

Seiten und Musterseiten

Seiten `page` und Musterseiten `masterSpread` sind wichtige InDesign-Objekte.

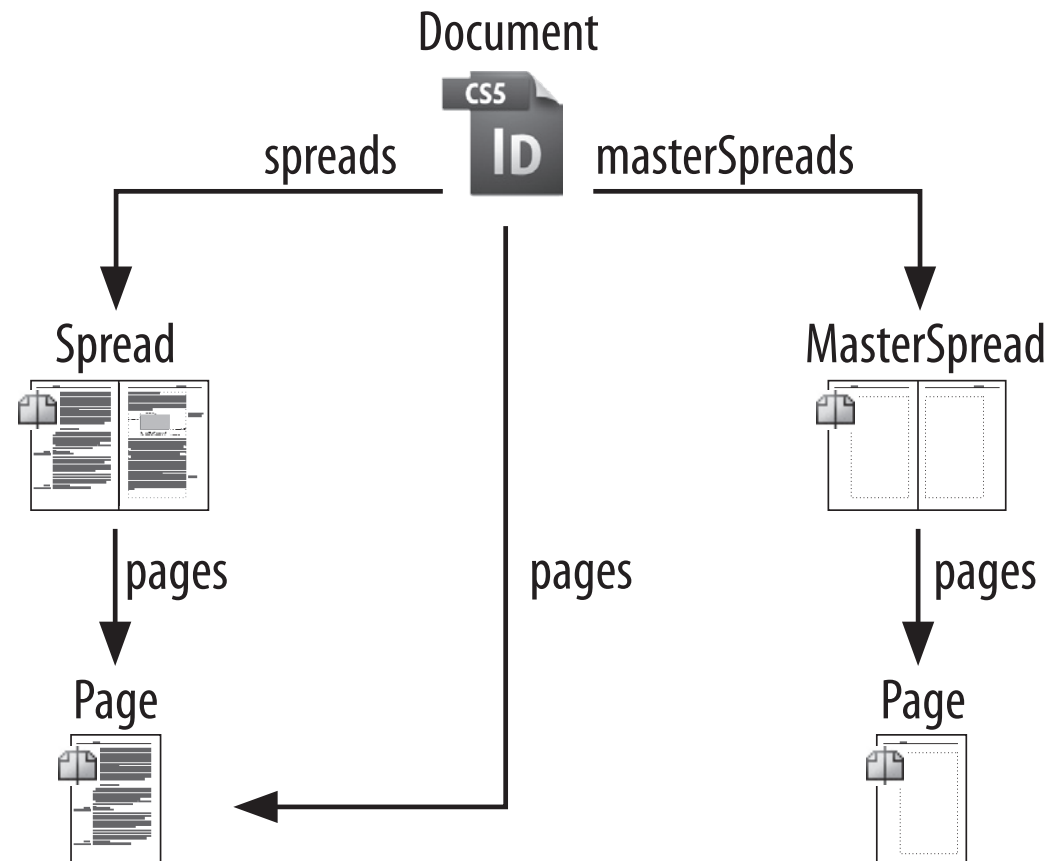
Nochmal konkret Sammlung vs. Objekt: `pages` vs. `page`

- 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 ihrer Seitenzahl finden
Mit `itemByRange(FROM, TO)` einen Array der Seiten anfordern.
- Dem Seitenobjekt `page` kann man:
 - Mit der Eigenschaft `documentOffset` die Position im Dokument ermitteln (zählt von 0)
 - Mit der Eigenschaft `pageItems` alle Objekte auf der Seite abrufen
 - Mit `textFrames` die Sammlung der Textrahmen abrufen
 - Mit `allGraphics` alle Grafiken der Seite abrufen

Systematik von Druckbögen, Seiten und Musterseiten

Streng hierarchisch müsste man Seiten (page) über den Druckbogen (spread) adressieren. Es geht aber auch direkt vom Dokument (document) aus.

`app.documents[0].pages[0]` entspricht `app.documents[0].spreads[0].pages[0]`



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 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");
```

Musterseitenobjekte

- Musterseitenobjekte müssen auf der Seite **gelöst** werden:

1. Objekt auf der Musterseite lokalisieren:

```
var _mpo = app.activeDocument.pages[0].appliedMaster.pageItems.itemByName("textf")
```

```
oder var _mpo = ...masterSpreads[0].pages[1].pageItems.itemByName("textf")
```

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/Intelligenter Textumfluss** ab CS4
Einstellung unter Eigenschaften -> Type/Eingabe

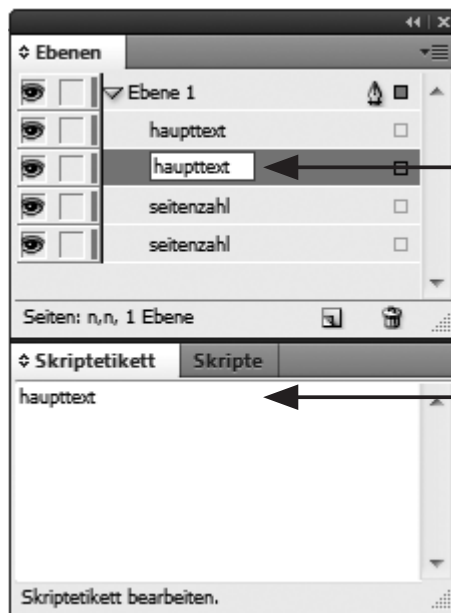


Objekte mit Namen versehen

Beim Skripting ist es hilfreich, auf Seitenobjekte direkt zugreifen zu können. Dafür gibt es in InDesign die Möglichkeit, Objekten einen Namen zuzuweisen.

- Mit der Version CS3 oder CS4, Bedienfeld **Skriptetikett**
- Ab CS5 müssen die Namen im Bedienfeld **Ebenen** eingetragen werden.
- Mit der Methode `itemByName("")` können die Objekte adressiert werden

Ebenen-
Bedienfeld



Name
Textfeld (CS5)

Skriptetikett
(CS3 & CS4)

Skriptetikett-
Bedienfeld

Ausgewählter Rahmen
`app.selection[0].name`



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.

Textrahmen und Texte

Textrahmen

Alte Bekannte: Sammlung TextFrames Objekt TextFrame

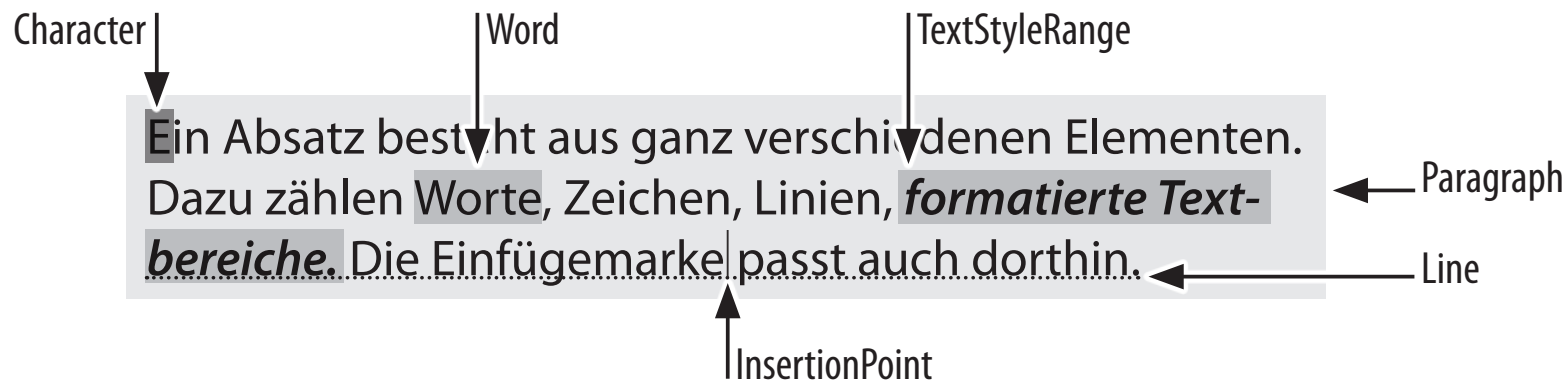
- Mit `place()` können Textdaten platziert werden
- Objektformate mit der Eigenschaft `appliedObjectStyle` zuweisen
- Ab CS5 gelangen Sie mit der Eigenschaft `parentPage` auf die Seite, auf der der Rahmen platziert ist. Die Eigenschaft `parent` führt im Regelfall zum Spread

Textverkettungen

- Zwei Textrahmen können über die Eigenschaft `nextTextFrame` (bzw. auch `previousTextFrame`) miteinander **verknüpft** werden.

```
_tf1.nextTextFrame = _tf2;
```

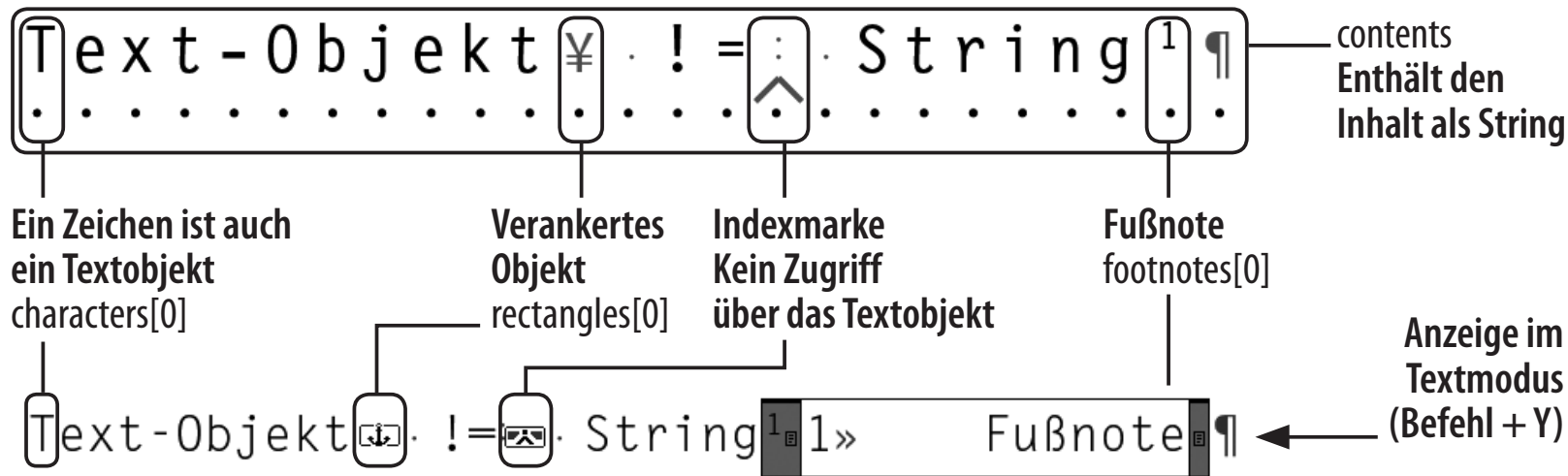
Absatz, Zeichen und Einfügemarke



- Absatz **Paragraph**; Zeile **Line**; Zeichen **Character**, Einfügemarke **InsertionPoint** etc. haben die gleichen Eigenschaften und Methoden.
- Mit der Eigenschaft `contents` kann der enthaltene Text abgefragt und neu gesetzt werden. `contents` liefert einen String des gesamten Inhalts aber nicht die Formatierung!
- Alle **Formatierungsoptionen** haben eine Entsprechung.
z. B.: `appliedFont`, `appliedParagraphStyle` `fontStyle`, `justification` (Ausrichtung), ...
- Textobjekte können mehr: `baseline` **vertikale Position** `horizontalOffset` horizontale Position und `endHorizontalOffset`
- Bereiche mit gleicher Formatierung `textStyleRanges`

InDesign Satzautomation

- Die Sammlung `characters` bietet die Möglichkeit **Textbereiche** anzusteuern
`characters.itemByRange(0,9).contents` // liefert die ersten 10 Zeichen
- Beachten Sie den Unterschied zwischen String und InDesign Text-Objekten



Sätze formatieren

- Die Dateien befinden sich im Ordner `03_termin`
- Öffnen Sie die Datei `14_textFormat.indd`, öffnen Sie das Skript `14_textFormat.jsx`
- ! Der Inhalt des Beispieldokuments ist nicht formatiert. Im ersten Absatz stehen die Autoren des Texts, im zweiten die Überschrift. Legen Sie in InDesign zwei verschiedene passende **Absatzformate** an und weisen Sie diese den beiden Absätzen zu.
- ! Legen Sie für den Grundtext ein Absatzformat mit einem Einzug in der ersten Zeile an und weisen Sie dieses den restlichen Absätzen zu.
- ! Eine weitere automatische Formatierung von Überschriften ist schwer/nicht möglich – warum?
- ! **Zusatz:** Mit der Eigenschaft `appliedParagraphStyle` werden lokale Formatierungen eines Absatzes überschrieben. Finden Sie heraus, wie Sie lokale Formatierungen mit der Methode `applyStyle()` beibehalten können.
- ! Mit der Eigenschaft `baseline` kann die vertikale Position (y-Achse) der ersten Zeile des Absatzes ermittelt werden. Erstellen Sie neben dem zweiten Absatz ein quadratisches schwarzes Rechteck in Schriftgröße (`pointSize`) mit einem Abstand von 3 mm zur linken Satzspiegelkante.

Textrahmen und Stories – eine schwierige Verwandtschaft

TextFrame- und Text-Objekte haben mehrere Eltern:

- **Text-Objekte** (paragraphs, words, characters, insertionPoints)

Direktes Eltern-Objekt mit parent ist `story`

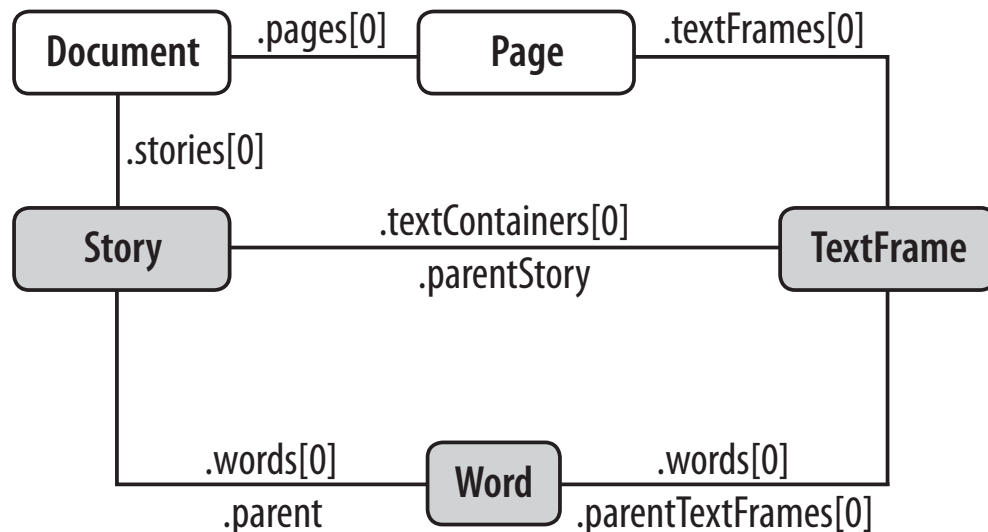
`parentTextFrames[0]` führt zum Textrahmen (`textFrame`)

- **Textrahmen**

Direktes Eltern-Objekt mit parent ist die Seite/ ab CS5 der Druckbogen oder eine Gruppe

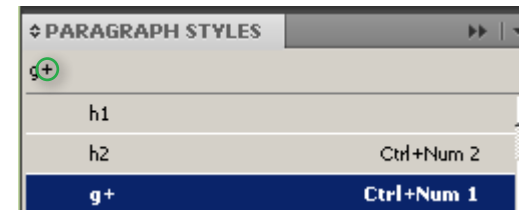
`parentStory` führt zur `story` und damit zu den Textobjekten

`parentPage` führt ab CS5 immer zur Seite.



Formatierung prüfen

- Öffnen Sie die Datei `15_texte.indd` in InDesign
- ! Wie kann man prüfen, ob das Absatzformat eines Absatzes im Layout nicht verändert wurde (kleines Plus bei Formatänderungen in der Absatzformatpalette)?
- ! Prüfen Sie das Dokument auf veränderte Absätze.
- ! Finden Sie jeweils den **Textbereich** im Absatz der verändert wurde.



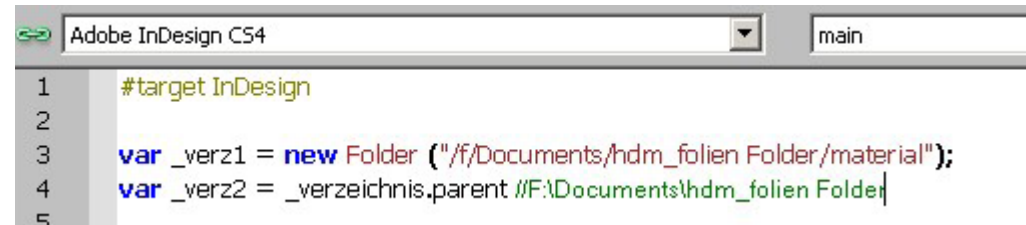
Dateien und Verzeichnisse

Aus Sicherheitsgründen (JS ist eine Websprache) hat JavaScript eigentlich keinen Zugriff auf lokale Dateien. Adobe hat die Objekte **File** und **Folder** implementiert um dies zu ermöglichen.

- Sehr leicht kann man eine Dateireferenz mit `File.openDialog()` herstellen. Der Rückgabewert ist dann schon ein richtiges File-Objekt.
- Ohne Benutzerinteraktion muss der Pfad als String übergeben werden

```
var _file = new File("/c/verzeichnis/unterverzeichnis/datei.indd")
```

 - Plattformabhängig Mac/PC
 - Immer / (Forward-Slash) verwenden nie \ (Back-Slash)
- Mit der Eigenschaft `exists` kann geprüft werden ob die Datei überhaupt existiert.
- Die Eigenschaft `parent` gibt bei Dateien den Ordner in dem die Datei liegt, bei Ordnern den übergeordneten Ordner zurück.



Dateien öffnen, platzieren, speichern, schließen...

- ! Stellen Sie mit der Funktion `File.openDialog()` eine Dateireferenz her.
- ! Stellen Sie die gleiche Referenz mit `new File ("/P/fad_zur/Datei")` her.
- ! Verwenden Sie die Eigenschaft `parent` um den Ordner der Datei zu ermitteln.
- ! Das Objekt `app` hat die Funktion `open(FILE)` als Ergebnis erhält man die Referenz auf das geöffnete Dokument.
Lassen Sie den Benutzer eine InDesign Datei auswählen und öffnen Sie diese mit der Funktion `open()`.
- ! Dokumente haben die Funktionen `save()` und `close()`, speichern Sie die Datei unter einem anderem Namen und schließen Sie die aktuell geöffnete Datei.
- ! **Zusatz:** Die Funktion `getFiles()` liefert alle Dateien eines Ordners zurück. Bereiten Sie einen Ordner mit mehreren Bild-Dateien vor und platzieren Sie diese in ihrem Dokument.