

Scripting  
JavaScript

Workflows

GREP

InDesign

IDML

CSV

XML

Datenzusammenführung

# InDesign »Beine machen«

7. Dezember 2015  
medien[plan]tage

Kontakt: [gregor.fellenz@publishingx.de](mailto:gregor.fellenz@publishingx.de)  
Folien: <http://www.publishingx.de/dokumente>

# Heute:

1. ■ GREP

2. ■ Datenzusammenführung

3. ■ XML und IDML

4. ■ Skripting



## Klassisch

- **Corporate Design** und **Arbeitsvorbereitung**
- **Formate** und **Mustervorlagen** verwenden.
- **GREP** und **GREP-Stile** verstehen und einsetzen.
- InDesign **Skripte** finden und für sich arbeiten lassen.
- **JavaScript** lernen und Skripte anpassen oder erstellen.

## Schnittstellen/Import

- Word-, Excel-, ...Import
- E-Books im Format **ePUB**
- **XML-Schnittstelle** oder **IDML**
- **Datenzusammenführung** mit **CSV**

## Plugins

- Database Publishing/  
Product Information System
- Redaktionssystem/  
Media Asset Management

**GREP**



# Schöner suchen und ersetzen

## Was ist GREP?

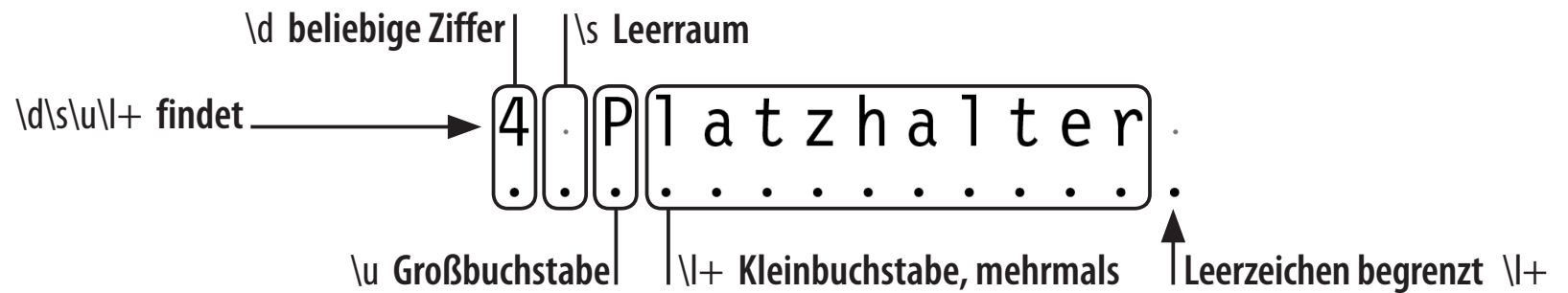
Die **Suche** nach Mustern. Die gefundenen Muster bei der **Ersetzung** wiederverwenden.

- InDesign bietet ab CS3 die Suche mit **Regulären Ausdrücken** bzw. **GREP**.
- Seit InDesign CS4 **GREP-Stile**.
- Große Augen bei den Kollegen :-)

## Beispiele für GREP-Muster

01\_Beispiele.indd

- **Alle Ziffern im Dokument:** `\d`  
Ziffern in kleinerer Schrift oder Überprüfung von Einheiten.
- **Zeichenauswahl** `[a-n]`  
Lokalisierung von bestimmten Kombinationen z.B. Produktnummern.
- **Variationen** `(Müller|Mue1ler)`  
Ein Name ist in unterschiedlich geschrieben worden.
- **Positionen** `^.\+\.`   
Den ersten Satz eines Absatzes finden.
- **Komplexe Ersetzungen** `Seite (\d+)` Ersetzen durch `→ $1`  
Seitenverweise in der Form `Seite 100` in die Form `→ 100` bringen.

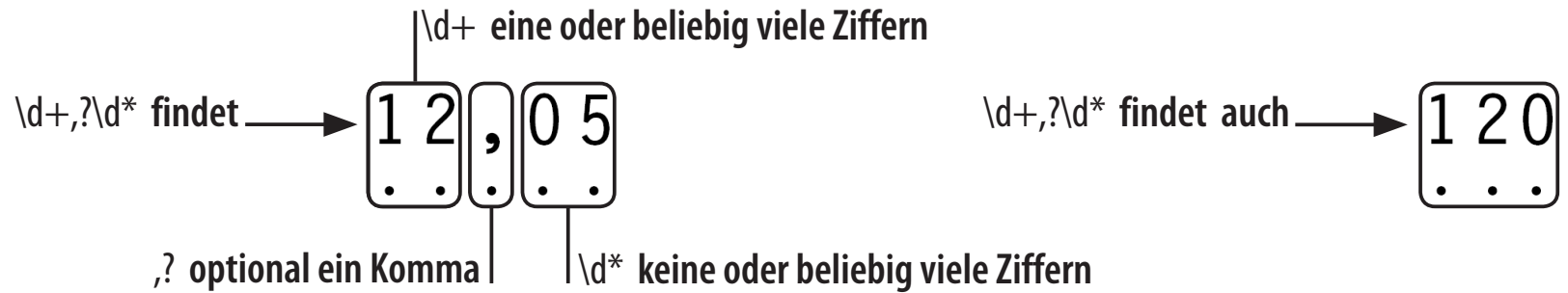


# Platzhalter/Joker

- Der einfachste **Platzhalter** ist der Punkt `.`.  
Er findet jedes beliebige Zeichen, mit Ausnahme von Umbrüchen.  
Mit `..` findet man zwei aufeinanderfolgende Zeichen.

Platzhalter repräsentieren eine bestimmte Menge an Zeichen, sie werden **Zeichenklassen** genannt.

- Bereits bekannt: Mit `\d` alle Ziffern lokalisieren.
- Mit `\h` können Leerräume gefunden werden.
- Mit `\u` findet man einen beliebigen Großbuchstaben, mit `\l` einen Kleinbuchstaben.



# Wiederholungen

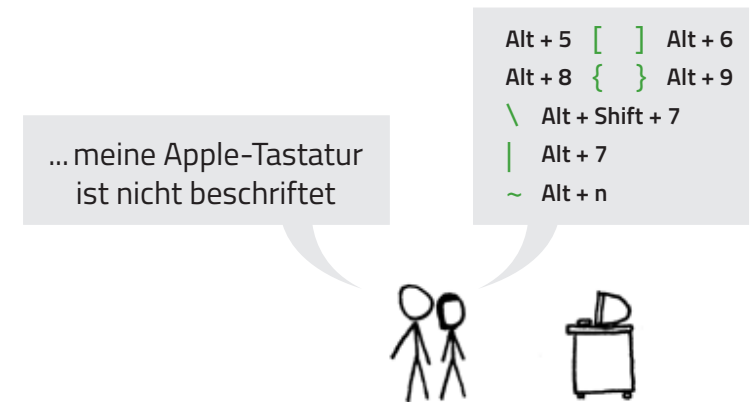
Das Wiederholungszeichen, auch Quantifizierer genannt, bezieht sich immer auf das vorhergehende Zeichen bzw. Zeichenklasse.

- Wiederholungszeichen **?** (Im Menü **Null oder ein Mal**)  
Mit **dass?** z.B. alle Vorkommen von **das** und **dass**
- Wiederholungszeichen **+** (Im Menü **Ein oder mehrere Male**)  
**\h+** mehrere Leerräume
- Wiederholungszeichen **\*** (Im Menü **Null oder mehrere Male** korrekt: Null oder beliebig oft)  
**\d+,\backslash d\*** findet sowohl **15** als auch **14,9999**

## Es ist gierig ...

- Wiederholungszeichen finden die maximale Anzahl Treffer.  
Kombination mit **?** findet die kürzest mögliche Entsprechung.  
**.+\.** Findet alle Sätze eines Absatzes.  
**.+?\.** Findet nur den ersten Satz eines Absatzes.





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

## Mehrfache Leerräume ersetzen

02\_Leerraum.indd

- **Schlechte Lösung:** `\s+`  
Trifft auch Umbruchzeichen.
- **Bessere Lösung:** `\h+`  
Spezialzeichen (Indexmarken, Notizen, XML-Tags ...) werden ggfs. gelöscht.
- **Noch Besser:** `\h(?:=\h)`  
Das `(?:=\h)` ist ein Lookahead.

~~Suche~~  
Ersetze

# Intelligenter ersetzen

Reguläre Ausdrücke bieten die Möglichkeit, das Suchergebnis bei der Ersetzung weiterzuverwenden. Der Zugriff auf die Fundstellen geschieht mit so genannten Rückwärtsreferenzen.

- **Vollständige Fundstelle** in der Ersetzung verwenden  
Suche nach Telefonnummer: `[\d\s+()\.\/\-\-]{6,}`  
Ersetzung durch: `Tel.: $0`
- Bestimmte Bereiche mit Klammern `()` zusammenfassen  
Suche: `(.+)\t(.+)`  
Ersetzung: `$2\t$1`

- 1.»Ein Aufzählungspunkt
- 2.»Der Zweite.
- 3.»Noch ein Punkt.



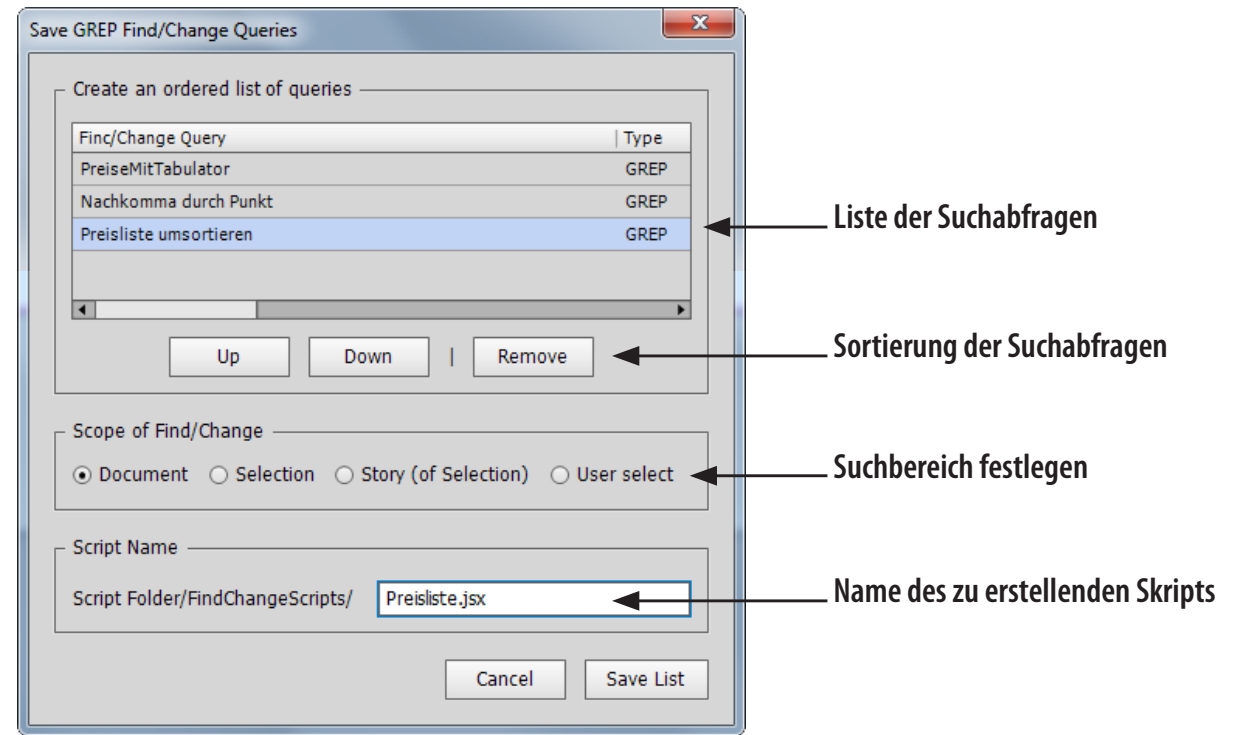
1. Ein Aufzählungspunkt
2. Der Zweite.
3. Noch ein Punkt.

## *Liste automatisch formatieren*

03\_Listen.indd

2. Listen/Aufzählungen finden `^\d+\.\t`
3. Zähler entfernen
4. Absatzformat anwenden
1. Und außerdem: Erstem Listenelement **neu nummerieren** zuweisen! `^1.\t`  
Muss man zuerst machen!

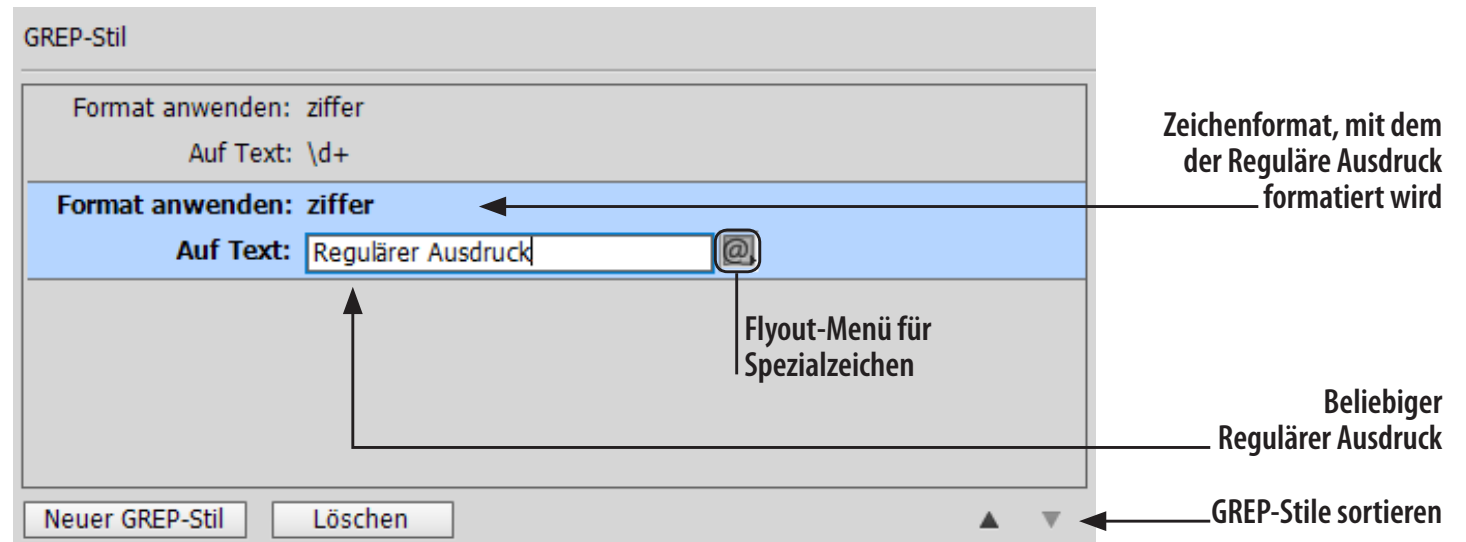
# Ein Skript zur Hilfe...



## ChainGREP.jsx

- GREP-Abfragen im Suchen/Ersetzen-Dialog speichern.
- Skript `ChainGREP.jsx` aufrufen  
<http://www.publishingx.de/download/ChainGREP.jsx>
- Liste der GREP-Abfragen sortieren
- Namen für das zu speichernde Skript eintragen.
- Die Skripte werden im Unterordner FindChangeScripts gespeichert.

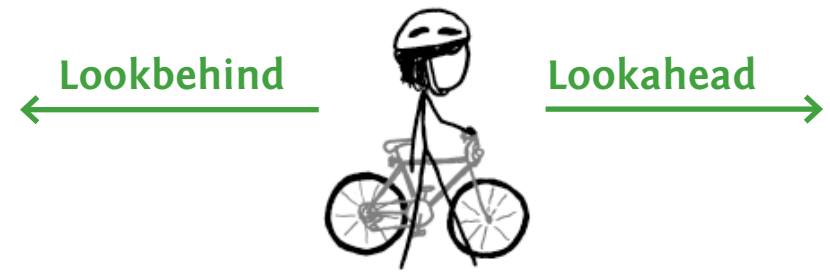
Absatzformat-  
optionen  
Abschnitt  
GREP-Stil



# GREP-Stile

Automatische Formatierung von gefundenen GREPs in Absätzen.

- Ziffernkleinerschaltung
- Produktnamen
- Brüche
- Symbolschriften



Lizenz: | Autor: Randall Munroe

# Ausschau halten

Bei der Verwendung von sogenannten **Look Around Assertions** wird geprüft, ob vor bzw. nach dem eigentlichen Ausdruck ein anderer Regulärer Ausdruck steht. Besonders wichtig für **GREP-Stile**

## Lookbehind

- **Positiv** Brüche: Ziffern nach einem Bruchstrich: `(?<=\/)\d+`
- **Negativ** Preise vor denen nicht EUR steht: `(?<!EUR)\h+\d+, \d\d`
- **Achtung:** Lookbehind **nur** mit fester Länge möglich (Widerholungen nicht einsetzbar)

## Lookahead

- **Positiv** Ziffern denen die Einheit cm folgt: `\d+(?=\hcm)`  
Mehrere Leerräume `\h(?=\h)`
- **Negativ** Geschäftsbericht, AG hinter dem Firmennamen prüfen: `Firma\h+(?!AG)`

19,99 → 19<sup>99</sup>

## *Preisauszeichnung mit GREP-Stilen*

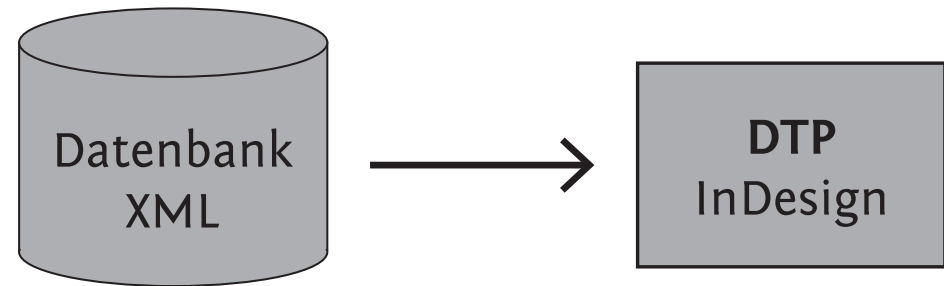
04\_GrepStil.indd

1. GREP-Stil einsetzen.
2. Nachkommastellen lokalisieren `(?<=,)\d\d`
3. Komma weiß schalten `, (?=\d\d)`

# Daten- zusammen- führung







# Was?

## Produkte aus variablen Daten

- **Persönliche Ansprache** der Empfänger  
Mailings, Einladungen, Eindrücke, Verpackungen ...
- **Individuelle Produktangebote**  
Namensschilder, Visitenkarten, ...

# Woher?

## Datenquellen erschließen ...

- Excel-Tabellen, Anmeldedaten, Adresslisten, E-Mail Programm
- PIM/Produktdatenbanken, CRM-Systeme
- ...

## Datenformate

- Excel, XML, Text (z.B. CSV, VCard) oder direkt aus der Datenbank

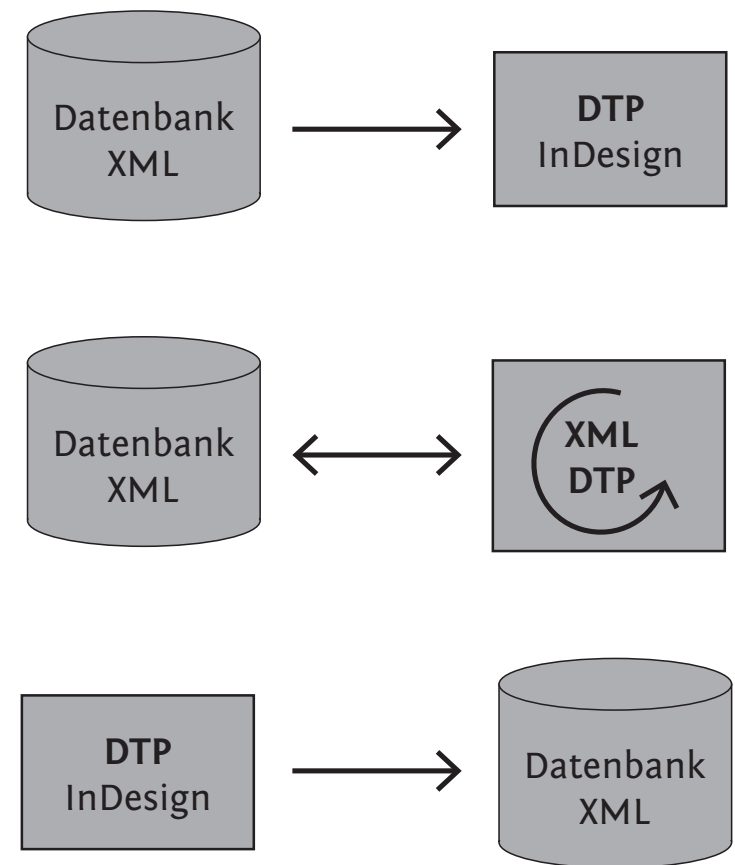
# Wie?

## Buzzword Database Publishing

- Teure Publishing-Systeme vs. Copy & Paste aus Excel
- Datenzusammenführung von InDesign

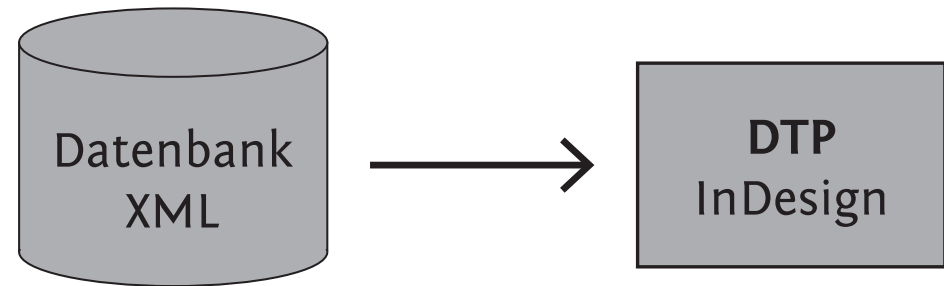
## Grundsätzliche Workflows

- Single Source Publishing
- Bidirektional
- Datenextraktion



# Eigentlich alles ganz einfach

1. Daten aus Excel/Datenbank holen
2. Daten in das Publishing System einspeisen



## *Visitenkarten aus Excel erstellen*

visika\_template.indd

### 1. Datenquelle auswählen

Fenster -> Hilfsprogramme -> Datenzusammenführung 

Nur CSV oder Textdateien möglich

Importoptionen wählen!

### 2. Datenfelder beim Cursor verknüpfen

Testen mit **Vorschau**

### 3. Formatierungen wie normaler Text möglich...

### 4. Ein **Ergebnisdokument** mit mehreren Seiten entsteht

Flyout -> Zusammengeführtes Dokument erstellen

## Bilder

- Bildspalte mit @ (in Excel: ,@ einfügen!) kennzeichnen  
Bilddatei im gleichen Ordner oder vollständigen Pfad angeben

## QR-Codes

- QR-Code-Spalte mit # kennzeichnen
  - Plain Text: Text
  - Hyperlink: **URL:**<http://www.google.com>
  - SMS: **SMSTO:**022198181435:Hallo
  - Email: **MATMSG:**\nTO:johndoe@adobe.com\nSUB:Hi;\nBODY:;;
  - Business Card: **BEGIN:VCARD**\nVERSION:2.1\nN:Smith;John\nFN:John Smith\nORG:Adobe Systems\nTITLE:Engineer\nTEL;CELL:+919876543210\nTEL;WORK;VOICE:123456789\nADR;WORK:;;Street ABC;Seattle;Washington;98101;US\nEMAIL;WORK;INTERNET:abc@adobe.com\nURL:www.adobe.com\nEND:VCARD

# Schon wieder Skripte

- PDF Export pro Seite `exportEveryPageAsPDF.jsx`
- Objekte im Textfluss verankern `inlineMerge.jsxbin`
- Tabelle aufbauen `CSV2TABLES.jsxbin`

# XML und IDML





# Extensible Markup Language

## Motivation?

- Für die **digitale Verarbeitung** wird eine Sprache zur Beschreibung von Dokumentstrukturen benötigt.
- **Lesbar** für Mensch und Maschine.
- Durch vorgegebene Strukturen sind die Inhalte **transformier-** und **konvertierbar**.

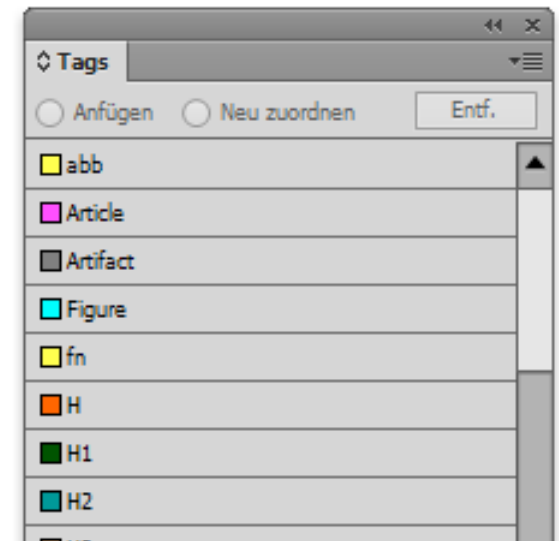
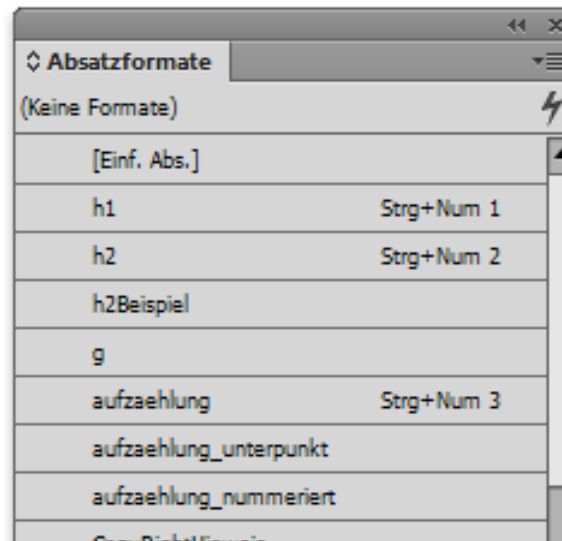
## XML als Standard für Datenstrukturierung

- W3C Recommendation seit dem 10. Februar 1998.
- Sprache zur Beschreibung von **Dokumentstrukturen** bzw. eine Metasprache, die eine abstrakte Syntax für Auszeichnungssprachen definiert.
- Austausch von Dokumenten, Definition von Datenstrukturen.  
Es wird unterschieden: **dokumentzentriert** vs. **datenzentriert**.

## *Produktinformation als XML*

01\_product9902131290.xml

- Strukturiert
- Semantik



# DTP vs. strukturierte Daten

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

## Aber

- Mit **Strukturierung** wurde schon immer das Erfassen des Informationsgehaltes erleichtert. Umgesetzt durch Formatierung wird die Struktur indirekt vom Leser wahrgenommen.
  - **Beispiel:** InDesign Absatz- und Zeichenformate

# XML in InDesign-Dokumenten

Seit **InDesign CS3** kann InDesign XML Daten (sinnvoll) verarbeiten.  
Ab InDesign CS4 sind Tabellen im Textmodus editierbar.

## 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



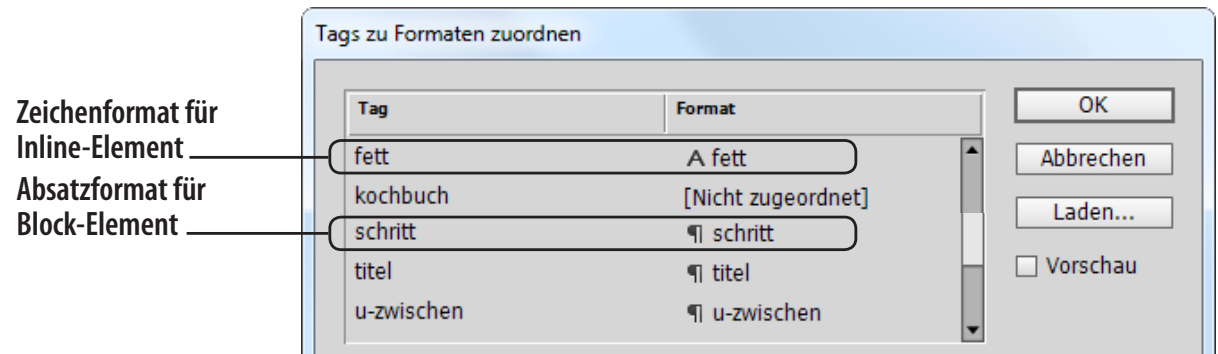
# Wie geht das in InDesign?

- **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



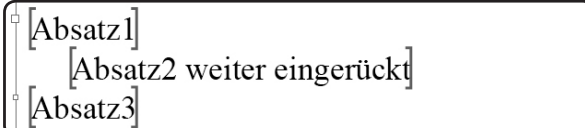
## Formatierung von XML Tags

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

#### XML-Quelldaten

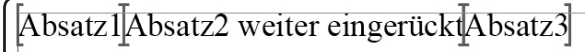
```
<abs>Absatz1</abs>  
  <abs>Absatz2 weiter eingerückt</abs>  
<abs>Absatz3</abs>
```

#### InDesign-Import



[Absatz1]  
[Absatz2 weiter eingerückt]  
[Absatz3]

#### Ohne Whitespace-Entfernen-Option



[Absatz1][Absatz2 weiter eingerückt][Absatz3]

#### Mit angewählter Whitespace-Entfernen-Option

# Whitespace Problem und Lösung.

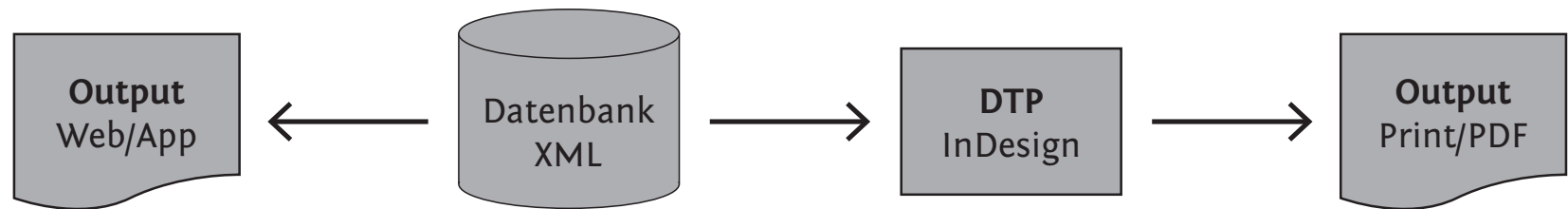
- **Whitespace** wird vollständig **importiert** und dargestellt **oder** wird vollständig ignoriert.
- **Lösung:** Import mit XSLT oder Importskript verwenden. `xmlImport.jsx`



## *Produkt für einen Katalog erstellen*

03\_xml\_template.indd

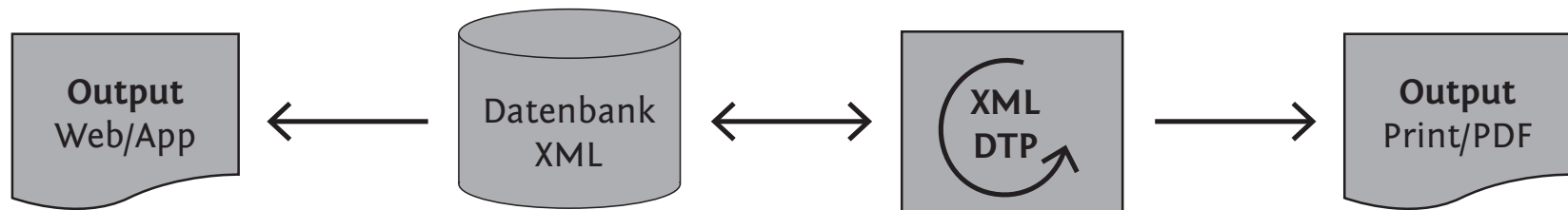
- InDesign Template.
- Formate werden mit XML-Elementen verknüpft.
- Direkte Import: Whitespace fehlerhaft.
- Einsatz des Import-Skripts.



# 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“

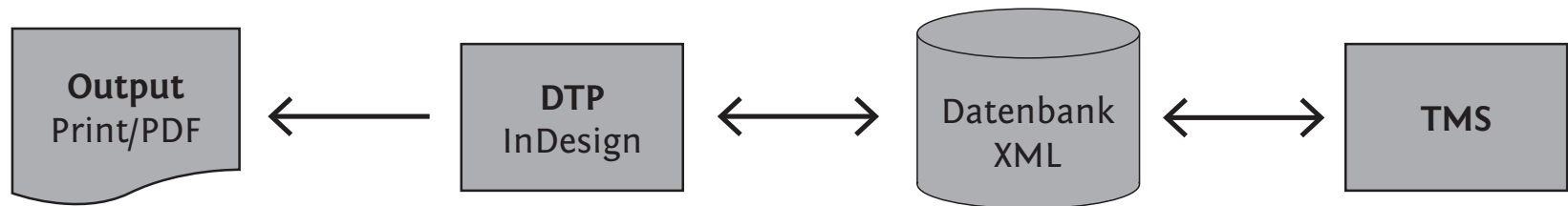
Beispiele sind Database Publishing Plugins und Workflows auf Basis von XML und Scripting.



# 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**.

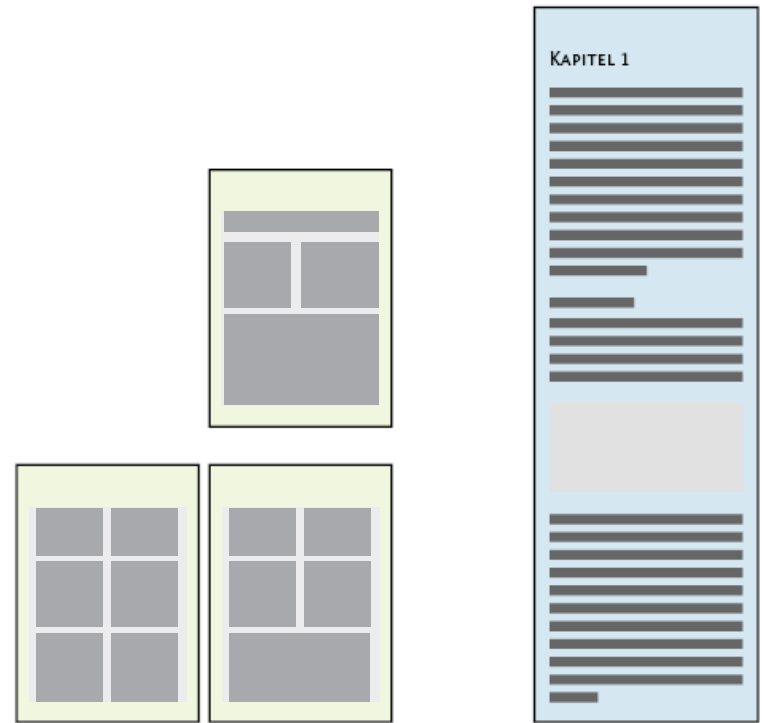
## Verwendung

- **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.

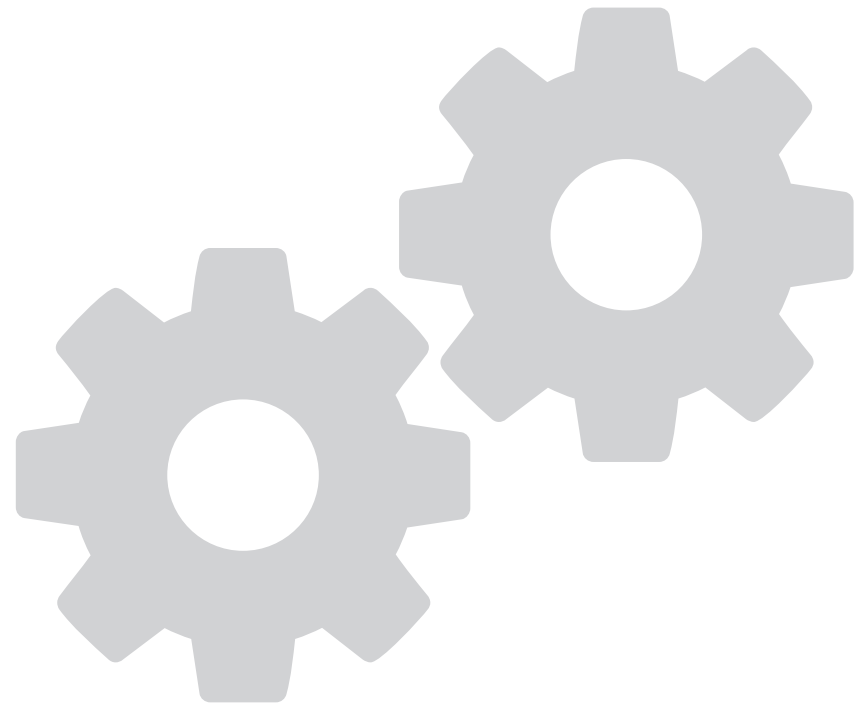
- Lesbar im Texteditor.
- Ohne gerenderte Voransichten.
- Ohne Umbruchinformationen der Render-Engine.
- Automatisierter Aufbau von **Dokumenten** mit XML-Daten.
- Maßeinheit ist immer **Points** (= 72 pt pro Inch).
- Minimale Angaben reichen aus (IDML-Voreinstellungen).
- Erstellt sich aus den gleichen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



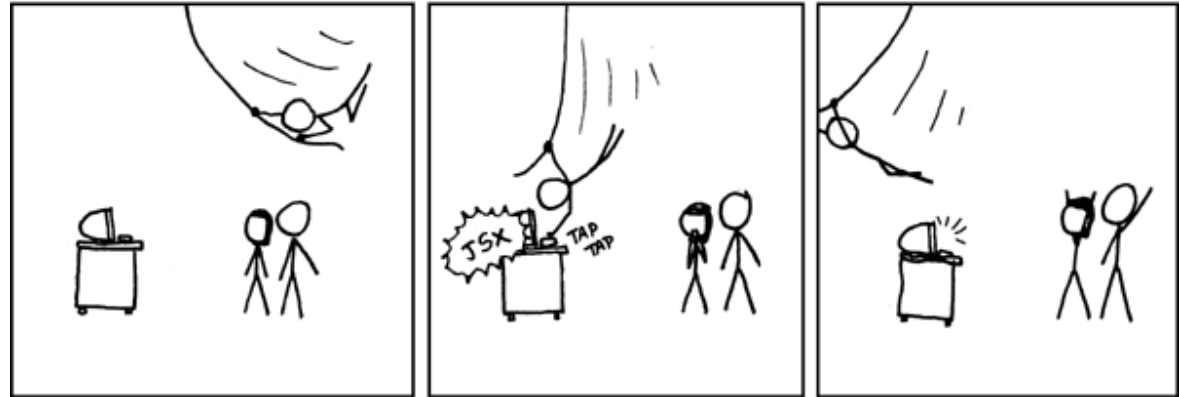
# Was geht nicht so gut?

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

# 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 ...

## ... 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.**

## *Ein paar Beispiele ...*

- Sprechblasen mit `Speech.jsx`
- Abweichungen anzeigen `showOverrides.jsx`
- Abweichungen löschen `clearOverrides.jsx`
- PDF Importieren `MultiPageImporter2.5-CS5.jsx`
- Mehrere Schritte rückgängig `UndoHistory.jsx`
- Analyse der verwendeten Formate `formatliste.jsx`

# 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.**

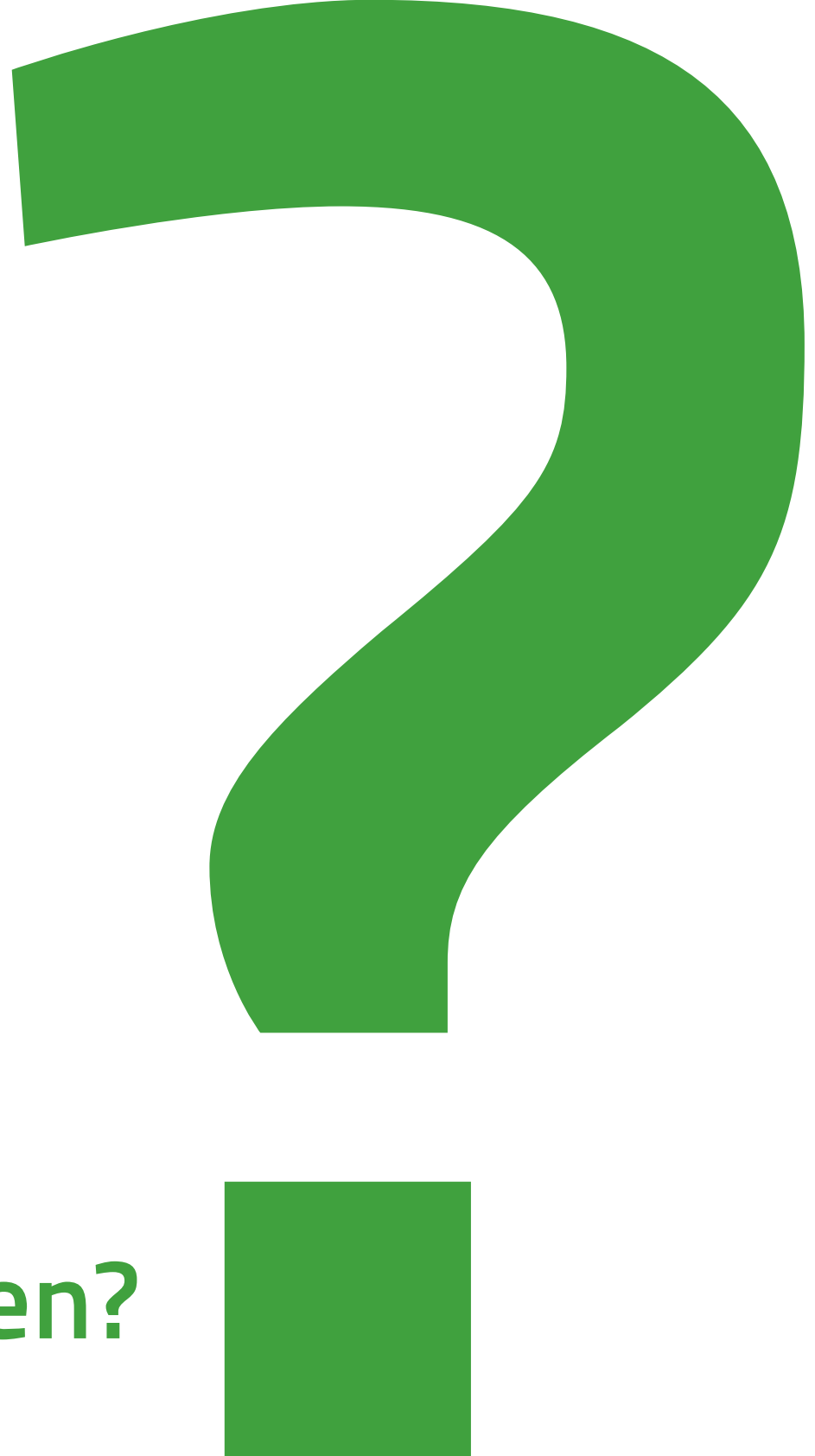
## 3. Entwickeln lassen

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!

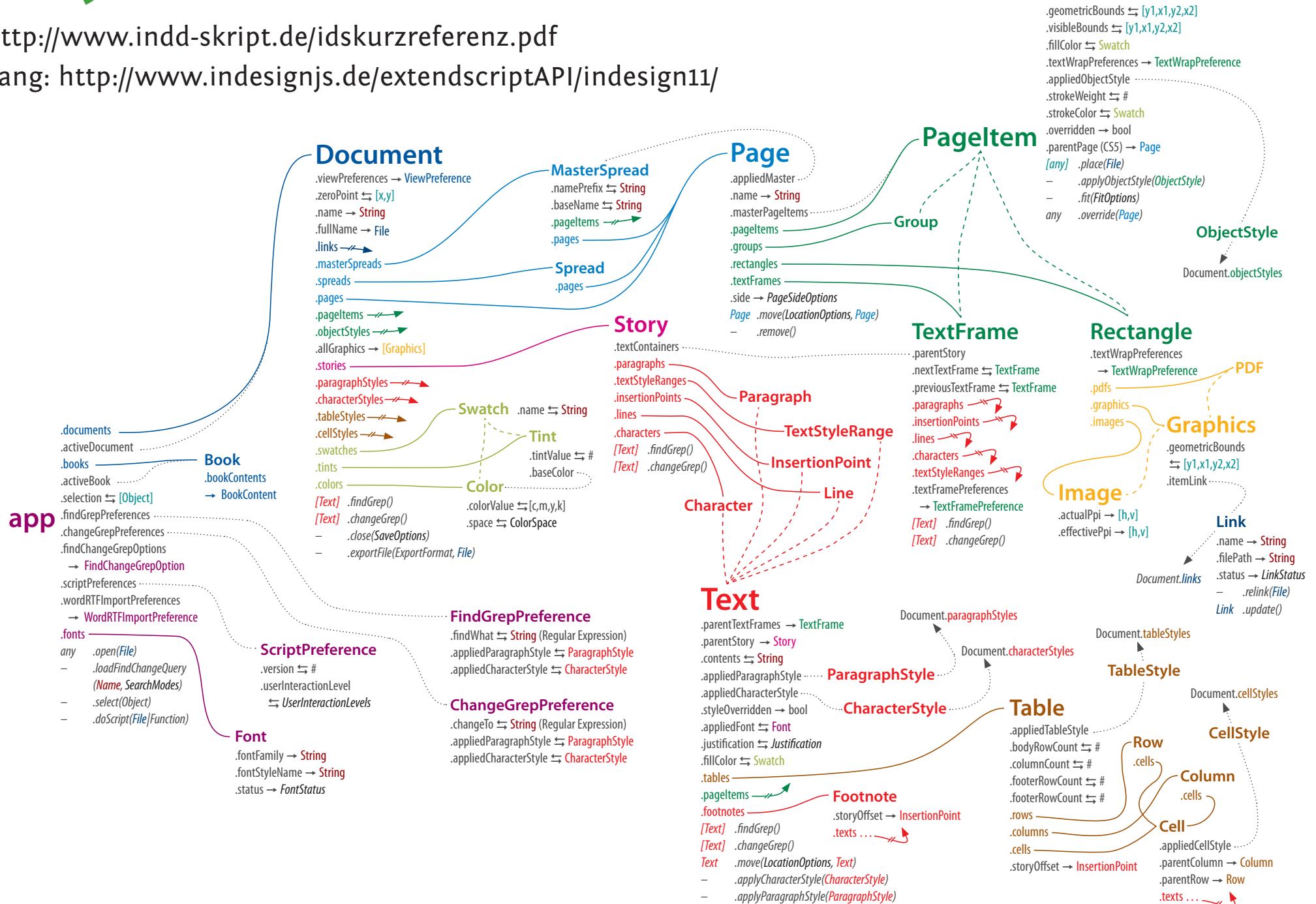
```
// Achtung: In InDesign muss mindestens ein Dokument geöffnet sein  
var tf = app.activeDocument.pages[0].textFrames.add();  
tf.geometricBounds = [10,10,100,100];  
tf.contents = "Hallo InDesign";
```



# Objektmodell Kurzreferenz

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

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



## *Katalog zusammenstellen*

01\_catalogTemplate.indd und placeSnippets.jsx

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

# In eigener Sache

## InDesign automatisieren – Skripting, GREG & Co. 2. Auflage

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

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



**Vielen Dank für Ihre Aufmerksamkeit!**

**Fragen und Anregungen?**

E-Mail: [gregor.fellenz@publishingx.de](mailto:gregor.fellenz@publishingx.de)

Twitter: [grefel](#)

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