

Scripting
JavaScript

Workflows

XML

InDesign

IDML

InDesign mit Skripten und XML automatisieren

8. Mai 2017

IDUG Stuttgart

Kontakt: gregor.fellenz@publishingx.de

Folien: <http://www.publishingx.de/dokumente>

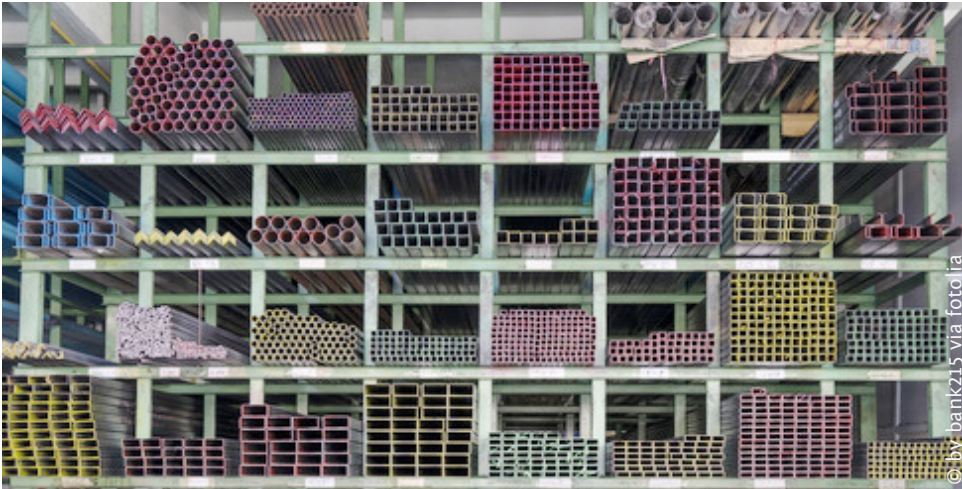
Heute:

1. ■ XML und IDML

2. ■ Skripting

XML und IDML





Was sind strukturierte Daten?

- Gleichartige Struktur vs. beliebig angeordnet
- Gehorchen eine Grammatik/Schema vs. gestalterische Anordnung
- Validierung vs. nicht prüfbar

Welche Datenquellen gibt es?

- Datenbanken
- CSV/Tabellenkalkulation
- JSON
- XML/Markup

Warum strukturierte Daten für das Publishing?



Old School Publishing

Ein Content → Ein Layout → Ein Medium

Crossmedia Publishing

Medienneutrale Workflows für unterschiedliche Ausgabeformate.

Chancen

- Ein Inhalt für mehrere Medien.
- Wiederverwendbarkeit von Struktur-Informationen.
- Einheitliche Datenspeicherung.
- Trennung von Content und Layout.

Weitere Anwendungsgebiete von strukturierten Daten...

Datenbank/CSV/Excel

- **Persönliche Ansprache** der Empfänger: Mailings, Einladungen, Eindrücke, ...
- **Vorlagenbasierte Individualisierung**: Namensschilder, Visitenkarten, ...
- **Database Publishing**: Automatisierte Kataloge, Preislisten, ...

JSON

- Dokumentsteuerung oder -auswertung via **Webservice**
 - Bildquellen aus Datenbank
 - Generierung von Micro-Templates z.B. Discounter

XML/Markup

- **Befüllung von Templates**: Visitenkarten, Datenblätter, Tabellen, ...
- **Automatisierter Satz**: Bücher, Zeitschriften, Kataloge, ...
- **Übersetzungen/Varianten** austauschen
- **Export aus InDesign**: Digitale Ausgabe Web oder App (CMS), E-Books,

Extensible Markup Language

Motivation?

- Sprache zur Beschreibung von **Dokumentstrukturen**.
- **Lesbar** für Mensch und Maschine.
- Durch vorgegebene Strukturen sind die Inhalte **transformier-** und **konvertierbar**.

XML als Standard für Datenstrukturierung

- W3C Recommendation seit dem 10. Februar 1998.
- Metasprache, die eine **abstrakte Syntax** für Auszeichnungssprachen definiert.
- Es wird unterschieden: **dokumentzentriert** vs. **datenzentriert**.

Produktinformation als XML

01_product9902131290.xml

- Die Daten sind strukturiert.
- Die Daten sind semantisch ausgezeichnet.

Festes
Format



Dynamischer
Seitenaufbau

Print vs. strukturierte Daten

Beim Einsatz von DTP-Programmen für strukturierte Dokumente prallen zwei unterschiedliche Philosophien aufeinander.

- **DTP ist medienspezifisch:** Layout einer fixen Seite.
- DTP Programme bieten kaum Tools zur **Strukturierung**.
- Im Endergebnis ist der **Aufbau der Seiten** nicht ersichtlich/wichtig.

XML in InDesign-Dokumenten

Seit **InDesign CS3** kann InDesign XML Daten (sinnvoll) verarbeiten.
Ab InDesign CS4 sind Tabellen im Textmodus editierbar.
Bekommt von Adobe aktuell nicht besonders viel ♥

Boardmittel

- **XML-Import**
 - Verknüpfung** von Tags mit Formaten, Textrahmen und Tabellen
 - Eingeschränkter **XSLT 1.0** Prozessor
 - Keine** Hierarchien, Bilder, Fußnoten, Verweise ...
 - Das **Whitespace** Problem
- **Skripting**
 - XML-Rules** mit XPath
 - Prinzipiell alles was in der **Datenstruktur** enthalten ist



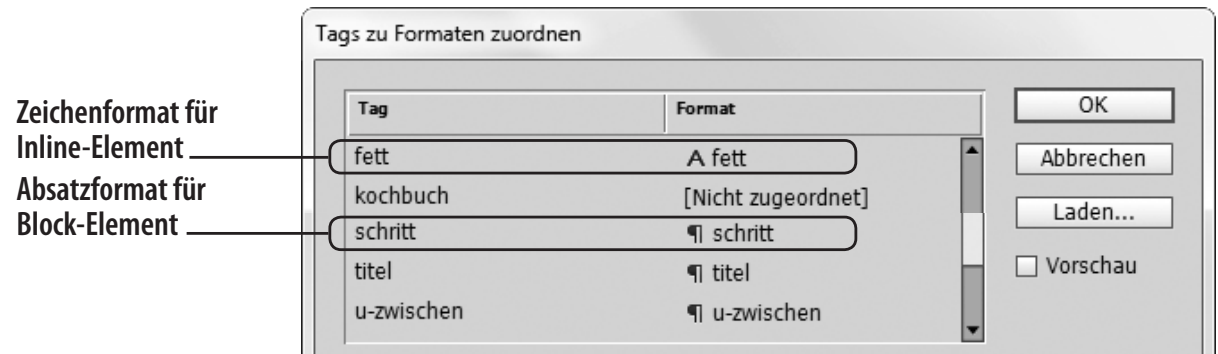
Wie geht das in InDesign?

- **Grundsätzlich:** Dokument- und XML-Inhalte auseinanderhalten.
- **Struktur Palette**
Übersicht über die XML-Inhalte, Verknüpfungsstatus, Strukturänderungen, Attribute, validieren
- Tag-Marken für Rahmen bzw. Text einblenden
- **Tags Palette**
Elemente anlegen, verwalten und zuweisen
- **Texteditor**
Mit eingblendeten Tag-Marken gute Ansicht der Textauszeichnung

Rezeptseite

02_rezept-beispiel.indd

- XML-Elemente in Layout- und Textmodusanzeige
- Strukturansicht
- Prüfung des Struktur/Validieren



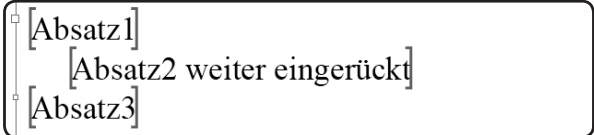
Formatierung von XML Tags

- **Tags zu Formaten** zuordnen (oder vice versa)
Einzelne XML-Elemente können Absatz- und Zeichenformaten zugeordnet werden.
- XSL-Transformationen und Skripting/XML-Rules.

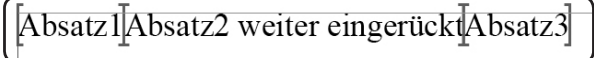
XML-Quelldaten

```
<abs>Absatz1</abs>  
  <abs>Absatz2 weiter eingerückt</abs>  
<abs>Absatz3</abs>
```

InDesign-Import



Ohne Whitespace-Entfernen-Option



Mit angewählter Whitespace-Entfernen-Option

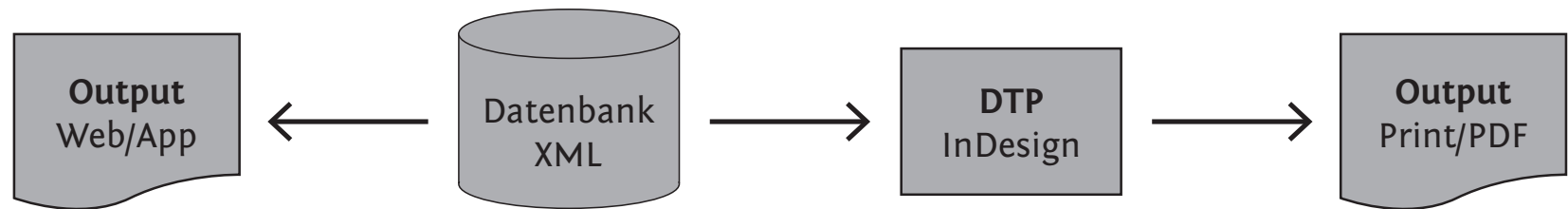
Whitespace Problem und Lösung.

- **Whitespace** wird vollständig **importiert** und dargestellt **oder** wird vollständig ignoriert.
- **Lösung:** Import mit XSLT oder Importskript verwenden. `xmlImport.jsx`

Produkt für einen Katalog erstellen

03_xml_template.indd

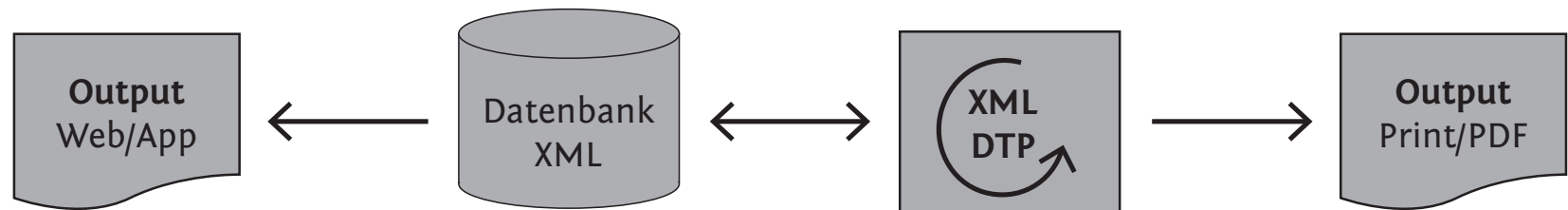
- InDesign Template.
- Formate werden mit XML-Elementen verknüpft.
- Direkte Import: Whitespace fehlerhaft.
- Einsatz des Import-Skripts.



Single Source Publishing

- **Doppelpflege:** Inhaltliche Änderungen sowohl im Layout als auch im Datenbestand
- **Hoher Automatisierungsgrad**
- **Layoutaufbau** zu einem späten Produktionszeitpunkt
- **Qualifikationsniveau Anwender**
Keine/Kaum Änderung – „normal weiter produzieren“

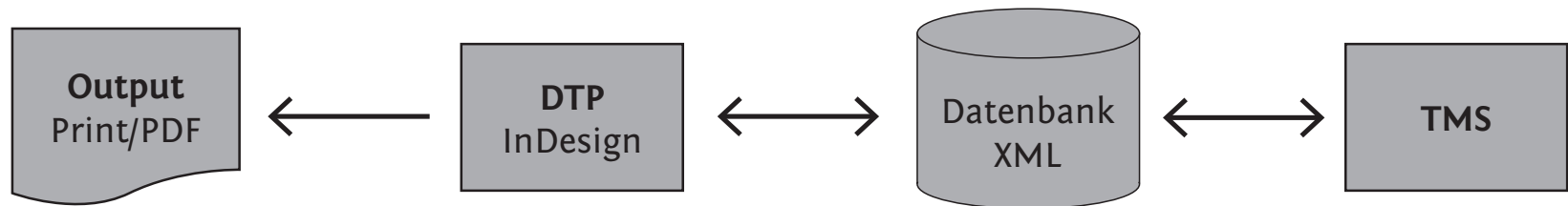
Beispiele sind Database Publishing Plugins und Workflows auf Basis von XML und Scripting.



Bidirektional

- Mehrwert der **Korrekturläufe** bleibt erhalten
- **Höhere Prozesskomplexität**
- Daten aus der Layoutsoftware werden wieder zurückgespielt.
- **Qualifikationsniveau Anwender**
Sehr hohe Anforderungen an den Bearbeiter/Mediengestalter

Beispiele sind Database Publishing Plugins und Workflows auf Basis von XML und Scripting.



Datenextraktion

Eine weitere Möglichkeit ist die Datenextraktion aus dem Layoutdokument.

- Klassische Produktion mit strikten Konventionen
- Datenformate oder die Inhaltsdaten analysieren
- **Qualifikationsniveau Anwender**
Etwas höher als normal, aber keine neuen Technologien

Diese Variante kommt z.B. für Übersetzungen oder Archivierungen zum Einsatz.



Die IDML-Familie

Mit **InDesign CS4** wurde das datenzentrierte XML-Format **IDML** (InDesign Markup Language) eingeführt. Familie mit **IDMS**, **ICML** und **ICMA**.

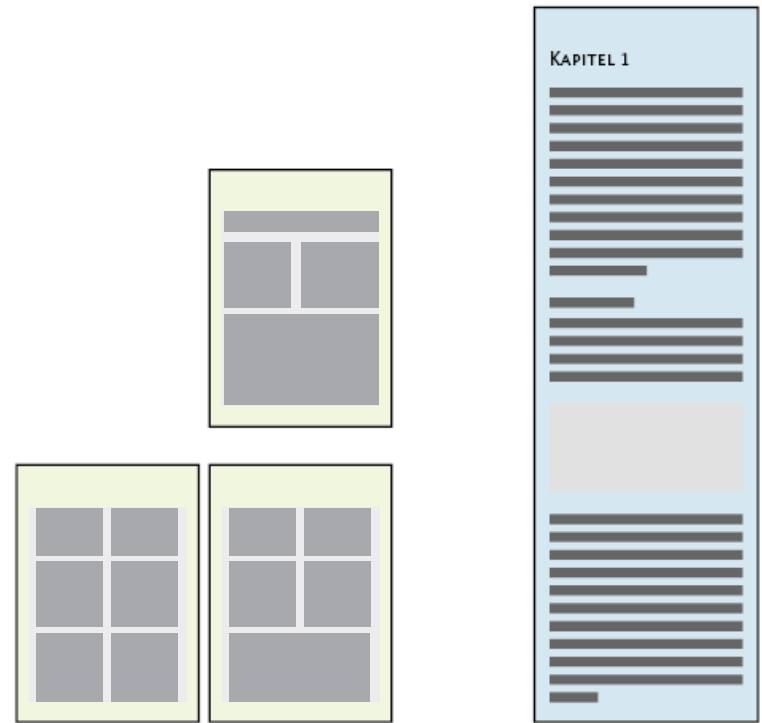
Verwendung

- **Austauschformat** zwischen verschiedenen InDesign-Versionen.
- **Katalogproduktion**: IDML aus XML-Daten bzw. Datenbanken automatisiert erstellen.
- **Templating** für Web2Print-Lösungen.
- **Übersetzungsworkflows** und Translation Memory Systeme (TMS).
- **Snippets** und **InCopy**

Designprinzip

Eine vollständige Repräsentation des InDesign Dateiformats in XML.

- Lesbar im Texteditor.
- Ohne gerenderte Voransichten.
- Ohne Umbruchinformationen der Render-Engine.
- Automatisierter Aufbau von **Dokumenten** mit XML-Daten.
- Maßeinheit ist immer **Points** (= 72 pt pro Inch).
- Minimale Angaben reichen aus (IDML-Voreinstellungen).
- Erstellt sich aus den gleichen Informationen wie das Scripting Object Model.



Was geht nicht so gut?

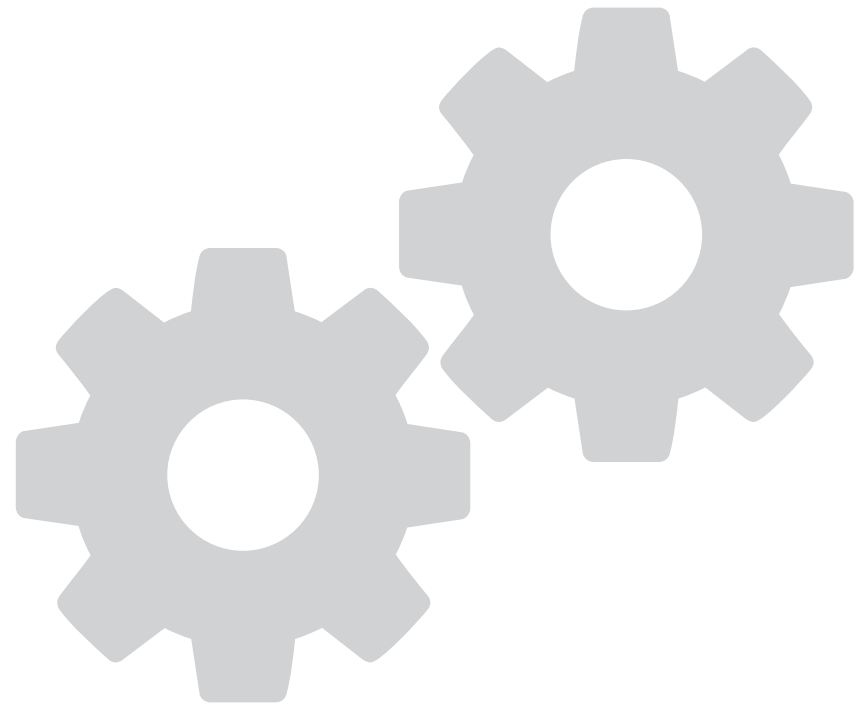
- Die Render-Engine ist bei der Erstellung nicht verfügbar.
- Layoutabhängige Berechnungen schwierig oder unmöglich.
- Lösung: **Mehrere Durchläufe** oder **Skripting**.

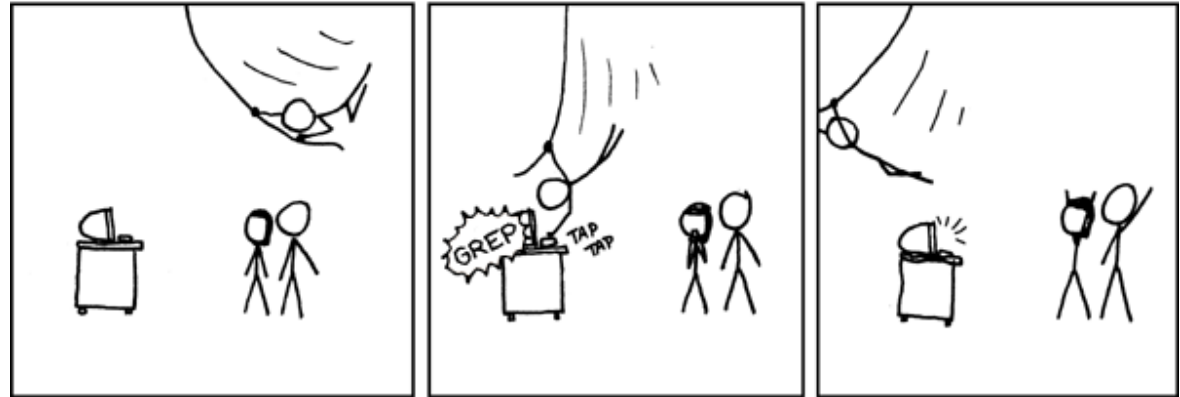
Katalog zusammenstellen

01_catalogTemplate.indd und placeSnippets.jsx

- Snippets pro Produkt.
- Automatisches einfließen in Katalog-Template.
- Aktualisierung auf Produktebene.

Skripte





Lizenz: © ⓘ ⓘ | Quelle: Remixed from <http://xkcd.com/208> | Autor: Randall Munroe

Immer wenn ...

... sich Arbeitsschritte wiederholen

Kleine und große Helferlein Bild und Bildunterschrift, komplette Formatierung einer Tabelle, Verknüpfung umbenennen, Endnoten, Bildexport, Querverweise erstellen, PDF/Bilder exportieren ...

... Arbeitsschritte fehleranfällig sind

Dokumentanalysen und erweitertes Preflight Unerwünschte Formatierungen, Falsche Umbrüche, Textlängen anzeigen ...

... es zeitkritisch ist

Layoutautomatisierung aus strukturierten Daten Kataloge, Magazine, Bücher, Tablet Publishing ...

Was können Skripte?

- **Zugriff auf InDesign-Objekte und Funktionen**
Regel: Was Sie manuell ausführen können, kann auch ein Skript erledigen.
- Einige wenige versteckte Funktionen aktivieren.
- Aber auch „**neue**“ **Funktionen**
Neue Dialoge und Menü-Einträge, Neue Funktionen wie z.B. automatische Bildunterschriften aus XMP-Metadaten oder EAN-Codes generieren.

Was können Sie nicht?

- **Neue Objekttypen oder Steuerzeichen**
Kein neuer Tabellen- oder Rahmentyp, echte Endnoten.
- **Eingriffe in Programmfunktionen**
Platzieren bleibt platzieren, der Inhalt eines Bedienfelds kann nicht angepasst werden.
- **Semantische oder grafische Entscheidungen treffen.**

Wie intelligent sind Skripte?

Einfacher Ablauf

Im wesentlichen **step by step** Anweisungen ausführen.

- **InDesign beschleunigen:** `speedUpInDesign.jsx`
<https://www.publishingblog.ch/indesign-beine-machen/>
- **Typo Kontrolle** `Typographische_Automatisierung.jsx`
- **Satz optimieren** `Umbruch_Automatisierung.jsx`
- **GREP Abfragen verketteten** `ChainGREP.jsx`
<http://www.indesignblog.com/2017/02/grep-ersetzungen-automatisieren/>

Regelbasierte Algorithmen

Gute Algorithmen lassen ein Programm schneller oder bessere Entscheidungen treffen.

- Bild Platzierung auf vorgegebener Fläche / Bin Packaging `placeImageOnPage.jsx`

Künstliche Intelligenz

Aktueller Ansatz: Maschinelles Lernen

- Google Photos, Flickr

Woher nehmen?

1. Kenne das Netz ...

Skripting Übersicht unter <http://grefel.github.io/indesignjs-resources/>

2. Selber entwickeln

Einstieg **HilfDirSelbst** Forum

und Buch **InDesign automatisieren – Skripting, GREP & Co.**

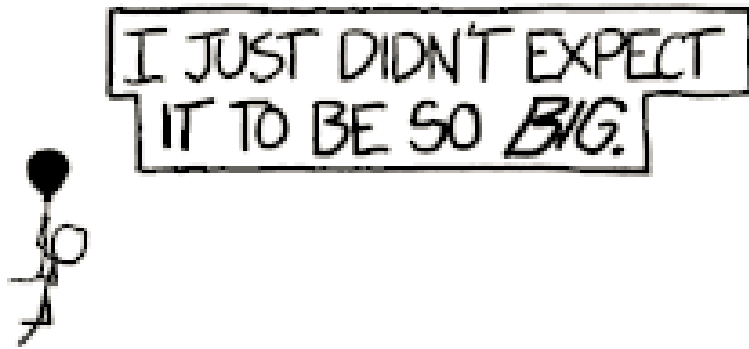
3. Entwickeln lassen

Verschiedene Anbieter im Netz...

Installieren

Ganz einfach: <http://www.publishingx.de/skripte-installieren>

Was muss ich lernen?



Programmieren

- Keine Angst – Sie müssen nicht **Informatik** studieren.
 - Logisch abstraktes bzw. »computerähnliches« denken.
 - Auf die **Syntax** der Programmiersprache einlassen.
- Erster Schritt: **Analyse** und **Anpassung** von vorhandenen Skripten.
- Fangen Sie mit **übersichtlichen** Aufgaben an.
- Keine Angst vorm **unperfekten** Skript.



Was brauche ich?

JavaScript

- **Scripting in InDesign** ist mit VB Skript, AppleScript und JavaScript möglich. Nur JavaScript ist **plattformunabhängig**,
- Eigentlich ExtendScript.
- Entwicklungsumgebung: **ExtendScriptToolkit** ab CC über die Cloud nachinstallieren.

InDesign Objektmodell

Mit Hilfe des **Objektmodells** kann man per Skript auf InDesign zugreifen, es ist die **Schnittstelle** zwischen JavaScript und dem Programm.

- Alle Programmfunktionen von InDesign, aber auch Seiten, Rahmen, Texte usw. stehen zur als **Klassen/Objekte** zur Verfügung.
- **Objektmodell** ist abhängig von der InDesign Version.
- **Problem:** Man kann nicht alle Funktionen und Einstellungen auswendig lernen.
- **Stattdessen:** Strategien zur Erschließung des Objektmodells erarbeiten.

Mein erstes InDesign Skript

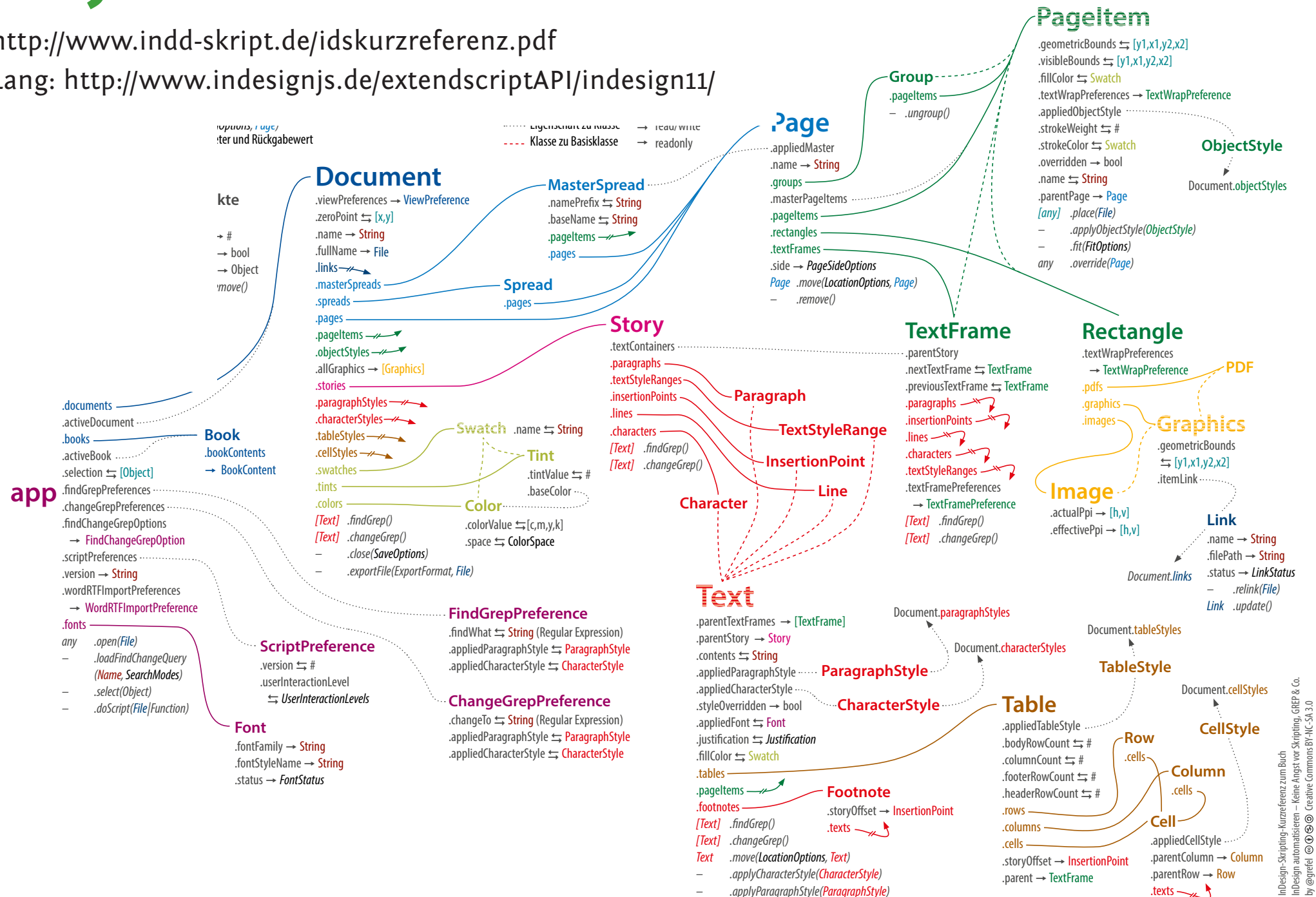
1. **InDesign** und **Extended Script Toolkit** starten.
2. Skript schreiben.
3. Entweder direkt aus dem ESTK oder im Bedienfeld *Skripte* ausführen.
4. Fertig!

```
// Achtung: In InDesign muss mindestens ein Dokument geöffnet sein  
var tf = app.activeDocument.pages[0].textFrames.add();  
tf.geometricBounds = [10,10,100,100];  
tf.contents = "Hallo InDesign";
```

Objektmodell Kurzreferenz

<http://www.indd-skript.de/idskurzreferenz.pdf>

Lang: <http://www.indesignjs.de/extendscriptAPI/indesign11/>



InDesign Skript zum XML Import

Import von strukturierten Daten.

- Auswahl eines Produktes aus der Liste z.B. 0223
- Einsetzen der Daten in ein Template
- Variable Anzahl von Kombinationsprodukten

In eigener Sache

InDesign automatisieren – Skripting, GREP & Co. 2. Auflage

Das Buch zur InDesign Automation mit einer Skripting Einführung und vielen Praxistipps zu EPUB, XML und GREP.

Auf der Homepage zum Buch <http://www.indesignjs.de/> gibt es Leseproben und alle Beispiele

- Klassisch auf Papier
ISBN: 978-3-86490-235-2
Preis: 36,90 Euro (D)
- EPUB
ISBN: 978-3-86491-684-7
Preis: 29,99 EUR



Kurse gibt es auch ...

<https://www.publishingx.de/termine/>

4-5. Juli InDesign XML, Heidelberg

18-19. Juli InDesign JavaScript, Heidelberg

8. November InDesign JavaScript, Stuttgart

Vielen Dank für Ihre Aufmerksamkeit!

Fragen und Anregungen?

E-Mail: gregor.fellenz@publishingx.de

Twitter: [grefel](#)

Blog: <http://www.indesignjs.de>