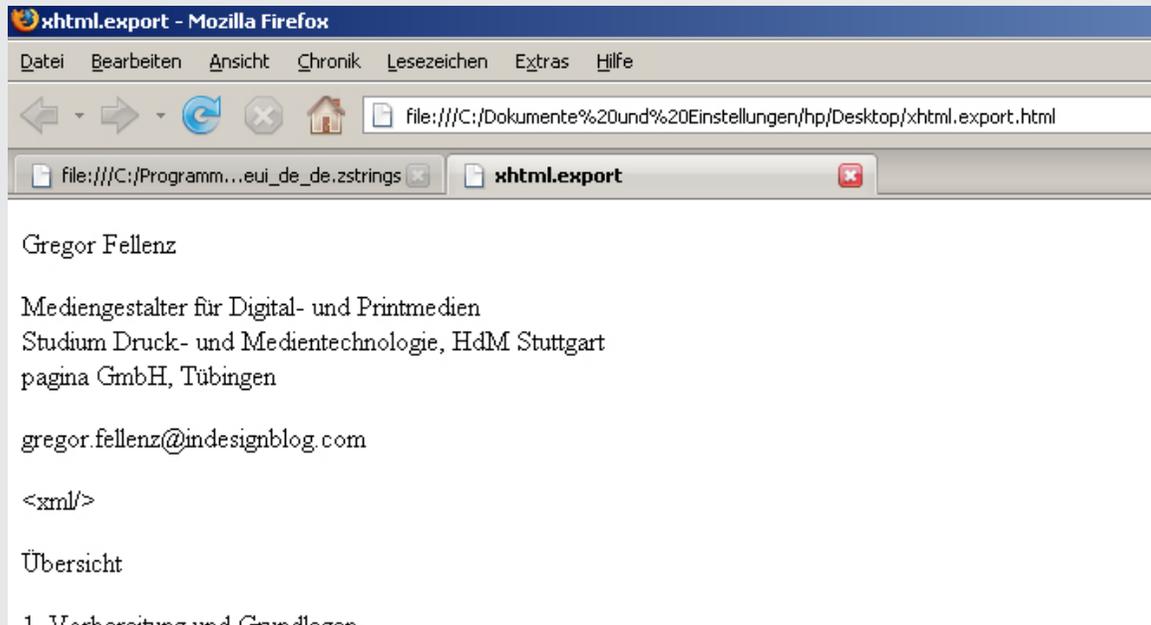
Werden auch exportiert, Downsampling möglich.

▪ Echte Fußnoten werden verworfen.



XHTML-Export CS3

- Skript von Adobe zum Export von InDesign Dokument als **XHTML**.
- Einfache Möglichkeit um Inhalte zu generisch zu **extrahieren**.
- Absatz- und Zeichenformate werden als CSS class Attribute übergeben.
- **Tabellen**, Fußnoten und Verweise werden exportiert!
- Listen aus Absatzformaten werden konvertiert.
- Keine **eingebetteten Bilder** (Bad Practice!).



InDesign Snippets (InDesign CS3)

InDesign Snippets beinhalten Seitenobjekte und deren relative Position zueinander.

- Kleine Bausteine aus InDesign Dokumenten exportieren/speichern
- Wiederverwendbarkeit von kleinen Layout-Kompositionen.
- vereinfacht Stehsatz etc.
- Snippet Dateien haben die Endung .inds
- Werden in XML gespeichert.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<?aid style="33" type="snippet" DOMVersion="5.0" readerVersion="4.0" featureSet="257"
product="5.0(458)" ?>
<?aid SnippetType="PageItem"?>
<SnippetRoot>
<colr clmd="e_prss" clsp="e_CMYK" clvl="x_4_D_0_D_0_D_0_D_0" ovrD="e_e0vP" clbs="o_n"
atcs="e_nasp" atvl="x_0" pnam="c_Paper" edbl="b_t" rmb1="b_f" pvis="b_t" swID="l_1f01"
self="rc_u12"/>
<colr clmd="e_prss" clsp="e_CMYK" clvl="x_4_D_0_D_0_D_0_D_100" ovrD="e_e0vB"
clbs="o_n" atcs="e_nasp" atvl="x_0" pnam="c_Black" edbl="b_f" rmb1="b_f" pvis="b_t"
swID="l_1f01" self="rc_u8"/>
```

XML-Export anwenden

- Die Dateien befinden sich im Ordner `03_indd`.
 - Öffnen Sie die Datei `04_xml_export.indd` in InDesign
- ! Testen Sie den XML-Export. Analysieren Sie die exportierte Datei in einem XML-Editor
`Datei` ⇒ `Exportieren` [Strg E]
- ! Wie ist vorzugehen, um ungetaggten Inhalt zu exportieren?
- ! **Demonstration:** Fußnoten in einen XML-Baum einfügen
(`fussnote.indd` und `insertFootnoteIntoXML.jsx`).
- ! **Demonstration InDesign CS3:** XHTML-Export – Textreihenfolgenproblematik
(`xhtml.export.indd`).
`Datei` ⇒ `Crossmedia Export` ⇒ `XHTML`
- ! **Demonstration InDesign CS3:** Snippet-Export.
`Datei` ⇒ `Export`

Fazit

- Struktur muss in InDesign **abbildbar** sein.
- XML-Daten müssen **vorhanden** sein, bzw. **generiert** werden können.
- Wirtschaftlicher Nutzen?

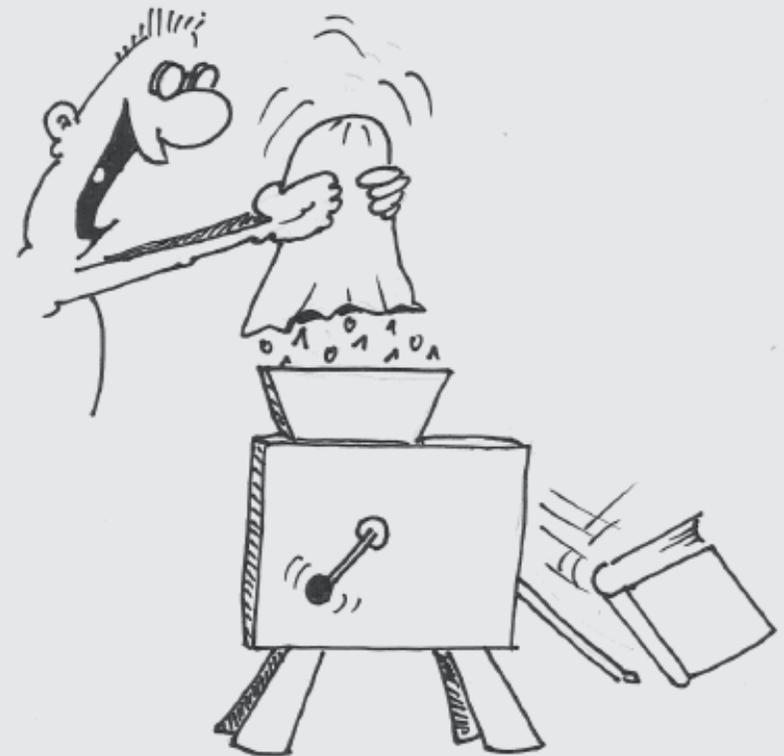
- **Vollständige Automatisierung** je nach Layout möglich.
- Für **Strukturierte Layouts** sinnvoll und wirtschaftlich.
- **Systematisches Arbeiten** wird unterstützt.

- **Tabellenunterstützung/Fußnoten/Verweise** suboptimal.
- **Bearbeitungsmöglichkeiten** verbesserungswürdig.
- **Export** setzt konsequentes Arbeiten mit XML voraus.

InDesign Programmierung

Erweiterte Möglichkeiten

- Keine Grenzen mehr fürs Layout – Aufbau **komplexer Strukturen** steuerbar.
- Satz von wiederkehrende Aufgaben entlasten – mehr Zeit für die Gestaltung.
- Verweise und Indexerstellungen möglich.
- **Scripting** oder Plugin-Entwicklung.
- Scripting in VB Skript, AppleScript und JavaScript.
- Adobe Scripting Guide.
- **Vereinfachen** von Tätigkeiten
- Komplexe oder immer wiederkehrende Abläufe zusammenfassen.
- Verborgene Funktionen nutzen.



Skripte in JavaScript erstellen

Moderne, schlanke, dynamisch typisierte, objektorientierte aber klassenlose Skriptsprache. JavaScript lässt sich sowohl prozedural als auch objektorientiert programmieren.

- Plattformunabhängig (Mac/PC/[Linux]).
- Kombination aus **Sprachkern** und **Document Object Model** (DOM).
- Sprache ist aus der **Web-Programmierung** bekannt.
Clientseitige Programmierschnittstelle (API) für den Zugriff auf HTML-Dokumente.
- **Informatisches Verständnis** sinnvoll, grundsätzliche Konstrukte sollten bekannt sein.
- Bei InDesign **prozedurale Programmierung** – es wird aber auf Objekte zugegriffen.
- **Kontrollstrukturen**
 - If ... else (Bedingte Anweisungen)
 - For-Schleife (Verarbeiten von Mengen)
- **Funktionen**
 - Eine Funktion ist ein Block von Anweisungen
- **Fehlerbehandlung**
 - Try Catch Blöcke

- **Document Object Model** DOM von InDesign

Objektorientiertes Klassenmodell, alle Objekte von der Basisklasse „Application“ abgeleitet

`app.documents.add()` ;

Objekt Referenz ist wichtigstes Hilfsmittel (Seit CS3 nicht mehr als PDF)

Vereinfacht: Alle Menu-Einträge/Auswahlen haben eine Entsprechung (und noch mehr ...)

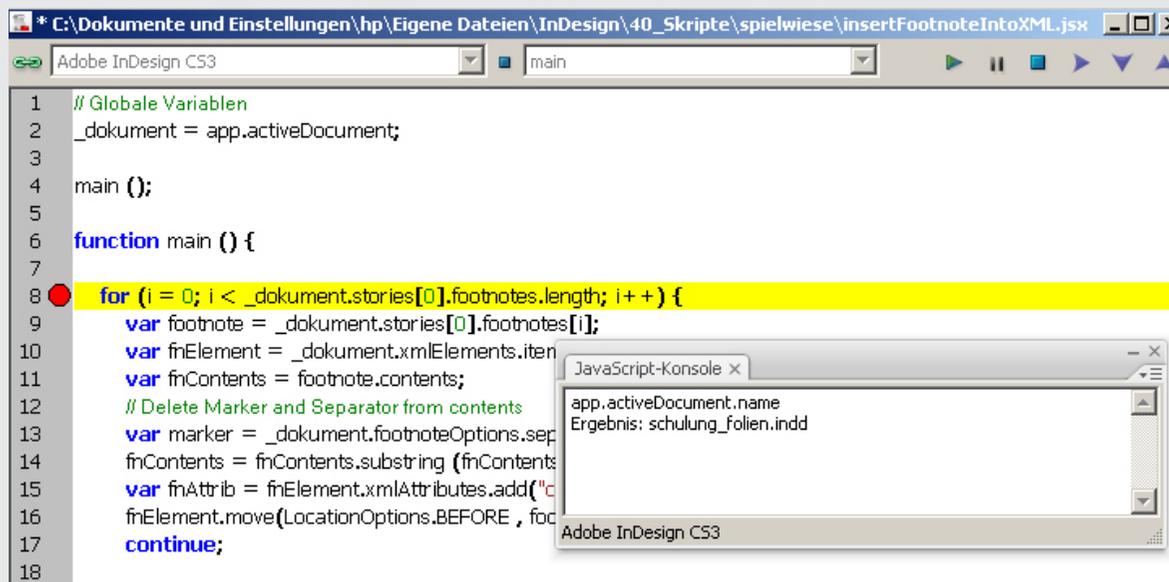
Hierarchisch aufgebaut (Eltern/Kind Beziehungen)

- **ExtendedScript Toolkit 2**

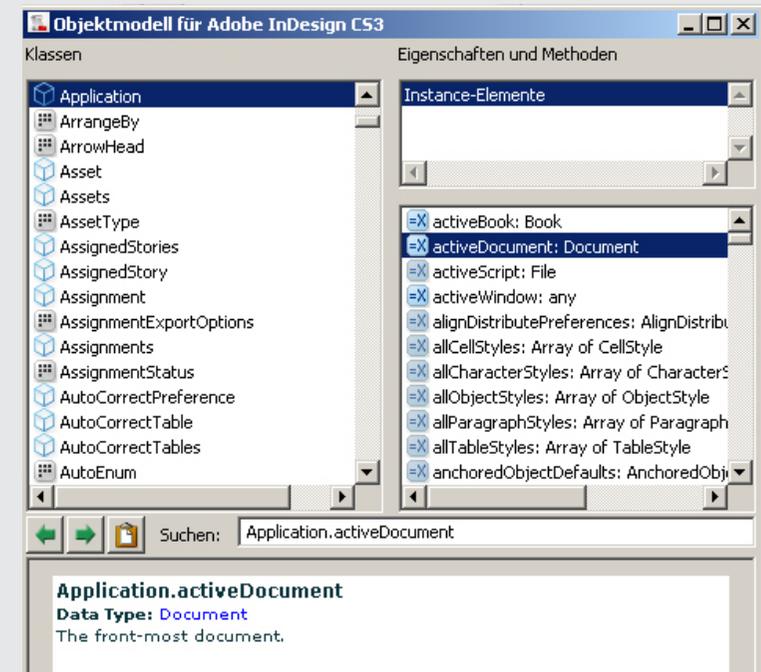
Editor/Mini **IDE** (Integrierte Entwicklungsumgebung) für InDesign.

Ideal zum Entwickeln von Skripten

Debugging und Hilfe mit Objektmodell



```
1 // Globale Variablen
2 _dokument = app.activeDocument;
3
4 main ();
5
6 function main () {
7
8   for (i = 0; i < _dokument.stories[0].footnotes.length; i++) {
9     var footnote = _dokument.stories[0].footnotes[i];
10    var fnElement = _dokument.xmlElements.iter
11    var fnContents = footnote.contents;
12    // Delete Marker and Separator from contents
13    var marker = _dokument.footnoteOptions.sep
14    fnContents = fnContents.substring (fnContents
15    var fnAttrib = fnElement.xmlAttributes.add("c
16    fnElement.move(LocationOptions.BEFORE , foc
17    continue;
18
```



Objektmodell für Adobe InDesign CS3

Klassen

- Application
- ArrangeBy
- ArrowHead
- Asset
- Assets
- AssetType
- AssignedStories
- AssignedStory
- Assignment
- AssignmentExportOptions
- Assignments
- AssignmentStatus
- AutoCorrectPreference
- AutoCorrectTable
- AutoCorrectTables
- AutoEnum

Eigenschaften und Methoden

Instance-Elemente

- activeBook: Book
- activeDocument: Document
- activeScript: File
- activeWindow: any
- alignDistributePreferences: AlignDistrib
- allCellStyles: Array of CellStyle
- allCharacterStyles: Array of CharacterS
- allObjectStyles: Array of ObjectStyle
- allParagraphStyles: Array of Paragraph
- allTableStyles: Array of TableStyle
- anchoredObjectDefaults: AnchoredObj

Suchen: Application.activeDocument

Application.activeDocument
Data Type: Document
The front-most document.

Das erste Skript

- **Speicherort:** Über das Skript Panel am einfachsten zu finden.
- **Musterskript** hineinkopieren.
- Mit **ESTK** editieren.
- Dokumentreferenz herstellen
- TextFrame erstellen
- Text einfügen

```
var _dokument = app.activeDocument;

main();
exot();

function main () {
    var tf = _dokument.pages[0].textFrames.add();
    tf.geometricBounds = [10,10,100,100];
    tf.insertionPoints[0].contents = „hallo welt“;
}
```

Ein einfaches Skript erstellen

- Die Dateien befinden sich im Ordner `04_scripts`.
 - Öffnen Sie die Datei `vorlage.jsx` im ESTK
-
- ! Erstellen Sie ein neues Dokument, einen TextFrame und füllen Sie diesen mit Inhalt.
 - ! Importieren Sie ein XML-File (Funktion `importXML(File)` des Dokument Objekts).
 - ! Platzieren Sie den importierten XML-Content in einen neuen TextFrame (Funktion `placeXML(textFrame)` des XML-Objekts)

XML-Rules kurz vorgestellt

XML-Rules eignen sich dazu dynamisch Layouts zu erstellen und zu formatieren.

- **XML-Rule-Sets** (ab CS3) sind kleine Skripte die bei bestimmten Knoten Aktionen auslösen.
- **XPath** gesteuert.
- Wenn die Regel zutrifft, wird eine bestimmte Aktion ausgelöst.
- VB Skript (Windows), AppleScript (MacOS) oder JavaScript (beide).
- **Beispiele**
 - Eine Regel für Absatzelemente überprüft, ob es der erste Absatz nach einer Überschrift ist und rückt ihn entsprechend ein.
 - Bei einer bestimmten Hierarchie-Ebene wird eine neue rechte Seite erstellt.

```
function formatTelefonnummer(){
  this.name = „formatTelefonnummer“;
  this.xpath = „//TelNummer“;
  this.apply = function(myElement, myRuleProcessor){
    with(myElement){
      insertTextAsContent(„Telefon:\t“, XMLElementPosition.ELEMENT_START);
      applyParagraphStyle(_dokument.paragraphStyles.item(„TELNUMMER“));
    }
  }
  return true;}
}
```

XML-Rules anwenden

- Die Datei `xmlRules.jsx` befindet sich im Ordner `04_scripts`.

! Demonstration InDesign CS3: XML Rules zum formatieren von XML-Dokumenten.

Literatur, Informationen und Quellen

InDesign und XML

- Adobe Informationen zu XML
http://www.images.adobe.com/www.adobe.com/products/indesign/scripting/pdfs/indesign_and_xml_technical_reference.pdf

InDesign Scripting

- Adobe Scripting Guide
<http://www.adobe.com/products/indesign/scripting/>
- Adobe Scripting Forum (sehr rege Community)
<http://www.adobeforums.com/webx?13@@.3bbf275d>

Datenbank Plugins

- **Übersicht** von Jörg Oyen:
http://publishing-report.com/database_publishing_tools.html
- **Empfehlung:** Easycatalog einfach und relativ günstig:
<http://www.65bit.com>

Vielen Dank für Ihre Aufmerksamkeit!

Fragen? Anregungen? Diskussionsbedarf?

Die Folien zum Download:

<http://www.indesignblog.com/tekom>

E-Mail:

gregor.fellenz@indesignblog.com