

werden die Befehle im Anweisungsblock hinter dem ebenfalls reservierten Wort `else` ausgeführt. Die grundsätzliche Syntax ist:

```
if (Vergleich) {  
    // Anweisungsblock  
}  
else {  
    // Anweisungsblock  
}
```

Im Falle eines zutreffenden Vergleichs wird der erste Anweisungsblock durchlaufen, in allen anderen Fällen der zweite Anweisungsblock. Der Skriptablauf wird oft mit dieser Methode beeinflusst.

Verzweigung

Eine Verzweigung besteht aus einer Bedingung und zwei Anweisungsblöcken. Wenn die Bedingung zutrifft, wird der erste Anweisungsblock ausgeführt, andernfalls der zweite.

Wenn in der `for`-Schleife gerade eine rechtsseitige Seite durchlaufen wird, wird auch der Textrahmen auf der rechtsseitigen Musterseite ausgewählt (`musterSeiten.pages[1] ...`). Ansonsten wird die erste, also linksseitige Musterseite gewählt.

Im konkreten Beispiel gibt es drei Möglichkeiten: rechts, links oder nur eine einzelne Seite. Das heißt, im Falle einer rechten Seite wird der `if`-Block durchlaufen, im Falle einer linken oder einer einzelnen Seite wird der `else`-Block durchlaufen. Dies ist auch so gewollt, weil die Musterseite mit nur einer Seite ebenfalls auf `pages[0]` liegt.

Weitere Informationen zu Verzweigungen finden Sie in Unterkapitel 6.7.

4.8 Suchen und Ersetzen per Skript

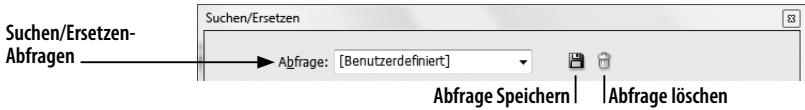
Im folgenden Unterkapitel wird gezeigt, wie man per Skript die Suchen/Ersetzen-Funktion steuern kann.

Dazu verwende ich im ersten Skript Suchen/Ersetzen-Abfragen, die man im Suchen/Ersetzen-Dialog speichern kann. Die Abfragen stehen programmweit für jedes Dokument zur Verfügung. Die genauen Optionen und Eigenschaften können bequem in der Registerkarte gesetzt und für die spätere Verwendung gespeichert werden. Im Skript wird dann eine bereits erstellte Abfrage geladen.

Um die Abfrage für das Skript zu erstellen, öffnen Sie den Suchen/Ersetzen-Dialog. Dort muss in der Registerkarte GREP die Suche nach dem Wort `Max` und die Änderung in `Moritz` erstellt werden. Diese Einstellungen werden als Suchen/Ersetzen-Abfragen mit dem Namen `MaxMoritz` gespeichert.

Suchen/Ersetzen-
Abfragen

Abb. 35
Suchen/Ersetzen-
Abfragen speichern



Diese Abfrage MaxMoritz wird dann im folgenden Skript geladen und vom Skript ausgeführt. Dieses Skript zeigt exemplarisch die Verwendung von Suchen/Ersetzen-Abfragen; produktiv wird es erst, wenn man mehrere Abfragen hintereinander ausführen lässt.

Listing 21
4-8_SuchenUnd
Ersetzen-1.jsx

```
1 #target InDesign
2 app.loadFindChangeQuery("MaxMoritz", SearchModes.GREP_SEARCH);
3 app.activeDocument.changeGrep();
```

Achten Sie auf die exakte Schreibweise des Namens.

2 Innerhalb des Skripts wird als Erstes die Abfrage MaxMoritz geladen. Die Methode `loadFindChangeQuery()` gehört zur Anwendung und wird direkt über `app` angesteuert. Die Methode übernimmt zwei Parameter. Der erste Parameter beinhaltet den Namen der Abfrage, der zweite Parameter den Typ der Abfrage. Für die vier verschiedenen Suchtypen muss einer der folgenden, sich selbst erklärenden Parameter verwendet werden:

- `SearchModes.TEXT_SEARCH`
- `SearchModes.GREP_SEARCH`
- `SearchModes.GLYPH_SEARCH`
- `SearchModes.OBJECT_SEARCH`

```
app.activeDocument.changeGrep();
```

3 Die Anforderungen aus der Suchen/Ersetzen-Abfragen werden mit der Methode `changeGrep()` im aktuell geöffneten Dokument ausgeführt. Die Methode muss ebenfalls mit dem Suchtyp korrespondieren.

- `changeText()` für die Text-Suche
- `changeGrep()` für die Suche mit GREP
- `changeGlyph()` für die Suche nach Glyphen
- `changeObject()` für die Suche nach Objekten

Die Einstellungen der Suche selbst setzen

Bei der Verwendung von Suchen/Ersetzen-Abfragen muss man sicherstellen, dass die Abfragen an dem Arbeitsplatz, an dem das Skript ausgeführt wird, verfügbar sind. Deswegen stelle ich in der Praxis oft die Eigenschaften für die gewünschte Abfrage direkt ein. Im Skript wird dann keine Abfrage geladen, sondern der Suchen/Ersetzen-Dialog direkt per Skript gesteuert. Das folgende Skript zeigt, wie die Einstellungen für das Suchen und Ersetzen per GREP vorgenommen werden.

Listing 22
4-8_SuchenUnd
Ersetzen-2.jsx

```
1 #target InDesign
2 app.findGrepPreferences = NothingEnum.nothing;
3 app.findGrepPreferences.findWhat= "Max";
4 app.changeGrepPreferences = NothingEnum.nothing;
5 app.changeGrepPreferences.changeTo= "Moritz";
6 app.activeDocument.changeGrep();
```

```
app.findGrepPreferences = NothingEnum.nothing;
```

2 Die Eigenschaft `findGrepPreferences` gehört zur Anwendung. Hierüber können alle Such-Einstellungen gesetzt werden, die Sie auch im normalen Dialog eingeben können.

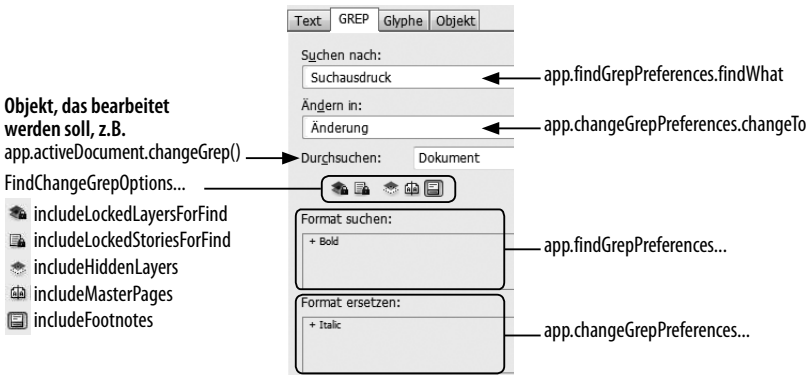


Abb. 36

Skripteinstellungen im Suchen/Ersetzen-Dialog

Als Erstes werden eventuell vorhandene Einstellungen im Bereich Suche mit dem Wert `NothingEnum.nothing` gelöscht. InDesign merkt sich die Einstellungen im Dialog, auch wenn dieser geschlossen ist. Dies ist im normalen Umgang mit der Funktion recht praktisch. Beim Skripting gibt es jedoch böse Überraschungen, wenn in den Such-Einstellungen noch Werte der vorherigen Suche gesetzt sind.

In Voreinstellungen (engl. preferences) können meist mehrere Werte gesetzt werden, deswegen stehen die Namen der Eigenschaften in der Pluralform. Es handelt sich hier aber ausnahmsweise nicht um Sammlungen. Der Aufruf `app.findGrepPreferences[0]` funktioniert nicht!

! In ...Preferences sind keine Sammlungen enthalten.

```
app.findGrepPreferences.findWhat= "Max";
```

3 Die Eigenschaft `findWhat` in dieser Zeile entspricht dem Texteingabefeld **SUCHEN** im Suchen/Ersetzen-Dialog. Es wird nach der Zeichenfolge **Max** gesucht.

```
app.changeGrepPreferences = NothingEnum.nothing;
```

4 Die Einstellungen für die Ersetzung müssen analog zu denen der Suche zurückgesetzt werden.

```
app.changeGrepPreferences.changeTo= "Moritz";
```

5 Die Eigenschaft `changeTo` entspricht dem Texteingabefeld **ÄNDERN**.

```
app.activeDocument.changeGrep();
```

6 Die Methode `changeGrep()` führt die Ersetzung im aktuellen Dokument aus, sie entspricht dem Button **ALLE ÄNDERN** aus dem Suchen/Ersetzen-Dialog. Die Methode kann von vielen Objekten aus gestartet werden, Sie können auch nur einen einzelnen Textrahmen oder Absatz durchsuchen.

**InDesign »merkt«
sich die Einstellungen.**

Nach der Ausführung des Skripts stehen noch Max und Moritz in den entsprechenden Feldern des Suchen/Ersetzen-Dialogs. Um das zu verhindern, müssen die `findGrepPreferences` und `changeGrepPreferences` am Ende des Skripts erneut auf den Wert `NothingEnum.nothing` gesetzt werden.

Um ein Skript zu überprüfen, kann es aber durchaus hilfreich sein, die Werte nach Beendigung nicht zu löschen. So kann man nach Ablauf des Skripts bequem nachprüfen, ob die Anfrage aus dem Skript korrekt im Suchen/Ersetzen-Dialog angekommen ist.

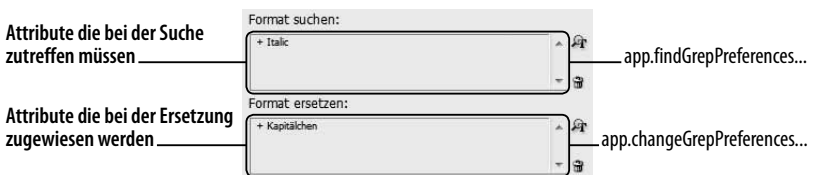
Such-Optionen

Die Such-Optionen, also z. B. ob gesperrte Ebenen oder Musterseiten mit in die Suche aufgenommen werden sollen, sind Eigenschaften von `app.findChangeGrepOptions` und müssen bei Bedarf ebenfalls eingestellt werden. Dies wird in Unterkapitel 7.13 gezeigt.

Formate suchen
und ersetzen

Im unteren Bereich des Suchen/Ersetzen-Dialogs können die Such-Einstellungen mit Formatangaben weiter verfeinert werden. Wenn nur Formatangaben und keine Suchbegriffe eingegeben werden, werden alle Textstellen, auf die die Formatangaben zutreffen, gefunden. Falls zusätzlich noch Formatangaben für die Änderung eingetragen werden, können die Formatierungen ersetzt werden.

Abb. 37
Änderung der Format-
angaben im Suchen/
Ersetzen-Dialog



Diese Einstellungen können natürlich auch in Skripten verwendet werden. Das folgende Beispiel zeigt, wie man alle Textstellen, auf denen der Schriftschnitt *Italic* angewendet ist, mit dem Schriftschnitt **Bold** formatiert – also alle kursiven Texte in fett gesetzte Texte umwandelt. Dies setzt natürlich voraus, dass die Schriftschnitte der Schriftarten den entsprechenden Namen haben und vorhanden sind.

Listing 23
4-8_SuchenUnd
Ersetzen-3.jsx

```

1 #target InDesign
2 app.findGrepPreferences = NothingEnum.nothing;
3 app.findGrepPreferences.fontStyle = "Italic";
4 app.changeGrepPreferences = NothingEnum.nothing;
5 app.changeGrepPreferences.fontStyle = "Bold";
6 app.activeDocument.changeGrep();

```

```
app.findGrepPreferences.fontStyle = "Italic";
```

3 Mit der Eigenschaft `fontStyle` wird der Schriftschnitt für die Suchvorgaben festgelegt. Der Name muss als String übergeben werden.

```
app.changeGrepPreferences.fontStyle = "Bold";
```

5 Bei den Vorgaben für die Ersetzung wird ähnlich verfahren, wobei man auch noch zusätzlich Angaben für die Formatierung vergeben könnte. In der Suche bzw. bei der Ersetzung können alle Formateinstellungen, die auf Absatz- und Zeichenebene eingestellt werden können, verwendet werden. Die Wichtigsten finden Sie in der folgenden Tabelle. Die Eigenschaften müssen Sie wie in den oben gezeigten Skripten an `findGrepPreferences` bzw. `changeGrepPreferences` mit dem Punkt anhängen.

Eigenschaft	Beschreibung, findet/ersetzt ...
<code>findWhat = "Suche"</code>	Suchanfrage (nur <code>findGrepPreferences</code>)
<code>changeTo = "Ersetzung"</code>	Ersetzungsangabe (nur <code>changeGrepPreferences</code>)
<code>appliedCharacterStyle = "Name des Formats"</code>	die angewendete Zeichenformatvorlage
<code>appliedParagraphStyle "Name des Formats"</code>	die angewendete Absatzformatvorlage
<code>appliedFont = "Name der Schrift"</code>	die Schriftart
<code>fontStyle = "Name des Schriftschnitts"</code>	den Schriftschnitt
<code>pointSize = 12</code>	die Schriftgröße
<code>position = Position.SUBSCRIPT</code>	tiefgestellten Text
<code>position = Position.SUPERSCRIPIT</code>	hochgestellten Text
<code>capitalization = Capitalization.ALL_CAPS</code>	Versalien
<code>capitalization = Capitalization.SMALL_CAPS</code>	Kapitälchen
<code>fillColor = "Name der Farbe"</code>	Schriftfarbe
<code>justification = justification.LEFT_JUSTIFIED</code>	Blocksatz
<code>justification = Justification.RIGHT_ALIGN</code>	rechtsbündigen Text
<code>justification = Justification.LEFT_ALIGN</code>	linksbündigen Text
<code>justification = Justification.CENTER_ALIGN</code>	zentrierten Text

Tab. 11
Eigenschaften für Suchen/Ersetzen-Einstellungen

Leider kann man in der InDesign-Suche nicht negativ suchen, also nach Texten, die eine bestimmte Eigenschaft nicht aufweisen. In Unterkapitel 11.6 finden Sie eine Möglichkeit, nach abweichenden Textstellen zu suchen.

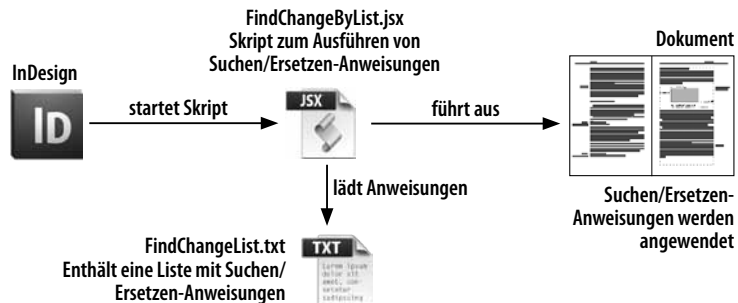
Mit InDesign CS3 wurde ein fehlerhaftes FindChangeByList-Skript ausgeliefert. Laden Sie eine korrigierte Version unter [☞ 21](#) herunter.

4.8.1 Adobe-Skript FindChangeByList

Mit dem Hintergrund ist nun die Anpassung des von Adobe mitgelieferten Skripts *FindChangeByList.jsx* kein Problem. Mit ihm können wiederkehrende Suchen/Ersetzen-Aufgaben hintereinander abgearbeitet werden. Das Skript finden Sie im Ordner *Samples*, Unterordner *Java-Script* des Skript-Bedienfelds.

Das Skript wird über eine Textdatei, die in jeder Zeile eine Suchen/Ersetzen-Aufgabe enthält, gesteuert. Die dafür benötigte Datei *FindChangeList.txt* befindet sich im Ordner *FindChangeSupport*. Die mitgelieferte Datei enthält eine englische Anleitung und ein paar Beispiele.

Abb. 38
Zusammenhang
InDesign, Skript
und Textdatei



Ich stelle zunächst den Aufbau der Textdatei vor und erläutere dann, wie die Datei erweitert werden kann.

Aufbau der Datei
FindChangeList.txt

Zur Analyse öffnen Sie die Textdatei *FindChangeList.txt* in einem beliebigen Texteditor oder im ExtendScript Toolkit. Jede Zeile setzt sich aus vier Teilen zusammen, die durch einen *Tabulatorschritt* getrennt werden müssen.

Der erste Teil definiert den *Suchtyp*. Hier stehen die Schlüsselwörter `text` für die normale Textsuche, `grep` für die Suche mit Regulären Ausdrücken und `glyph` für die Suche nach bestimmten Glyphen zur Verfügung.

1. Die zweite Spalte enthält den Suchausdruck. Die Syntax ist in der Form `{findWhat:"SUCHE"}` festgelegt. Innherhalb der Anführungszeichen muss die Suchfolge, die im InDesign-Dialog in das Textfeld `SUCHEN` eingetragen würde, geschrieben werden.
2. Die dritte Spalte enthält die Ersetzungsanweisung in einer ganz ähnlichen Syntax: `{changeTo:"ERSETZUNG"}`. Der Bereich zwischen den Anführungszeichen korrespondiert mit dem Textfeld `ÄNDERN`.
3. Im vierten Teil kann die Suche genauer eingegrenzt werden, die Optionen entsprechen den Buttons unterhalb der Suchfelder.
4. Optional kann danach noch ein beliebiger Text zur Beschreibung der Aufgabe verwendet werden.

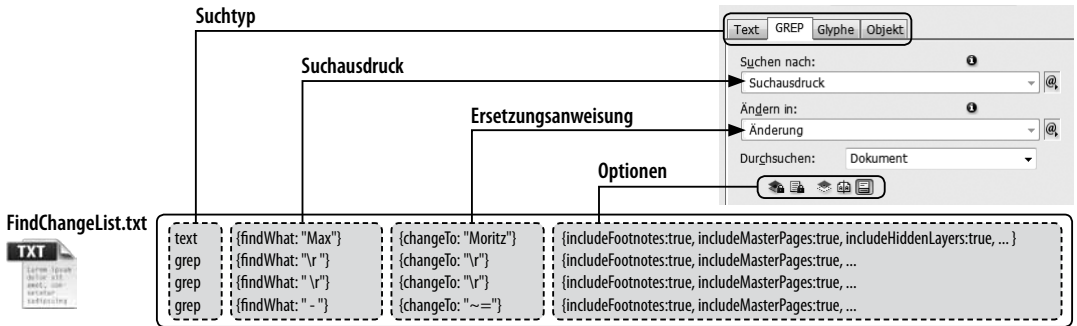


Abb. 39
Zusammenhang
zwischen Suchen/
Ersetzen-Dialog und
FindChangeList.txt

Die Anweisungen der Liste innerhalb der geschweiften Klammern entsprechen den JavaScript-Eigenschaften.

Alle Zeilen, die mit zwei Schrägstrichen // (Forward Slash) beginnen, werden ignoriert und können für Kommentare verwendet werden.

Als Beispiel, wie sich die Liste erweitern lässt, zeige ich die GREP-Suche nach einem fälschlich gesetztem Bindestrich, der in einen Gedankenstrich geändert und mit einem geschützten Leerzeichen vor dem Zeilenumbruch geschützt werden soll. Aus diesen vier Angaben ergibt sich die folgende Zeile, die man am besten nach der letzten Zeile in der Datei hinzufügt.

Die Datei *4-8_FindChangeList.txt* enthält bereits die folgenden Anpassungen, sie kann mit dem Dokument *4-8_FindChangeListTest.indd* getestet werden.

```

grep {findWhat:" - "} {changeTo:"~S~="} {includeFootnotes:true,
includeMasterPages:true, includeHiddenLayers:true}

```

Speichern Sie diese Datei mit dem Namen *FindChangeList.txt* im Ordner *FindChangeSupport* bei den Adobe-Beispielskripten.

Zunächst muss man sich für einen *Suchtyp* entscheiden. Für die GREP-Suche muss die Zeile mit dem Wort *grep* beginnen.

Dann sollte die eigentliche Aufgabe mit dem normalen InDesign-Dialog getestet werden.

- In der Registerkarte GREP wird nach der Zeichenfolge `-` gesucht.
- Geändert wird die Zeichenfolge in `~S~=`, wobei `~S` für ein geschütztes Leerzeichen `^` und `~=` für den Gedankenstrich steht.

Für den Suchausdruck in der ersten geschweiften Klammer ergibt sich `{findWhat:" - "}`, für die Ersetzungsanweisung in der zweiten `{changeTo:"~S~="}`. Für die *Such-Optionen* werden die Standardwerte von den bereits vorhandenen Abfragen übernommen.

Einige übliche Fehler wie falsch gesetzte Satzzeichen oder Klammern sind im folgenden Listing umgesetzt.

Listing 24

Weitere Suchanfragen
für den Suchtyp grep
aus der Datei
4-8_FindChangeList.txt

```

14 grep {findWhat:" +\\."} {changeTo:"."}
    {includeFootnotes:true, includeMasterPages:true,
    includeHiddenLayers:true}
15 grep {findWhat:" +;"} {changeTo:";"} {...}
16 grep {findWhat:" +,"} {changeTo:","} {...}
17 grep {findWhat:"\\( +"} {changeTo:"("} {...}
18 grep {findWhat:" +\\)"} {changeTo:")"} {...}

```

! Der Backslash muss
hier doppelt maskiert
werden.

Achten Sie auf die Maskierung der Zeichenklasse Punkt und der runden Klammer in Zeile **14**, **17** und **18**. Sie ist mit einem doppelten Backslash \\ ausgeführt. Dies entsteht dadurch, dass die Textdatei zunächst als JavaScript-Zeichenkette interpretiert wird. Dort wird der Backslash aber bereits für die Maskierung von Steuerzeichen verwendet und hat ebenfalls eine besondere Bedeutung. Damit der Backslash bei der GREP-Suche ankommt, muss er zunächst für JavaScript und dann für die GREP-Suche maskiert werden. Dies klingt kompliziert, für die Arbeit mit der Textdatei kann man sich aber einfach merken, dass alle umgekehrten Schrägstriche verdoppelt werden müssen.

Ich empfehle direkt nach dem Platzieren einer externen Textquelle, wie z.B. einem Word-Dokument, die Daten mit den angeführten Suchanfragen automatisch zu bereinigen. Wenn Ihr Dokument bereits layoutet ist, kann es dabei zu ungewollten Umbruchänderungen kommen.

Falls Sie für verschiedene Projekte unterschiedliche Listen anlegen und diese bei der Ausführung auswählen wollen, verändern Sie einfach den Namen des Ordners *FindChangeSupport*. Danach fragt das Skript bei jeder Ausführung nach der zu verarbeitenden Liste.

In Unterkapitel 10.13 wird eine Steuerliste mit Festabständen für mikrotypografische Korrekturen vorgestellt.

4.9 Suchen, finden und verändern

Für komplexere Anforderungen reichen die Möglichkeiten der Suchen/Ersetzen-Funktion von InDesign nicht aus. Insbesondere wenn bestimmte Stellen in einem Dokument nicht einfach nur ersetzt, sondern nach bestimmten Regeln bearbeitet werden müssen, stößt die Funktion an ihre Grenzen. Die Einschränkungen können mit dem folgenden Skript aufgehoben werden.

Mit Suchergebnissen
arbeiten

Die Idee für das Skript ist, die Suche wie gewohnt von InDesign ausführen zu lassen, die Ersetzung aber über eine eigene Funktion zu steuern. Für das Skript müssen die folgenden Schritte umgesetzt werden:

1. die Suchen/Ersetzen-Abfragen definieren
2. Anweisungen für die Ersetzung programmieren