

2., aktualisierte und
erweiterte Auflage

Gregor Fellenz

InDesign automatisieren

Keine Angst vor Skripting, GREP & Co.

dpunkt.verlag



Gregor Fellenz studierte nach seiner Ausbildung zum Mediengestalter an der Hochschule der Medien in Stuttgart Druck- und Medientechnik. Während des Studiums konzentrierte er sich auf die Themen XML und Cross-Media-Publishing.

Seit 2004 beschäftigt er sich beruflich mit Publishing-Workflows für Print und digitale Medien. Sein Schwerpunkt liegt auf der automatisierten Erstellung von Publikationen mit InDesign und XML. Neben der Implementierung von Publishing-Workflows ist er als Berater, Trainer und Projektleiter tätig. Unter @grefel twittert er über die aktuellen Entwicklungen im Bereich InDesign-Skripting.

Gregor Fellenz

InDesign automatisieren

**Keine Angst vor Skripting, GREP & Co.
2., aktualisierte und erweiterte Auflage**

Lektorat: Barbara Lauer, Bonn
Copy-Editing: Alexander Reischert (Redaktion Aluan, Köln)
Satz: Bidirektionaler XML-Workflow, Gregor Fellenz
Herstellung: Susanne Bröckelmann
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: Media-Print Informationstechnologie, Paderborn
Bildnachweis: Nora Klein: Seite 2; Teresa Wurtz: Seite 295; Gregor Fellenz: alle weiteren.

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-86491-683-0

2. Auflage 2015
Copyright © 2015 dpunkt.verlag GmbH
Wieblinger Weg 17
69123 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Papier
plus⁺
PDF.

Zu diesem Buch – sowie zu vielen weiteren dpunkt.büchern –
können Sie auch das entsprechende E-Book im PDF-Format
herunterladen. Werden Sie dazu einfach Mitglied bei dpunkt.plus⁺:

www.dpunkt.de/plus

Inhalt

Einleitung	9
Teil I Einstieg in die InDesign-Automatisierung	17
1 Schöner suchen und ersetzen mit GREP	18
1.1 Der Suchen/Ersetzen-Dialog	19
1.2 Die Suche mit Regulären Ausdrücken	21
1.2.1 Zeichen mit spezieller Bedeutung	21
1.2.2 Variable Zeichen auswählen	22
1.2.3 Sonderzeichen	26
1.2.4 Wiederholungen	28
1.2.5 Genaue Positionen ermitteln	30
1.2.6 Noch mehr finden	31
1.3 Praxisbeispiel Preisliste	32
1.4 Intelligenter ersetzen	33
1.5 GREP-Stile	35
2 Skripte in InDesign verwenden	36
2.1 Skripte installieren	36
2.2 Versteckte Perlen – InDesigns Skripte	38
2.3 Skripte zum Download	39
2.3.1 Skriptsammlungen	39
2.3.2 Skripte und Informationen	40
3 Adobe ExtendScript Toolkit	41
4 Skripte verstehen und verändern	46
4.1 Die wunderbare Welt der Programmierung	46
4.2 Hello World	48
4.3 Textrahmen formatieren	49
4.3.1 Rahmengrößen anpassen	55
4.3.2 Seitenabhängige Formatierung	59
4.4 Spaltenbreite festlegen	61
4.5 Mit Text arbeiten	69
4.6 Bilder auf Blitzler prüfen	75
4.7 Seiten und Objekte finden	83
4.7.1 Objekte mit Namen versehen	86
4.7.2 Objekte von der Musterseite lösen	88

4.8	Suchen und Ersetzen per Skript	90
4.9	Suchen, finden und verändern	95
5	Fehlersuche	102
5.1	Häufige Fehler und Lösungen	102
5.2	Hilfe finden	105
Teil II InDesign-Programmierung mit JavaScript		107
6	Einführung in JavaScript	108
6.1	Warum JavaScript?	108
6.2	Was ist JavaScript?	108
6.3	Wie es aussehen muss – Syntax	109
6.4	Variablen	111
6.5	Zahlen und Zeichenketten	114
6.6	Datenkraken – Objekte	119
6.7	Entscheidungen treffen – Abfragen und Verzweigungen	122
6.8	Arbeit wegschaffen – Schleifen	127
6.9	Arrays	130
6.10	Schleifen über Arrays und Sammlungen	132
6.11	Eigene Funktionen und Methoden	136
6.12	Etwas ausprobieren – Fehlerbehandlung	140
6.13	Mit Dateien arbeiten	142
6.13.1	Textdateien einlesen und erstellen	145
	Exkurs Unicode	148
7	InDesign-Objektmodell	150
7.1	Objekte und Klassen	150
7.2	Das InDesign-Objektmodell verstehen	151
7.2.1	Aufbau des Objektmodells	152
7.2.2	Im Objektmodell navigieren	156
7.3	Objekte analysieren	160
7.4	Sammlungen	162
7.5	Gemeinsame Methoden und Eigenschaften	167
7.6	Voreinstellungen	168
7.6.1	Voreinstellungen für das Skripting	169
7.6.2	Dokument- und Ansichtseinstellungen	170
7.6.3	Einstellungen für Import und Export	172
7.7	Seiten und Mustervorlagen	174
7.8	Rahmen und Seitenobjekte	177
7.9	Textrahmen	182
7.10	Mit Texten arbeiten	185
7.10.1	Formatvorlagen	186
7.11	Tabellen	188
7.11.1	Tabellen- und Zellenformate	191
7.12	Bilder und Verknüpfungen	192

7.13	Interaktive Seitenobjekte	194
7.14	Suchen und Ersetzen	196
7.15	Dokumente	199
7.16	Benutzerinteraktion und Dialogfenster	201
7.17	Target und Session	206
7.18	Events	206
8	Debugging mit dem ESTK	209
9	Programmierkonzepte	213
9.1	Format- und Mustervorlagen	213
9.2	Schritte der Skripterstellung	215
9.3	Best Practice	216
9.4	Ein Skript-Template nutzen	221
Teil III Automatisierung in der Praxis		223
10	Noch mehr finden mit GREP	224
10.1	GREP-Abfragen automatisieren	224
10.2	Die besten GREP-Tools	225
10.3	Grenzen und Übergänge	226
10.4	Doppelte Wörter finden	227
10.5	GREP-Performance	227
10.6	Ausschau halten	228
10.7	Unicode und GREP	232
10.8	Weitere Zeichenklassen	233
10.8.1	Posix-Ausdrücke	234
10.8.2	Unicode-Properties	234
10.9	Mit der Zwischenablage arbeiten	235
10.10	GREPs formatieren und kommentieren	235
10.11	Vornamen abkürzen	236
10.12	Anführungszeichen und Apostrophe	237
10.13	Zifferngruppen bilden	240
10.14	Festabstände	241
10.15	GREP-Stile in alle Absatzformate kopieren	243
10.16	Leerraum vereinheitlichen	244
11	Skripting-Kochrezepte	249
11.1	Laufweite und Umbruch	249
11.1.1	Zeilen einsparen	249
11.1.2	Formatattribute außerhalb eines Bereichs	254
11.2	Marginalien	256
11.3	Transparente Absatzlinien	259
11.4	Fußnoten in Endnoten umwandeln	262
11.5	Bilder und Metadaten	267
11.5.1	Bildunterschriften aus Metadaten	268
11.5.2	Bildquellenverzeichnis erstellen	269

11.6	Bildunterschriften interaktiv	272
11.7	Tabellen suchen und ersetzen	275
11.8	CSV-Dateien importieren	278
11.9	Zoom per Skript	283
11.10	Alle Dokumente schließen	283
11.11	Stapelverarbeitung eines Ordners	284
11.12	Backup beim Speichern anlegen	287
11.13	Eigene Einträge im Menü erstellen	290
	11.13.1 Menü-Einträge	290
	11.13.2 Skripte beim Start von InDesign laden	293
	11.13.3 Menü-Befehle per Skript ausführen	294
12	Skript-Workflow mit Word-Dateien	295
12.1	Vorhandene Daten und Zielstellung	296
12.2	Das Skript planen	297
12.3	Word-Dateien importieren	299
12.4	Formatierte Texte auswerten	301
12.5	Den Index generieren	308
13	Dokumente für E-Books optimieren	310
13.1	Reflowable E-Books im EPUB-Format exportieren	311
	13.1.1 Aufbau einer EPUB-Datei	311
	13.1.2 EPUB aus InDesign exportieren	313
13.2	Problematische Zeichen prüfen	318
13.3	Lokale Formatabweichungen	325
13.4	Format- und Dateinamen in Ordnung bringen	327
	13.4.1 Formatnamen aufräumen	327
	13.4.2 Namen von Verknüpfungen bereinigen	330
13.5	Verwendete Formate anzeigen und einsammeln	333
13.6	Bilder im Textfluss verankern	335
13.7	Bilder-Preflight für EPUB	338
14	InDesign und XML	341
	Exkurs XML-Grundlagen	342
14.1	XML in InDesign	346
14.2	Zusammenspiel von XML und InDesign	355
14.3	Skripting mit XML	359
14.4	XML-Elemente mit XPath suchen	365
	Exkurs Elemente mit XPath adressieren	365
	14.4.1 Skripting mit XPath	366
	Anhang	371
A1	GREP-Referenz	371
A2	Ressourcen und Literatur	379
A3	Index	381

Einleitung

Vorwort zur 2. Auflage

Nachdem sich Anfang 2014 abzeichnete, dass die erste Auflage bald ausverkauft sein würde, sagte ich zu meiner Lektorin Barbara Lauer, dass ein Update schnell gemacht wäre. Knapp ein Jahr später weiß ich, dass diese Einschätzung falsch war. Auch wenn der bewährte Aufbau des Buches grundlegend gleich geblieben ist, sind nur wenige Seiten von einer Überarbeitung gänzlich verschont worden.

Das hat verschiedene Gründe: Zunächst natürlich viele Neuigkeiten und Erleichterungen in InDesign CS6 und CC – alle Referenzen auf ältere Versionen habe ich entfernt, damit das Buch übersichtlich bleibt. Aber auch Erfahrungen aus Schulungen, Gesprächen und Feedback zur ersten Auflage wollten eingearbeitet werden.

Besonders zu erwähnen ist das aus meinem Schulungskonzept übernommene Beispieldokument, das die Einführung in JavaScript in Kapitel 6 begleitet. Einige Kochrezepte sind entfallen, dafür neue Beispiele aus meiner täglichen Praxis hinzugekommen – erwähnenswert ist der Umgang mit CSV, die Realisierung von Stapelverarbeitungen sowie das Verketteten von GREP-Abfragen. Außerdem wurde das Buch mit Beispielen zum Skripting der interaktiven Funktionen erweitert.

Im Bereich E-Book hat sich so viel verändert, dass das komplette Kapitel ausgetauscht werden musste. Hier habe ich den Fokus auf das Skripting des Exports gelegt, die eigentliche Erstellung von E-Books musste entfallen – hier gibt es inzwischen viele andere gute Quellen.

Mehr aus InDesign herausholen

Adobe InDesign hat sich zu einem der wichtigsten Desktop-Publishing-Programme entwickelt und ist aus der professionellen Gestaltung und Produktion von Druckerzeugnissen kaum mehr wegzudenken. Die meisten Anwender verwenden allerdings nur die offensichtlichen und gebräuchlichsten Funktionen und Features. Viele Möglichkeiten gehen in der Fülle der Funktionen unter oder es bleibt schlicht keine Zeit, sich während des Tagesgeschäfts damit auseinanderzusetzen. Selbst gestandene InDesign-Profis lernen immer wieder neue Arbeitsweisen hinzu.

Die Produkte, die mit InDesign erstellt werden, sind außerordentlich vielfältig. Die Bereiche der professionellen Anwendung gehen von der klassischen Druckvorstufe über die Erstellung von digitalen Medien bis zu serverbasierten Web2Print-Lösungen. Entsprechend vielseitig sind auch die Szenarien und Workflows, in denen InDesign eingesetzt wird.

Wer sollte InDesign automatisieren?

Eine Automatisierung lohnt sich meist nur, wenn wiederkehrende Aufgaben zu lösen oder große Datenmengen zu verarbeiten sind. Klassische Beispiele sind die Katalogproduktion, technische Dokumentationen oder die Buchherstellung. Aber auch die Erstellung von Zeitschriften, Magazinen, Broschüren sowie ganz allgemein von Periodika lässt sich durch Automatisierung gut unterstützen. Im Bereich der digitalen Medien können sowohl E-Books als auch digitale Magazine mit Hilfe von Skripting effizienter erstellt werden.

Es liegt in der Natur der Sache, dass sich das kreative Gestalten eines neuen Corporate Designs, einer Werbeanzeige oder Ähnlichem kaum automatisieren lässt. Doch selbst in diesem Bereich lassen sich mit Skripting-Kenntnissen Variationen einer Gestaltung erstellen, beispielsweise kann man generell alle Schriftgrößen um einen bestimmten Prozentwert verringern. Denn auch hier gilt: Statt ständig zehnmal klicken, lieber einmal programmieren.

Unabhängig davon kann eine Automatisierung auch die regelbasierte Gestaltung unterstützen, da Ausnahmen einen höheren Aufwand bedeuten und dadurch neu hinterfragt werden. Es muss auch nicht immer eine ausgefeilte Layoutautomatisierung sein – manchmal helfen schon kleine Skripte oder das Wissen über die beeindruckenden Möglichkeiten der Suchen/Ersetzen-Funktion mit GREP weiter.

Suchen und ersetzen mit GREP

GREP bietet unabhängig vom Skripting eine oft übersehene Möglichkeit, Anforderungen, die an Dokumente gestellt werden, schnell und effizient zu lösen. Mit GREP können komplexe Suchanfragen und Ersetzungsanweisungen definiert werden, die den Arbeitsalltag ganz erheblich erleichtern. Dazu werden Reguläre Ausdrücke verwendet, die auch in anderen Programmen zum Einsatz kommen.

Skripting

Eine weitere Möglichkeit ist die Programmierung von InDesign mit selbst erstellten Skripten. Wer »langweilige« oder sich immer wiederholende Arbeiten automatisieren kann, spart viel Zeit, um sich auf die wesentlichen Aufgaben zu konzentrieren. Ich würde sogar sagen, dass die Möglichkeit der Programmierung oder besser gesagt des Skriptings das beste und produktivste Feature von InDesign ist.

Fast alles, was Sie mit der normalen Benutzeroberfläche realisieren können, kann auch über ein Skript erreicht werden. Ob Sie eine neue Seite anlegen, einen Textrahmen aufziehen oder einen Absatz formatieren wollen – per Skript ist das alles kein Problem. Skripte erlauben sogar noch mehr: Mit ihnen kann man eigene Menüs definieren oder auf Benutzeraktionen reagieren.

Neben GREP und Skripting wird die Zweitverwertung von gedruckten Inhalten in digitaler Form immer wichtiger. Die hier verwendeten Technologien EPUB für E-Books und XML als Schnittstelle zu strukturierten Datenbeständen oder sogar Datenbanken werden meist mit Hilfe von Skripten optimiert oder überhaupt erst möglich gemacht.

E-Books und XML

Gerade in diesem Bereich zeigt sich auch die Entwicklung hin zu schnelleren Produktionszyklen und dem damit einhergehenden Zeit- und Kostendruck. Oft müssen die elektronischen Publikationen zeitgleich mit dem Printwerk erhältlich sein. Mit Kenntnissen im Bereich Automatisierung kann man diesen Entwicklungen entspannt entgegensehen.

Das Skripting von InDesign ist in den Versionen CS6 bis CC 2014 weitgehend identisch. Das hier vorgestellte JavaScript kann plattformunabhängig auf Windows und Mac OS eingesetzt werden.

Welche Versionen werden beschrieben?

Wenn sich Unterschiede in der Programmierung ergeben, sind diese im Buch durchgehend gekennzeichnet. Beispieldaten wurden gegebenenfalls für alle notwendigen Versionen erstellt.

Jeder kann InDesign automatisieren! Das Buch setzt lediglich solide InDesign-Grundkenntnisse voraus. Erfahrungen in einer Skript- oder Programmiersprache sind von Vorteil, werden aber nicht vorausgesetzt. Ich habe den Anspruch, auch Anfänger auf dem Weg zur InDesign-Automatisierung zu begleiten!

An wen richtet sich das Buch?

Die Einführung in die Automatisierung kann jedem gelingen. Die Beispiele sind aus dem Arbeitsalltag von Mediengestaltern, Grafikern und Verlagsherstellern entnommen. Aber auch Medieninformatiker oder Webdesigner werden viele praxisorientierte Hinweise für die Produktion von Printmedien finden.

Viele Anwender schrecken vor der Hürde der Programmierung zurück. Doch liegen gerade darin unermessliche Möglichkeiten, sich das Arbeitsleben einfacher zu machen. Etwas überspitzt könnte man sagen: »Nur ein fauler Mensch macht produktive Skripte.« Denn wer kein Problem damit hat, hundertmal denselben Arbeitsschritt auszuführen, anstatt einen Kaffee zu trinken, der braucht auch keine Automatisierung.

Wer kann skripten?

Zugegeben, das erste Mal sind die nackten Codezeilen eher abschreckend und man weiß oft nicht, wo man anfangen soll. Wer sich allerdings auf die Reise begibt, wird bald feststellen, dass das Themengebiet logisch aufgebaut ist und sich mit dem richtigen Einstieg meistern lässt.

Außerdem macht Programmieren Spaß. Das glauben Sie nicht? Sobald Sie Ihren eigenen Skripten bei der Arbeit zuschauen, werden Sie Ihre Meinung ändern. Lassen Sie sich nicht vom ersten Fehler entmutigen – das Verständnis fürs Programmieren entwickelt man leider nicht über Nacht.

JavaScript In diesem Buch wird das Skripting anhand der Skriptsprache JavaScript erläutert, die sowohl auf Windows- als auch Mac OS-Systemen eingesetzt werden kann. Genauer gesagt wird die erweiterte JavaScript-Implementierung von Adobe mit dem Namen ExtendScript verwendet.

JavaScript wird auch für die Programmierung von Webseiten benutzt. Leider ist nur der Sprachkern identisch, der Umgang mit dem Browser bzw. InDesign aber völlig verschieden.

Danksagung

Ich möchte mich bei allen, die an der Entstehung dieses Buches beteiligt waren, bedanken. Dazu zählt vor allem Rebecca, die das Projekt von der ersten Idee bis zur Endkorrektur unterstützt hat.

Besonders hervorheben möchte ich auch meine Lektorin Barbara Lauer, die mir stets motivierend und konstruktiv zur Seite stand, die Gutachter der ersten Auflage Sarah Schäfer, Nadine Thiele, Marco Morgenthaler, Marko Hedler, Martin Fischer, Gerald Singelmann und Tobias Fischer, die mir viele wichtige Anregungen und Hinweise gegeben haben. Ein Dank gilt auch meinen Eltern, die das komplette Manuskript Korrektur gelesen haben. Für die zweite Auflage kamen wichtige Inspirationen von Yves Apel, Klaas Posselt, Kai Rübsamen, Uwe Laubender und Stefan Göbel.

Viele weitere Personen, die Anregungen für einzelne Skripte geliefert haben, werden im Verlauf des Buches genannt.

Verwendung dieses Buchs

Das Buch ist als Praxis- und Lernbuch aufgebaut, das didaktische Konzept beruht auf den Erfahrungen verschiedener Schulungen und meiner Lehrveranstaltung an der Hochschule der Medien in Stuttgart. Alle Skripte werden besprochen und, wann immer möglich, durch Beispieldokumente erläutert. Unabhängig davon kann vor allem der zweite Teil auch als Nachschlagewerk verwendet werden.

Das Buch ist in drei Teile gegliedert. Der erste Teil bietet einen sanften Einstieg in die Programmierung und Automatisierung von InDesign. Der zweite Teil ist eine systematische Einführung in die InDesign-Programmierung mit JavaScript. Der dritte Teil enthält praxisorientierte Kochrezepte für konkrete Problemstellungen.

I. Teil

Der Einstieg über die Suche und Ersetzung mit GREP in **Kapitel 1** bietet eine in sich geschlossene Heranführung an die abstrakte Formulierung von Lösungen, wie sie auch beim Skripting üblich ist. **Kapitel 2** zeigt, wie Skripte in InDesign installiert werden und wie Sie diese produktiv einsetzen können. In **Kapitel 3** wird das ExtendScript Toolkit, das Werkzeug für die Erstellung von Skripten, vorgestellt. In **Kapitel 4**

werden einfache Skripte für alltägliche Praxisprobleme im Detail besprochen. Alle Skripte werden zeilenweise diskutiert und können anhand von Beispieldokumenten ausprobiert werden. Dieses Kapitel enthält alle wichtigen Programmiermethoden, behält aber den Fokus auf einem möglichst sanften Einstieg. An vielen Stellen verweist es auf den systematischen Einstieg im zweiten Teil des Buchs. **Kapitel 5** enthält nützliche Tipps zur Fehlersuche und verrät, wo man Hilfe bekommen kann.

II. Teil

Der zweite Teil beginnt in **Kapitel 6** mit einer systematischen Einführung in die Programmiersprache JavaScript. Hier werden alle wichtigen Programmiermethoden und -konzepte für die InDesign-Automatisierung vorgestellt. Das Kapitel schließt mit einem Exkurs zum Thema Unicode, dem Standard für die Kodierung von Zeichen in Schriften und Dateien. **Kapitel 7** dreht sich um das InDesign-Objektmodell. Dieses Kapitel enthält den Schlüssel zur InDesign-Programmierung, es ist als praxisorientierte Referenz aufgebaut. Zu allen Themen finden Sie wieder dokumentierte Beispielskripte.

In **Kapitel 8** wird das Debugging, das »Fehlerfinden«, mit dem ExtendScript Toolkit besprochen. **Kapitel 9** stellt Programmierkonzepte für die Automatisierung von Printprodukten vor und zeigt Best-Practice-Beispiele für gute Skripte.

III. Teil

Der dritte Teil ist thematisch in fünf Bereiche aufgeteilt. Zunächst wird das Thema GREP in **Kapitel 10** vertieft und mit Praxisbeispielen abgerundet. Hier sei auch auf die GREP-Referenz Anhang A1 verwiesen, die alle Metazeichen für die Suche mit GREP enthält.

Kapitel 11 enthält Rezepte für konkrete Aufgabenstellungen. Sie können die einzelnen Unterkapitel losgelöst voneinander bearbeiten. Für alle Skripte gibt es Beispieldokumente.

In **Kapitel 12** wird ein umfassender Automatisierungs-Workflow mit Word-Dateien vorgestellt. Hier wird eine Word-Datei per Skript in ein nahezu fertig gestaltetes InDesign-Dokument überführt.

Kapitel 13 stellt den Export für E-Books im Format EPUB vor. Der Schwerpunkt liegt auf Skripten für die Optimierung des Exports.

In **Kapitel 14** wird der Umgang von InDesign mit XML vorgestellt. Neben den normalen Programmfunktionen werden auch die Möglichkeiten per Skripting erörtert.

Wegweiser durch das Buch

Für einen umfassenden Einstieg können Sie Teil I und II durchgehend lesen. Alternativ können Sie das Buch auch mit dem Schwerpunkt GREP bzw. Skripting durcharbeiten.



Schwerpunkt GREP Falls Sie sich zunächst nur für die Suche und Ersetzung mit Regulären Ausdrücken interessieren, ist Kapitel 1 der richtige Einstieg für Sie. Im Unterkapitel 10.1 wird gezeigt, wie man mehrere GREP-Abfragen hintereinander ablaufen lassen kann. Weitergehende GREP-Techniken finden Sie in Kapitel 10, das sich in einen Theorie- (→ Unterkapitel 10.2 bis 10.10) und Praxisteil aufteilt (→ Unterkapitel 10.11 bis 10.16). Eine vollständige GREP-Referenz finden Sie auf Seite 371.

Schnelleinstieg Skripting Wenn Sie bereits Programmiererfahrung haben und direkt mit dem Skripting beginnen wollen, ist der folgende Fahrplan empfehlenswert: Informieren Sie sich in Unterkapitel 2.1 über Installation und Ausführung von Skripten. Springen Sie dann zu Kapitel 3 und erlernen Sie die Verwendung der Entwicklungsumgebung ExtendScript Toolkit. Nach einem kurzen Abstecher zum Skript *Hallo Welt* im Unterkapitel 4.2 können Sie sich in Teil II systematisch in die Programmierung mit JavaScript einarbeiten. Auf wichtige Techniken, die im ersten Teil vorgestellt wurden, wird im Text verwiesen. Bei Ihrem ersten Skriptfehler springen Sie zurück zu Kapitel 5, das sich mit häufigen Fehlern beschäftigt.


Schnelleinstieg JavaScript-Profi Wenn Sie schon Erfahrung in JavaScript gesammelt haben, empfehle ich die folgenden Kapitel: Nach der Übersicht über die Installation von Skripten in Unterkapitel 2.1 informieren Sie sich in Kapitel 3 über die IDE ExtendScript Toolkit. Nun können Sie direkt in die Beschreibung des InDesign-Objektmodells in Kapitel 7 einsteigen. Bei Ihrem ersten Skriptfehler springen Sie nochmal zurück zu Kapitel 5, das sich mit häufigen Fehlern beschäftigt. Werfen Sie auch ein Blick auf die Best-Practice-Konzepte in Kapitel 9.

Webseite zum Buch

Für dieses Buch habe ich die Webseite <http://www.indesignjs.de> eingerichtet. Hier finden Sie alle digitalen Informationen zum Buch sowie aktuelle Neuigkeiten. Über meinen Twitter-Account *@grefel* veröffentliche ich außerdem aktuelle Informationen zum Buch und zur InDesign-Automatisierung



Linkverkürzer Da niemand zeilenlange Internetadressen aus einem Buch abtippen möchte, habe ich für die meisten Adressen einen Linkverkürzer eingerichtet. Sie erkennen diese Links an dem  Symbol. Die darauffolgende Zahl müssen Sie mit der Adresse *indesignjs.de* kombinieren. Für 18 müssen Sie also <http://www.indesignjs.de/18> in die Adresszeile des Browsers eingeben.

Skripte und Beispieldaten


Ich habe alle Skripte und die Beispieldaten, die im Buch erwähnt werden, auf GitHub zur Verfügung gestellt. Sie können die Daten unter <https://github.com/grefel/indesignjs> 2 herunterladen.

Mit  Download ZIP alle Daten als ZIP-Archiv herunterladen

Schriften

In allen Beispieldokumenten werden die Schriften *Alegreya* und *Alegreya Sans* von Juan Pablo del Peral verwendet  139. Sie können bei www.fontsquirrel.com  140 kostenfrei heruntergeladen werden.

InDesign-Skripting-Kurzreferenz

Zum Buch gehört eine InDesign-Skripting-Kurzreferenz mit einer Übersicht der wichtigsten Objekte, Eigenschaften und Methoden des InDesign-Objektmodells. Sie können ein PDF unter <http://www.indesignjs.de/idskurzreferenz.pdf>  1 herunterladen.

... und außerdem

Die notwendige Reduktion der Komplexität führt leider manchmal dazu, dass die Skripte nur in den beschriebenen Situationen funktionieren. Um unerwünschte Nebeneffekte auszuschließen, sollten Sie die Skripte nur dann in einem produktiven Umfeld einsetzen, wenn Sie verstanden haben, wie diese genau funktionieren. Spätestens am Ende des ersten Teils sollten Sie die Fähigkeit erlangt haben, die Wirkungen und Nebenwirkungen von Skripten zu verstehen, aber vor allem können Sie dann die Skripte an Ihre eigenen Anforderungen anpassen!

In vielen Fällen kommt man bei der Automatisierung mit Technologien in Berührung, über die man ein eigenes Buch schreiben könnte. Damit das Buch auch für Leser ohne Vorkenntnisse lesbar bleibt, habe ich an diesen Stellen kurze Exkurse eingefügt. Diese bieten nur einen minimalen Einstieg und ersetzen nicht das Studium der Technologie. Im Anhang finden Sie Internetressourcen und Buchempfehlungen.

 **Hinweis**

Exkurse

Einige Themen und Konzepte richten sich an fortgeschrittene Anwender. Diese Bereiche sind wie dieser Absatz hervorgehoben und in der Marginalspalte gekennzeichnet.

 **Experten**

Das Buch ist sicher nicht umfassend und trotz vieler Tests nicht fehlerfrei. Eventuelle Fehler werden auf <http://www.indesignjs.de> im Bereich *Errata* veröffentlicht.


Errata

Zu den vorgeschlagenen Lösungen gibt es sicher auch Alternativen, die im Rahmen dieses Buches nicht alle dargestellt werden können. Gerne veröffentliche ich interessante Lösungsalternativen auf der Webseite zum Buch. Schicken Sie Ihre Ideen mit einem Hinweis, dass diese zur Veröffentlichung gedacht sind, an gregor.fellenz@publishingx.de. Sie werden zusammen mit weiteren Themen im Bereich *Neuigkeiten* veröffentlicht.

Tastenkombinationen

Die geläufigen Tastenkombinationen von InDesign setze ich voraus, besonders wichtige oder praktische erwähne ich im Text. Die meisten Tastenkombinationen von InDesign sind unter Windows und Mac identisch. Für die Taste STRG unter Windows wird unter Mac OS CMD bzw. ⌘ verwendet. Im Buch fasse ich die beiden Tasten mit BEFEHL zusammen.

Formatierungen und deren Bedeutung

Hervorhebungen, wichtige Begriffe, Konzepte oder Techniken sowie Dateinamen und Pfade	<i>Kursiv</i>
Navigation zu Befehlen über InDesign-Menüs und Shortcuts sowie Bezeichnungen von InDesign-Dialogen, Bedienfeldern und Menüs	DATEI → NEU
Suchanfragen und Ersetzungsanweisungen für die Arbeit mit GREP	Such- oder Ersetzungstext
Code oder Codebestandteile, die im Text erwähnt werden	Code
Hervorhebungen im Code	Fett
Hinweise auf wichtige Themen in der Marginalspalte	!j
Hinweis auf fortgeschrittene Themen in der Marginalspalte	++
Linkverkürzer, die Zahl in Kombination mit der Adresse http://www.indesignjs.de führt Sie zur Webseite.	 18

Feedback

Ich freue mich über jedes Feedback, Kritik oder Verbesserungsvorschläge und natürlich auch über Fehlermeldungen. Sie können mich unter gregor.fellenz@publishingx.de erreichen.

Teil I Einstieg in die InDesign-Automatisierung

In diesem Teil möchte ich Sie bei Ihren ersten Schritten begleiten. Der Einstieg ist in drei Bereiche aufgeteilt. Zunächst wird das Suchen und die Ersetzung mit GREP vorgestellt (→ Kapitel 1). Im zweiten Bereich liegt der Fokus auf dem Einsatz von Skripten und wie diese verändert werden können (→ Kapitel 2 bis Kapitel 4). Den Abschluss bildet eine Übersicht zur Fehlersuche in Skripten (→ Kapitel 5).

Der Einstieg über die GREP-Funktion bringt Ihnen die »Programmiererlogik« anhand von konkreten Beispielen in einem überschaubaren Themengebiet näher. Die dann folgenden Kapitel zum Thema Skripting sind einfach und übersichtlich gehalten. Sie werden in den Beschreibungen viele Verweise auf den zweiten Teil finden, in dem der Fokus auf einer detaillierten und systematischen Einführung in die Sprache JavaScript und das InDesign-Objektmodell liegt.

Wenn Sie Kapitel 4 durchgehen und dabei das erste Mal programmieren, sollten Sie diese Verweise ignorieren und sich zunächst darauf konzentrieren, ein Gefühl für den Programmablauf und die Zusammenhänge zu bekommen. Zeit für Details bleibt danach immer noch.

Wer schon andere Programme geskriptet oder in anderen Zusammenhängen programmiert hat, dem wird vieles bekannt vorkommen. Für die Leser mit Hintergrundwissen sind auch die Verweise in den zweiten Teil des Buches gedacht. Die Einführung kann in diesem Fall trotzdem hilfreich sein, da sie zeigt, wie Sie Ihre Programmiererfahrung in InDesign einsetzen können. Außerdem erhalten Sie einen guten Überblick, was und wie mit Skripting sinnvoll automatisiert werden kann.

1 Schöner suchen und ersetzen mit GREP

Bei der Arbeit in InDesign kommt man oft mit der Suchen/Ersetzen-Funktion in Berührung. Viele Anwender beschränken sich auf die Suche nach bestimmten Texten, die gegebenenfalls durch andere Texte ersetzt werden. Dies ist allerdings nur das kleinste und einfachste Werkzeug aus dem großen Suchen/Ersetzen-Werkzeugkasten.

Reguläre Ausdrücke

InDesign bietet ab der Version CS3 neben der normalen Suchfunktion die Suche mit *Regulären Ausdrücken* bzw. GREP*. Hier haben Sie die Möglichkeit, nach Textmustern (z.B. Zahlen, Leerräumen, Großbuchstaben etc.) zu suchen. Ein Beispiel ist die Suche nach allen Zahlen von 0 bis 9 mit einem einzigen Suchbefehl. Die gefundenen Zeichen können dann bei der Ersetzung wiederverwendet werden. Stellen Sie sich vor, Sie müssten alle Seitenverweise in einem Dokument von der Form (S. 100) in die Form → Seite 100 bringen. Mit GREP brauchen Sie dafür nur eine Ersetzungsanweisung, weil die unterschiedlichen Seitenzahlen in der Ersetzung wieder eingesetzt werden.

Seit InDesign CS4 gibt es zusätzlich die Möglichkeit, in Absatzformaten so genannte GREP-Stile festzulegen. Hier werden bestimmte Zeichenfolgen definiert, die besonders formatiert werden sollen. Typisch wäre die Kleinerschaltung von Ziffern im Buchsatz oder die Hervorhebung von Namen. In InDesign CS6 wurden zwei Befehle hinzugefügt, die bei der Suche nach Leerräumen und Absätzen hilfreich sind.

Auf den ersten Blick sehen die Suchanfragen unübersichtlich und kompliziert aus. Dies ist der Abstraktion geschuldet, die gleichzeitig erst die Mächtigkeit der Suchausdrücke ermöglicht. Solange die Anfragen präzise und logisch formulierbar sind, kann man praktisch alles finden.

Reguläre Ausdrücke sind mit einigen Grundlagen schnell zu erlernen. Für den produktiven Einsatz ist es auch gar nicht notwendig, die letzten Details und Kniffe zu kennen. Statt theoretisch Möglichkeiten zu diskutieren, ist es oft hilfreicher, das grundsätzliche Prinzip zu verstehen und bei schwierigen Anfragen zu wissen, wie man weiterkommt.

* Die genaue Bedeutung von GREP ist umstritten, ich finde **General Regular Expression Parser** passend. Wer es genau wissen will, kann unter ↗ 122 nachsehen.

Im Folgenden stelle ich typische Suchanfragen vor, die man am besten mit GREP lösen kann. Im weiteren Verlauf des Buches werden noch komplexere Such- bzw. Ersetzungsanweisungen (→ Kapitel 10) und Möglichkeiten zum Skripten von GREP-Anfragen besprochen (→ Unterkapitel 4.8).

1.1 Der Suchen/Ersetzen-Dialog

Wenn Sie mit BEFEHL + F den Suchen/Ersetzen-Dialog aufrufen, sehen Sie die vier Registerkarten für die verschiedenen Suchtypen und die Bedienelemente für die Ausführung der Anweisungen. Der Dialog ist übrigens nicht modal, d. h., Sie können zwischen Dokument und Dialog hin und her wechseln und z. B. per Copy & Paste Text, Leerräume und Sonderzeichen aus dem Dokument übernehmen.

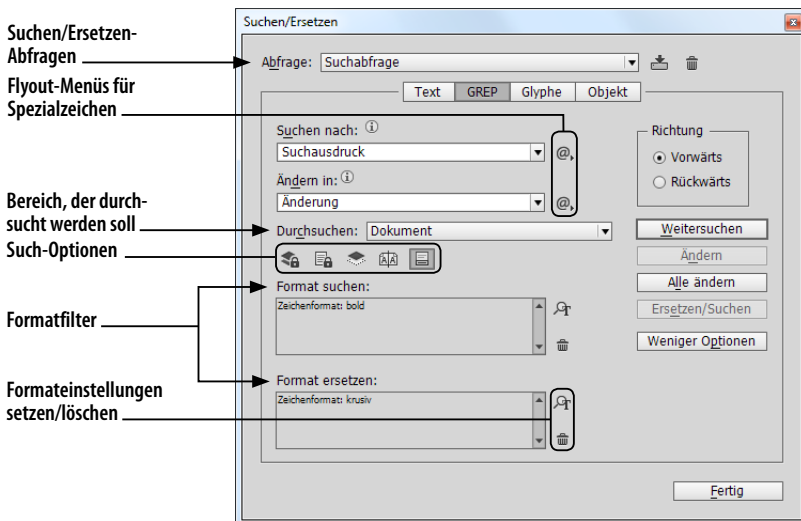


Abb. 1
Suchen/Ersetzen-Dialog

Mit den Registerkarten TEXT und GREP können Texte durchsucht und Ersetzungen vorgenommen werden. Der Aufbau und die Such-Optionen der beiden sind nahezu gleich. In der Registerkarte GREP hat man die Möglichkeit, mit regulären Ausdrücken zu arbeiten.

Mit der Registerkarte GLYPHE können bestimmte Zeichen anhand ihrer ID, mit der Registerkarte OBJEKT Rahmen anhand von Attributen gesucht werden. Diese Registerkarten werden in diesem Buch nicht vorgestellt, weil man sie in der Praxis nur selten benötigt.

Die Registerkarte GREP enthält die Texteingabefelder SUCHEN NACH und ÄNDERN IN, die Buttons zur Einstellung der Such-Optionen und die Felder für die Formateinstellungen.





Die Registerkarte GREP

Im Suchen- bzw. Ändern-Feld werden die regulären Ausdrücke für Suchanfragen bzw. Ersetzungsanweisungen eingetragen. Für die Arbeit

mit Regulären Ausdrücken werden einige Platzhalter und Zeichen mit einer speziellen Bedeutung benötigt. Diese finden Sie neben den Feldern im Flyout-Menü des @,-Buttons.

Unter den Texteingabefeldern kann mit einer Dropdown-Liste der Bereich, der durchsucht werden soll, festgelegt werden. Im Normalfall finden Sie hier die Auswahl zwischen allen Dokumenten und dem aktiven Dokument. Wenn sich die Einfügemarke beim Aufruf des Dialogs in einem Text befindet, finden Sie zusätzlich die Möglichkeiten, den aktuellen Textabschnitt oder den Textabschnitt von der Einfügemarke bis zum Ende zu durchsuchen. Der Textabschnitt beinhaltet den Text des Textrahmens oder den Text der miteinander verketteten Rahmen. Falls Sie einen Text ausgewählt haben, gibt es noch die Möglichkeit, die Auswahl zu durchsuchen.

Such-Optionen

Unterhalb der Dropdown-Liste können die Such-Optionen eingestellt werden. Mit ihnen kann festgelegt werden, welche Bereiche durchsucht werden sollen. Im Normalfall ist nur die Suche innerhalb von Fußnoten eingeschaltet. Weiterhin kann man für die Suche und Ersetzung auch die Musterseiten  und ausgeblendete Ebenen  mit einbeziehen. Gesperrte Ebenen bzw. Objekte  und gesperrte Textabschnitte  können nur bei der Suche berücksichtigt werden. Zu den Such-Optionen zählt auch die ab InDesign CC vorhandene Möglichkeit, VORWÄRTS bzw. RÜCKWÄRTS zu suchen.

Im Gegensatz zu der Textsuche sind in der Registerkarte GREP keine Buttons für die Suche nach einem ganzen Wort und für die Beachtung der Groß-/Kleinschreibung vorhanden. Diese Einstellungen werden bei Regulären Ausdrücken im Suchausdruck festgelegt.





Die Suchen/Ersetzen-Funktion von InDesign geht weit über das Suchen und Ersetzen von Buchstaben, Wörtern oder Satzteilen hinaus. InDesign bietet die Möglichkeit, die Suche mit der Angabe von Textformatierungen und Absatzattributen weiter zu verfeinern. Bei der Ersetzung können alle diese Formateinstellungen dann geändert bzw. gesetzt werden. Diese Einstellungen werden in den Feldern FORMAT SUCHEN bzw. FORMAT ERSETZEN angezeigt. Mit der Lupe  gelangen Sie zur Formatauswahl, mit dem Mülleimer  können Sie die Formateinschränkungen wieder entfernen.

Abb. 2
Fette Textstellen
kursiv formatieren



Vermutlich ist Ihnen auch schon aufgefallen, dass der Dialog alle Einstellungen nach dem Schließen beibehält. InDesign vergisst die Einstellungen erst beim Neustart des Programms.

Es können auch komplette Suchen/Ersetzen-Abfragen abgespeichert werden. Dazu finden Sie im Dialog eine Dropdown-Liste und den Button  zum Speichern bzw.  Löschen der aktuellen Abfrage. Interessante Abfragen kann man so für die spätere Verwendung speichern. Beim Skripting kann man auf die so gespeicherten Abfragen wieder zugreifen. Mit Hilfe des Skripts *ChainGREP.jsx* können diese Abfragen verkettet und gesammelt ausgeführt werden (→ Unterkapitel 10.1).

Suchen/Ersetzen-
Abfragen abspeichern

1.2 Die Suche mit Regulären Ausdrücken

Zum Einstieg empfiehlt es sich, einfach Suchanfragen auszuprobieren. Wenn Sie alle Vorkommen in einem Dokument ersetzen, sollten Sie allerdings Vorsicht walten lassen, da die Gefahr besteht, dass ungewollte Ersetzungen durchgeführt werden.

Zur besseren Übersichtlichkeit sind alle Suchanfragen und Ersetzungsanweisungen in diesem Kapitel grau hinterlegt.

Das Einfachste vorab: Ganz normale Suchtexte funktionieren auch mit GREP. Wenn nach `Max` gesucht wird, kann der Name durch `Moritz` ersetzt werden.

1.2.1 Zeichen mit spezieller Bedeutung

Mit Regulären Ausdrücken kann man variable Zeichenketten suchen und Suchergebnisse präzise eingrenzen. Diese Flexibilität wird dadurch erreicht, dass einige Zeichen eine besondere Bedeutung haben.

Wenn man z.B. im Texteingabefeld Suchen `~$` eingibt, findet InDesign geschützte Leerzeichen `^` und nicht die beiden Zeichen `~` und `$`. Hier wird ein Sonderzeichen durch die Kombination aus dem Tilde-Zeichen mit einem anderen Buchstaben dargestellt. Das Tilde-Zeichen ist hier ein *Metazeichen*. Metazeichen stehen nicht für sich selbst, sondern haben eine besondere Bedeutung.

Metazeichen

Neben dem Tilde-Zeichen gibt es noch einige weitere Zeichen, die eine spezielle Bedeutung haben. So findet man z.B. mit dem Punkt `.` nicht nur das Satzzeichen, sondern fast jedes beliebige Zeichen. Wenn nur nach einem Punkt gesucht wird, muss die besondere Bedeutung aufgehoben werden. Dies geschieht mit einem vorangestellten Backslash. Nach einem Punkt muss also mit `\.` gesucht werden. Diese Aufhebung der Bedeutung von Zeichen mit spezieller Bedeutung nennt man *Maskierung* (engl. escape) – sie wird übrigens ganz ähnlich auch beim Skripting benötigt.

Maskierung

In Regulären Ausdrücken gibt es Zeichen, um Sonderzeichen zu erzeugen, die Suche genauer zu steuern oder etwas im Suchausdruck zu markieren. Diese Zeichen müssen alle mit dem Backslash maskiert werden, wenn sie buchstäblich gefunden werden sollen. Der Backslash zählt auch dazu und wird mit `\\` gefunden.

! Metazeichen bei der Suche mit GREP

! In InDesign CS6 funktioniert \\$ nicht. Verwenden Sie [\$]

\ . * + ? () { } [^ \$ | ~

Wenn man wo bist du? finden möchte, muss man nach wo bist du\? suchen. Zunächst ist es nur wichtig, die Metazeichen zu kennen. Sie werden im Verlauf des Kapitels vorgestellt.

Die Idee, die Suchanfragen mit Metazeichen zu verfeinern, gab es auch schon vor InDesign CS3 in der normalen Textsuche. Dort gibt es z.B. die Möglichkeit, mit ^p nach einem Absatzende oder mit ^? nach einem beliebigen Zeichen zu suchen. Hier wird das Caret-Zeichen (Zirkumflex) ^ zur Kennzeichnung von Zeichen mit spezieller Bedeutung verwendet. Wenn man es finden möchte, muss man es ebenfalls mit ^^ maskieren. Diese Möglichkeiten sind für die Textsuche in der Registerkarte TEXT erhalten geblieben.

Die Problematik mit den Metazeichen ist der einzige Grund, zuweilen noch die normale Suche in der Registerkarte TEXT zu verwenden. Wenn man nur eine bestimmte Zeichenfolge sucht, muss man sich dort (mit Ausnahme des Caret-Zeichens) nicht um die Maskierung kümmern.

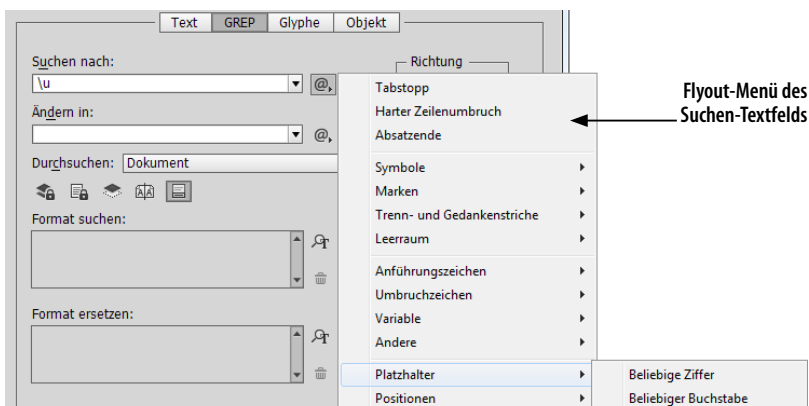
1.2.2 Variable Zeichen auswählen

Platzhalter

Die wichtigsten GREP-Befehle sind die so genannten *Platzhalter* (engl. wildcard), mit denen unterschiedliche Zeichen gefunden werden. Sie ermöglichen eine unscharfe Suche nach Zeichengruppen wie allen Kleinbuchstaben oder allen Ziffern.

Die im Folgenden vorgestellten Zeichen mit einer speziellen Bedeutung finden Sie im Flyout-Menü @, neben dem Suchen/Ersetzen-Textfeld.

Abb. 3
Platzhalter
Flyout-Menü



Beliebige Zeichen
finden

Der wichtigste und einfachste Platzhalter ist der Punkt `.`. Mit ihm kann jedes beliebige Zeichen, mit Ausnahme vom Absatzende (hierzu zählen der normale Zeilenumbruch ¶ und auch alle anderen Umbruchzeichen wie z. B. Seitenumbruch) und dem harten Zeilenumbruch (Soft-Return) ↵, gefunden werden. Mit `..` würde man zwei beliebige aufeinanderfolgende Zeichen finden. Platzhalter repräsentieren eine bestimmte Menge an Zeichen und werden deswegen als *Zeichenklassen* bezeichnet.

Die Suche nach `Max.` findet nicht nur `Max.`, sondern auch `Maxi` mit `i` und ohne Punkt sowie alle anderen Formen von `Max0` bis `Max:`. Mit der Suche nach `d.` könnte man z. B. alle Wörter finden, die mit `d` beginnen und mindestens drei Buchstaben lang sind.

Ein Platzhalter trifft genau ein Zeichen. Wenn Sie einen Punkt in das Suchen-Feld eingeben und die Suche mehrmals ausführen, wird ein Zeichen nach dem anderen gefunden. Um die Verwendung von Zeichenklassen zu verdeutlichen, greife ich auf das nächste Unterkapitel vor und führe die Möglichkeit ein, mit dem Plus-Zeichen mehrere Zeichen vom gleichen Typ zu suchen. Wenn man ein `+` hinter eine Zeichenklasse setzt, wird das erste und alle folgenden zusammenhängenden Zeichen der Zeichenklasse gefunden. Wenn man das Plus-Zeichen mit der Zeichenklasse Punkt kombiniert `.+`, werden entsprechend alle Zeichen bis zum Absatzende oder einem harten Zeilenumbruch gefunden.

Zeichenklassen

Neben dem Punkt gibt es noch einige andere nützliche Zeichenklassen. Diese werden alle aus Kombinationen mit dem Backslash gebildet.

Mit `\u` findet man einen beliebigen Großbuchstaben, mit `\l` einen Kleinbuchstaben. Mit `M\l+` findet man entsprechend alle Wörter, die mit einem großen `M` beginnen und denen Kleinbuchstaben folgen: `Max`, `Moritz`, `Montag`, `Mars` usw., aber nicht `MacOS`.

Entsprechend findet man mit `\u\l+` alle Wörter, die mit einem Großbuchstaben beginnen, mit `\l+` alle kleingeschriebenen.

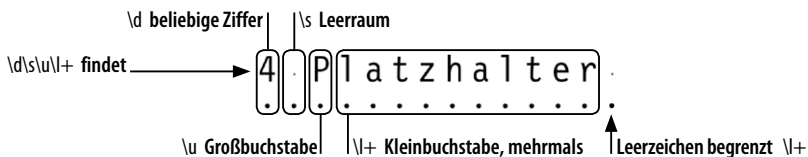


Abb. 4
Verhalten von
Platzhaltern

Eine weitere wichtige Zeichenklasse sind Zahlen. Alle Ziffern von 0 bis 9 können mit `\d` gefunden werden. Alle beliebigen Ziffern in einem Text finden Sie mit `\d+`. Später zeige ich noch die Möglichkeit, auch Ziffern mit Nachkommastellen in den Suchausdruck miteinzubeziehen.

Mit `\s` findet man alle Leerräume, also u. a. Leerzeichen, geschützte Leerzeichen `^`, Festabstände wie z. B. Halbgeviert `␣`, den Tabulator `»` und Zeilenschaltungen `¶` bzw. `↵`.

Mit InDesign CS6 hat Adobe die Platzhalter `\h` für alle horizontalen Leerräume (Leerzeichen, Tabulator etc.) und `\v` für alle vertikalen Leerräume (Absatzende und harter Zeilenumbruch) eingeführt. Bei anspruchsvollen Layouts, in denen mikrotypografische Anpassungen vorgenommen werden sollen, kann man mit `\h` sehr gut verschiedene Kombinationen suchen und durch eine Ersetzung vereinheitlichen. Mit dem Suchausdruck `\d\hmm` findet man z. B. alle Kombinationen einer

Die Platzhalter `\h` und `\v` wurden in InDesign CS6 eingeführt.

Zahl und der angehangenen Maßeinheit mm unabhängig vom Leer-
raum, der dazwischen verwendet wurde.

Zeichenklassen
negieren

Außerdem hat man die Möglichkeit, die Bedeutung der Zeichenklassen
umzudrehen, d.h. also alle Zeichen, die ihr *nicht* entsprechen, zu su-
chen. So kann man mit `\D` alles außer Ziffern finden – hier findet man
neben Buchstaben auch Leerräume, Satzzeichen etc.

Tab. 1
Oft verwendete
vordefinierte
Zeichenklassen

Zeichen- klasse	Beschreibung
.	Ein beliebiges Zeichen, aber nicht das Absatzende (Return) oder einen erzwungenen Zeilenumbruch (Soft-Return)
\u	Alle Großbuchstaben (engl. uppercase). Findet keine elektronisch versal TT gestellten Texte.
\l	Alle Kleinbuchstaben (engl. lowercase)
\d	Alle Ziffern (engl. digits) 0–9. Keine Brüche und nur elektronisch hoch- oder tiefgestellte Zahlen.
\s	Alle Leerräume, also Leerzeichen, alle Festabstände, Tabulatoren und Umbruchzeichen (engl. space). Fälschlicherweise auch das selten ver- wendete »Einzug bis hierhin«- Zeichen †.
\h	Alle horizontalen Leerräume: Leerzeichen, alle Festabstände und den normalen Tabulator (nicht den Tabulator für rechte Ausrichtung).

Zeichenauswahl

Im Flyout-Menü finden
Sie die Zeichenauswahl
unter **ENTSPRECHUNG** →
ZEICHENSATZ

Etwas genauer als mit Zeichenklassen kann man mit einer *Zeichen-
auswahl* suchen. Hier können die Zeichen, die gesucht werden sollen,
selbst festgelegt werden – man kann sozusagen seine eigene Zeichen-
klasse definieren. Eine Zeichenauswahl wird innerhalb von eckigen
Klammern `[]` festgelegt. So findet man mit dem Ausdruck `S[kc]ript` die
Variationen Skript und Script. Innerhalb der eckigen Klammern kön-
nen beliebig viele Zeichen eingetragen werden, die im Suchtreffer an
der entsprechenden Position vorkommen können. Ein anderes Beispiel
wäre die Suche nach `Buch[st][st]abe`, mit der man z.B. Buchstaben-
dreher zwischen s und t auflösen kann. Mit der Suche nach `[gjpqy]`
wiederum würde man alle Buchstaben mit Unterlängen finden – bei
kursiven Schriften kämen noch *f* und *ß* hinzu.

Noch flexibler wird eine Zeichenauswahl durch die Möglichkeit, Be-
reiche festzulegen. Bereiche werden mit einem Bindestrich zwischen dem
ersten und letzten Zeichen des Bereichs definiert. Die Anfrage `200[5-9]`
findet alle Jahreszahlen zwischen 2005 und 2009. Mit der Anfrage `[A-N]`
findet man alle Großbuchstaben zwischen A und N. Solche Bereiche kann
man gut zur Formatierung von Listen, in denen z.B. Produktnummern
nach einem bestimmten Schema festgelegt sind, verwenden. Vorstellbar
ist z.B. eine Produktnummer, die mit 1 oder 2 beginnt, dann mit einem
Buchstaben zwischen A und N weitergeht, gefolgt von noch zwei weite-
ren Ziffern. Der passende Ausdruck dafür wäre `[12][A-N]\d\d`.

Weiterhin können auch negative Zeichenmengen angegeben werden. Hier kann man Zeichen festlegen, die nicht vorkommen dürfen. Dies wird mit dem Zirkumflex bzw. Caret-Zeichen `^` erreicht. Die Anfrage `InDesign CS[^3-9]` findet alle InDesign CS-Versionen vor CS3, wobei die Version CS auch ohne die Ziffer 1 gefunden wird, da das folgende Zeichen – vermutlich ein Leerzeichen – ebenfalls nicht in der Menge aller Zeichen zwischen 3 und 9 enthalten ist. Genauso könnte man mit `Abbildung\h[^d]` alle Abbildungslegenden finden, die noch keine Nummerierung haben.

Innerhalb der Zeichenauswahl gibt es andere Metazeichen, hier haben nur die schließende eckige Klammer `]`, der Zirkumflex `^`, das Tilde-Zeichen `~` und der Bindestrich (Divis) `-` eine besondere Bedeutung. Für die Maskierung braucht man wieder den Backslash. Zwischen den eckigen Klammern sind nur die folgenden Zeichen reserviert:

`] ^ - \ ~`

Die im vorhergehenden Abschnitt genannten Metazeichen haben hier keine besondere Bedeutung mehr und müssen nicht maskiert werden. Die Suche nach `[.\~]` findet Punkte und Tilde-Zeichen. Die Maskierung des Bindestrichs kann am Ende der Zeichenauswahl entfallen `[.\~-]`.


Innerhalb einer Zeichenauswahl können auch Zeichenklassen eingesetzt werden. Mit `[\d,]` findet man Zahlen mit Nachkommastellen – eventuell muss noch der Punkt hinzugefügt werden.

Mit `[\dA-N]+` würde man wieder die oben erwähnte Produktnummer finden. Diese Anfrage ist aber weit weniger präzise, weil Reihenfolge und Länge der Nummer nicht festgelegt sind.

Mit dem Ausdruck `[\u1\-\-]+` könnte man die Zeichenklassen von Groß- und Kleinbuchstaben sowie den Bindestrich suchen, also praktisch alle Wörter. Vielleicht kennen oder finden Sie die Zeichenklasse `\w`, die »offiziell« für diesen Zweck gedacht ist. Mit `\w` findet man jedoch alle Buchstaben, Zahlen und den Unterstrich, für die Suche nach Wörtern ist das eher unbrauchbar.

Als weitere praktische Anwendung stelle ich die Suche nach einer E-Mail-Adresse vor. Dreh- und Angelpunkt ist das `@`-Zeichen. Vor dem `@` dürfen sich Buchstaben, Zahlen und Unterstriche befinden, die mit `\w` gefunden werden können. Da aber auch Punkte und Bindestriche erlaubt sind, muss eine eigene Zeichenklasse gebildet werden `[\w.-]+`. Nach dem `@`-Zeichen sind die gleichen Zeichen erlaubt. Zur genaueren Eingrenzung sollte die Domainendung, die sich hinter einem Punkt befindet und nur noch aus den Zeichen `a-z` bestehen darf, angefügt werden `\.[a-z]+`. Zusammengefügt ergibt sich der Ausdruck `[\w.-]+@[\w.-]+ \.[a-z]+`. Bitte beachten Sie, dass E-Mail-Adressen sehr komplex sein können und mit der hier vorgestellten GREP-Suche nur die üblichen Varianten gefunden werden.

Negative
Zeichenauswahl

 **Metazeichen bei
der Festlegung einer
Zeichenauswahl**

Wörter finden

E-Mail-Adressen finden

Alternativen

Wenn man nicht nach Zeichenmengen, sondern nach variierenden Wortteilen suchen möchte, müssen die Variationen in runde Klammern geschrieben und mit dem senkrechten Strich | unterteilt werden.

Die Suche nach `(Java|Apple|VB)Script` findet alle drei Programmiersprachen innerhalb des Textes. Genauso kann natürlich auch einfach nach zwei alternativen Wörtern gesucht werden – `(Nela Malou|Silke)` findet beide Vorkommen im Text. Es liegt auf der Hand, dass diese Suchanfrage deutlich präziser ist als die ebenfalls erfolgreiche Suche nach allen Buchstaben der beiden Namen mit `[NMSe|aouik]+`.

1.2.3 Sonderzeichen

Viele Zeichen können nicht direkt in die Such- bzw. Ersetzungsanfrage eingegeben werden. So ist es z.B. nicht möglich, eine Zeilenschaltung direkt in das Texteingabefeld einzugeben. Bei anderen Zeichen ist dies zwar prinzipiell möglich, aber ein geschütztes Leerzeichen wäre im Texteingabefeld nicht von einem normalen Leerzeichen zu unterscheiden. Die dritte Klasse bilden die Zeichen mit einer besonderen Bedeutung. Wenn die eigentlichen Zeichen gefunden werden sollen, müssen diese, wie am Anfang beschrieben, mit dem Backslash maskiert werden.

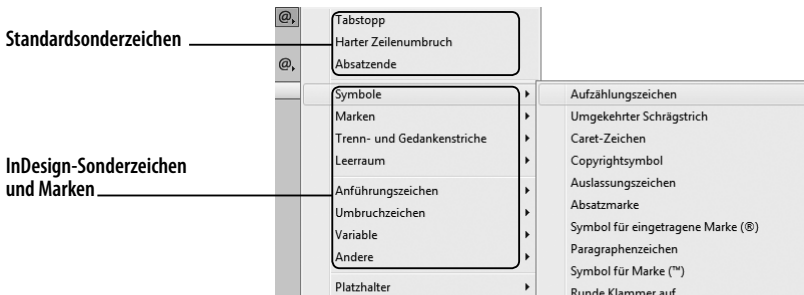
Tabulator und Zeilenumbrüche werden in Regulären Ausdrücken mit dem Backslash erzeugt. Einen oder mehrere Tabulatoren findet man z.B. mit dem Suchausdruck `\t+`.

Tab. 2
Standardsonderzeichen
für die Suche mit
Regulären Ausdrücken

GREP	Zeichen	Beschreibung
<code>\t</code>	»	Tabulator
<code>\r</code>	¶, ↵, ⏎ ...	Absatzende bzw. Zeilenschaltung (Return), aber auch Rahmenumbruch, Seitenumbruch etc.
<code>\n</code>	↵	Erzwungener Zeilenwechsel (Soft-Return)

Alle weiteren Sonderzeichen werden mit der Tilde ~ erzeugt. Alle Sonderzeichen, die so erzeugt werden, finden Sie nach Bereichen sortiert im zweiten und dritten Bereich des Flyout-Menüs neben dem Suchenfeld.

Abb. 5
Sonderzeichen-
Auswahl im Suchen/
Ersetzen-Dialog



Nach einem geschützten Leerzeichen sucht man z. B. mit `~S`, nach bedingten Trennzeichen mit `~-`. Interessant ist die Möglichkeit, dass auch Indexmarken `~I`, verankerte Objekte `~a` und Textvariablen `~v` gesucht werden können.

Grundsätzlich können Sie alle Zeichen auch per Copy & Paste in die Texteingabefelder des Suchen/Ersetzen-Dialogs einfügen. InDesign erzeugt dann automatisch die richtige Kombination mit der Tilde. Die folgende Tabelle zeigt einige oft gebräuchliche Sonderzeichen, eine vollständige Übersicht finden Sie im Anhang A1.

GREP	Zeichen	Beschreibung
<code>~S</code>	^	Geschütztes Leerzeichen
<code>~m</code>	□	Geviert
<code>~></code>	◌◌	Halbgeviert
<code>~<</code>	◌◌	Achtelgeviert
<code>~b</code>	¶	Standardzeilenumbruch
<code>~=</code>	—	Halbgeviertstrich, Gedankenstrich
<code>~-</code>	-	Bedingter Trennstrich
<code>~y</code>	␣	Tabulator rechte Ausrichtung
<code>~I</code>	⋈	Indexmarke
<code>~a</code>	⌘	Marker für verankerte Objekte
<code>~v</code>		Alle Textvariablen
<code>\~</code>	~	Tilde

Tab. 3
InDesign-
Sonderzeichen in
Regulären Ausdrücken

Kombinationen mit der Tilde gelten nur in InDesign und können nicht in anderen Programmen, die Reguläre Ausdrücke unterstützen, eingesetzt werden.

Sonderzeichen und Zeichenklassen können in einem alternativen Suchausdruck oder einer eigenen Zeichenklasse kombiniert werden. So könnte man der Zeichenklasse `\h` mit dem Ausdruck `(\h|~y)` bzw. `[\h~y]` den fehlenden Tabulator rechte Ausrichtung hinzufügen.

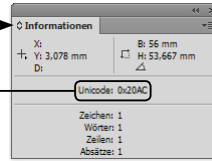
Neben diesen festgelegten Zeichen gibt es auch die Möglichkeit, ein Zeichen über den *Unicode*-Codepoint festzulegen. Dazu muss der hexadezimale Codepoint des Zeichens bekannt sein. Nach dem Zeichen `¾` suchen Sie mit `\x{00BE}`, das €-Symbol finden Sie mit `\x{20AC}`. Den Codepoint eines markierten Zeichens kann man in InDesign in der Mitte des Informationsfensters ablesen (FENSTER → INFORMATIONEN). Weitere Informationen zu Unicode finden Sie auf Seite 148. Im Kapitel 10.7 werden weitere Suchtechniken vorgestellt.

++
Unicode-Zeichen
finden

Abb. 6
Informationen-
Bedienfeld

Informationen-Bedienfeld (F8)

Unicode-Codepoint des
markierten Zeichens



Markieren Sie das gewünschte
Zeichen in InDesign#

Wenn der Codepoint dort zu sehen ist, können Sie das Zeichen aber genauso gut per Copy & Paste in die Texteingabefelder des Dialogs befördern. Die Definition eines Zeichens mit `\x{####}` benötigt man nur, wenn ein Unicode-Zeichen gesucht wird, dessen Codepoint aus einer Tabelle oder einem anderen Programm bekannt ist.

1.2.4 Wiederholungen

Eine weitere wichtige Technik in Regulären Ausdrücken ist es, die Anzahl von Wiederholungen eines Zeichens oder einer Zeichenklasse festzulegen. Das Wiederholungszeichen `+`, mit dem man mehrere Zeichen finden kann, wurde bereits verwendet.

Abb. 7
Wiederholung
Flyout-Menü

Wiederholung	Null oder ein Mal _____?
Entsprechung	Null oder mehrere Male _____*
Modifizierer	Ein oder mehrere Male _____+
Posix	Null oder ein Mal (kürzeste Entsprechung) _____??
	Null oder mehrere Male (kürzeste Entsprechung) _____*?
	Ein oder mehrere Male (kürzeste Entsprechung) _____+?

Wiederholungszeichen

Immer wenn man nicht genau weiß, ob ein Suchbegriff oder Zeichen im Text vorkommt, kann man das Wiederholungszeichen `?` verwenden (im Menü NULL ODER EIN MAL). So könnte man z.B. alle Vorkommen von `das` und `dass` mit der Suchanfrage `dass?` prüfen. Mit `(Max)? Mustermann` finden Sie jede Erwähnung von Herrn Mustermann – mit und ohne Vorname. Das Wiederholungszeichen, auch *Quantifizierer* genannt, bezieht sich immer auf das vorhergehende Zeichen, eine Zeichenklasse oder das vorhergehende Wort in runden Klammern. Es kann nicht alleine stehen.

Mit einem `+` wird festgelegt, dass ein Suchausdruck mindestens einmal vorkommen muss, aber beliebig oft vorkommen darf (im Menü EIN ODER MEHRERE MALE). Wenn Sie z.B. mehrere Leerzeichen hintereinander zu einem einzigen vereinheitlichen wollen, kann man mit `+` eine beliebige Anzahl von Leerzeichen finden. Verwenden Sie dazu nicht `\s+`, da in der Zeichenklasse auch die Absatzschaltungen enthalten sind und Sie eventuell die Kombination Leerzeichen Return entfernen würden.

Ein Plus bietet sich oft an, um den Rest eines Absatzes oder den variablen Text zwischen zwei bekannten Zeichenfolgen zu finden. Wenn ein beliebiger Text, z.B. die Beschreibung nach einer Produktnummer

oder vor dem Preis, gesucht wird, helfen die speziellen Zeichenklassen meist nicht weiter, da die Beschreibung fast alle Zeichen enthalten kann. Angenommen einer wie oben geformten Produktnummer folgt eine Produktbeschreibung, z. B. 1G78 Staubsauger. Die Nummer inklusive Beschreibung finden Sie jetzt mit `[12][A-N]\d\d\h.+` – es wird sogar 1G79 Staubsauger mit Beutel gefunden oder 1H32 Wischmops.

Wenn man nicht genau weiß, ob das Zeichen oder die Zeichenklasse überhaupt vorkommt, kann mit einem `*` auch nach beliebig vielen Vorkommen gesucht werden (im Menü NULL ODER MEHRERE MALE, korrekt wäre die Bezeichnung BELIEBIG OFT). Der Suchbegriff kann keinmal, einmal oder beliebig oft vorkommen. Im Unterschied zum `?` findet man mit dem `*` eine beliebige Anzahl von Zeichen – mit `jaa*` findet man ja, jaa, jaaa usw. Dies ist z. B. nötig, um nach einer Zahl zu suchen, die eventuell Nachkommastellen haben kann. Der Suchausdruck `\d+,\?\d*` findet sowohl 15 als auch 14,9999, wobei die Anzahl der Nachkommastellen nicht relevant ist. Im Suchausdruck könnte man auch das Komma mit einem `*` versehen. Dies wäre aber nicht besonders präzise, weil bei einer Zahl mit Nachkommastellen genau ein, bei einer Zahl ohne Nachkommastellen genau kein Komma enthalten ist.

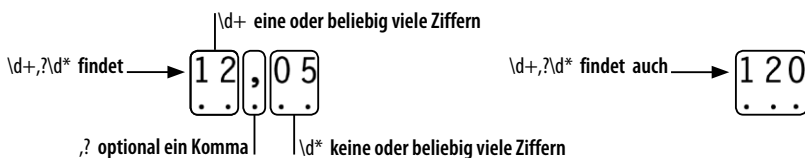


Abb. 8
Wiederholungen

Achten Sie darauf, dass das Sternchen in Kombination mit nur einem Zeichen immer trifft – eine solche Suche ist also nicht sinnvoll. Beispielsweise findet `x*` an jeder Position des Textes entweder ein `x` oder kein `x`.

Wiederholungszeichen	Beschreibung
<code>?</code>	Das Zeichen oder die Zeichenklasse ist optional, d. h., sie kann einmal oder keinmal vorkommen.
<code>+</code>	Der Suchbegriff muss mindestens einmal, kann aber beliebig oft vorkommen.
<code>*</code>	Der Suchbegriff kann beliebig oft vorkommen – also auch keinmal.

Tab. 4
Wiederholungsfaktoren
in Regulären
Ausdrücken

Suchverhalten der Wiederholungszeichen

Die Wiederholungszeichen `+` und `*` finden immer so viele Zeichen wie möglich. Sie suchen so lange weiter, bis sie den größtmöglichen passenden Suchausdruck gefunden haben. Das ist auch der Grund, warum man mit `.*` nicht den ersten Satz eines Absatzes auswählen kann, wie man vielleicht vermuten könnte. In diesem Ausdruck sammelt das `+` beliebige Zeichen vom Absatzbeginn bis zum letzten Vorkommen eines

InDesign versucht normalerweise den längsten Treffer zu ermitteln.

Das Fragezeichen hat je nach Kontext unterschiedliche Bedeutungen.

Punkts und damit meist den ganzen Absatz. Dieses Verhalten wird als *gierig* (engl. greedy) bezeichnet.

Um die Wiederholungszeichen zu zügeln, kann ein Fragezeichen verwendet werden. Hinter einem Wiederholungszeichen bedeutet es, dass nur die kleinstmögliche Anzahl Treffer gefunden werden soll. Mit dem Ausdruck `.+?\.` wird nur noch der erste Satz eines Absatzes gefunden. Das Fragezeichen hat also mehrere Bedeutungen: Hinter einem Zeichen oder einer Zeichenklasse bewirkt es, dass das Suchzeichen im Treffer ein- oder keinmal vorkommen darf. Hinter einem Wiederholungszeichen regelt es, dass so wenige Zeichen wie möglich gefunden werden.

Das Fragezeichen sollte immer eingesetzt werden, wenn die Gefahr besteht, dass die nächste Fundstelle nach einem Wiederholungszeichen nicht eindeutig ist. Wenn z.B. in Klammern gesetzter Text gesucht wird (also dieser hier), ist es durchaus möglich, dass innerhalb des Absatzes noch ein weiterer Text in runden Klammern steht (hier ist der zweite). Wenn man nun mit einem gierigen Wiederholungszeichen `\(.+\)` sucht, findet man den gesamten Text von der ersten bis zur letzten runden Klammer. Hier sollte man stattdessen mit dem Ausdruck `\(.+?\)` arbeiten. Achten Sie auch darauf, dass die spezielle Bedeutung der Klammern durch die Maskierung mit dem Backslash aufgehoben werden muss.

Genau Anzahl der Vorkommen festlegen

Noch genauer kann mit geschweiften Klammern `{}` festgelegt werden, wie oft ein Zeichen oder eine Zeichenklasse vorkommen darf. Eine Zahl innerhalb der geschweiften Klammern legt fest, wie viele Vorkommen des Zeichens oder der Zeichenklasse erlaubt sind. So findet man mit `\d{4}` alle vierstelligen Zahlen.

Wenn man einen Wiederholungsbereich festlegen möchte, muss die minimale und maximale Anzahl der Vorkommen angegeben werden. Beide Werte werden durch ein Komma getrennt. Zwei- oder dreistellige Zahlen findet man entsprechend mit `\d{2,3}`.

Tab. 5
Genau Anzahl von
Wiederholungen
festlegen

Wiederholungen	Beschreibung
<code>{n}</code>	Der Ausdruck muss exakt n-mal vorkommen.
<code>{n,m}</code>	Der Ausdruck muss mindestens n-mal und darf maximal m-mal vorkommen.
<code>{n,}</code>	Der Ausdruck muss mindestens n-mal und darf öfter vorkommen.

1.2.5 Genau Positionen ermitteln

Suchbereich

Eine Suche kann oft durch Positionsangaben im Suchbereich noch genauer eingegrenzt werden. Standardmäßig endet der Suchbereich nach dem Absatzende oder dem harten Zeilenumbruch – dem ersten Zeichen, das nicht von `.` gefunden wird.