InDesign Satzautomation

11748 Vorlesung von Gastdozenten
22.10., 12.11., 19.11.und 03.12.
10 Uhr bis 17 Uhr in Raum 171

Kontakt: gregor.fellenz@publishingx.de

Funktionen/Methoden

- Methoden von Objekten „tun etwas“ z. B. haben viele Objekte die Methode `.add()`
- Funktionen können auch selber geschrieben werden z. B. um Codeteile öfter zu verwenden!
- Machen den Code übersichtlicher!
- Es können **Argumente** (Parameter) übergeben werden, die innerhalb der Funktion verwendet werden können.
- Funktionen können einen **Rückgabewert/Ergebnis** zurückliefern
  Dazu wird die Anweisung `return` verwendet; `return` beendet die Funktion.
- ```
  var _zurueck = hallo("Moritz");
  alert(_zurueck);
  function hallo (_name) {
   var _gruss = "Hallo " + _name;
   return _gruss;
  }
  ```
- **Best Practice:** Funktion `main()` den gesamten Code kapseln.
  Mehr Übersicht und separierte Variablen!
  ```

Ein Register erstellen!

- Die Dateien befinden sich im Ordner `02_termin`
- Öffnen Sie die Datei `08_IndexUndArray.jsx` im Extended Script Toolkit
- Öffnen Sie die Datei `08_indexTest.indd` in InDesign

! Alle Zeichenfolgen, die mit dem Zeichenformat `caps` ausgezeichnet sind, sollen in den Index aufgenommen werden.
Die Funktion `register (_indexEintrag)` nimmt den im Parameter übergebenen Text in den Indesign-Index auf.
Die Variable `_ergebnisArray` enthält alle Textstellen die mit dem Zeichenformat `caps` augezeichnet sind.

! **Zusatz:** Erweitern Sie die Funktion `index` so, dass alle Textstellen die mit **der**, **die** oder **das** beginnen in der Form **Eintrag, der** etc. in den Index aufgenommen werden.

Geltungsbereich/Scope

Werden Variablen außerhalb einer Funktion deklariert, sind sie global verwendbar. D.h. sie sind in dem gesamten Skript erreichbar. Innerhalb einer Funktion deklarierte Variablen gelten auch nur in dieser und sind von außen nicht erreichbar.

Beispiel:

```
var a;
a = 1;
function test() {
   var a;
   a=2;
   alert(a); // ergibt 2
}
test();
alert(a); // ergibt 1
```

Interaktion mit dem Anwender

Es gibt drei sehr einfache Funktionen, um während der Ausführung eines Skripts mit dem Benutzer des Skripts zu kommunizieren:

- Mit der Funktion `alert()` kann der Benutzer auf etwas hingewiesen werden.
- Mit `confirm()` kann der Benutzer etwas gefragt werden
  ```
  var _ergebnis = confirm("Sind Sie sicher?");
  ```
- Mit der Funktion `prompt()` können einfache Abfragen gestaltet werden.
  ```
  var _input = prompt();
  ```

InDesign Satzautomation

Übungsaufgabe

Kommunikation

- Erstellen Sie ein neues Skript.

! Testen Sie die drei Interaktionsfunktionen.

! Lassen Sie den Anwender Ihres Programms entscheiden ob ein Textrahmen mit grüner oder roter Hintergrundfarbe erstellt werden soll.
Die Eigenschaft für die Referenz auf den Textrahmen lautet:
`_tf.fillColor = "Name der Farbe";`
Sie müssen zunächst die beiden Farben rot und grün erstellen und bennen!

! Der Funktion `prompt()` können bis zu drei Argumente übergeben werden, finden Sie heraus, wozu diese dienen!

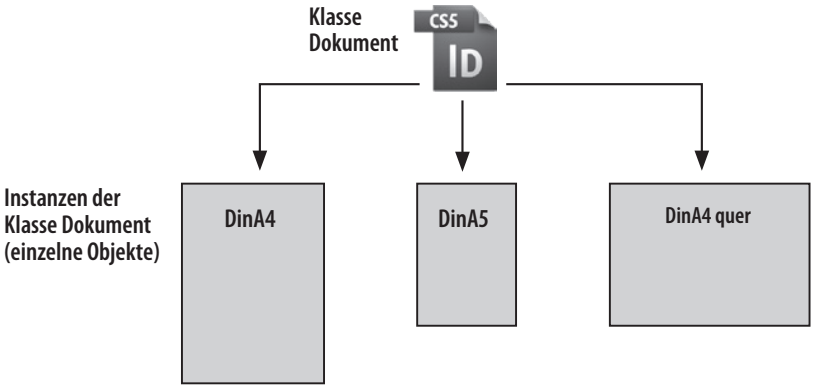
! Lassen Sie den Anwender mit der Funktion `prompt()` den Text für einen Textrahmen bestimmen.

InDesign Objektmodell

Das Objektmodell ist die **Schnittstelle** zwischen JavaScript und InDesign.

- Alle Programmfunktionen von InDesign, aber auch Seiten, Rahmen, Texte usw. stehen zur als **Klassen/Objekte** für die Programmierung zur Verfügung.

Exkurs: Klassen und Objekt



InDesign Satzautomation
Übersicht | JavaScript | InDesign Objektmodell | InDesign Programmierung | InDesign XML

Eigenschaften von Objekten/Klassen

- Objekte/Klassen haben eine oder mehrere **Eigenschaften**, sie beschreiben den Zustand z. B. eine zugewiesene Schriftart, Schriftschnitt oder Schriftgrad; aber auch der oben erwähnte Umfang `length` ist eine Eigenschaft.
- Einige Eigenschaften können nur ausgelesen (**read only**) werden, andere können zusätzlich auch neu beschrieben (**read/write**) werden.

Objektmethoden

- Im Gegensatz zu Eigenschaften erledigen **Methoden** bzw. **Funktionen** eine Aufgabe und liefern im Normalfall ein Ergebnis.
- Viele Objekte haben z. B. die Funktion `add()` die ein neues Objekt erstellt.

Aufbau des Objektmodells

Man kann sich das Objektmodell als **Baum** vorstellen. Ein Stamm der immer weiter bis zu den Blättern verzweigt ist. Baumstruktur wird oft verwendet: **Filesystem** (Dateien, Verzeichnisse).

- Ursprung und Einstiegspunkt: **Application** (`app`) Zugriff auf Klassen mit dem **Punktoperator** z.B. `app.documents`
- Wenn es mehrere Objekte geben kann, befinden sie sich in **Sammlungen**. Erkennbar am Plural des Klassenamens.
- Adressierung wie Arrays. **Objekt/Instanz einer Klasse** mit eckigen Klammern `[]` z.B. `app.documents[0]`

- **Objektmodell** ist abhängig von der InDesign Version
- Das Objektmodell kann in der Hilfe und im **Datenbrowser** des ESTK erkundet werden.
- Alternativ gibt es ein in HTML umgesetztes Objektmodell von Teus de Jong
  Download: `http://www.jongware.com/idjshelp.html`

InDesign Satzautomation
Übersicht | JavaScript | InDesign Objektmodell | InDesign Programmierung | InDesign XML

Objektmodell als Baum

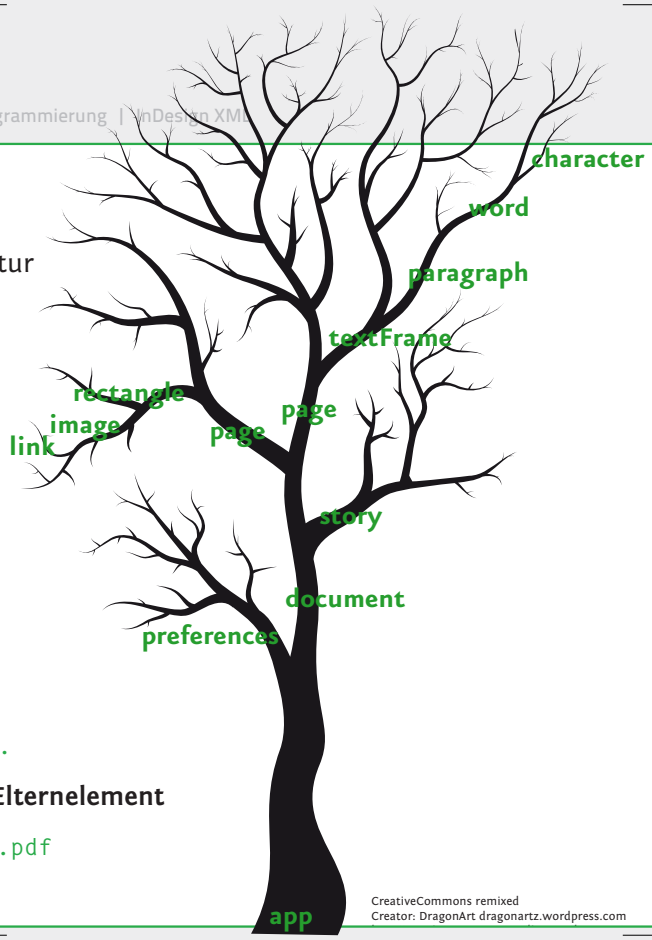
- Famillienstammbaum/Hierarchische Struktur
  - Eltern (parents)
  - Kinder (children)

Stamm	`app`

Dicke Äste	`documents`
	aber auch:
	`books, documentPreferences`

Dokumentäste	`textFrames, rectangles, stories, links, fonts, ...`
Blätter	`geometricBounds, paragraphs, characters, ...`

- Die Eigenschaft `parent` führt immer zum **Elternelement**
- **Kurzreferenz:** `KurzReferenz_Objektmodell.pdf`

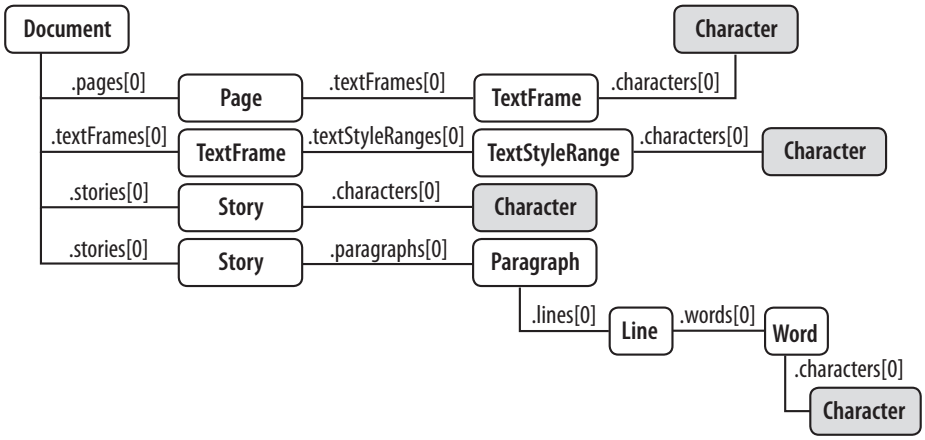


CreativeCommons remixed
Creator: DragonArt dragonartz.wordpress.com

Hangeln in Bäumen

Problem: Verschieden Eltern können die gleichen Kinder haben!

Das folgende Bild zeigt verschiedene Möglichkeiten, an das erste Zeichen im ersten Textrahmen in einem neuen Dokument zu gelangen.



Navigation im Baum

- Erstellen Sie ein neues Skript.

! Das Programm **InDesign** kann über das Root-Element `app` angesprochen werden.
Was liefert die Eigenschaft `name`?
Was liefert die Eigenschaft `activeDocument`?

! Dokumente haben ebenfalls die Eigenschaft `name`. Was liefert diese?
Die Seiten eines Dokuments können über die Sammlung `pages` angesprochen werden.
Wie kann man zur ersten Seite des Dokuments gelangen?
Wieviele Seiten hat ein Dokument?

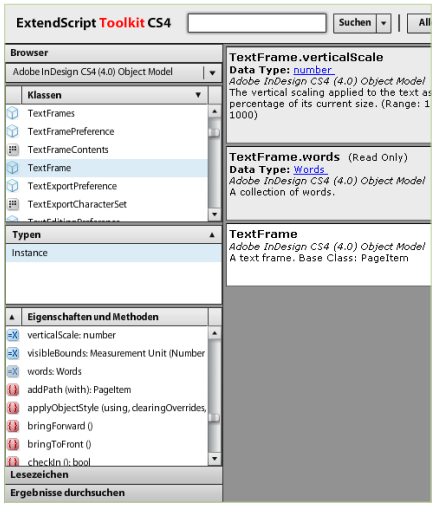
! Eine Seite kann verschiedene `pageItems` enthalten.
Was haben all diese `pageItems` gemeinsam?

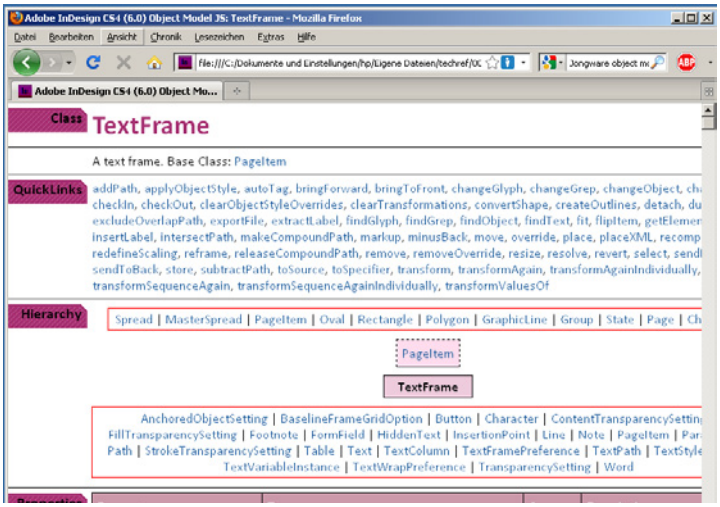
! Navigieren Sie zum zweiten Textrahmen auf der dritten Seite im aktuellen Dokument.
Erstellen Sie dazu vorher ein enstprechendes Dokument per Hand.

Objektmodell

! Vergleichen Sie den Objektmodell Viewer vom ESTK und die HTML Version.

! Suchen Sie bestimmte Objekte, verwenden Sie Copy & Paste um Objekte oder Eigenschaften in Ihre Skripte zu übernehmen!





Sammlung von Objekten

Sammlungen (Datentyp Collection) sind sehr ähnlich zu **Arrays**!

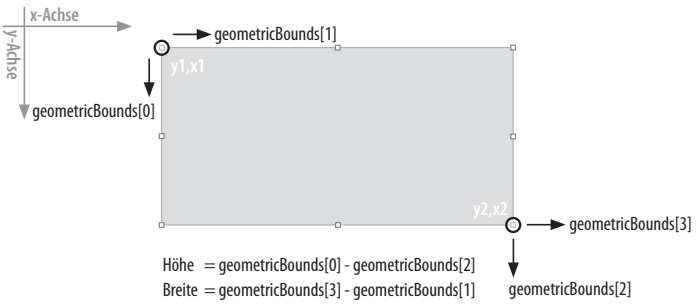
- **Sammlungen** sind eine Datenstrukur, in der alle Objekte versammelt sind.
  z. B. `app.selection[0].paragraphs[0].words` enthält alle Worte eines Absatzes.
  - Anhand eines Indizes kann weiter eingegrenzt werden
  - `app.selection[0].paragraphs[0].words[-1]` gibt das letzte Wort zurück.
- Der **Umfang** von Sammlungen kann mit `length` ermittelt werden.
- Sammlungsobjekte haben die Funktion `add()` mit der ein neues Objekt erstellt werden kann:
  `...textFrames.add(); //liefert einen neuen TextFrame zurück`
- Sammlungen haben meist die gleichen Funktionen/Methoden:
  `remove(), itemByName(); itemByRange(); move();`

Wichtige Objekte

Rahmen

InDesign ist ein **rahmenbasiertes** DTP Programm. Sowohl Texte als auch Bilder werden in Rahmenobjekten auf Seiten platziert.

Die Dimensionen/Größen werden im Skripting nicht über Höhe und Breite angegeben sondern immer mit Hilfe von **Koordinaten**.



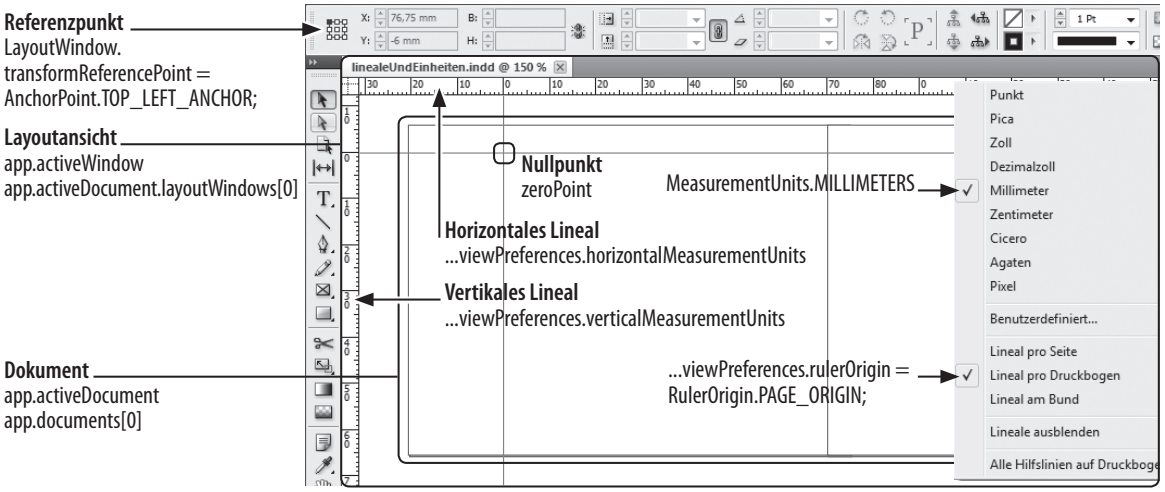
Die Eigenschaft `geometricBounds` ist ein Array mit den vier Koordinaten `y1,x1,y2,x2`

Daraus leiten sich auch Höhe und Breite ab:

```
_höhe = geometricBounds[2] - geometricBounds[0]
_breite = geometricBounds[3] - geometricBounds[1]
```


Übersicht der Einstellungen

Die Koordinaten aller Rahmenobjekte orientieren sich an den **Linealen** (!) des Dokuments. Die Einstellungen können in den `viewPreferences` des Dokuments per Skript eingestellt werden.



Koordination

- Die Dateien befinden sich im Ordner `02_termin`
- Öffnen Sie die Datei `11_koordinaten.jsx` im Extended Script Toolkit

! Prüfen Sie die Einstellungen der Koordinaten im Skript.

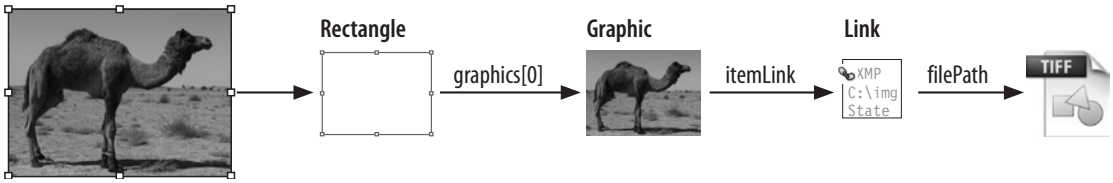
Für automatisierte Abläufe sollten diese Einstellungen immer geprüft werden.

! Erstellen Sie einen Textrahmen an der Position
x = 15 mm; y = 25 mm mit der Höhe 75 mm und der Breite 60 mm.

! **Bezugspunkt** für geometricBounds:
`app.layoutWindows[0].transformReferencePoint = AnchorPoint.TOP_LEFT_ANCHOR`
oder einfach im Template das für die Automatisierung verwendet wird umstellen.

Rahmen , Bilder und Verknüpfungen

- In InDesign werden Bilder nicht direkt eingebunden. Sie sind über **Verknüpfungen** (`Links`) mit den Originaldateien verbunden.
  `...rectangles[0].graphics[0].itemLink`
  `app.documents[0].links`



- Bilder können mit der Funktion `place()` platziert werden
  `...rectangles[0].place(File)`
- Anpassen der Bilder mit der Funktion `fit()` und der Aufzählung `FitOptions`
  `_auswahl.fit(FitOptions.CENTER_CONTENT);`

Rahmen bauen

- Verwenden Sie das Skript `12_rahmenbauen.jsx`.
- Die verwendete Textdatei und das Bild liegen im Ordner `Material`

! Erstellen Sie ein neues Dokument

! Erstellen sie einen **Textrahmen** (`textFrame`) und ein **Rechteck** (`rectangle`)

! Positionieren/verschieben Sie beide Objekte auf der Seite
und weisen Sie beiden eine **Breite** 100 mm und **Höhe** 50 mm zu

! Laden Sie Text und Bild aus dem Ordner `Material` in die jeweiligen Objekte.
Sie benötigen dazu die Funktion `place()` des Rahmenobjekts
Der Funktion `place(fileName)` muss eine Datei übergeben werden,
Erstellen Sie dazu eine Variable mit der **Dateireferenz**. Der folgende Code öffnet
einen „Datei öffnen Dialog“ und gibt die Dateireferenz der ausgewählten Datei zurück:
`File.openDialog()`

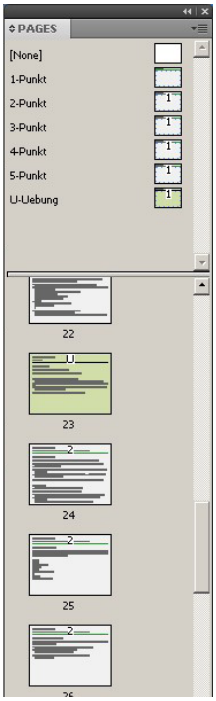
InDesign Satzautomation
Übungsaufgabe

! Machen Sie sich mit den Eigenschaften `parentTextFrames` und `parentStory` vertraut.

! Warum muss `parentTextFrames` über den Index adressiert werden?

! Erstellen Sie einen zweiten Textrahmen und verknüpfen Sie diesen mit dem ersten Textrahmen.

Seiten und Musterseiten

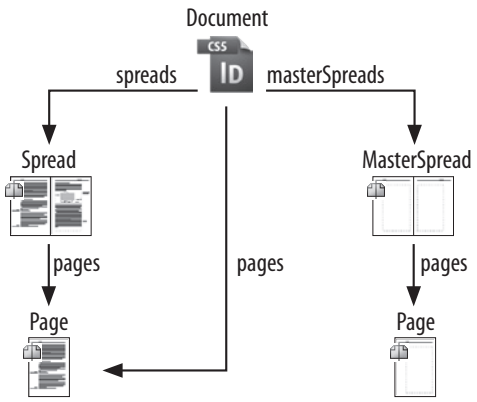


Seiten **page** und Musterseiten **masterSpread** sind wichtige InDesign-Objekte! Nochmal konkret: `pages` vs. `page`

- Mit der Sammlung `pages` kann man:
  - Mit der Eigenschaft `length` die Länge der Sammlung ermitteln z. B. für eine `for` Schleife
  - Mit der Funktion `add()` Seiten hinzufügen
  - Mit `itemByName()` Seiten anhand ihrere Seitenzahl finden
    Mit `itemByRange(FROM, TO)` einen Array der Seiten anfordern.
- Dem Seitenobjetk `page` kann man:
  - Mit der Eigenschaft `documentOffset` die Position im Dokument ermitteln
  - Mit der Eigenschaft `pageItems` alle Objekte auf der Seite abrufen
  - Mit `textFrames` die Sammlung der Textrahmen abrufen
  - Mit `allGraphics` alle Grafiken der Seite abrufen

InDesign Satzautomation
Übersicht | JavaScript | InDesign Objektmodell | InDesign Programmierung | InDesign XML

Systematik von Druckbögen, Seiten und Musterseiten



Musterseiten anwenden

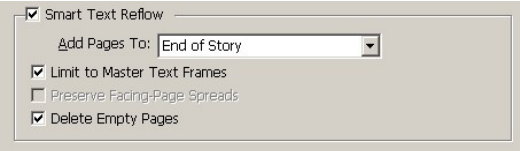
- Mit `appliedMaster` die angewendete Musterseite setzen/abrufen
- Mit `side` herausfinden ob es sich um `PageSideOptions.RIGHT_HAND`, `.LEFT_HAND` oder `.SINGLE_SIDED`
- Mit der Funktion `place()` einen Text in den Musterseitenrahmen platzieren
- Mit `move()` die Seite verschieben
- Mit `remove()` die Seite entfernen

Das Konzept der **Musterseiten** bietet sich für automatisierten Satz an! In einem Template können beliebige Musterseiten angelegt werden, die dann für den Satz verwendet werden.

- Die Sammlung **masterSpreads** hat die üblichen Sammlungsobjekte.
  Mit der Funktion `itemByName()` kann man einfach auf die einzelnen Musterseiten zugreifen.
  ```
  var _mspread = app.activeDocument.masterSpreads.itemByName("A-Master");
  ```
- Musterseiten können mit der Eigenschaft page.`appliedMaster` einer Seite zugewiesen werden
- Alle Musterseitenobjekte sind in der Sammlung masterPageItems

InDesign Satzautomation
Übersicht | JavaScript | InDesign Objektmodell | InDesign Programmierung | InDesign XML

- Musterseitenobjekte müssen auf der Seite **gelöst** werden:
  1. Objekt auf der Musterseite auswählen:
     ```
     var _mpo = app.activeDocument.pages[0].appliedMaster.pageItems.itemByName("textf")
     ```
     oder `var _mpo = ...masterSpreads[0].pages[1].pageItems[0].contents`
  2. Objekt auf der Seite lösen
     ```
     _mpo.override(app.activeDocument.pages[0])
     ```
  3. **Optional**: Nach dem Lösen von Objekten den Textfluss aktualisieren:
     ```
     app.activeDocument.recompose();
     ```
- **Smart Text Reflow** ab CS4
  Einstellung unter Eigenschaften -> Type



Mit Seiten spielen

- ! Erstellen Sie ein neues Skript!
- ! Erstellen Sie **5 neue Seiten**. Im Objektmodell ist beschrieben wie mit dem Parametern `LocationOptions und reference` an verschiedenen Positionen im Dokument Seiten erstellt werden können. Machen Sie sich damit vertraut.
- ! Verwenden Sie die Funktion `itemByRange()` um nur die Seiten 3-5 mit Seitenzahlen per Skript zu nummerieren. Verwenden Sie die Eigenschaft `side` um die Seitenzahlen jeweils am äußeren Rand zu platzieren. Um den Array zu erhalten muss die Funktion wie folgt aufgerufen werden `itemByRange().getElements()`
- ! Verschieben und entfernen Sie Seiten im/aus dem Dokument!
- ! Legen Sie mehrere **Musterseiten** an. Weisen Sie verschiedenen Seiten unterschiedliche Musterseiten zu.
- ! Lösen Sie einen Musterseitentextrahmen, platzieren das Word-Dokument aus dem Ordner `03_material` und testen Sie die Funktion Smart Text Reflow.
- ! Auch eine Seite hat die Funktion `place()`. Was macht diese? Machen Sie sich mit den Parameter
  `[autoflowing: bool=false]` vertraut.