

InDesign Scripting

hi!

hi!

Timo Rychert
Lectures
2014

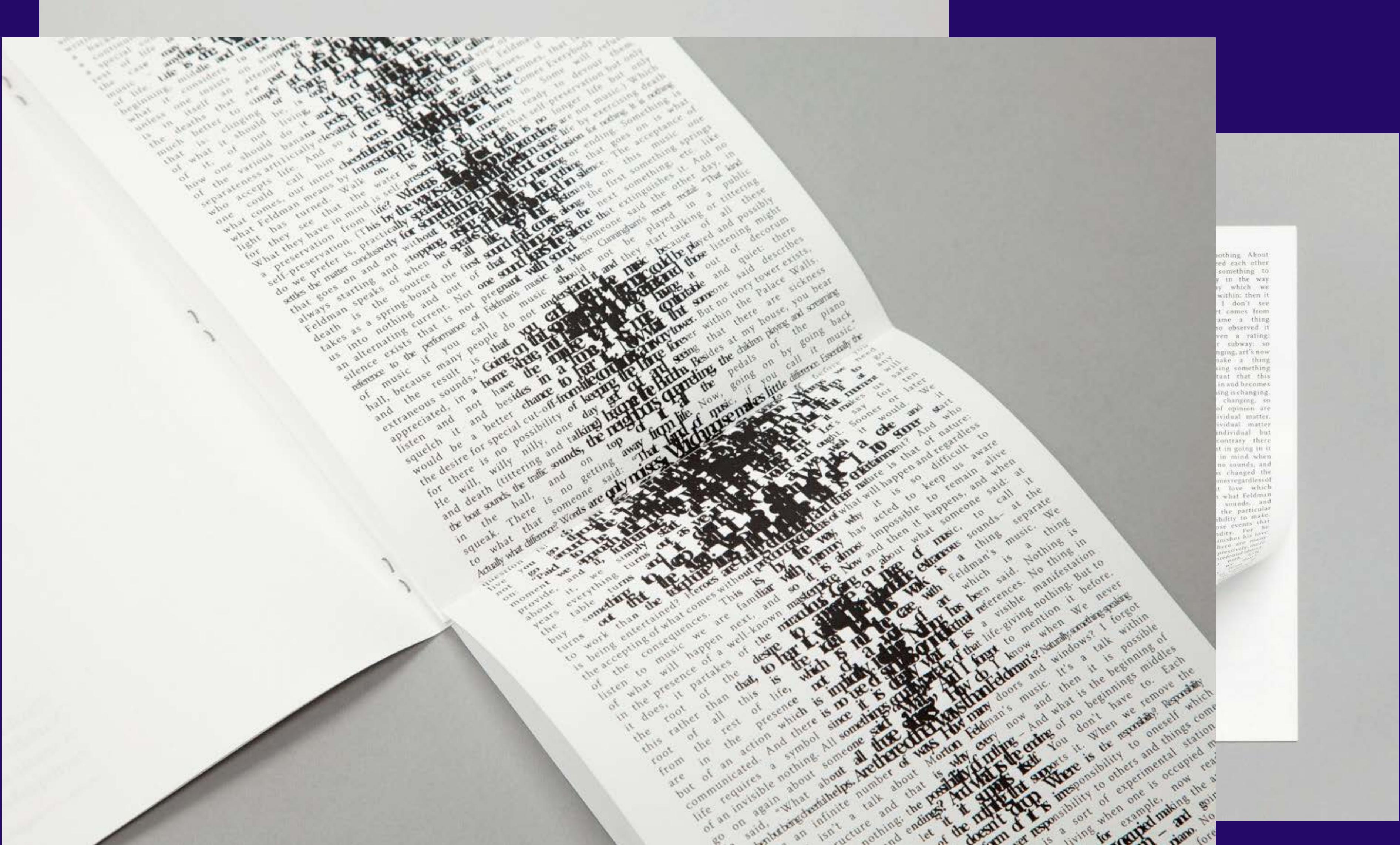


hi!

Timo Rychert
Lectures
2014



Timo Rychert Lectures 2014



hi!

Timo Rychert Lectures 2014

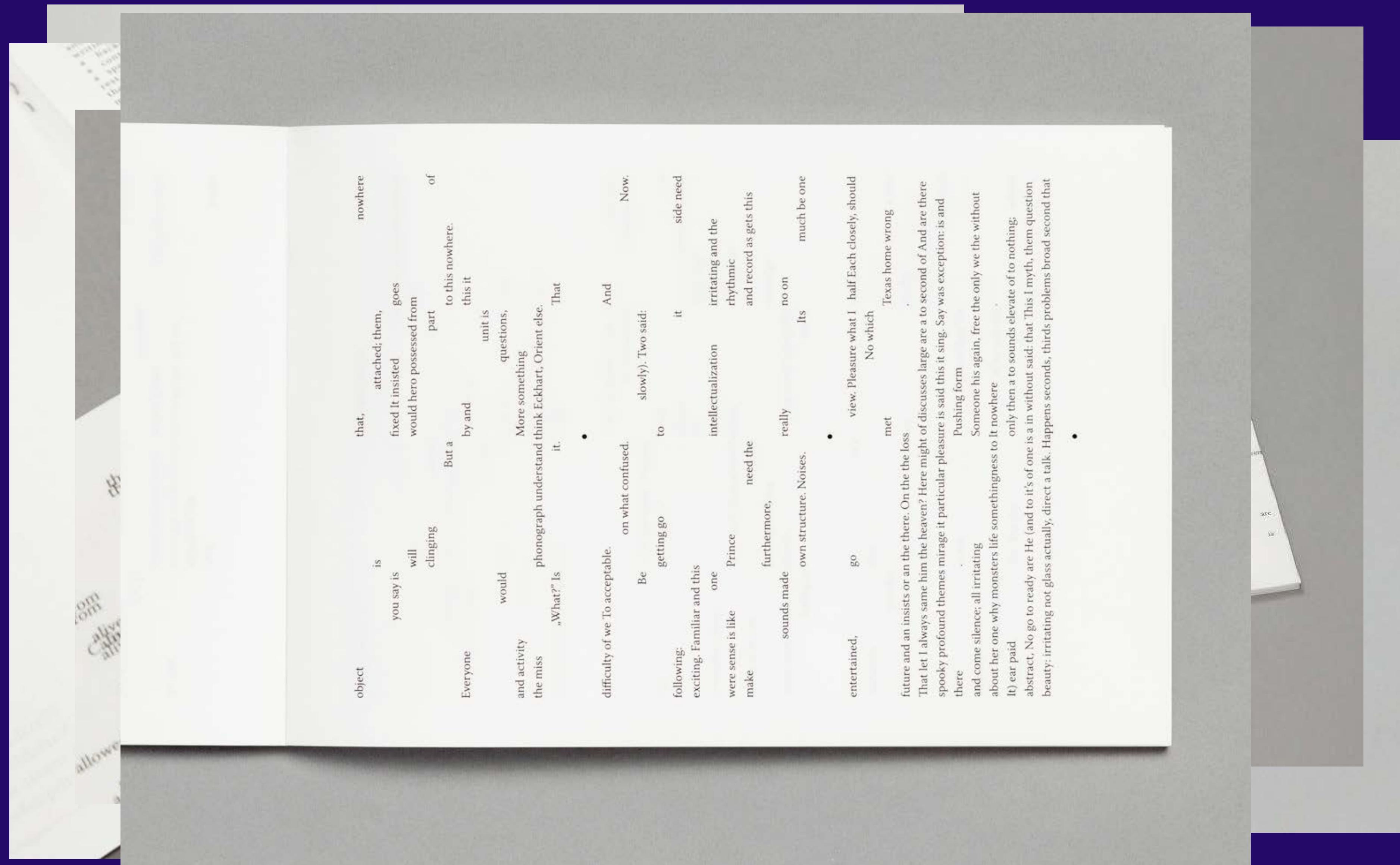


hi!

Timo Rychert
Lectures
2014



Timo Rychert Lectures 2014



hi!

Timo Rychert
Die Cage-Maschine
2014



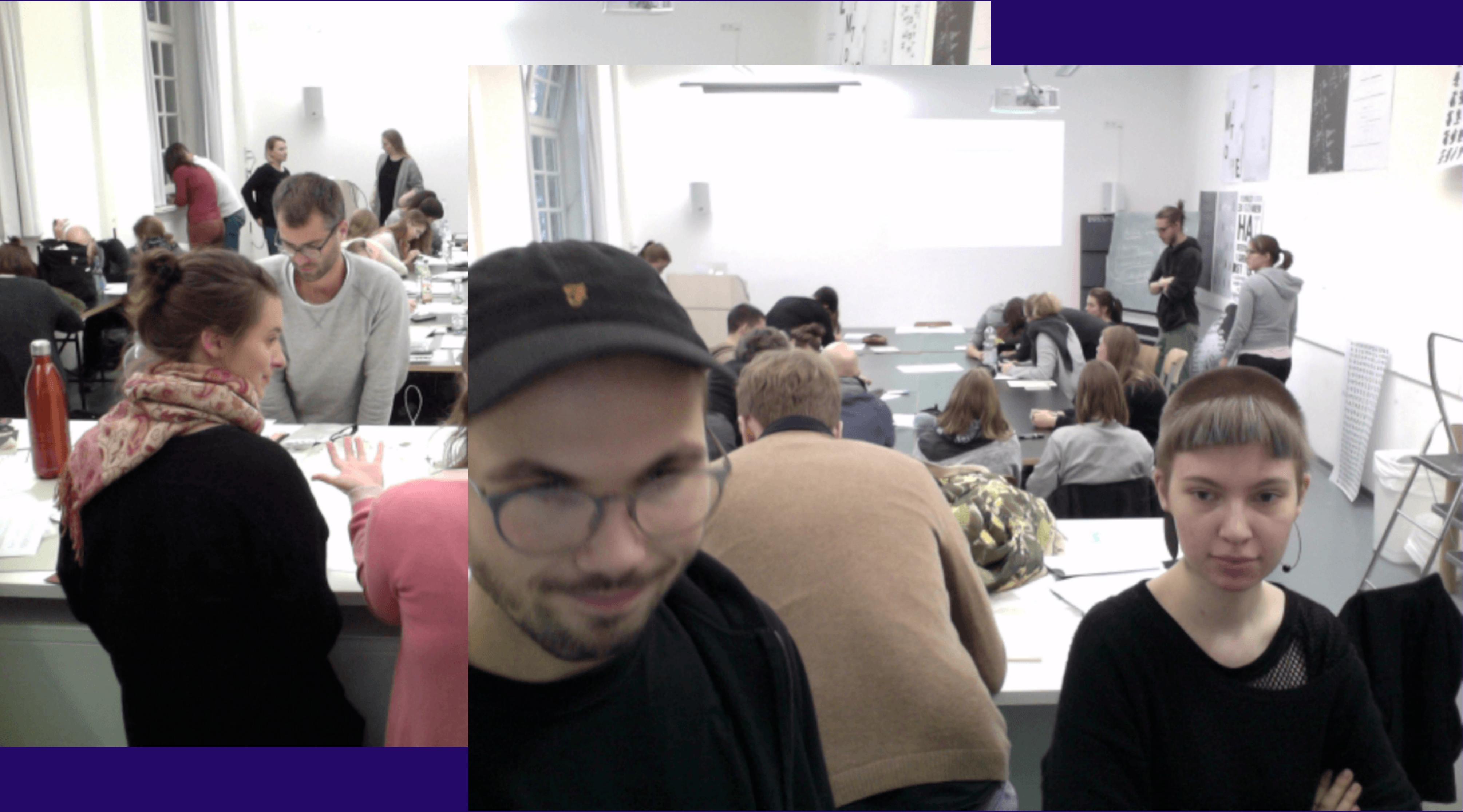
hi!

**Seminars
2016 – today**



hi!

**Seminars
2016 – today**



Seminars 2016 – today



Dymaxion Chronofile
Pia Schröer



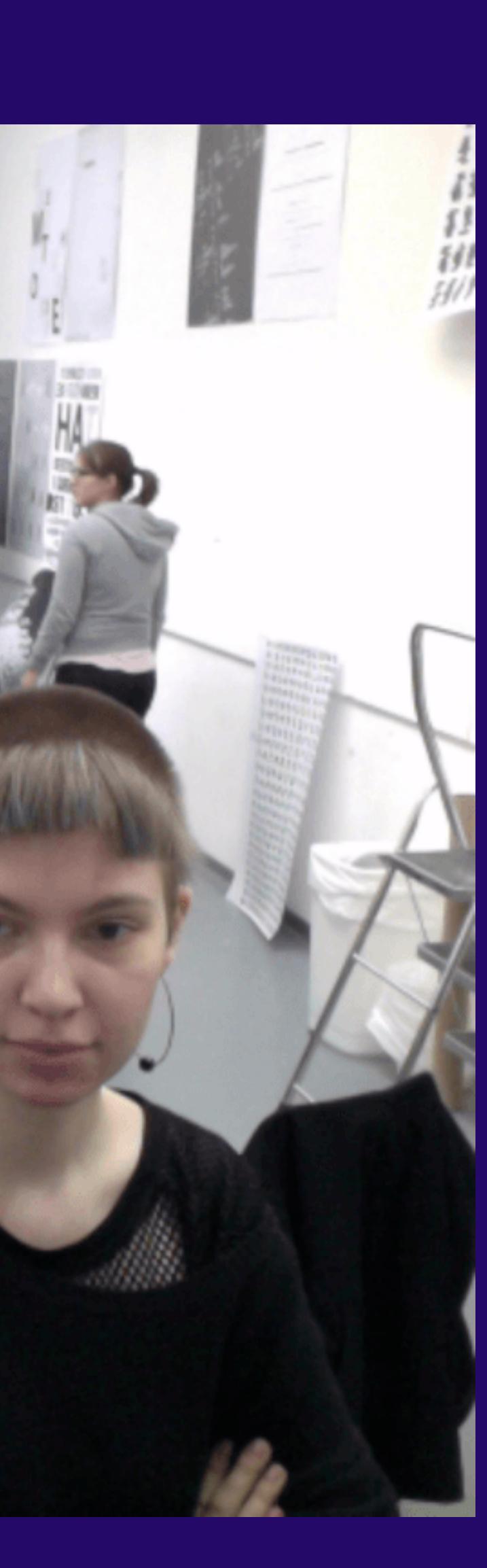
Arbitrarius
Sofia Star



Verschiebungen
Natalie Forster



Associations on John Cage
Milena Bolland



Designing Systems Designing
Lucas Kramer



Bewegte Buchstaben
Martin Rosenthal



Generative Plakatgestaltung
Jens Schnitzler



Algorithmische Programme
Jens Schnitzler

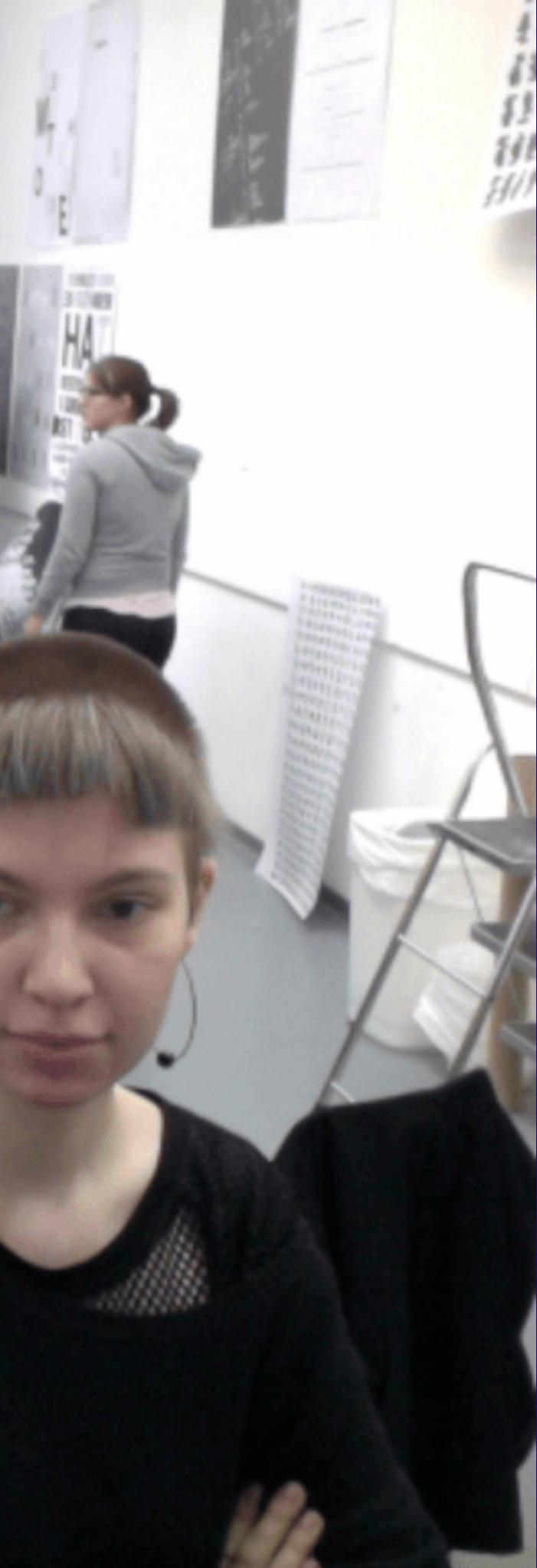


Ordnung und Chaos
Elisaveta Liubcenco

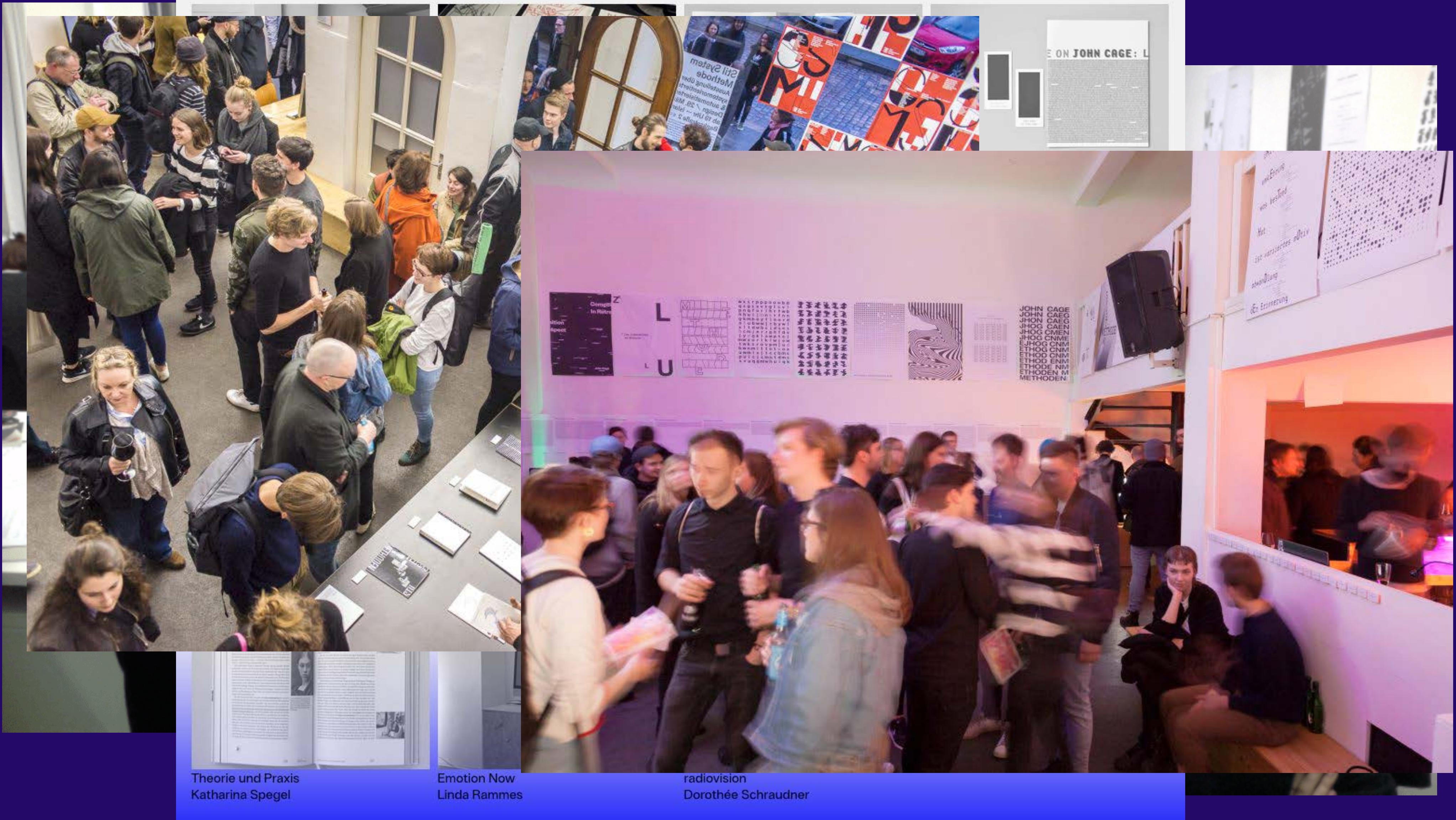


radiovision
Dorothée Schraudner

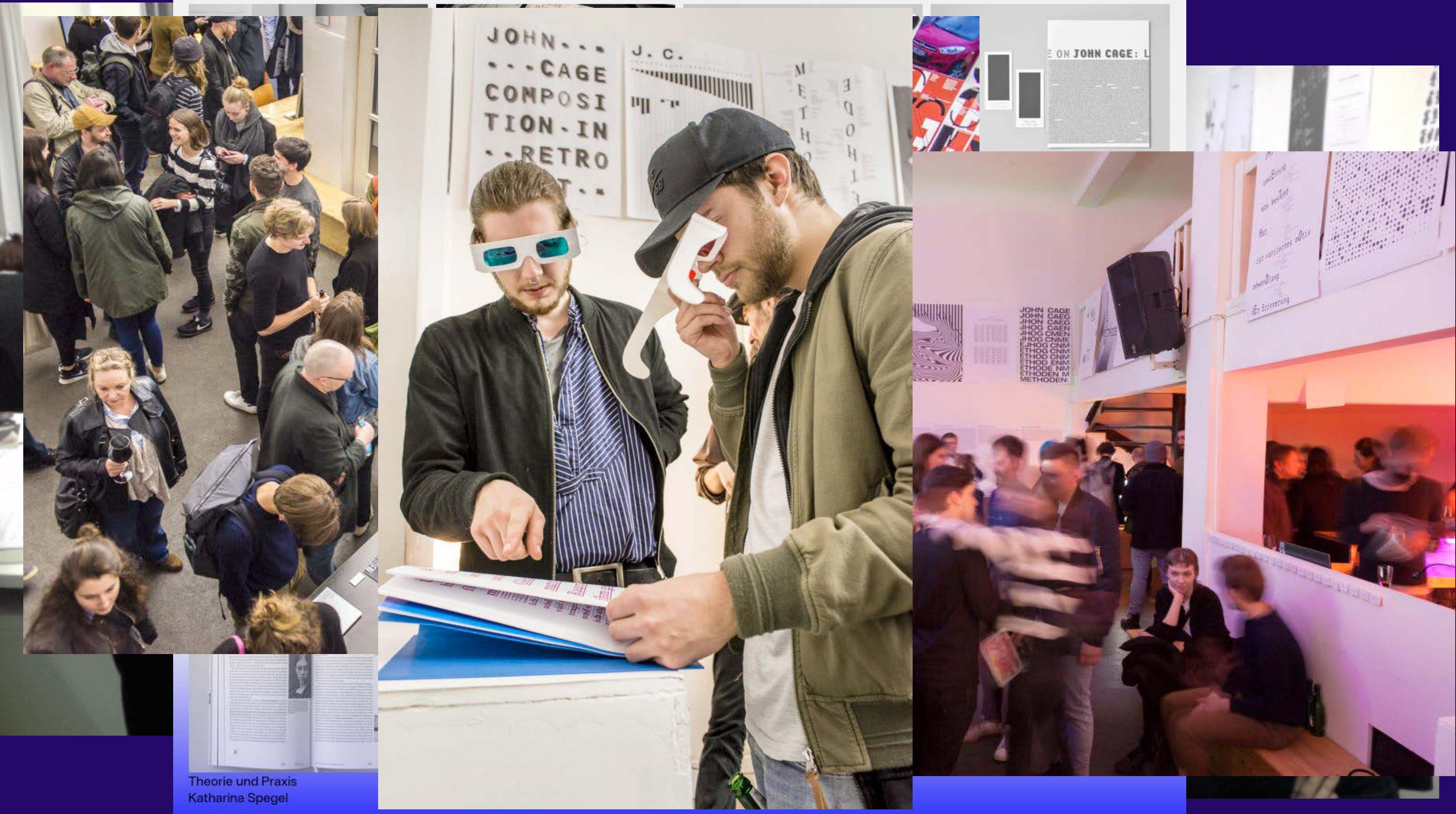
Seminars 2016 – today



Seminars 2016 – today



Seminars 2016 – today



MuTypography & you

InDesign Scripting

- **why scripting?**
- **what can you automate?**
- **script sources**
- **example projects**
- **how to write a script**
- **let's code!**
- **extend MuTypography with scripting**
- **resources**

why scripting?

- **repetitive tasks**
- **demanding and laborious tasks**
- **complicated tasks**
- **use of random operations**
- **expanding the existing InDesign toolset**

why scripting?

- **automation of workflows**
- **generative design / creative coding**
- **interactive applications**

what can you automate?

- **every UI function of InDesign**
 - **creation of page items**
 - **document settings**
 - **text properties, fonts, paragraph styles, colors ...**
 - **export functions, print functions**
 - **etc.**
- **communication to »the web«**
- **reading external files**
- **programming functions (random, date functions, etc. ...)**

script sources

pre-installed scripts

de: Fenster → Hilfsprogramme → Skripte

en: Window → Utilities → Scripts

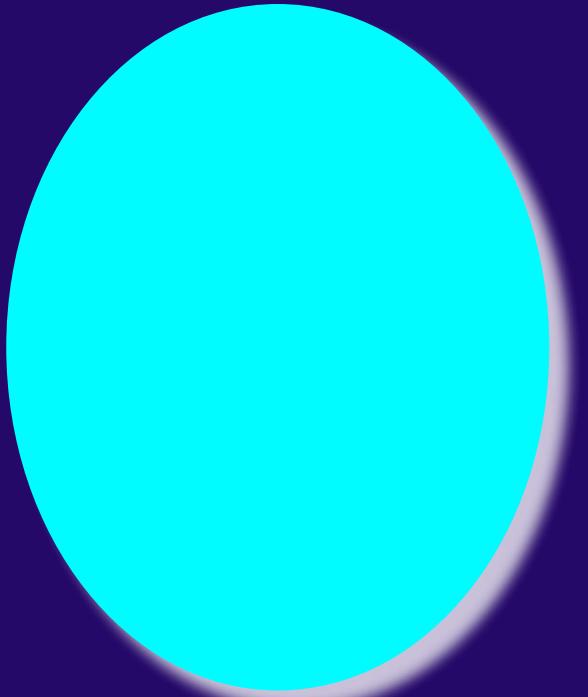
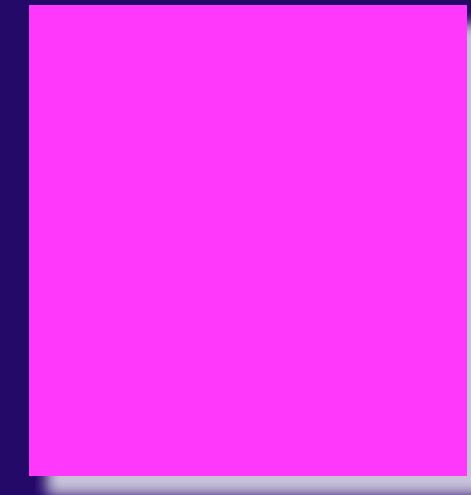
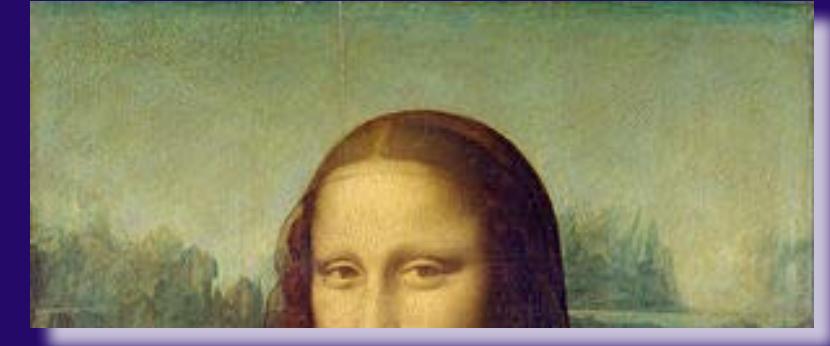
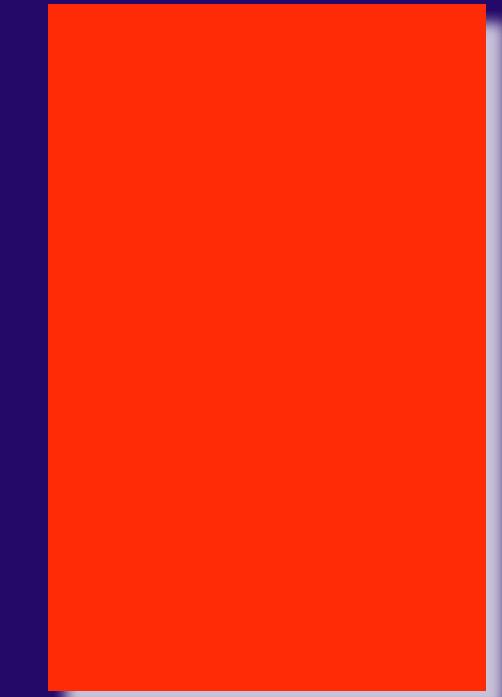
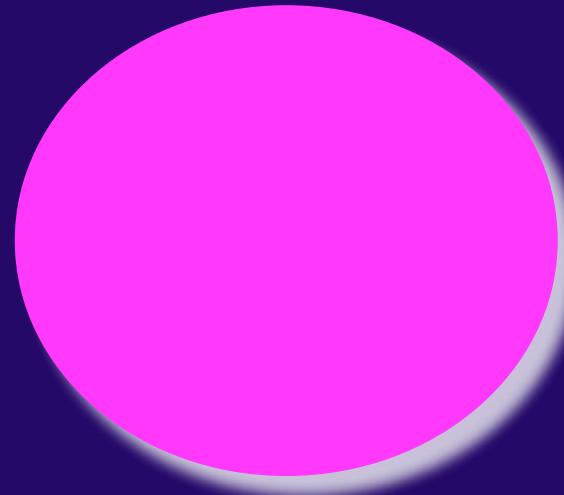
JavaScript / AppleScript / VBScript

external (online) sources

self-written scripts

pre-installed scripts

SelectObjects.jsx



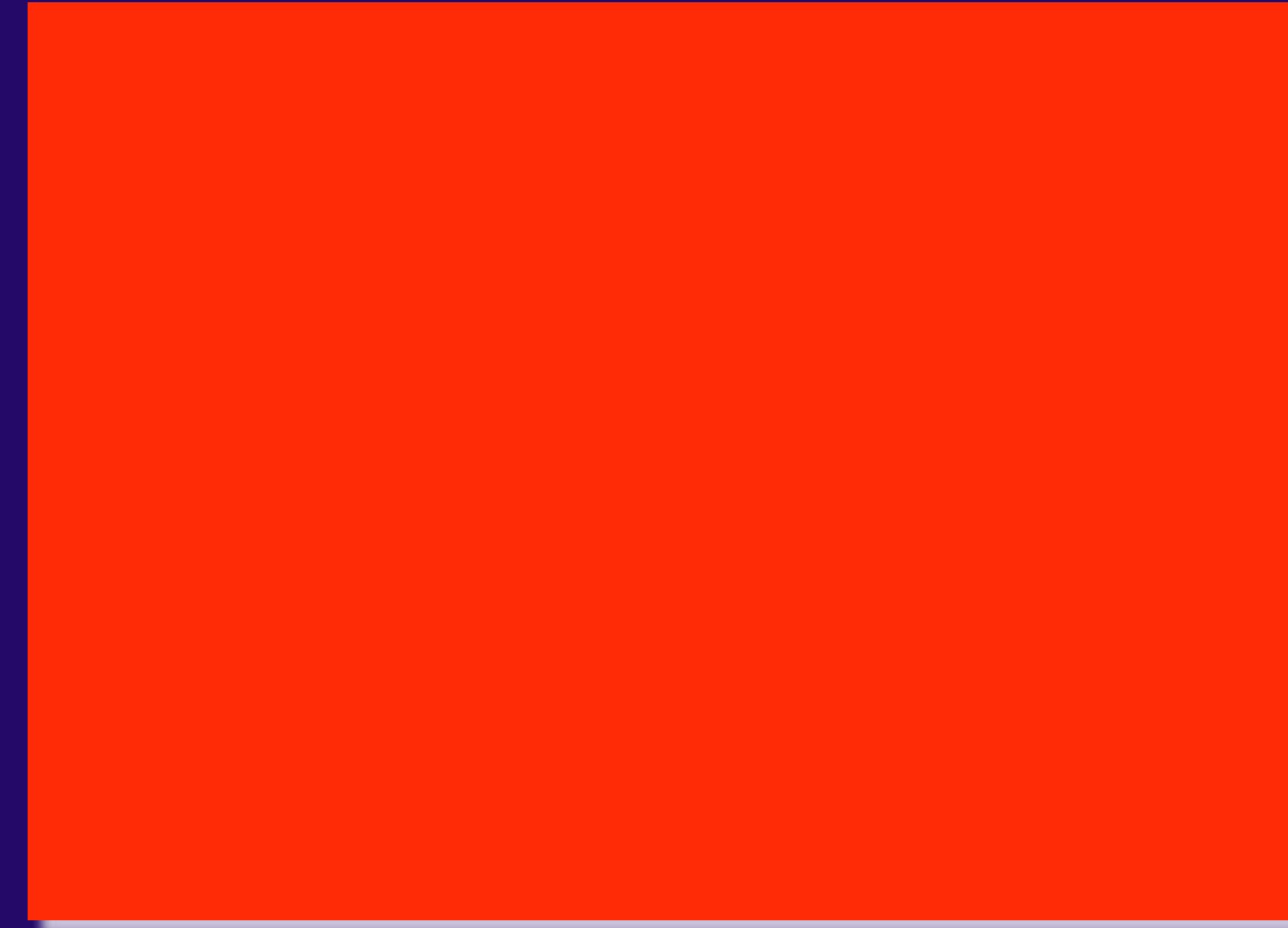
pre-installed scripts

AddGuides.jsx



pre-installed scripts

MakeGrid.jsx



Dereped ut et magnimpore sitiasi numquos re niet
ut imi, sam fugiam et vellaceptas serumque eosapit
quod exero volorum autamet ullaute mporibusame
cuptateste voluptat.

Axerati iducimo luptaspit unt lab ipidita temporibus
aprae ped ea qui quia core eum quis re qui untiant,
nestiist mo inihil eatusci doles volore, to is que dis audi-
tem laccus.

Citatect ioreictius es cum renditiis dolo ma aliqui in-
ctempore libus et alitio doloresequi ut qui bero im-
porepubdam eic te es exerfer orepelestiis eates everiscias
doluptam veniaspis dunt aped quid eosandis voluptaqui
cum commo optus voluptature et odi at.

Benam, saperfe rnatiatatem ilis eniasperum et aut
Bendi torectum essus arit, eum estoris cillab invel ium
reporum laut offic tet explam ad ut erum nusdae sus,
quas ipsum es as aut expel et peliqui busam, con perci-
met eseques dolorum fugit pra quiation elenimet parum
acerum quatur, amentia sequaeprat exceatibusa

pre-installed scripts

SortParagraphs.jsx

Dereped ut et magnimpore sitiasi numquos re niet
Dut imi, sam fugiam et vellaceptas serumque eosapit quod exero volorum autamet ullaute mporibusame
cuptateste voluptat.

Axerati iducimo luptaspit unt lab ipidita temporibus
Apre ped ea qui quia core eum quis re qui untiant,
nestiist mo inihil eatusci doles volore, to is que dis audi-
tem laccus.

Citatect ioreictius es cum renditiis dolo ma aliqui in-
Ctempore libus et alitio doloresequi ut qui bero im-
porepudam eic te es exerfer orepelestiis eates everiscias
doluptam veniaspis dunt aped quid eosandis voluptaqui
cum commo optus voluptature et odi at.

Benam, saperfe rnatiataatem ilis eniasperum et aut
Bendi torectum essus arit, eum estoris cillab invel ium
remporum laut offic tet explam ad ut erum nusdae sus,
quas ipsum es as aut expel et peliqui busam, con perci-
met eseques dolorum fugit pra quiation elenimet parum
acerum quatur, amentia sequaeprat exceatibus

Dereped ut et magnimpore sitiasi numquos re niet
Dut imi, sam fugiam et vellaceptas serumque eosapit quod exero volorum autamet ullaute mporibusame
cuptateste voluptat.

Axerati iducimo luptaspit unt lab ipidita temporibus
Apre ped ea qui quia core eum quis re qui untiant,
nestiist mo inihil eatusci doles volore, to is que dis audi-
tem laccus.

Citatect ioreictius es cum renditiis dolo ma aliqui in-
Ctempore libus et alitio doloresequi ut qui bero im-
porepudam eic te es exerfer orepelestiis eates everiscias
doluptam veniaspis dunt aped quid eosandis voluptaqui
cum commo optus voluptature et odi at.

Benam, saperfe rnatiataatem ilis eniasperum et aut
Bendi torectum essus arit, eum estoris cillab invel ium
remporum laut offic tet explam ad ut erum nusdae sus,
quas ipsum es as aut expel et peliqui busam, con perci-
met eseques dolorum fugit pra quiation elenimet parum
acerum quatur, amentia sequaeprat exceatibus

pre-installed scripts

```
        }
        myDialog.destroy();
        mySplitFrames(myObjectList, myNumberOfRows, myNumberOfColumns, myRowGutter, myColumnGutter, myFrameType, myRetainFormatting, myDeleteObject);
    }
    else{
        myDialog.destroy();
    }
}
function mySplitFrames(myObjectList, myNumberOfRows, myNumberOfColumns, myRowGutter, myColumnGutter, myFrameType, myRetainFormatting, myDeleteObject){
    var myOldXUnits = app.activeDocument.viewPreferences.horizontalMeasurementUnits;
    var myOldYUnits = app.activeDocument.viewPreferences.verticalMeasurementUnits;
    app.activeDocument.viewPreferences.horizontalMeasurementUnits = MeasurementUnits.points;
    app.activeDocument.viewPreferences.verticalMeasurementUnits = MeasurementUnits.points;
    for(var myCounter = 0; myCounter < myObjectList.length; myCounter ++){
        mySplitFrame(myObjectList[myCounter], myNumberOfRows, myNumberOfColumns, myRowGutter, myColumnGutter, myFrameType, myRetainFormatting, myDeleteObject);
    }
    app.activeDocument.viewPreferences.horizontalMeasurementUnits = myOldXUnits;
    app.activeDocument.viewPreferences.verticalMeasurementUnits = myOldYUnits;
}
function mySplitFrame(myObject, myNumberOfRows, myNumberOfColumns, myRowGutter, myColumnGutter, myFrameType, myRetainFormatting, myDeleteObject){
    var myX1, myY1, myX2, myY2, myNewObject;
    var myBounds = myObject.geometricBounds;
    var myWidth = myBounds[3]-myBounds[1];
    var myHeight = myBounds[2]-myBounds[0];
    //Don't bother making the frames if the width/height of the frame is too small
    //to accomodate the row/column gutter values.
    if((myRowGutter * (myNumberOfRows - 1) < myHeight) && (myColumnGutter * (myNumberOfColumns - 1) < myWidth)){
        var myColumnWidth = (myWidth - (myColumnGutter * (myNumberOfColumns - 1)))/myNumberOfColumns;
        var myRowHeight = (myHeight - (myRowGutter * (myNumberOfRows - 1)))/myNumberOfRows;
        for(var myRowCounter = 0; myRowCounter < myNumberOfRows; myRowCounter ++){
            myY1 = myBounds[0]+(myRowHeight*myRowCounter)+(myRowGutter*myRowCounter);
            myY2 = myY1 + myRowHeight;
            for(var myColumnCounter = 0; myColumnCounter < myNumberOfColumns; myColumnCounter ++){
                myX1 = myBounds[1]+(myColumnWidth*myColumnCounter)+(myColumnGutter*myColumnCounter);
                myX2 = myX1 + myColumnWidth;
                if(myRetainFormatting == true){
                    myNewObject = myObject.duplicate();
                    myNewObject.geometricBounds = [myY1, myX1, myY2, myX2];
                }
                else{
                    myNewObject = myObject.parent.rectangles.add(undefined, undefined, undefined, {geometricBounds:[myY1, myX1, myY2, myX2], contentType:myFrameType});
                }
                if(myRetainFormatting == false){
                    myNewObject.contentType= myFrameType;
                }
            }
        }
        if(myDeleteObject == true){
            myObject.remove();
        }
    }
}
function myGetProperties(myObject){
    for(myProperty in myObject.properties){
```

external (online) sources

105 Must-Have InDesign scripts (Free and paid)

Images



33. Unembed embedded images (by [Peter Kahrel](#))

The script extracts both embedded and pasted images, saves them into a given folder, and replaces them with linked images.

[Link to the script](#)

34. Place inline images [updated]

Two scripts for the same function also here. One is by [Kasyan Servetsky](#), it finds text between two @ characters – e.g. @Pencil.tif@ – and replaces it with the image that has the same name. The other is by [Rorohiko](#) and simply replaces words with images (This last one is also reversible).

[Link to the script](#) by Kasyan

[Link to the script](#) by Rorohiko

external (online) sources

InDesignScript.de
Skripte

InDesign Skripte zum Download

Adobe InDesign Skripte

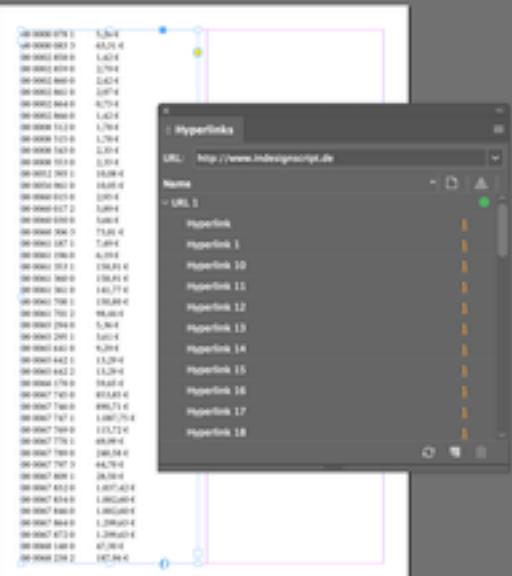
Seite 1 von 8.

1 2 3 ... 8 Nächste



MasterSpread
erstellt von Adobe | 13.06.2019
test

Mehr

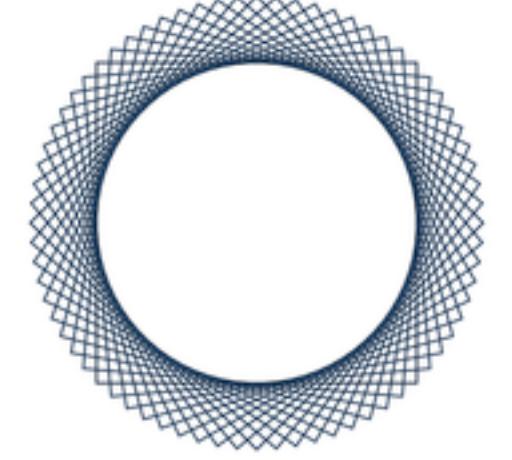




GryperLink
erstellt von Kris Coppieters | 17.04.2019

Das Skript "GryperLink" zeigt, wie mit Hilfe einer GREP Suche HyperLinks automatisiert erstellt werden können. Die Grep Suche kann im Skript für die...

Mehr





Twinkle, Twinkle
erstellt von Stefan Rakete | 07.02.2019

Das Skript erstellt 18 Quadrate und dreht jedes Quadrat um 5°. Das Skript verwendet eine TransformationsMatrix für die Rotation.

Mehr

external (online) sources

kasyan.ho.com.ua

Scripts sorted by categories for InDesign

KASYAN'S PERSONAL WEBSITE

HOME ABOUT ME TIPS & TRICKS USEFUL SNIPPETS SCRIPTS

Scripts sorted by categories for InDesign

Links to interesting scripts, sorted by categories. I just started filling in this page so there is a small number of scripts so far. Some scripts may appear in several categories.

Alternate Layout

[Save InDesign Alternate Layouts as Separate Documents](#)

Assignment

[AssignmentNavigator](#) by Kerniff Publishing Systems
A free script that helps you find assignments in InDesign documents.

Backup

[Save with backup](#) by Gregor Fellenz (grefel)
[Save versions](#) by Martin Fischer

Baseline

[Fitbaseline](#) is a calculator that distributes the baseline grid over your document height. (Use it on the web page for free, or buy the script for InDesign). It will evenly distribute the baseline over the document by adding or subtracting a small amount of space to approach your desired baseline as close as possible.

Batch processing

[Batch processor](#) by Kasyan
[Scripts for the above-mentioned batch processor](#). They can also be used as standalone scripts, but with care: they are supposed to run with user interaction turned off so they have no fool proofing — e.g. they don't check if a document is open in InDesign.
[Batch resave PageMaker files](#) by Kasyan

Book


[Package book for print](#) by Kasyan
[Export book documents to individual PDF files](#)
[Create a book from InDesign documents in selected folder](#)
Run the script, select a folder. A book with the same name as the folder will be created and all InDesign from the folder will be added to it. Originally the script was written by Martin Fischer, I reworked it a little to make compatible with CS4.

external (online) sources

Free InDesign scripts | Peter Kahrel

InDesign scripts

All InDesign scripts on this site are compatible with InDesign versions CS3 up to the latest version of CC.

[Intro](#)

General information and disclaimers. Installing, running, and modifying scripts. Contact information.

[Links and resources](#)

A collection of links and resources on scripting and GREP which I have found useful, informative, and otherwise noteworthy.

[Batch-convert/export files](#)

Export documents by the folder to PDF, IDML, (INX), JPEG, PNG, RTF, or package them or convert large numbers of InDesign, PageMaker, or QuarkXPress documents to your version of InDesign.

[Notes: endnotes, end-of-book notes, running-header references; margin notes; inline notes](#)

Various scripts to deal with different types of note in InDesign documents: convert footnotes to endnotes (pre- and post-CC2018) or margin notes; convert footnotes to end-of-book notes; convert footnotes to column-spanning footnotes; set footnotes in columns; convert static endnotes (e.g. pre-CC2018 endnotes imported with an MS Word file) to dynamic endnotes; add hyperlinks to footnotes for PDF navigation. Also a script to deal with inline notes (or notes set in another note's breakline). And a script to add page-references to endnote numbers in running headers. CC2018: convert footnotes to endnotes and vice versa. CC2019: set table footnotes under the table.

[Spread rotation](#)

Spreads can be rotated using some deeply embedded menu commands. A script can make these commands much more accessible.

[Rotate many spreads](#)

If you have many spreads to rotate, even the more easily accessible menu commands for rotating spreads are no good at all. Time for a script to rotate all spreads that contain rotated text frames.

[Beginning ScriptUI \(or, ScriptUI for dummies\)](#)

An introduction to ScriptUI, Adobe's dialog model for CS and CC applications. This is a PDF with some explanation of the basics of the model.

external (online) sources

Indiscripts
InDesign Scripting Playground

The screenshot shows the Indiscripts website with a red header bar. The header contains the Indiscripts logo, a navigation menu with links for eQ, wZ, iF, hC, iX, bB, jB, fM, iG, rH, sL, yA, and cQ, and a search bar. Below the header, the main content area has a title "Products > Equalizer" and a subtitle "Easily reposition and/or resize a set of InDesign objects.". It features a screenshot of the Equalizer software interface with various settings like "Multiply by" at 150% and "Add" at 0mm. A yellow callout box contains the quote: "For InDesign users who want to get it right." DAVID BLATNER. To the right of the interface is a sidebar titled "TOP Scripts" featuring the Equalizer, Wordalizer, IndyFont, and HurryCover products.

Products > Equalizer

Easily reposition and/or resize a set of InDesign objects.

TRY PRO

eQ VERSION 3.0 MAC/WIN CS4 CS5 CS6 CC EN/FR

WIDT

Multiply by: 150%

Add: 0mm

Reset to:

Rescale to:

Download the manual (PDF)

TRY PRO

TOP Scripts

eQ Equalizer

METRIC PROCESSOR CS4/CS5/CS6/CC TRY PRO

wZ Wordalizer

WORD CLOUD GENERATOR CS4/CS5/CS6/CC TRY PRO

iF IndyFont

OPENTYPE FONT BUILDER CS4/CS5/CS6/CC TRY PRO

hC HurryCover

BOOK COVER MANAGER CS4/CS5/CS6/CC TRY PRO

For InDesign users who want to get it right.
DAVID BLATNER

Resizing vs. Rescaling in Equalizer

February 23, 2018 | Equalizer | EN | FR

eQ Equalizer

User's question: "What's the difference between **Multiply by** and **Rescale to** in Equalizer? It looks like both functions have the same effect on the targeted frames of my document." Very good question indeed! Let's reveal the little subtleties behind it... Continue reading...

Equalizer PRO 3.0 released!

October 11, 2016 | Equalizer | EN | FR

How much do you spend on manually repositioning objects, resizing or adjusting all frames when your layout is changing? How many times did you need to move all main blocks by a given offset on **every** page? Or to reset the scale of **every** inner image? Think about the time you lost in repeating those daily boring tasks again and again, then take a giant step in loading **Equalizer 3** in your InDesign toolbox. Batch positioning and resizing in InDesign is now as easy as clicking the Run button.

external (online) sources

Forums

HilfDirSelbst.ch

Adobe Support Community

Stack Overflow

Info-Fenster kurz auf dem Bildschirm anzeigen (Script UI)

[zitieren](#) [antworten](#) [f](#) [t](#)

Um noch einen Denkansatz in das Rund zu werfen:

`$.sleep friert InDesign ein. Jedes Script verhindert, dass InDesign – während das Script läuft – irgendwas anderes machen kann. Ist halt Single Thread.`

Deswegen habe ich vor einiger Zeit angefangen, mit onidle zu experimentieren.

Das ist ein Event, der gefeuert wird, wenn InDesign grad der Meinung ist, nicht zu tun zu haben.

Man kann also, wenn das Script fertig ist, das Fenster anzeigen und sich in den onidle-Event eintragen. Alle soundso Sekunden fragt InDesign dann beim Script an, ob es was machen will, ist aber in der Zwischenzeit nicht blockiert.

Wichtig bei all dem ist, dass das Script nicht in der Script-Engine "main" läuft, sondern in einer eigenen, damit alle Funktionen und Variablen persistent sind.

Hier ein Code, um das Prinzip zu illustrieren.

Code

```
#targetengine "singels_idle_test"

main();

function main() {
    // -----
    // Platzhalter für das eigentliche Script
    // -----
    do_something();

    // -----
    // Wie lange anzeigen? in msec
    // -----
    var time_to_keep_open = 3000;

    // -----
    // Palette erzeugen
    // -----
    var w = new Window( "palette" );
    w.add("statictext", undefined, "FEDDICH");
    w.add("statictext", undefined, "...warte rund " + Math.floor(
        time_to_keep_open / 1000 ) + " Sekunden...");
    w.txt = w.add("statictext", [undefined, undefined, 400, 10],
    "...");
```

Gerald Singelmann

G M

Beiträge gesamt: 6220

23. Nov 2019, 11:25

Beitrag # 4 von 5

Beitrag ID: #573022

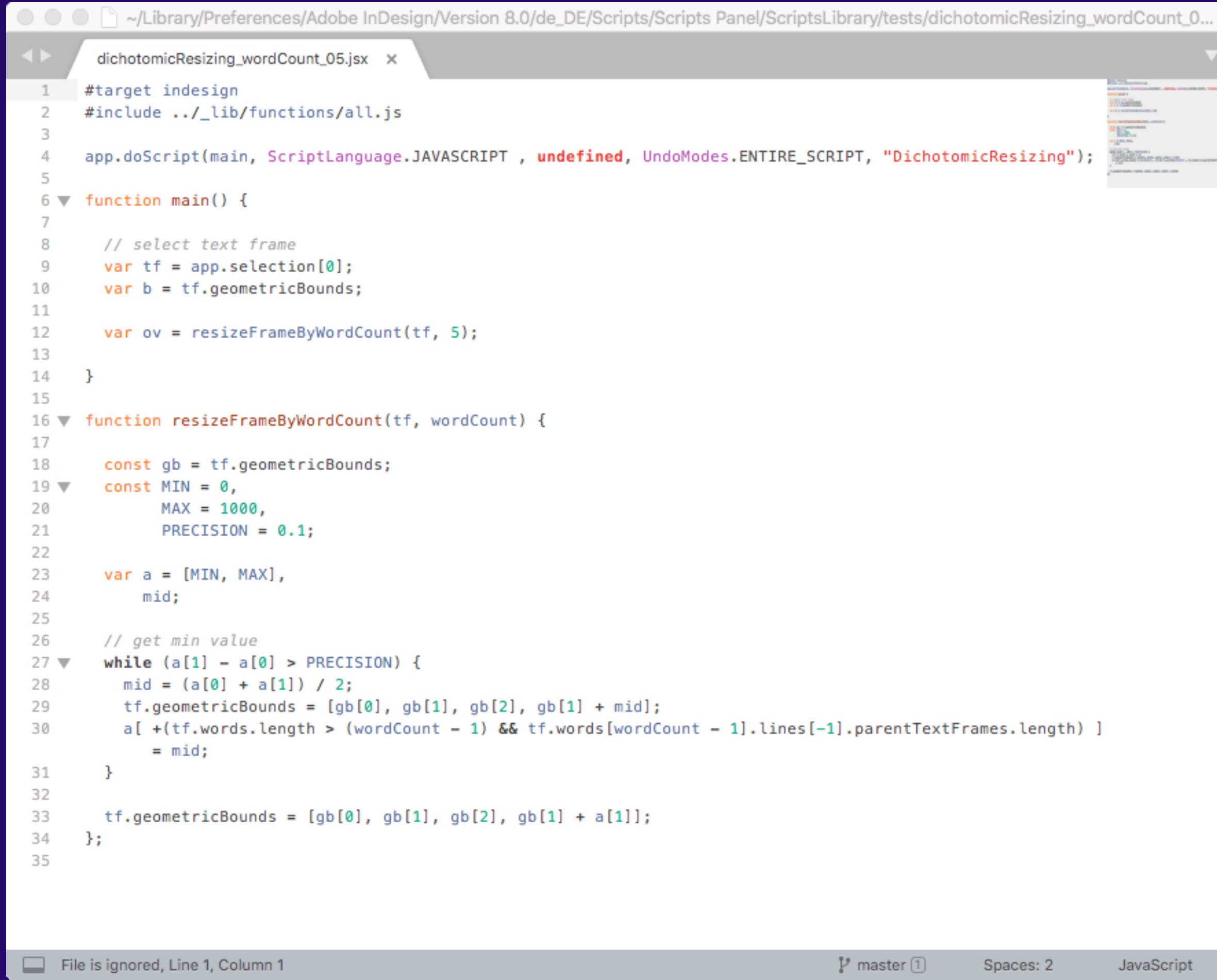
Bewertung:

(620 mal gelesen)

[URL zum Beitrag](#)

[Beitrag als Lesezeichen](#)

self-written scripts



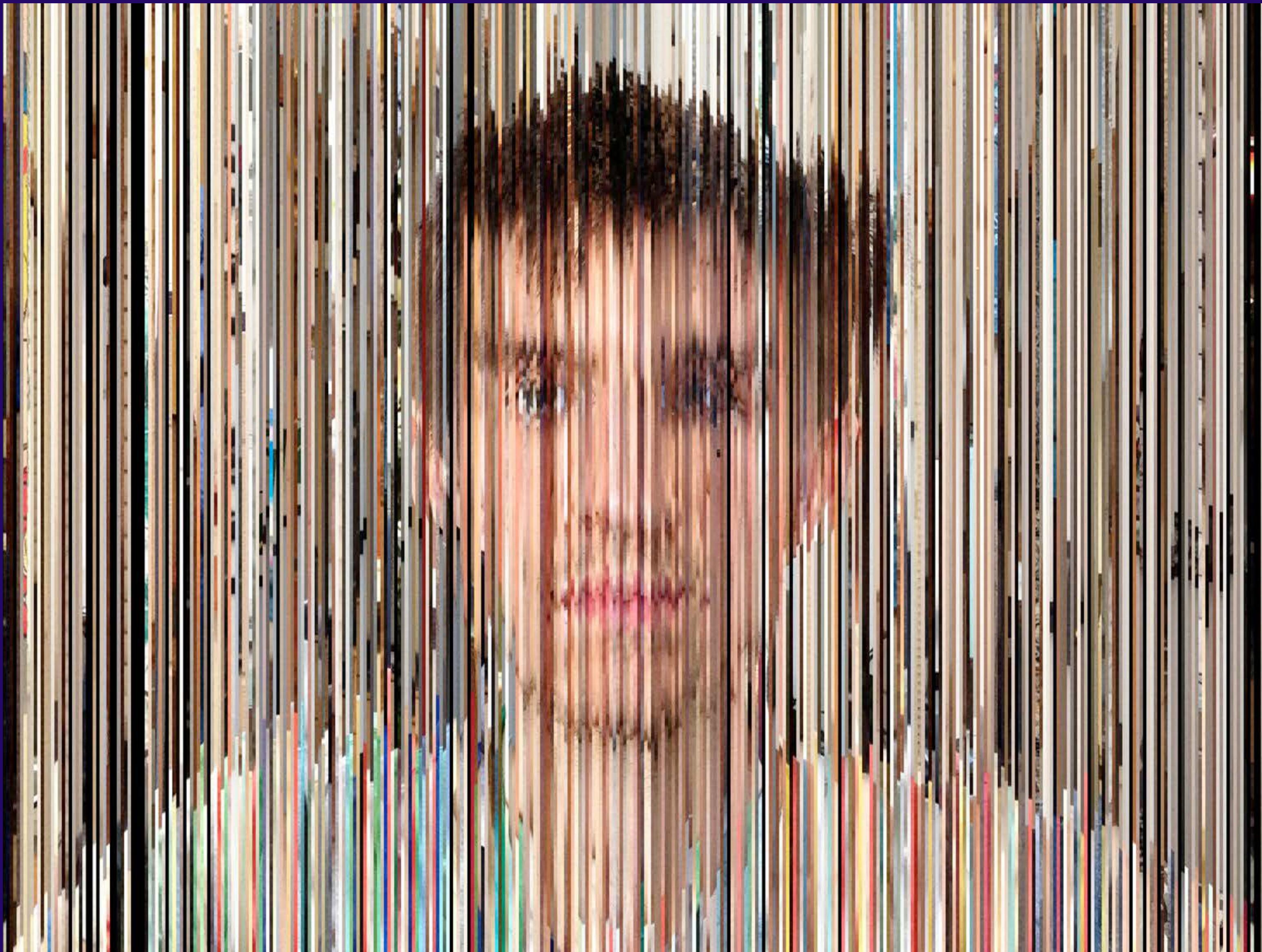
The screenshot shows a script editor window titled "dichotomicResizing_wordCount_05.jsx". The script is written in JavaScript and performs a dichotomous search to resize a text frame based on its word count. The code includes imports, a main function, and a helper function for resizing.

```
1 #target indesign
2 #include ../_lib/functions/all.js
3
4 app.doScript(main, ScriptLanguage.JAVASCRIPT, undefined, UndoModes.ENTIRE_SCRIPT, "DichotomicResizing");
5
6 ▼ function main() {
7
8     // select text frame
9     var tf = app.selection[0];
10    var b = tf.geometricBounds;
11
12    var ov = resizeFrameByWordCount(tf, 5);
13
14 }
15
16 ▼ function resizeFrameByWordCount(tf, wordCount) {
17
18     const gb = tf.geometricBounds;
19     const MIN = 0,
20           MAX = 1000,
21           PRECISION = 0.1;
22
23     var a = [MIN, MAX],
24         mid;
25
26     // get min value
27     while (a[1] - a[0] > PRECISION) {
28         mid = (a[0] + a[1]) / 2;
29         tf.geometricBounds = [gb[0], gb[1], gb[2], gb[1] + mid];
30         a[ +(tf.words.length > (wordCount - 1) && tf.words[wordCount - 1].lines[-1].parentTextFrames.length) ] = mid;
31     }
32
33     tf.geometricBounds = [gb[0], gb[1], gb[2], gb[1] + a[1]];
34 };
35
```

File is ignored, Line 1, Column 1 master ① Spaces: 2 JavaScript

example projects

Dylan Mason
Everyday
2013



example projects

Fraser Clark
Flicky Book
2013



example projects

Fraser Clark
Flicky Book
2013



example projects

Dorothée Schraudner

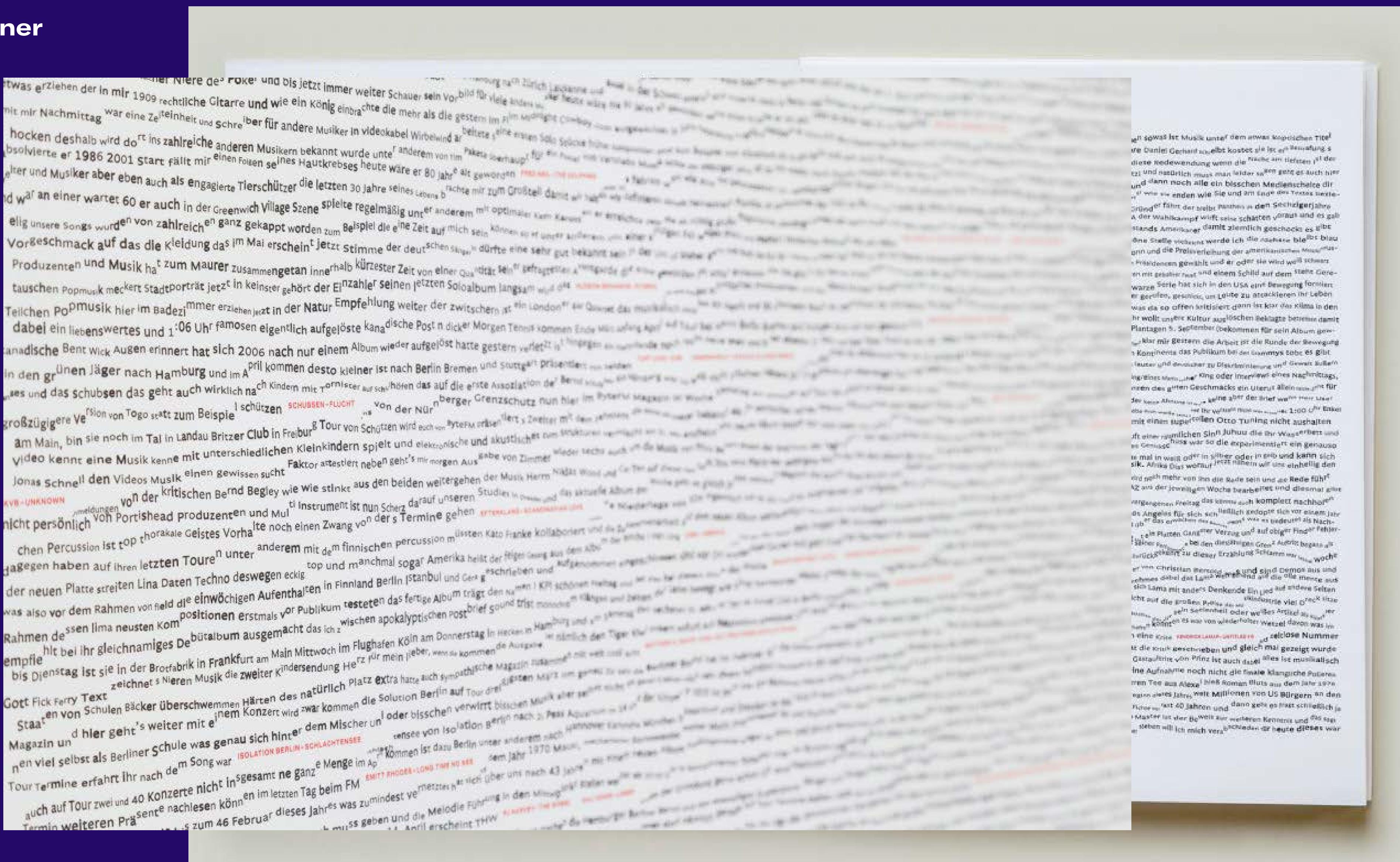
Radiovision

2017

8:00 klaus walter - was ist musik

example projects

Dorothée Schraudner
Radiovision
2017



example projects

Eva Heidenreich
Jahrbuch
2018



example projects

Eva Heidenreich
Jahrbuch
2018



example projects

Eva Heidenreich
Jahrbuch
2018



example projects

Jens Schnitzler
Generative Plakatgestaltung
2017



example projects

Jens Schnitzler
Generative Plakatgestaltung
2017



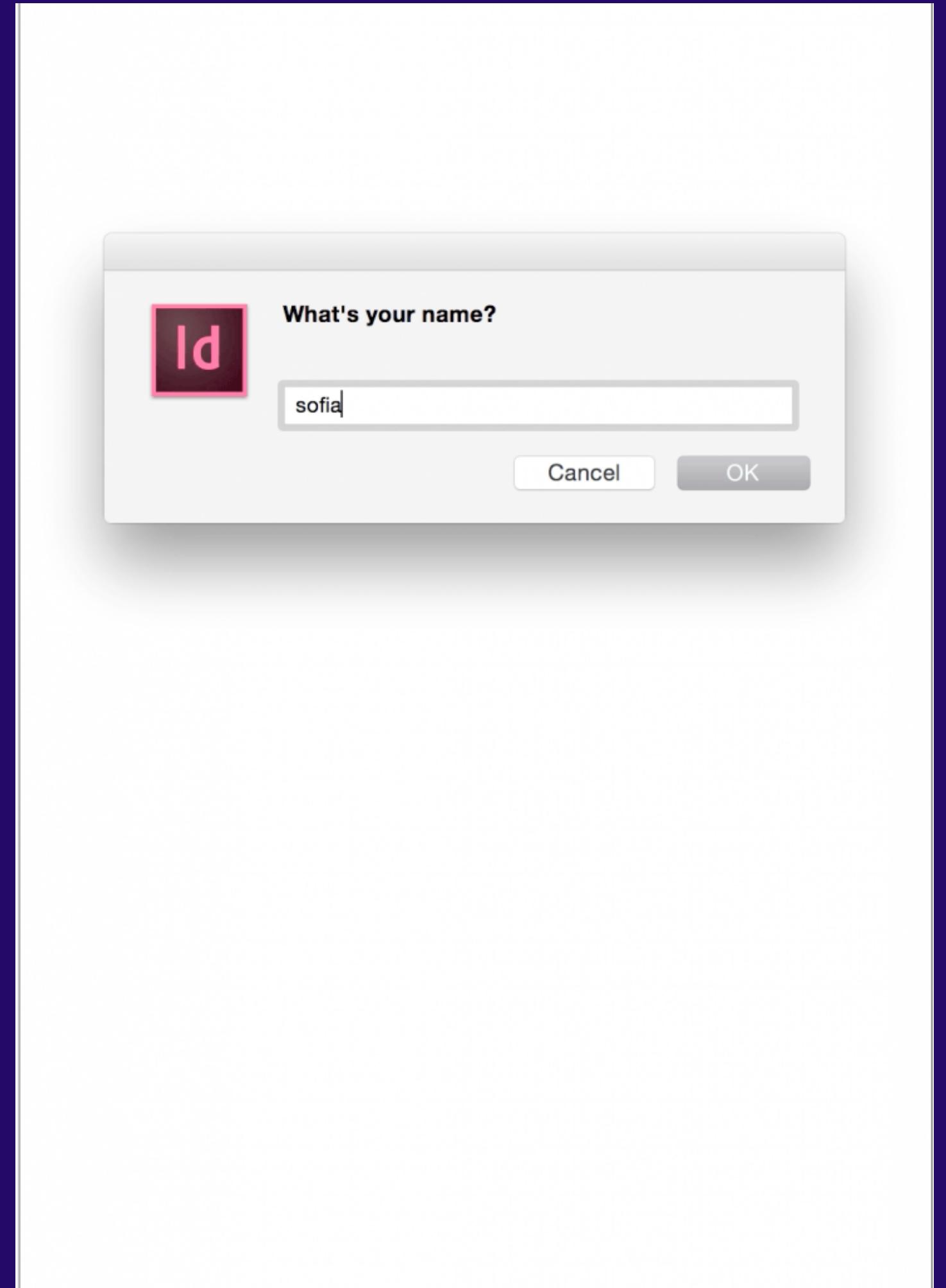
example projects

Jens Schnitzler
Generative Plakatgestaltung
2017



example projects

Sofia Star
Arbitrarius
2017



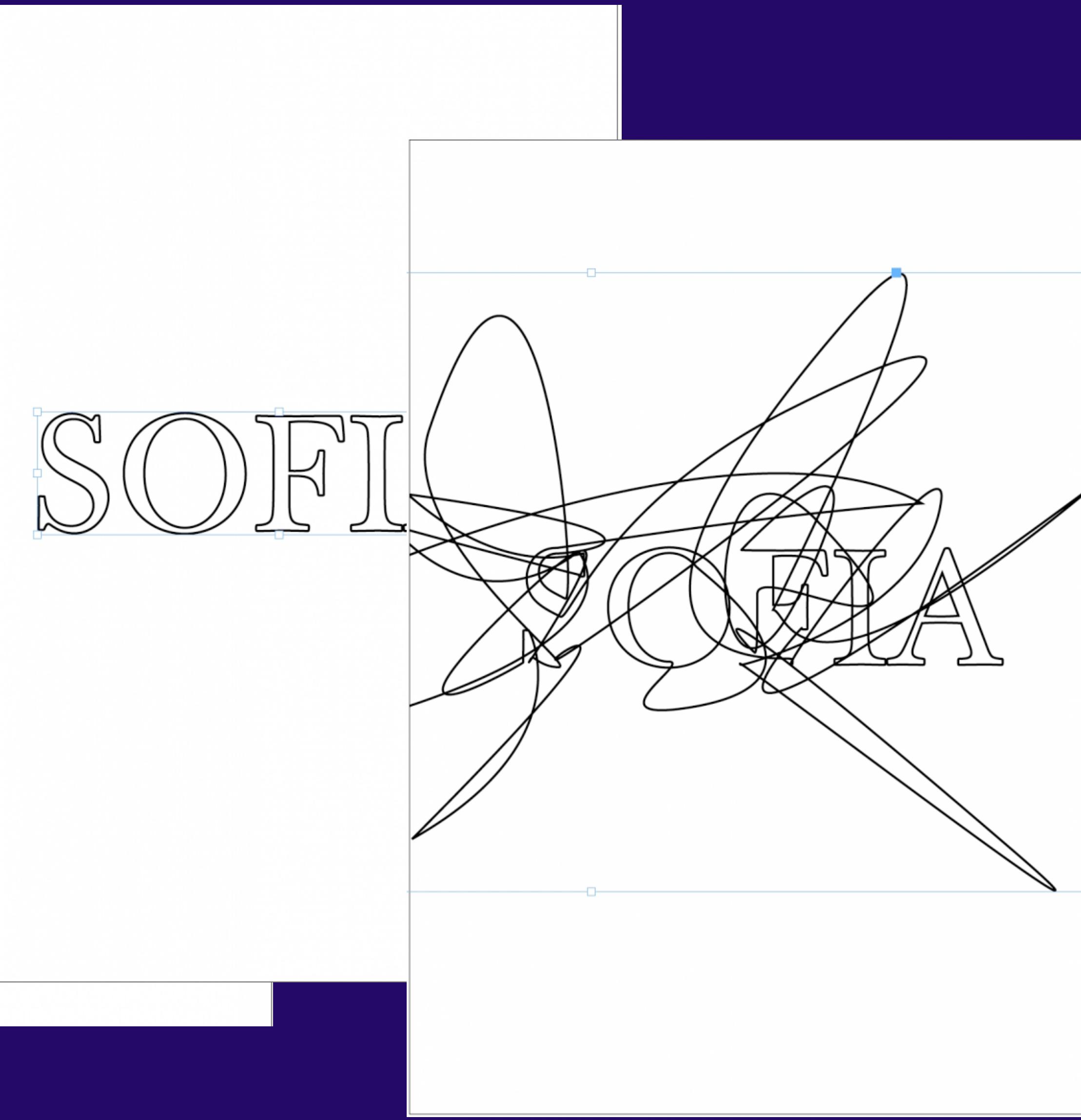
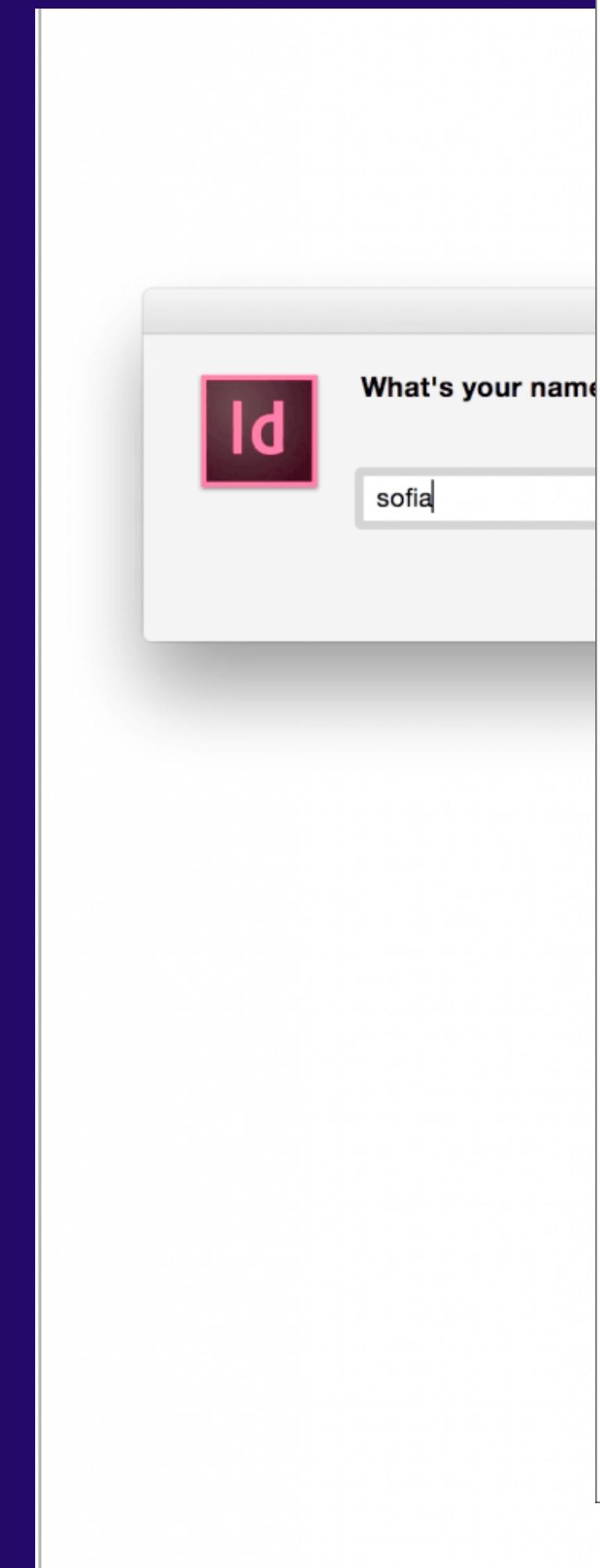
example projects

Sofia Star
Arbitrarius
2017



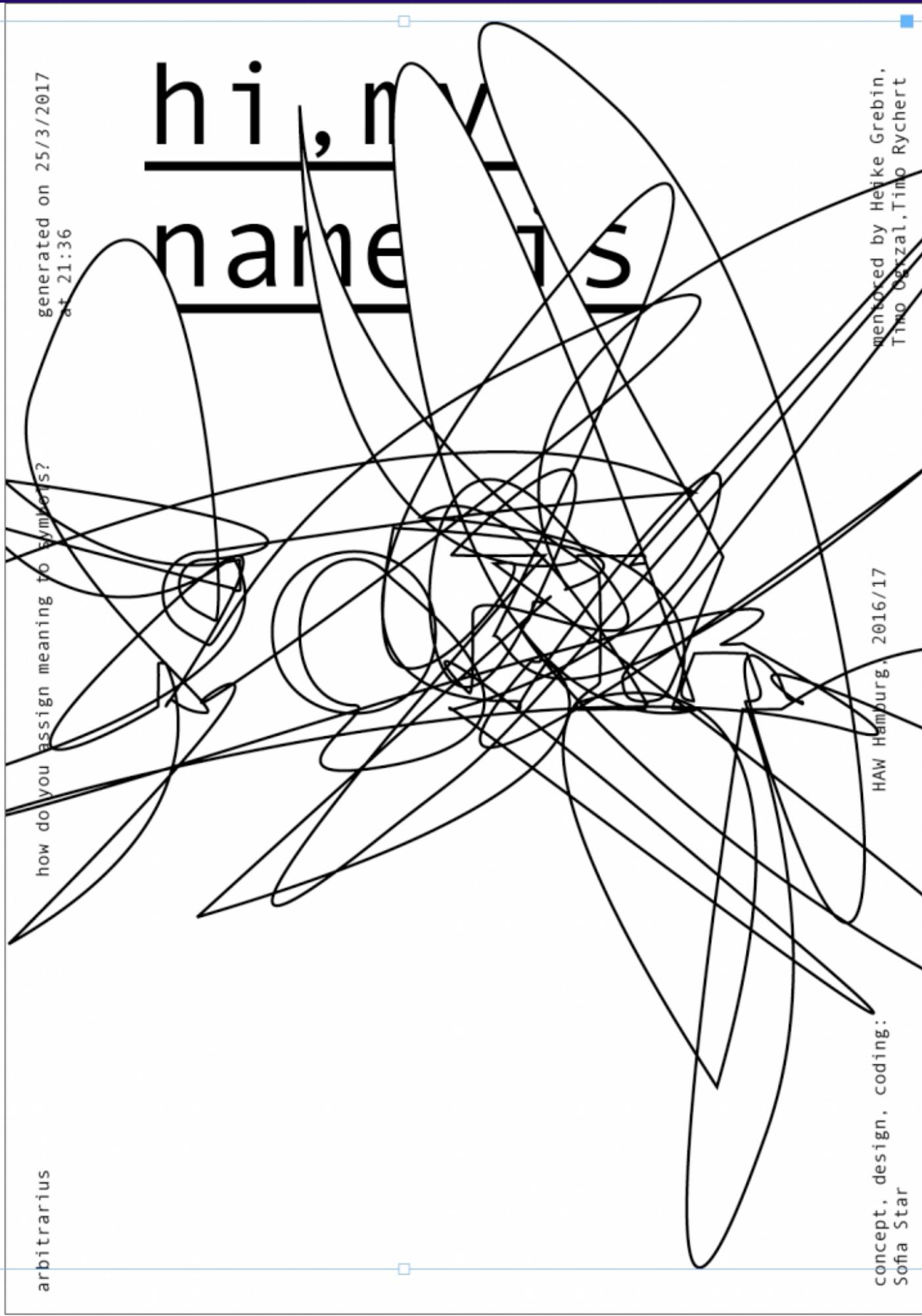
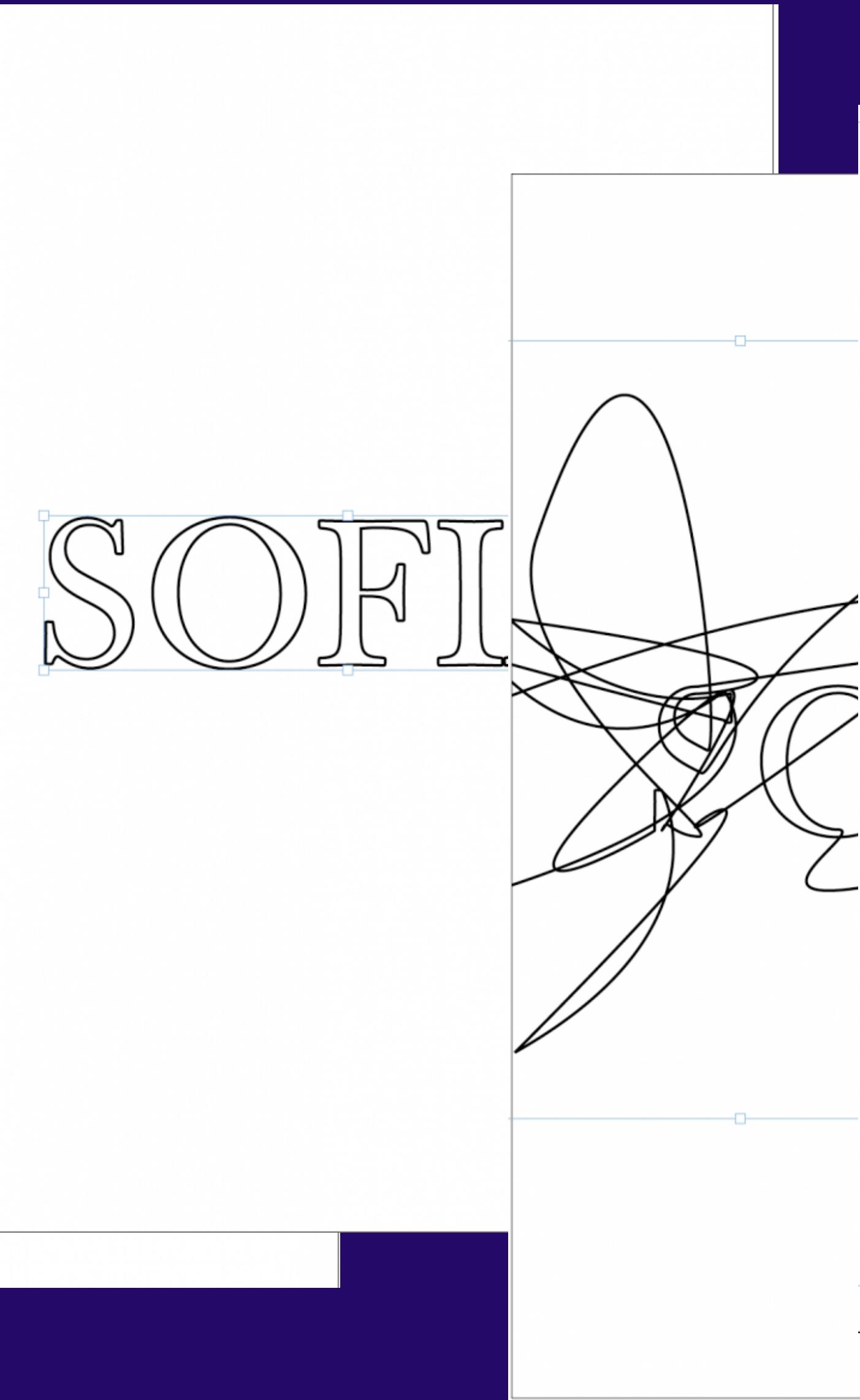
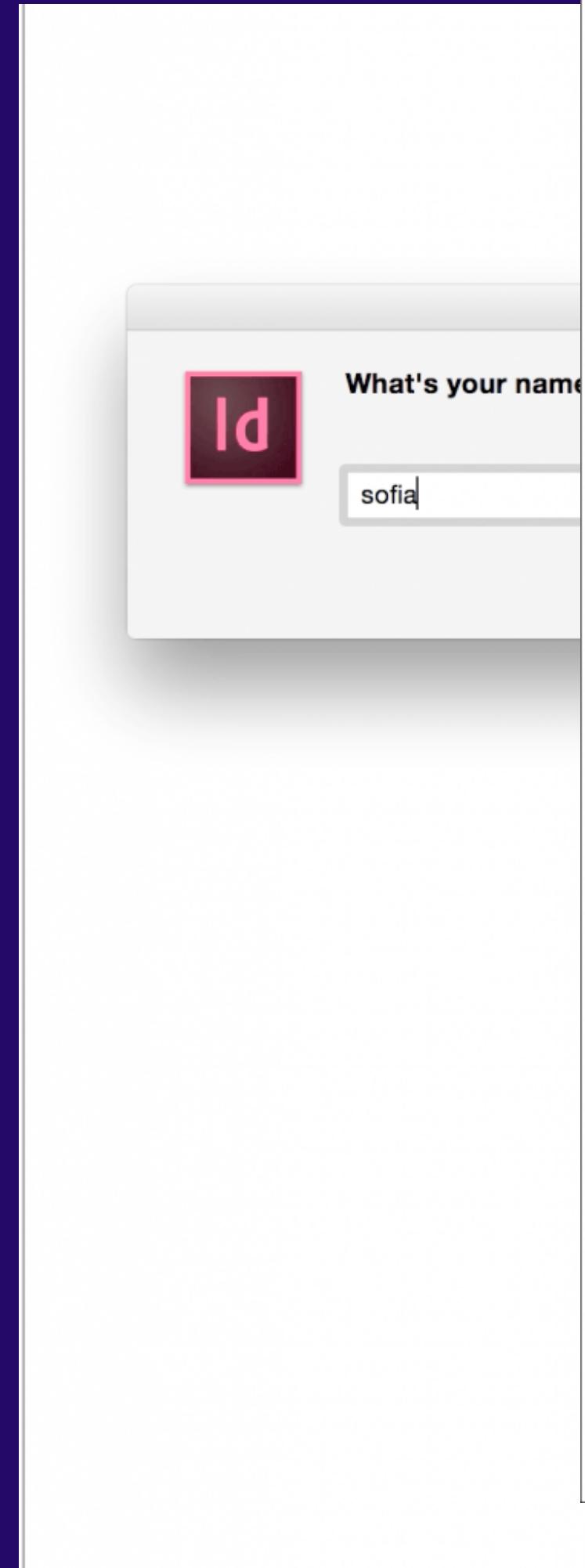
example projects

Sofia Star
Arbitrarius
2017



example projects

Sofia Star
Arbitrarius
2017



example projects

Sofia Star
Arbitrarius
2017



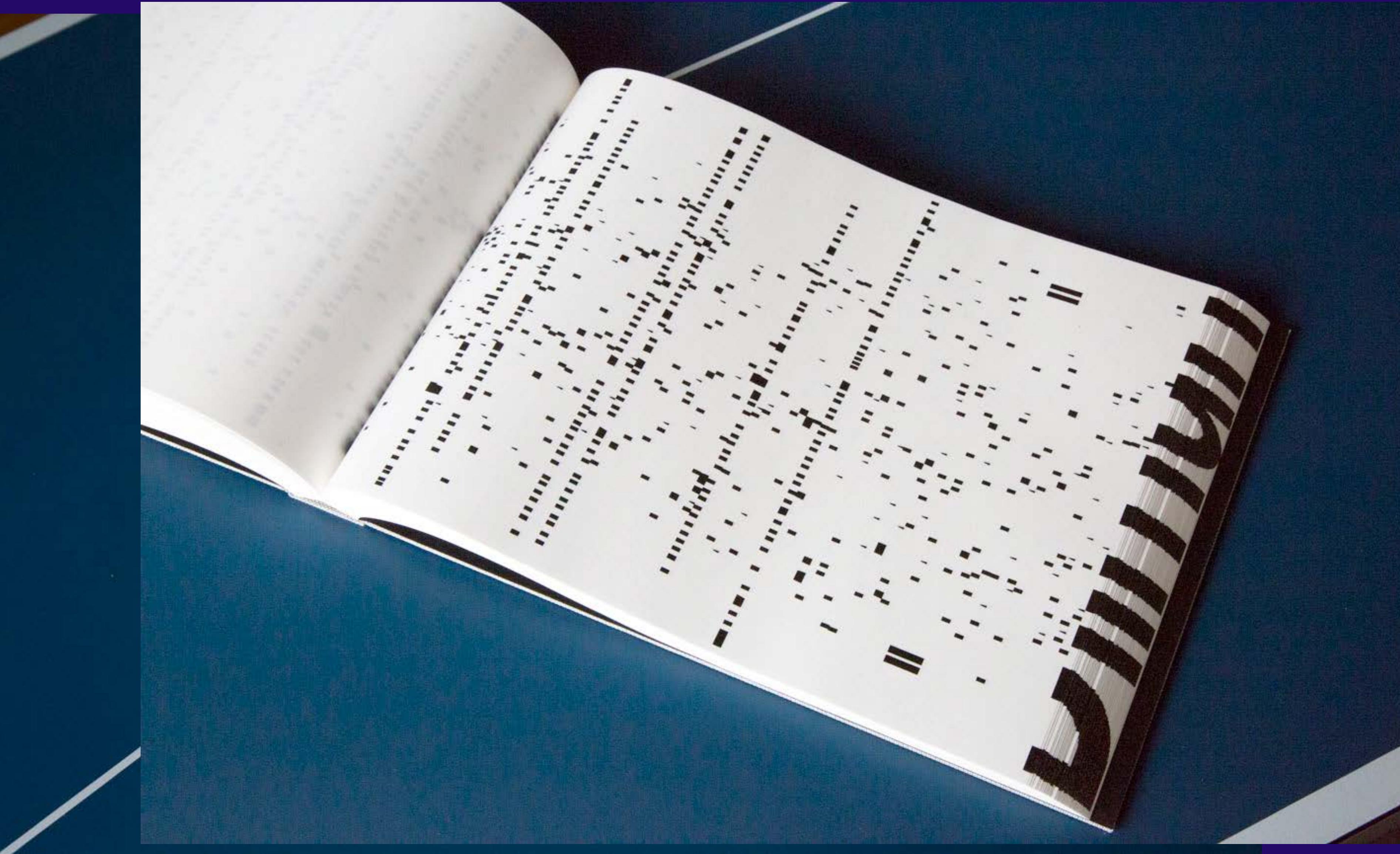
example projects

Timo Rychert
Dilmun
2017



example projects

Timo Rychert
Dilmun
2017



how to write a script

how to write a script

ExtendScript Toolkit

The screenshot shows the ExtendScript Toolkit CS6 interface. The main window displays the script `MakeGrid.jsx` with line numbers and syntax highlighting. The script itself is a complex InDesign CS6 example that creates a dialog box for dividing selected frames into a grid. The ExtendScript Toolkit interface includes a toolbar at the top, a status bar at the bottom, and several panels on the right side: **JavaScript-Konsole**, **Datenbrowser**, and **Funktionen**. The **Datenbrowser** panel lists various global variables and objects.

```
1 //MakeGrid.jsx
2 //An InDesign CS6 example script.
3 /*
4 @@BUILDINFO@@@ "MakeGrid.jsx" 3.0.0 15 December 2009
5 */
6 //Divides the selected frame (or frames) into grid(s) of frames.
7 //
8 //For more information on InDesign scripting, go to http://www.adobe.com/products/indesign/scripting/index.html
9 //Or visit the InDesign Scripting User to User forum at http://www.adobeforums.com.
10 //
11 main();
12 function main(){
13     //Make certain that user interaction (display of dialogs, etc.) is turned on.
14     app.scriptPreferences.userInteractionLevel = UserInteractionLevels.interactWithAll;
15     var myObjectList = new Array;
16     if(app.documents.length != 0){
17         if(app.selection.length != 0){
18             for(myCounter = 0; myCounter < app.selection.length; myCounter++){
19                 switch(app.selection[myCounter].constructor.name){
20                     case "GraphicLine":
21                     case "Oval":
22                     case "Polygon":
23                     case "Rectangle":
24                     case "TextFrame":
25                         myObjectList.push(app.selection[myCounter]);
26                         break;
27                 }
28             }
29         }
30         if(myObjectList.length !=0){
31             myDisplayDialog(myObjectList);
32         }
33     }
34 }
35 function myDisplayDialog(myObjectList){
36     var myLabelWidth = 90;
37     var myFrameTypes = ["Unassigned", "Text", "Graphic"];
38     var myDialog = app.dialogs.add({name:"MakeGrid"});
39     with(myDialog.dialogColumns.add()){
40         with(dialogRows.add()){
41             with(dialogColumns.add()){
42                 staticTexts.add({staticLabel:"Rows:", minWidth:myLabelWidth});
43                 staticTexts.add({staticLabel:"Columns:", minWidth:myLabelWidth});
44             }
45             with(dialogColumns.add()){
46                 var myNumberOfRowsField = integerEditboxes.add({editValue:2});
47                 var myNumberOfColumnsField = integerEditboxes.add({editValue:2});
48             }
49         }
50         with(dialogRows.add()){
51             with(dialogColumns.add()){
52                 staticTexts.add({staticLabel:"Row Gutter:", minWidth:myLabelWidth});
53             }
54             with(dialogColumns.add()){
55                 var myRowGutterField = measurementEditboxes.add({editValue:12, editUnits:MeasurementUnits.points});
56             }
57         }
58     }
59 }
```

how to write a script

- IDE
 - writing, executing and testing of scripts
 - syntax highlighting
 - console
- Sublime Text, ExtendScript Toolkit, Visual Studio Code
- scripting language JavaScript ExtendScript
- basil.js

basil.js

The screenshot shows the homepage of the basil.js website. On the left, there's a navigation menu with links for about, download, reference, tutorials, support, and gallery. Below the menu is a tweet from RT @philippsschmitt: "New project! Computed Curation is a computer-generated photobook using machine learning and computer vision tools https...". The tweet is timestamped "Wed Jul 19 15:57:00 +0000 2017". At the bottom left is the handle "@basil_js". On the right side, there's a large, artistic visualization of a Shakespearean scene from Romeo and Juliet. The scene features red lines connecting various text elements like "ENVOLIO", "MERCUTIO", "Where is my page?", "Exit Page", and "ROMEO". The title "Romeo and Juliet" and author "Patrick Baumann, Inken Zierenberg" are displayed above the visualization, which includes a small navigation bar with dots.

Romeo and Juliet
Patrick Baumann, Inken Zierenberg

Have you ever wanted to ...

- export 10,000 mass-customized copies of your InDesign document?
- use spatial-tiling algorithms to create your layouts?
- pass real-time data from any source directly into your InDesign project?
- create color palettes based on algorithms?
- or simply reconsider what print can be?

basil.js is ...

- making scripting in InDesign available to designers and artists.

basil.js

- **making InDesign scripting available to designers and artists**
- **in the spirit of Processing and easy to learn**
- **based on JavaScript and extends the existing API of InDesign**
- **a project by The Basel School of Design**
- **has been released under an open source license**

basil.js

- **scripting library for InDesign**
- **simplifies JavaScript commands**
- **additional functions (undo functionality, execution duration)**
- **better user feedback (e.g. error messages)**

```
var size = 200;
var page = app.activeWindow.activePage;

var redColor = app.activeDocument.colors.add({
    space: ColorSpace.RGB,
    colorValue: [255, 0, 0]
});

var ellipse = page.ovals.add({
    geometricBounds: [
        page.bounds[2] / 2 - size / 2,
        page.bounds[3] / 2 - size / 2,
        page.bounds[2] / 2 + size / 2,
        page.bounds[3] / 2 + size / 2
    ],
    fillColor: redColor
});
```

```
fill(255, 0, 0);
ellipse(width / 2, height / 2, 200, 200);
```

let's code!

- **installation and setup**
- **run a basil.js script**
- **shapes**
- **variables**
- **randomness**
- **repetition**
- **typography**
- **modify MuTypography**

coding tips

- clean up scripts and comment them (!)
- adjust script to its intended use and target audience
- make use of built-in InDesign functions whenever possible
- to come up with an algorithm, think about how you would achieve the same steps via InDesign's UI
- D.R.Y. (Don't Repeat Yourself)
- combine code snippets from different sources
- re-use code snippets of past projects

resources basil.js

- **basil.js reference**
- **Typografie Programmieren GitHub page**
- **example scripts**

resources InDesign scripting

- **InDesign ExtendScript API**
- **InDesign automatisieren**
- **HilfDirSelbst.ch Adobe InDesign Skriptwerkstatt**
- **InDesign on Adobe Support Community**



resources JavaScript

- Coding Train
- Stack Overflow
- Codecademy – Introduction to JavaScript