

Greetings from



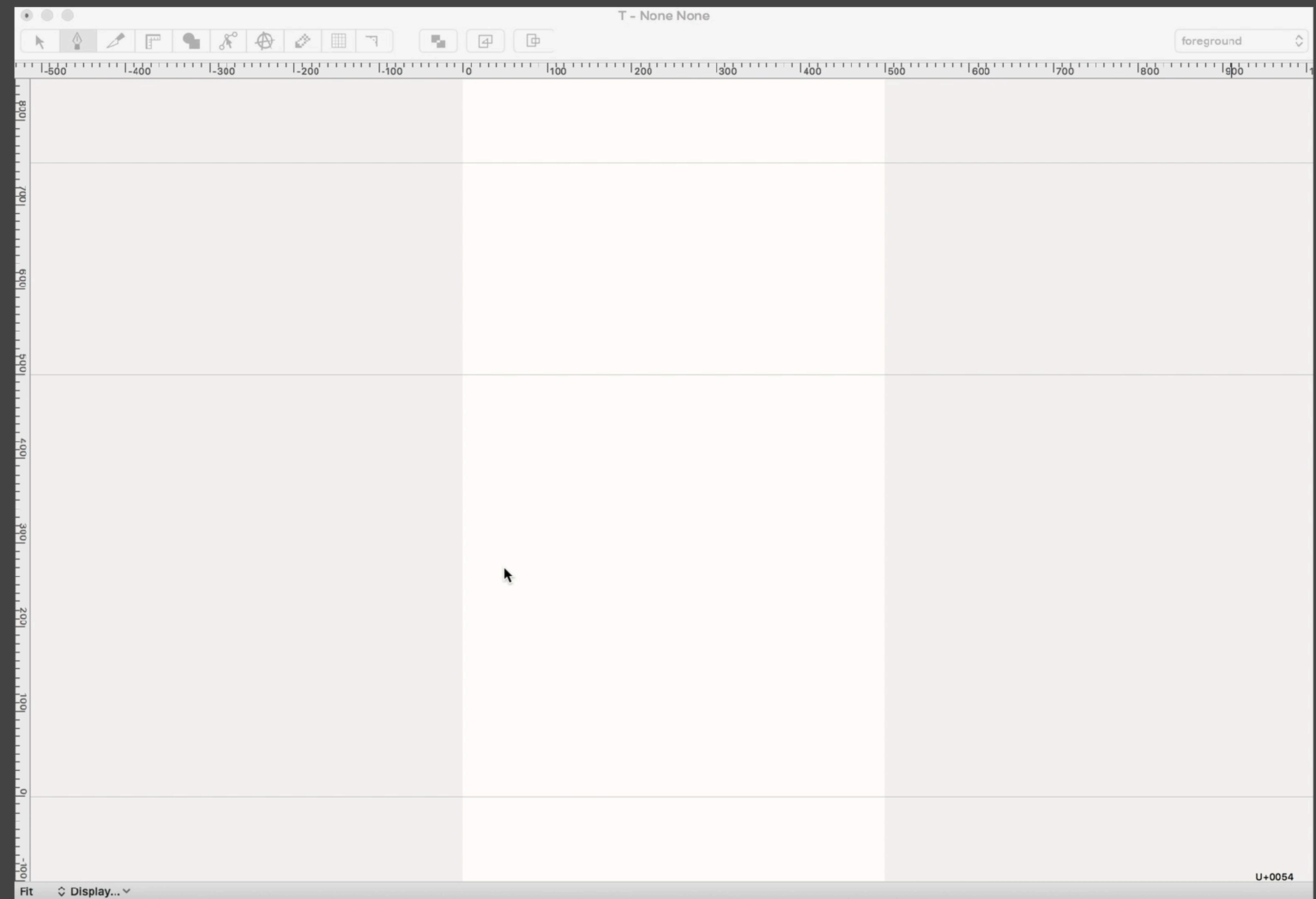
16199

B-3



Conway, MA

Absolutely
everything I know
about Python is
because of fonts.



The screenshot shows a web browser window for the URL <https://djr.com>. The page features a black header bar with the text "DAVID JONATHAN ROSS" on the left and navigation links "TYPEFACES", "CLUB", "NOTES", and "ABOUT" on the right. A white sidebar on the left contains the "Font of the Month Club" logo, which consists of the words "FONT", "OF THE", "MONTH", and "CLUB" stacked vertically in bold, black, sans-serif font, with horizontal red bars separating the words. The main content area has a blue grid background. On the right side, there is a promotional message: "Join my Font of the Month Club to get a fresh new font in your inbox each and every month. Starting at only \$6/month!". Below this, a large, dark blue section titled "GIMLET X-RAY > Discover the inner-workings of a variable font" displays a complex diagram. This diagram is a "Gimlet X-Ray" visualization of a variable font's internal structure. It shows multiple vertical columns of characters, likely 'G' and 'I', with their outlines and internal points (nodes) highlighted in purple against a dark background. The nodes are small circles connected by lines, forming the intricate paths of the letters.

The screenshot shows a web browser window for the URL fontofthemonth.club/. The page features a large, bold, black title "FONT OF THE MONTH CLUB" centered on the left side. Below this, a red horizontal bar contains the text "THIS MONTH'S FONT". To the right, there is a section titled "JOIN THE CLUB!" with descriptive text about the club's offerings and a "START YOUR MEMBERSHIP TODAY!" button. A gift icon and the text "Or give a gift subscription to your favorite font lover!" are also present. At the bottom, a preview of the font "Klooster Thin" is shown with the tagline "Taking the uncial in new directions", along with a "JOIN TODAY! \$6" button and a "MORE" link. The top navigation bar includes links for "DAVID JONATHAN ROSS", "TYPEFACES +", "CLUB +", "ABOUT", "NOTES", "Sign In", and "Cart".

DAVID JONATHAN ROSS

TYPEFACES + CLUB + ABOUT NOTES Sign In Cart

FONT OF THE MONTH CLUB

THIS MONTH'S FONT

Klooster Thin
Taking the uncial in new directions

JOIN TODAY! \$6 MORE

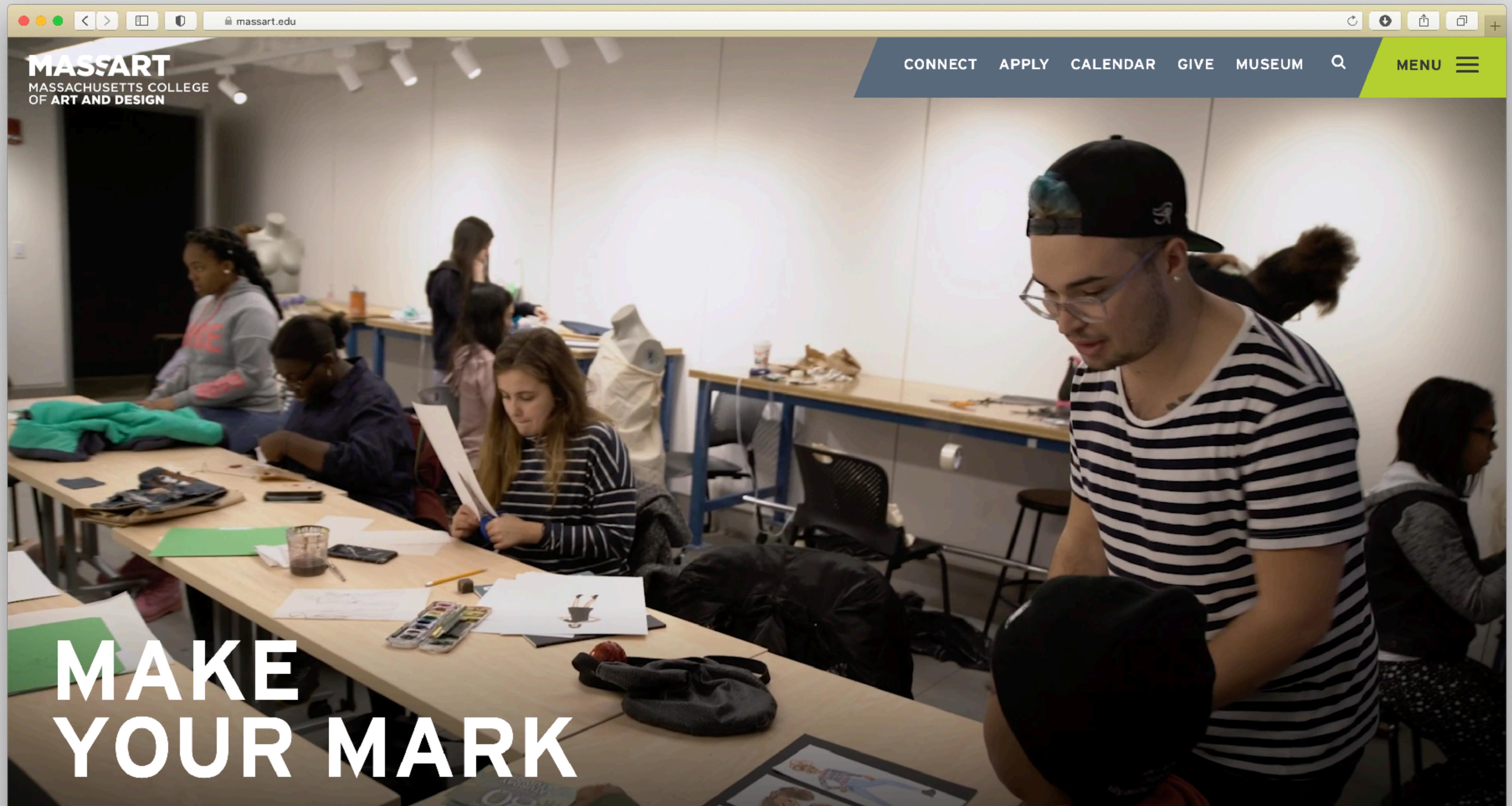
JOIN THE CLUB!

Font of the Month Club sends you a fresh new font every single month! Fonts of the month include distinctive display faces, experimental designs, and exclusive previews of my upcoming retail typeface families. Each font is lovingly designed and produced by me, David Jonathan Ross.

By diversifying your font collection at a minimal cost, the club can push you to try new and interesting type in your work.

START YOUR MEMBERSHIP TODAY!

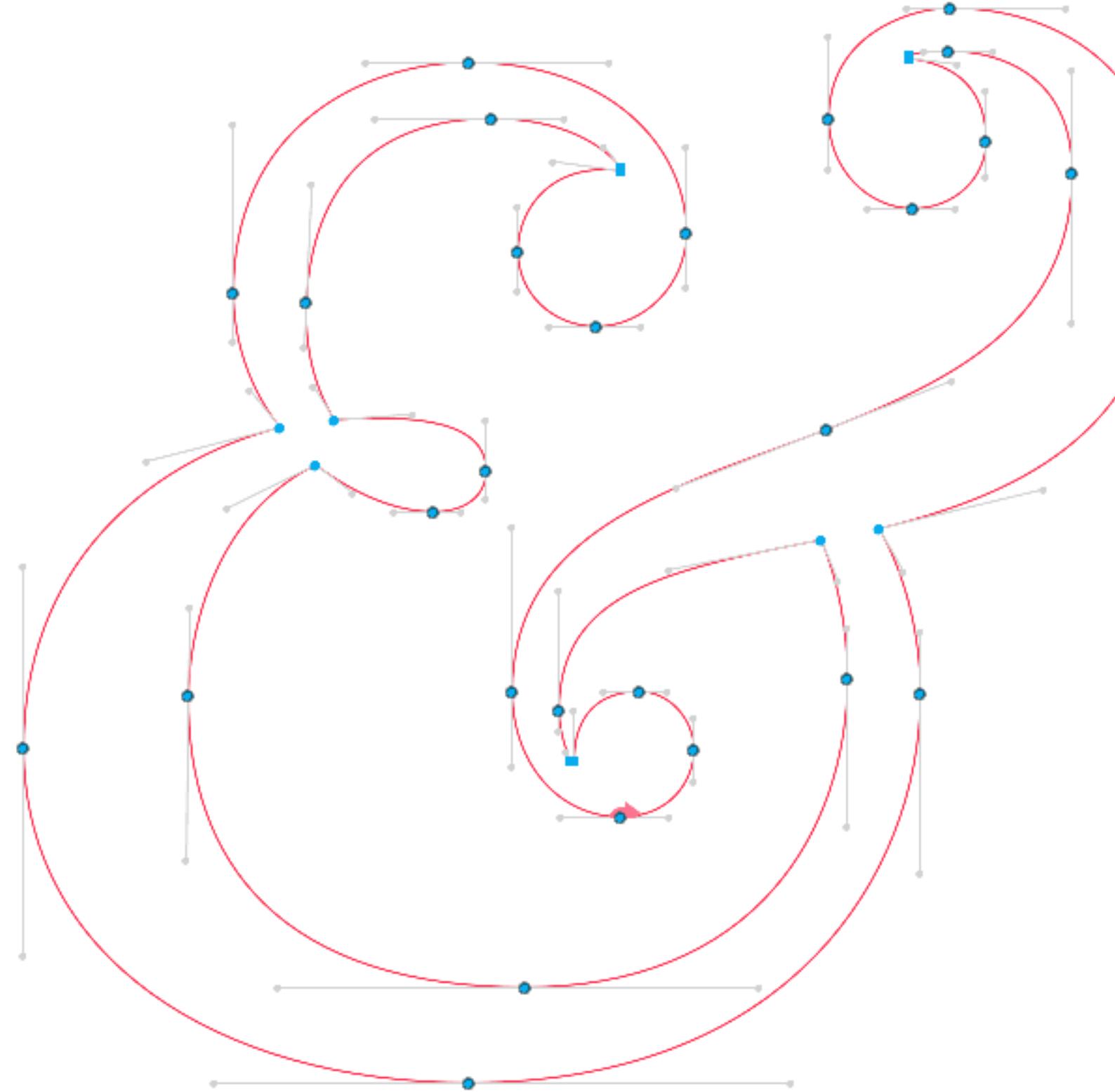
Or give a gift subscription to your favorite font lover!



The screenshot shows a web browser window for coopertype.org. The title bar reads "Not Secure — coopertype.org" and "Type@Cooper – Typeface Design certificate program from Cooper Union". The page content includes the text "Typeface Design certificate program from Cooper Union" and a navigation menu with links to Home, Extended Program, Condensed Program, Lecture Series, Public Workshops, Instructors, and FAQ. To the left of the main text is a graphic of several red circles with blue dots and connecting lines, resembling a complex typeface outline or a network diagram. To the right is the large, stylized text "Type@Cooper" where the '@' symbol is in blue. Below this, a blue text box contains the sentence: "The Continuing Education Department of The Cooper Union offers a Postgraduate Certificate in Typeface Design." To the right of this text is a small logo for "THE COOPER UNION" featuring a colorful geometric cube-like shape.

Typeface Design certificate program from Cooper Union

Home Extended Program Condensed Program Lecture Series Public Workshops Instructors FAQ



Type@ Cooper

The Continuing Education Department of The Cooper Union offers a Postgraduate Certificate in Typeface Design.

Top industry professionals lead a highly focused and comprehensive study of key typeface design principles: technique, technology, aesthetics, expression, history, and theory. Students explore the foundation of typography in depth by creating their own typefaces in hands-on classes, while developing a broad understanding of the field through lectures, discussions, and research. Electives are offered each term, focusing on topics such as pen and brush lettering, Python programming, and advanced tools for font development. A series of guest lectures round out the curriculum, allowing students deeper insight into specific relevant topics.

Participants leave the program with the specialized skills to design professional-quality digital typefaces and lettering.



GOUPARÉ

PEABAN

GYMPE

BUCKEYE MEADOW

CRESCENT HEIGHTS

WEST BURAVISTA

GATEWAY TOWERS

PASEO DEL PUERTO

WEST SILVER LAKE

A HISTORY OF THE HOLY ROMAN EMPIRE

JRIER

l Modèle
national

rant les années 1970,
pe l'informatique dist
is en plus réclamée pa
stèmes 32, 34, 36, 810
même temps IBM pas
-informatique. Après
machines de succès dive

adém
çaises

mission qui lui est assig
agine, et qui sera précis
r 1635 par lettres paten
l, est de fixer la langue
donner des règles, de le
compréhensible par tou

Na celém světě se vyrá
stovky typů sýra. Exist
více různých systémů dě
sýrů. Různé druhy a příc
sýrů jsou výsledkem pou
mléka různých savců nel

Roslindale

=365=

Vegetarian Barbecue Recipes

CALL ME ISHMAEL. Some years ago – never mind how long precisely – having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen and regulating the circulation. Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I meet; and especially whenever my hypos get such an upper hand of me, that it requires a strong moral principle to prevent me from deliberately stepping into the street, and methodically knocking people's hats off, then, I account it high time to get to sea as soon as I can.

This is my substitute for pistol and ball. With a philosophical flourish Cato throws himself upon his sword; I quietly take to the ship. There is nothing surprising in this. If they but knew it, almost all men in their degree, some time or other, cherish very nearly the same feelings towards the ocean with me. There now is your insular city of the Manhattoes, belted round by wharves as Indian isles by coral reefs, commerce surrounds it with her surf.

NO, WHEN I GO TO SEA, I go mast, plumb down into the fore-c head. True, they rather order me a spar to spar, like a grasshopper in sort of thing is unpleasant enough particularly if you come of an old Van Rensselaers, or Randolphs, or if just previous to putting your hand lording it as a country schoolmaster awe of you. The transition is a keen master to a sailor, and requires a Stoics to enable you to grin and bear

What of it, if some old hunk broom and sweep down the decks? weighed, I mean, in the scales of the archangel Gabriel thinks anything and respectfully obey that old hun ain't a slave? Tell me that. Well, may order me about – however they have the satisfaction of knowing th

```
#coding=utf-8
"""
SHOW MOUSE COORDINATES
"""

from vanilla import *
from defconAppKit.windows.baseWindow import BaseWindowController
from mojo.events import addObserver, removeObserver

class ShowMouseCoordinatesTextBox(TextBox):
    def __init__(self, *args, **kwargs):
        super(ShowMouseCoordinatesTextBox, self).__init__(*args, **kwargs)
        addObserver(self, "mouseDragged", "mouseDragged")
        addObserver(self, "mouseUp", "mouseUp")

    def mouseDragged(self, info):
        point = info["point"]
        positionSymbol = unichr(8982)
        deltaPoint = info["delta"]
        angle = math.degrees(math.atan2(deltaPoint.y, deltaPoint.x))
        distance = math.hypot(deltaPoint.x, deltaPoint.y)
```

Input
input.fontbureau.com

free download!

RoboFont

Frederik Berlaen

Turnip-Black.ufo

beta (123 days left)

Views

All Glyphs 769

Font Features Font Info Space Center Kerning Groups Sort Mark Edit With...

Selection to Set Save Set +

Glyph Inspector - colon

Name: colon

Unicode: 003A

Width: 337

Left: 58 Right: 61

Mark: Exclude during export

Colon Sidebar

colon

58 → 61

= → = ← =

→ ← 2 ← →

Increment

Preview

Layers

Layer Name Color

a - Turnip Black

Font Glyph Python Extensions Scripts Window Help

Segments

- Alpha...tin UC 124
- Alpha Latin LC 127
- Alpha Latin SC 124
- Sorts MC 48
- Sorts UC 16/22
- Sorts SC 12/41
- Figures MC 34
- Figures SC 7/34
- Figures LC 28/34
- Figur...C Tab 37/45
- Figur...th MC 21/22
- Diacritics MC 16
- Diacritics UC 12
- Diacritics SC 0/12
- Figure...ps/sinf 20
- Figures frac 22
- Spaces 5
- Alpha supers 27
- Ligatures 15
- Discre...tatures 1/4

Dev

Codepages

Unicod...anges

- Basic Latin 95
- Latin-1...lement 95
- Latin...nded-A 125
- Latin E...nded-B 6
- IPA Extensions 1
- Phonet...ensions 0
- Phonet...lement 0
- Spacin...Letters 8
- Modifi...Letters 0
- Combi...al Marks 2
- Combi...plement 0
- Greek...d Coptic 4
- Coptic 0
- Cyrillic 0
- Cyrilic...plement 0

Colon (1 selected)

Fit Display...

The screenshot shows the RoboFont application interface. The top menu bar includes RoboFont, File, Edit, Font, Glyph, Python, Extensions, Scripts, Window, Help, and a status bar indicating 'beta (123 days left)'. The main workspace features a grid-based 'All Glyphs' panel on the left containing numerous characters and symbols, many of which are bold black versions of standard characters. A specific character, the colon (colon), is selected and highlighted with a blue border. To the right of the grid is an 'Inspector' panel titled 'Inspector - colon' showing details for the selected character: Name: colon, Unicode: 003A, Width: 337, Left: 58, Right: 61, and a 'Mark' field with a red line icon. Below the inspector are 'Sidebar' controls for the colon character, including a sequence of numbers and arrows (58 → 61, = → = ← =, → ← 2 ← →) and an 'Increment' button. Further down are 'Preview' and 'Layers' sections. In the bottom right corner, there is a detailed view of the 'colon' glyph's structure, showing its internal strokes and nodes, with a color-coded stroke order diagram overlaid on the outline.

MERIT BADGE

IS A VARIABLE
COLOR FONT.



Just van Rossum



Frederik Berlaen

drawBot



drawbot.com

```
# Clip
clipPath(iconPath)
# Move to the center of the canvas
translate(256, 256)
circleCount = 30
for i in range(circleCount):
    f = i/circleCount
    angle = (f * 360) + (360 * timeFactor)
    x = 120 * sin(radians(angle+90))
    y = 120 * cos(radians(angle+90))
    colorFactor = * f * f + 0.1 # An exponential curve for the color factor
    stroke(None)
    fill(*interpolateColor(colorFactor))
    #shadow((0, 0), 50, interpolateColor(colorFactor)) # Extra smoothness?
    oval(x-150, y-150, 300, 300)
    restore()
```

code

console

**So what is
drawbot?**

DrawBot is a powerful,
free application for
MacOS that invites you
to write simple Python
scripts to generate two-
dimensional graphics.

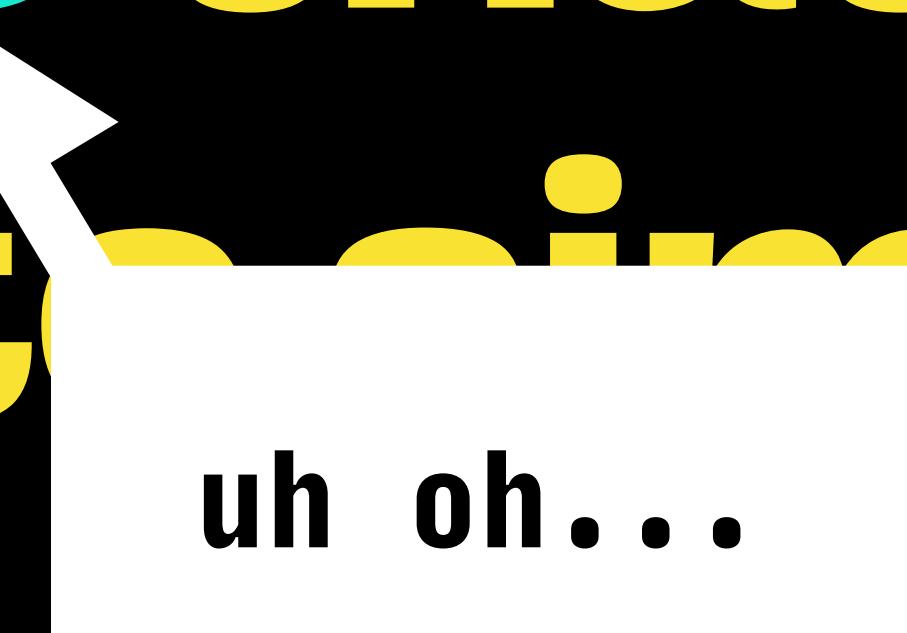
**DrawBot is a powerful,
free application for
MacOS that invites you
to write simple Python
scripts to generate two-
dimensional graphics.**



**DrawBot is a powerful,
free application for
MacOS that invites you
to write simple Python
scripts to generate two-
dimensional graphics.**

IDE FTW

**DrawBot is a powerful,
free application for
MacOS that invites you
to write simple Python
scripts to generate two-
dimensional graphics.**

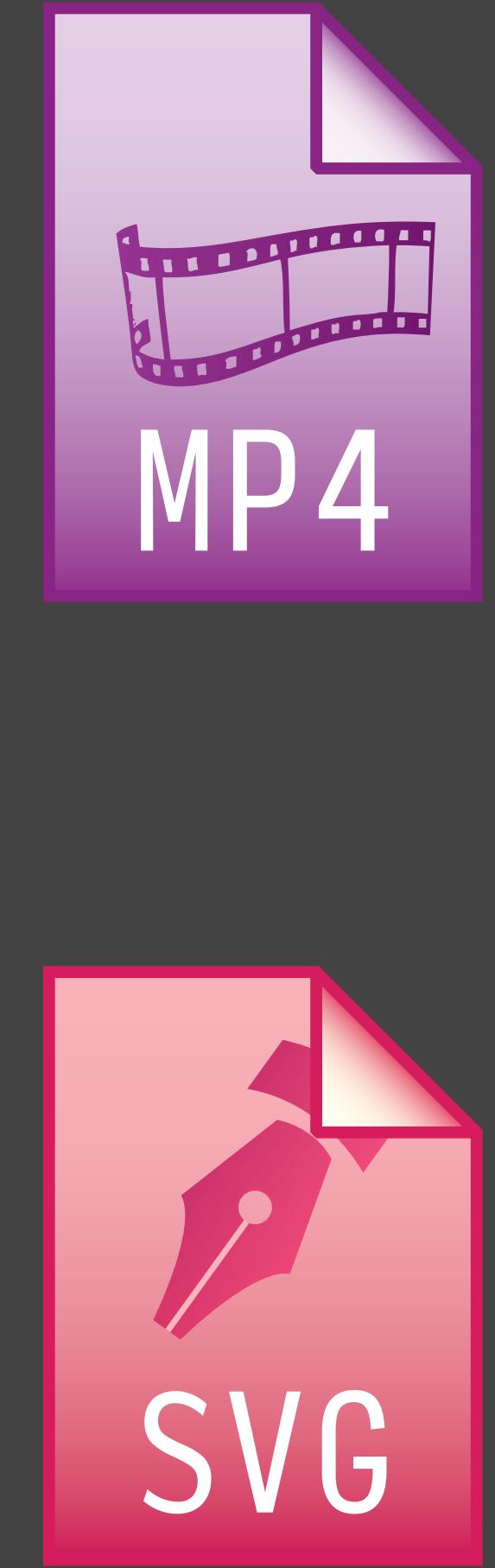
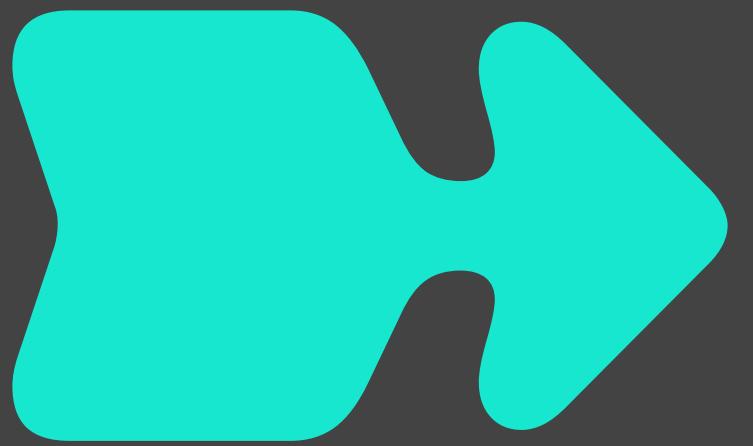


uh oh...

**DrawBot is a powerful,
free application for
MacOS that**  **you
to write simple Python
scripts to generate two-
dimensional graphics.**

**DrawBot is a powerful,
free application for
MacOS that invites you
to write simple Python
shape, text, image
statements to generate two-
dimensional graphics.**

**what can
it make?**



pyPDF reportlab pdfkit

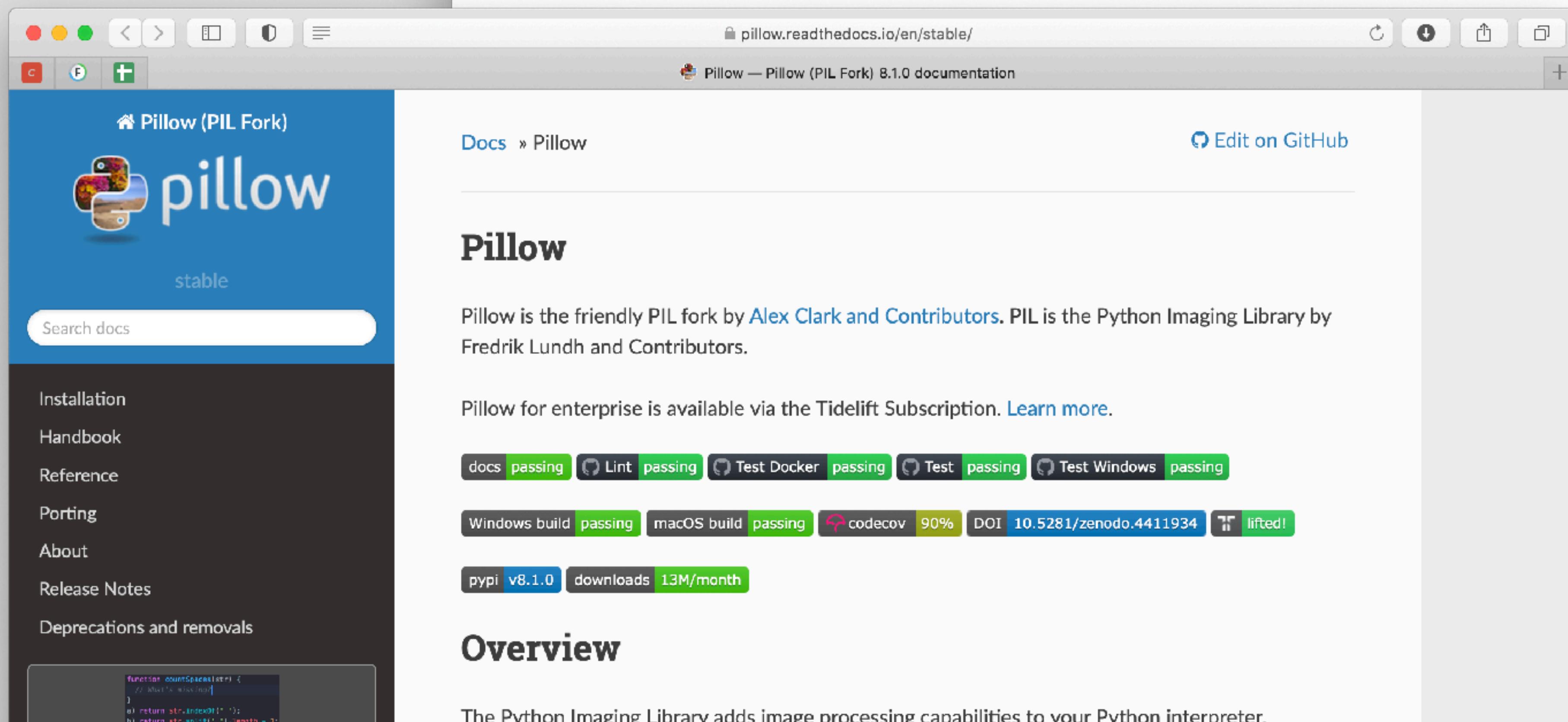
The screenshot shows the PyPI project page for reportlab 3.5.59. The title "reportlab 3.5.59" is prominently displayed at the top. Below it is a "pip install reportlab" button. The main content area includes sections for "Navigation" (Project description, Release history, Download files), "Project links" (Homepage, Download), and "Statistics". A sidebar on the left lists "Project links" (Homepage) and "Statistics".

The screenshot shows two PyPI project pages. The first is for pyPdf 1.13, which is described as a "PDF toolkit". It features a "pip install pyPdf" button and a "Latest version" button. The second is for pdfkit, which is described as a "Pure-Python library built as a PDF toolkit". It lists various capabilities such as extracting document information, splitting documents, merging documents, cropping pages, merging multiple pages, and encrypting/decrypting PDF files. Below the project description is a section for "Project links" (Homepage, Download). To the right of these pages is a GitHub repository page for "JazzCore/python-pdfkit". The repository has 83 issues, 19 pull requests, and 35 actions. It shows a list of commits, including one from alanhamlett and another from fbataill. The repository has 1 branch and 0 tags.

pycairo PIL/pillow



The screenshot shows a web browser window with the URL www.cairographics.org/pycairo/. The page features the Cairo logo (two orange beetles carrying a grid) at the top. Below it is a navigation bar with links for News, Download, Documentation, Contact, and Examples. The main content area is titled "Pycairo" in orange. It states: "Pycairo is a set of Python 2 & 3 bindings for the [cairo graphics library](#). Since version 1.11.0 pycairo has moved to GitHub and pycairo and py2cairo have been merged back into one project."



The screenshot shows a web browser window with the URL pillow.readthedocs.io/en/stable/. The page title is "Pillow — Pillow (PIL Fork) 8.1.0 documentation". On the left, there's a sidebar with links for Installation, Handbook, Reference, Porting, About, Release Notes, and Deprecations and removals. The main content area starts with a "Pillow" section, followed by a paragraph about Pillow being a friendly PIL fork. It also mentions Tidelift Subscription, build status (Windows, macOS, Docker, Test, Windows), DOI (10.5281/zenodo.4411934), and package details (pypi v8.1.0, downloads 13M/month). The bottom of the page includes footer links for bindings, cairo-ocaml, cookbook, and last edit information.

Processing p5.js

The screenshot shows the Processing 3.4 IDE interface. The title bar reads "sketch_181215a | Processing 3.4". The menu bar includes File, Edit, Sketch, Debug, Tools, and Help. The top toolbar features a play button, a stop button, a code count indicator (88), and a "Java" dropdown. The main workspace is titled "sketch_181215a" and contains a vertical list of line numbers from 1 to 19. Below the workspace is a toolbar with left and right arrows. At the bottom, there are tabs for "Console" and "Errors".

P Processing/p5.js

- Cross-platform
- 2D+3D (WebGL)
- Interactive
- Big community
- Web integration
- Pixel-based
- Basic typography

D DrawBot

- MacOS
- 2D
- Hardly interactive
- Small community
- Print-ready output
- Vector-based
- Advanced typography

Adobe InDesign

Adobe InDesign 2020

Font: Pomfret

Text Properties: TT 108 pt, TA 76 pt, Metrics, 100%, 100%, Aa 0 pt, T 0°, English: USA

Text: BERTRAM GOODHUE AND IN PARTICULAR IS 1892 RING R OF RANT

Character Panel: Pomfret, Regular, TT 108 pt, TA 76 pt, Metrics, 100%, 100%, Aa 0 pt, T 0°, Language: English: USA

Page Number: 13 Pages in 7 Spreads

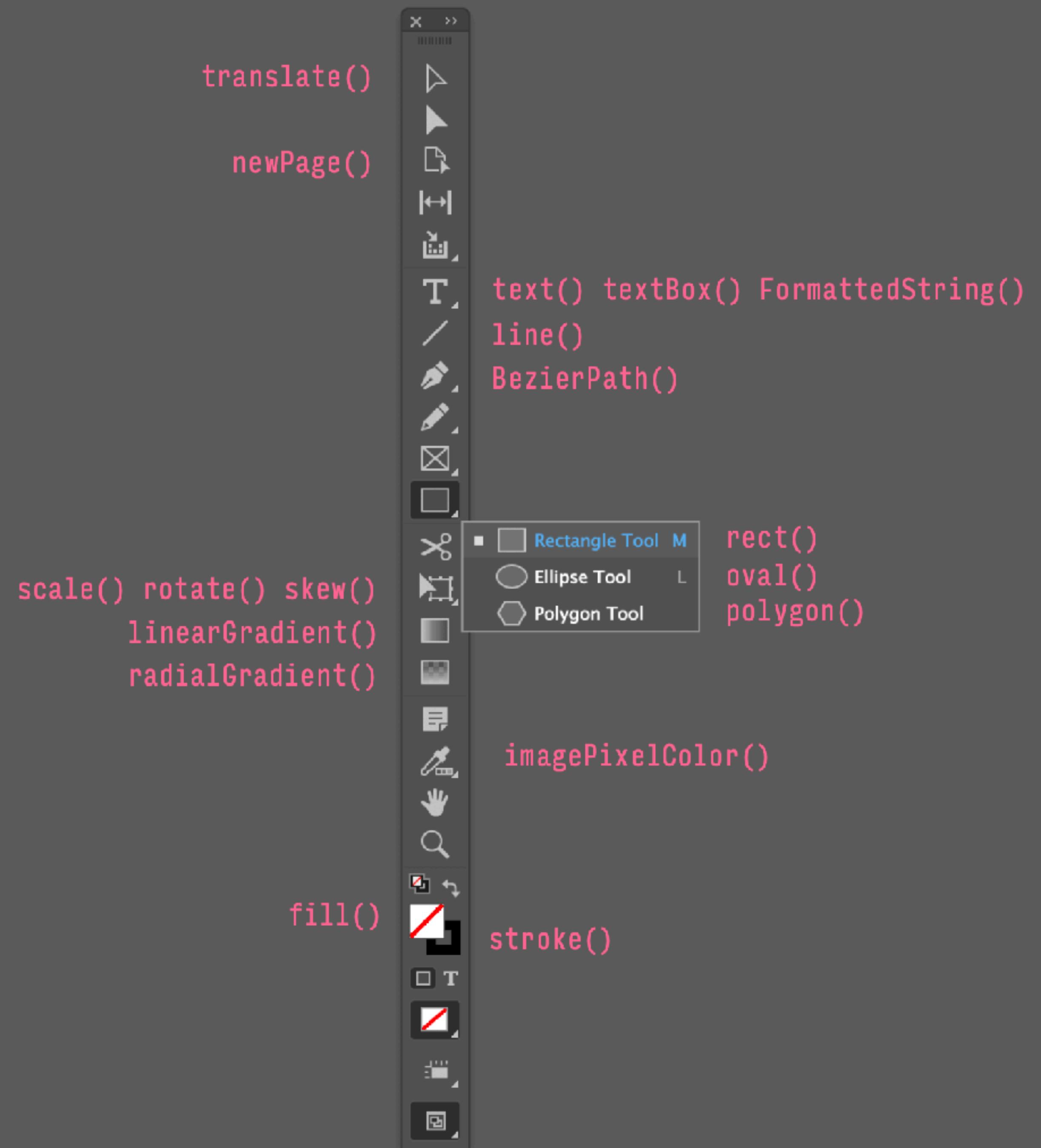
Specimen 1: BARRANCA STOPBACK

Specimen 2: EXPLAINER RECREANT

Page Number: 1 error

Page Number: 13 Pages in 7 Spreads

Page Number: 1 error



Lowlands 2018

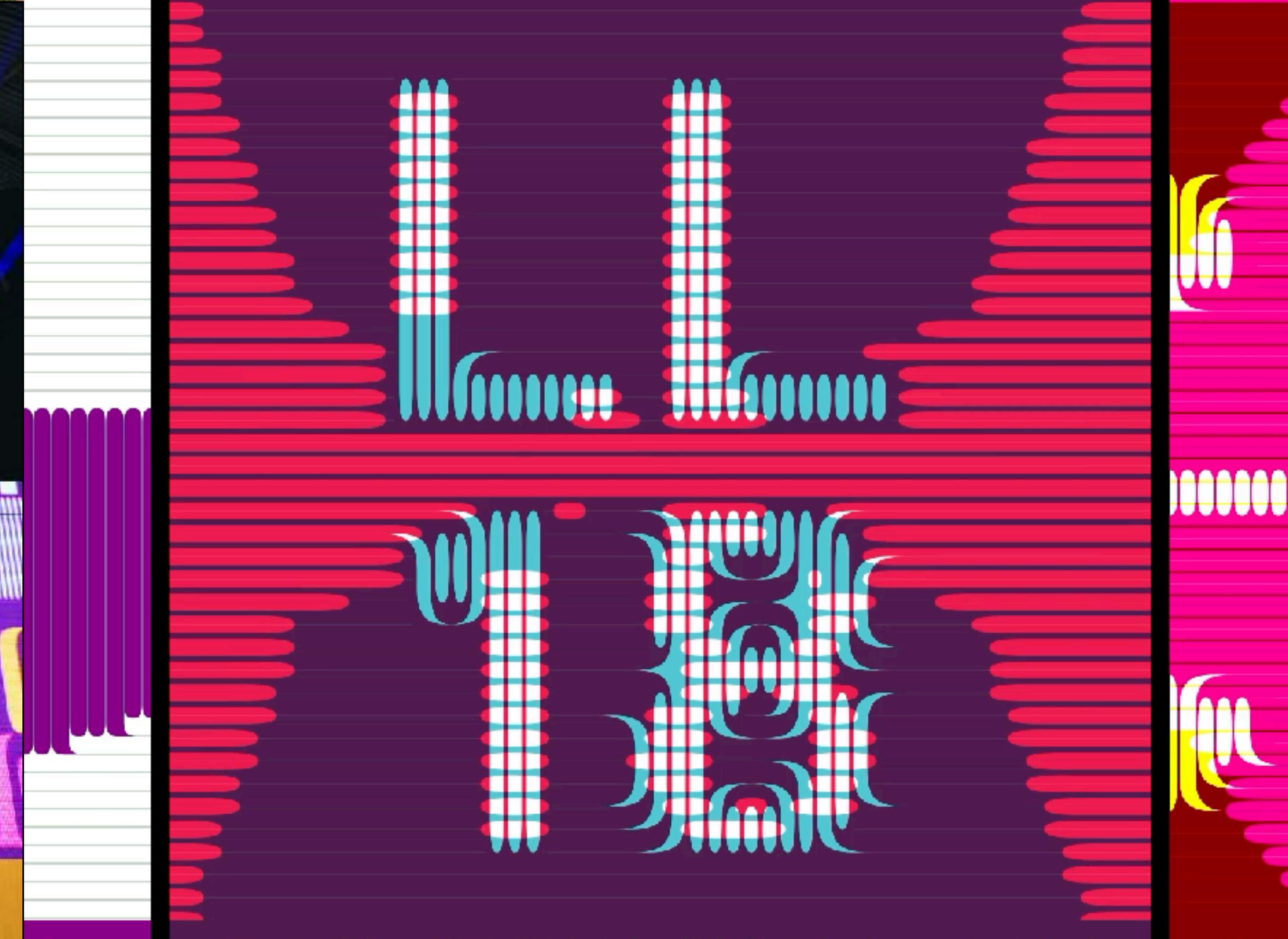
Hansje van Halem, Just van Rossum

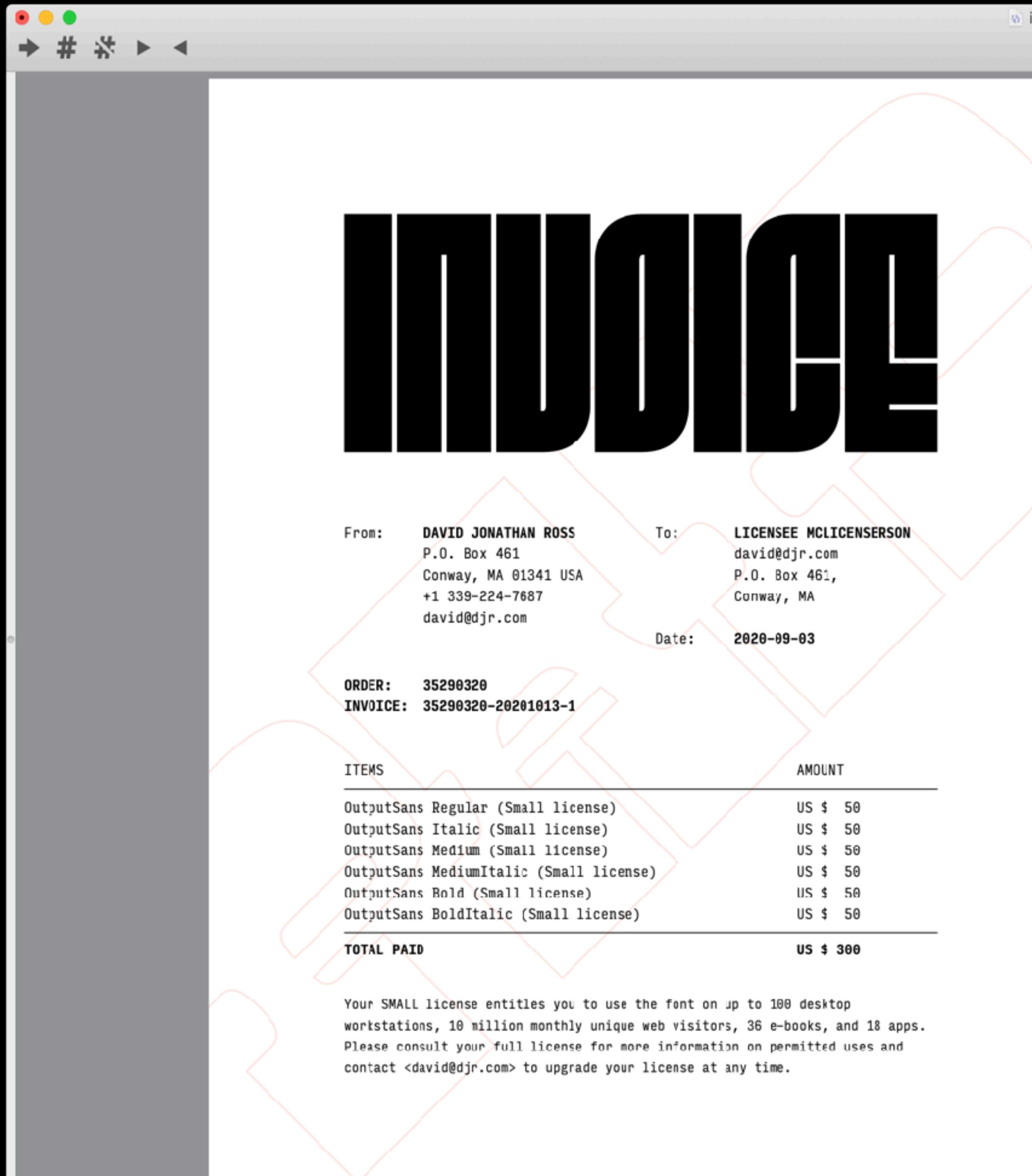


Lowlands Festival 2018

Hansje van
Halem, Just
van Rossum

hansje.net





```
Q discount
Done Replace
183     *digitDiff + price
184
185     if price == 'None':
186         productLine = ''
187         productLine += '\r'
188         productInfo.append(productLine,
189         font=textFont)
190
191
192     if data.get('discount'):
193         productInfo.append('Discount: %s\tUS $ - %s\r' %(str(data['discount']['name']),
194         str(data['discount']['amount-saved'])),
195         font=inputFont, lineHeight=textLineHeight*2)
196
197
198     if paid:
199
200         OutputSans-Regular
201         OutputSans-Italic
202         OutputSans-Medium
203         OutputSans-MediumItalic
204         OutputSans-Bold
205         OutputSans-BoldItalic
```

Typographics 2015





Typographics 2016

2016.typographics.com

HANS
JOSEPH
VAN HALM

KARRA
HAUPT

Typographics 2018





P

p Benton Sans – Condensed Bold Italic

a Shogun – Extended Black

U Rocky – Condensed Black Italic

v Anisette – Medium

S Salvo Serif – Condensed Italic

Z Rocket – Regular

2 Antenna – Condensed Italic

m Williams

Font Trading Cards

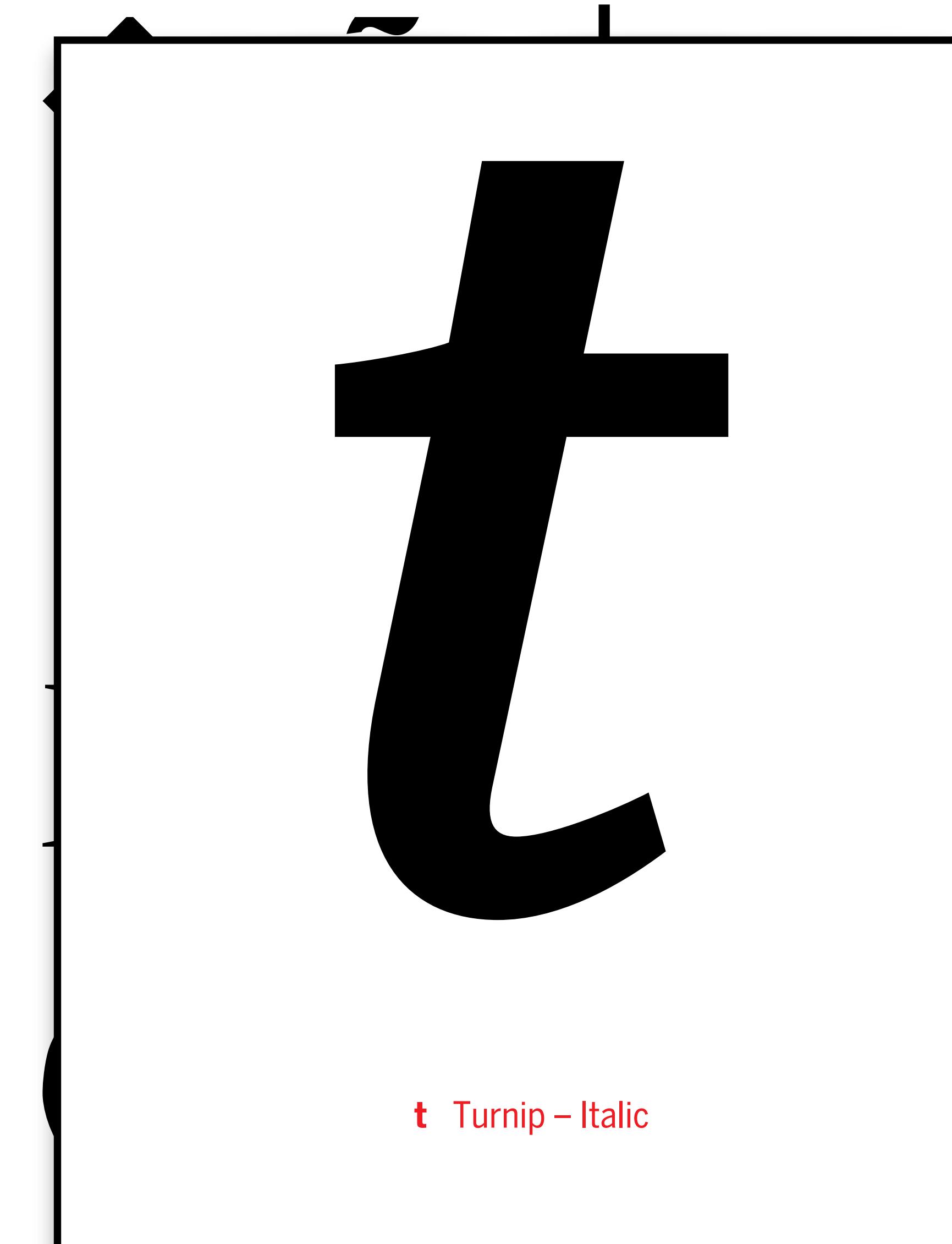
2 S

> w

H +

b c

B ě



t Turnip – Italic

LATIN SMALL LETTER T

Family: Turnip (16 styles)
Style: Italic
Unicode Value: U+0074
Unicode Block: Basic Latin
Unicode Category: Letter, Lowercase
Units Per Em: 1000
Advance Width: 361
Sidebearings: 16 / 24
Number of Contours: 1
Points: 16 oncurve / 10 offcurve

Where classic serif faces are urbane and well-mannered, Turnip is coarse and down-to-earth — the kind of type that feels completely at home with salty language. David Jonathan Ross crafted an energetic tension between Turnip's inner and outer shapes to give it vigor, while strong strokes and sturdy serifs yield dense paragraphs with a horizontal emphasis. This ain't your grandpa's book face, but it reads just as well; FB 2012

fontbureau.com
@fontbureau
randomglyph.tumblr.com
@randomglyph



Font Proofs



hh hoho oo
hha hohoao oo
hhB hohoobo oo
hhchohoco oo
hhD hohodoo
hhe hohoe oo
hhF hohofoo
hhG hohogoo
hhh hohoboo
hhI hohoi oo

hhJ hohoj oo
hhK hohokoo
hhI hoholoo
hhM hohomoo
hhN hohodoo
hhO hohoooo
hhP hohopoo
hhQ hohooqoo
hhR hohoroo
hhS hohosoo

2021-01-15 | Klooster-Thin | Spacing proof

2

Adidas N3XT L3V3L

Grand Army

Andy Clymer



grandarmy.com/projects/n3xt-l3v3l

International Center of Photography, Philadelphia

Pentagram

Andy Clymer, Yours Truly

ICP Logo Generator v 1.1

1 2

Width Height

Inches Units

Measure the full height
 Interior height only

Measure the full height of the logo, or only measure from the baseline to the x-height

1 Pixels/Points

Counter size Units

Automatic counter size

14 inch viewing distance —

Include buffer around logo

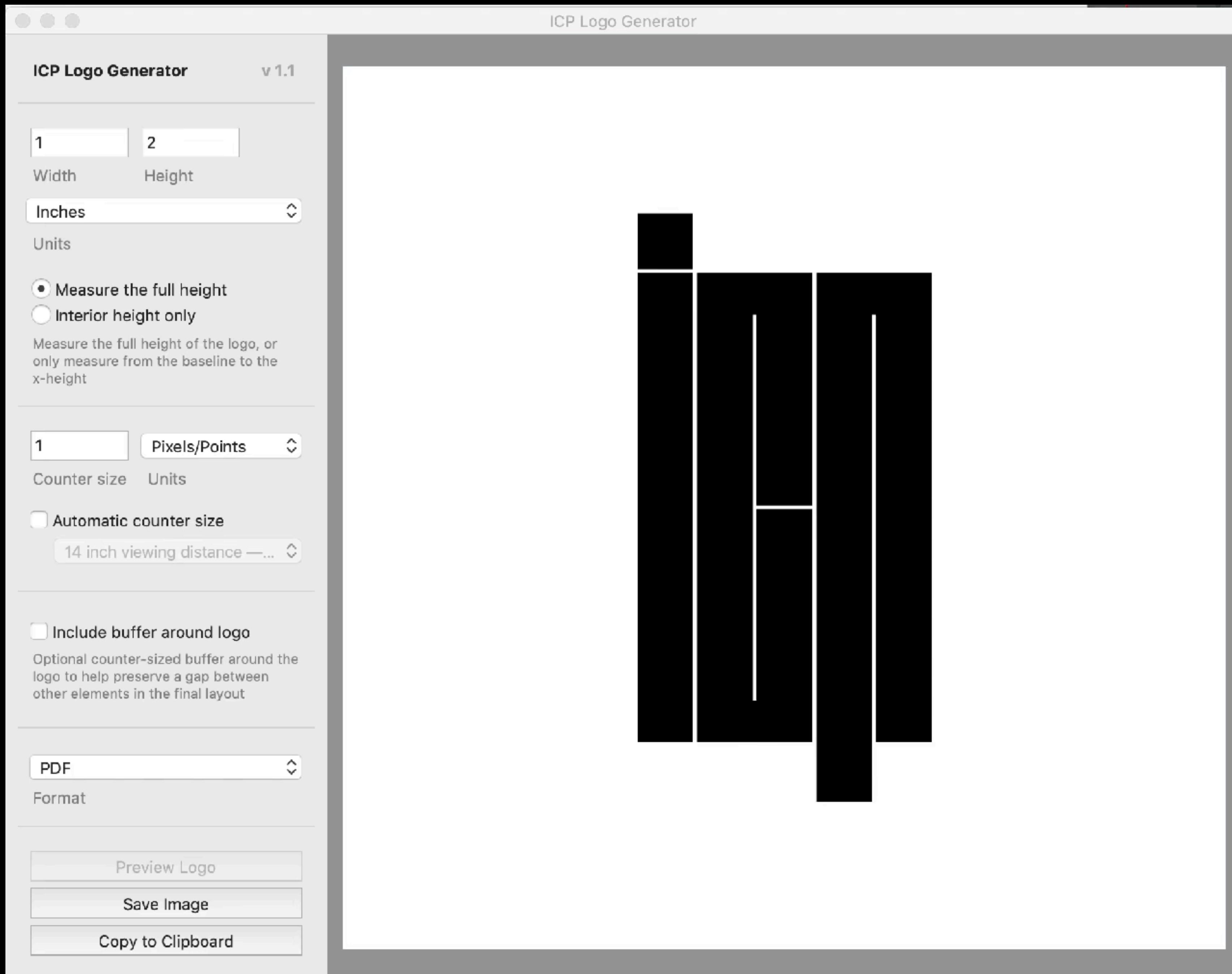
Optional counter-sized buffer around the logo to help preserve a gap between other elements in the final layout

PDF Format

Preview Logo

Save Image

Copy to Clipboard



Hobeaux Rococeaux Borders

ohnotype.co/fonts/hobeaux-rococeaux

James Edmondson



The image shows a Mac OS X desktop with a terminal window open. The terminal window has a title bar "Hobeaux-Rococeaux-Borders.py" and a status bar "dictElements". The main pane of the terminal contains a Python script demonstrating how to use the font. The script includes comments explaining the steps: 1. Including the font with the download, 2. Typing something, 3. Choosing a style (a number between 1 and 25), and 4. Setting padding. It also includes a tip about selecting numbers and dragging them to see live results.

```
# Included with the download.

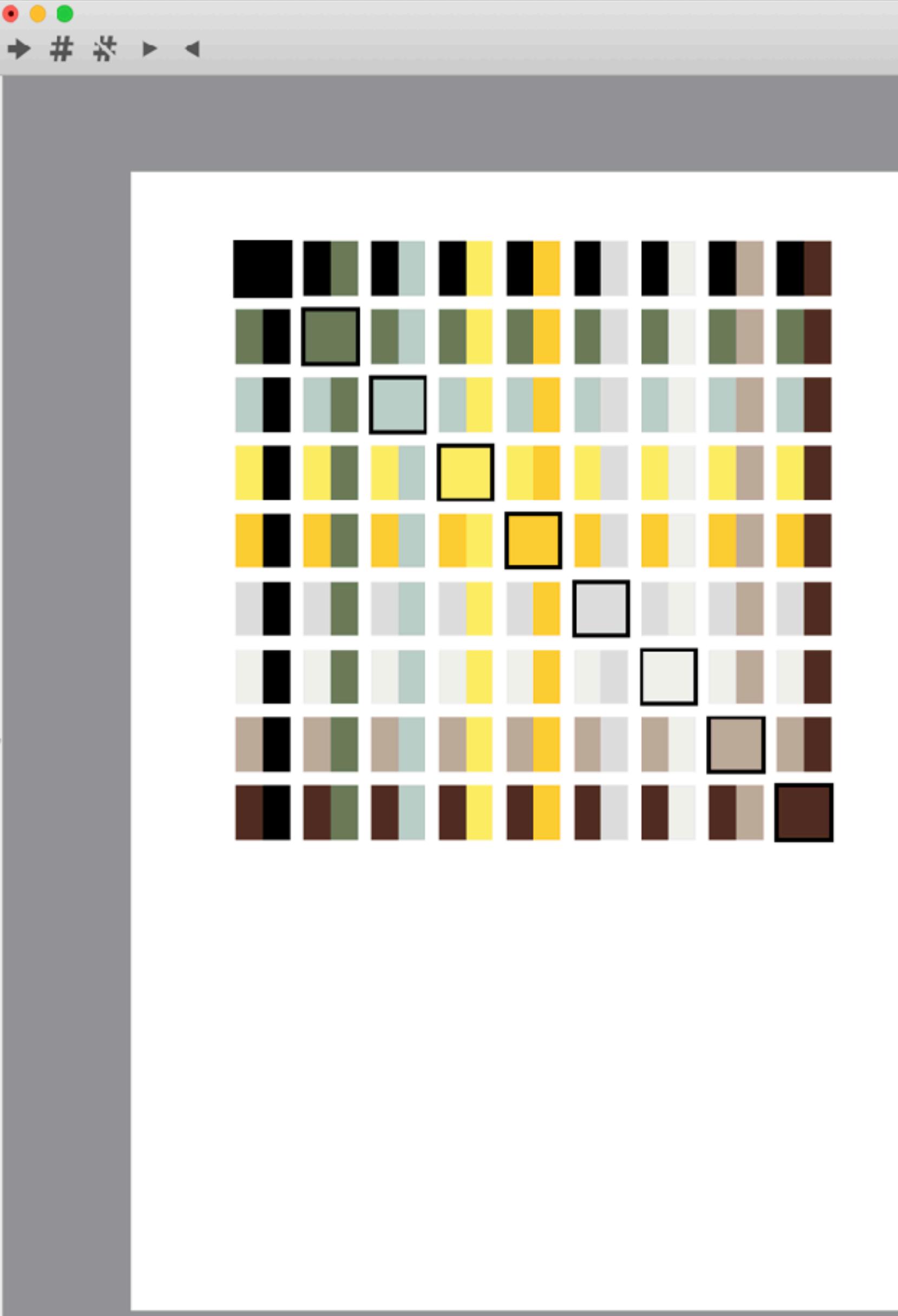
# STEP 2 - TYPE SOMETHING
# Include any desired line breaks.

txt = """Ohno
Type Co."""

# STEP 3 - CHOOSE A STYLE
# This number can be between 1 and 25. The
# lower, the less exciting.

style = 8

# STEP 4 - SET YOUR PADDING
# Adjust these numbers to get the right fit
# Tip: Select both numbers, hold command, and
# click and drag the numbers to see live
```



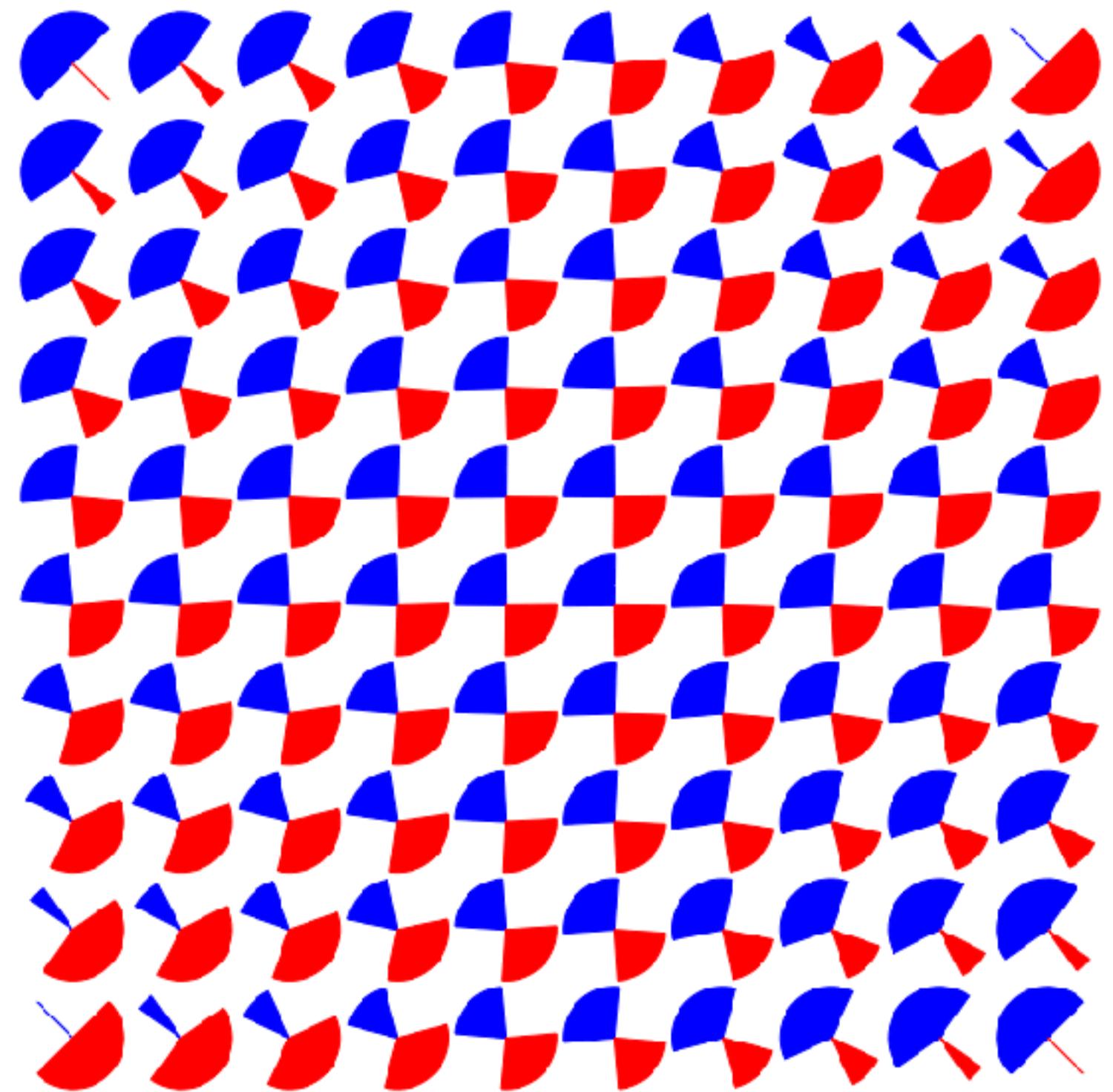
```
colorSpace(colorspace)
translate(startX,startY)

for c1 in colors:
    r1, g1, b1 = c1
    for c2 in colors:
        translate(spaceX,0)
        if c1 != c2:
            r2, g2, b2 = c2
            stroke(None)
            fill(r1,g1,b1)
            rect(0, 0, swatchWidth/2, swatchHeight)
            fill(r2,g2,b2)
            rect(swatchWidth/2, 0, swatchWidth/2, swatchHeight)
        else:
            stroke(0)
            strokeWidth(3)
```

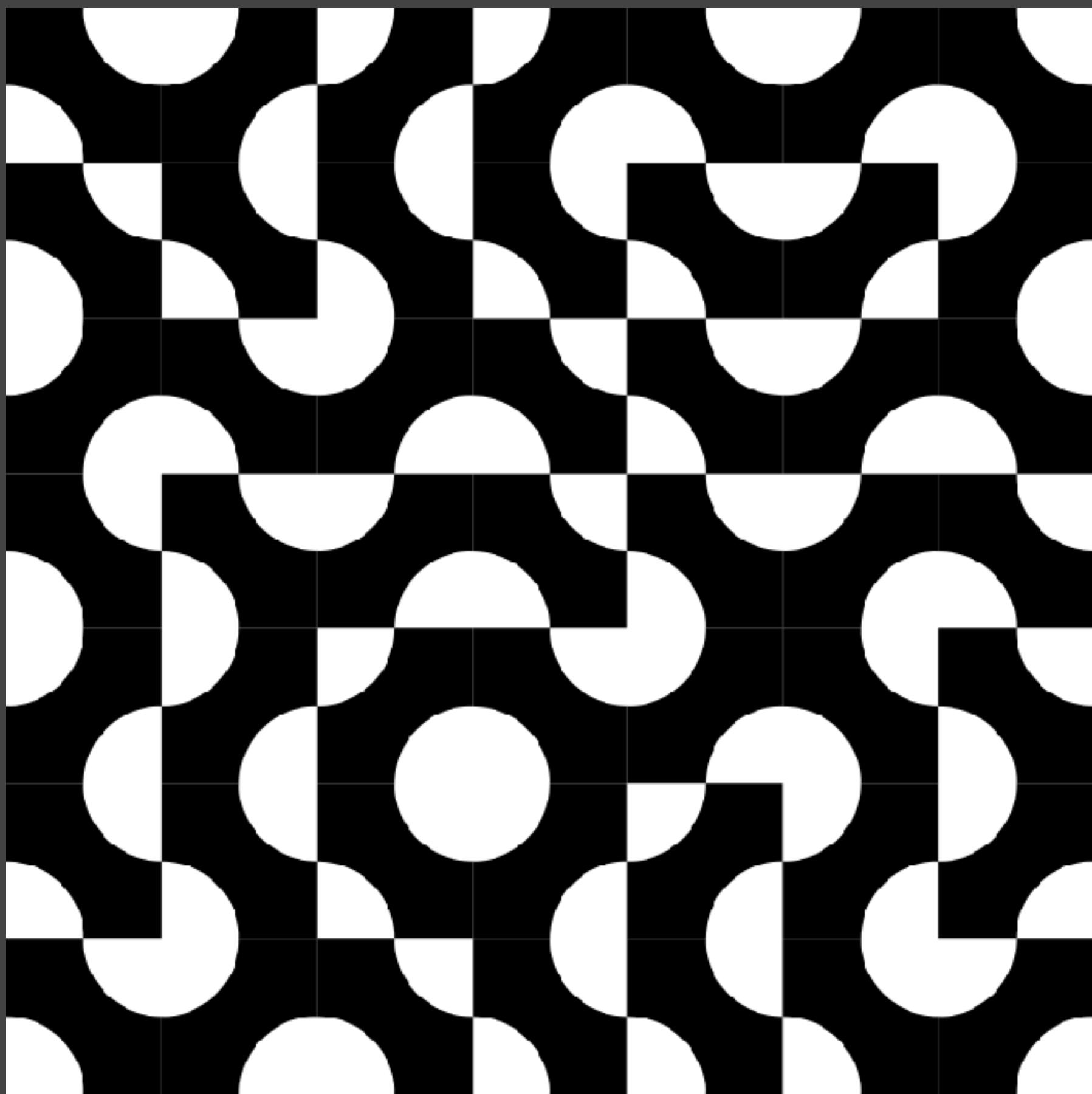
Drawbot Animations

Maurice Meilleur

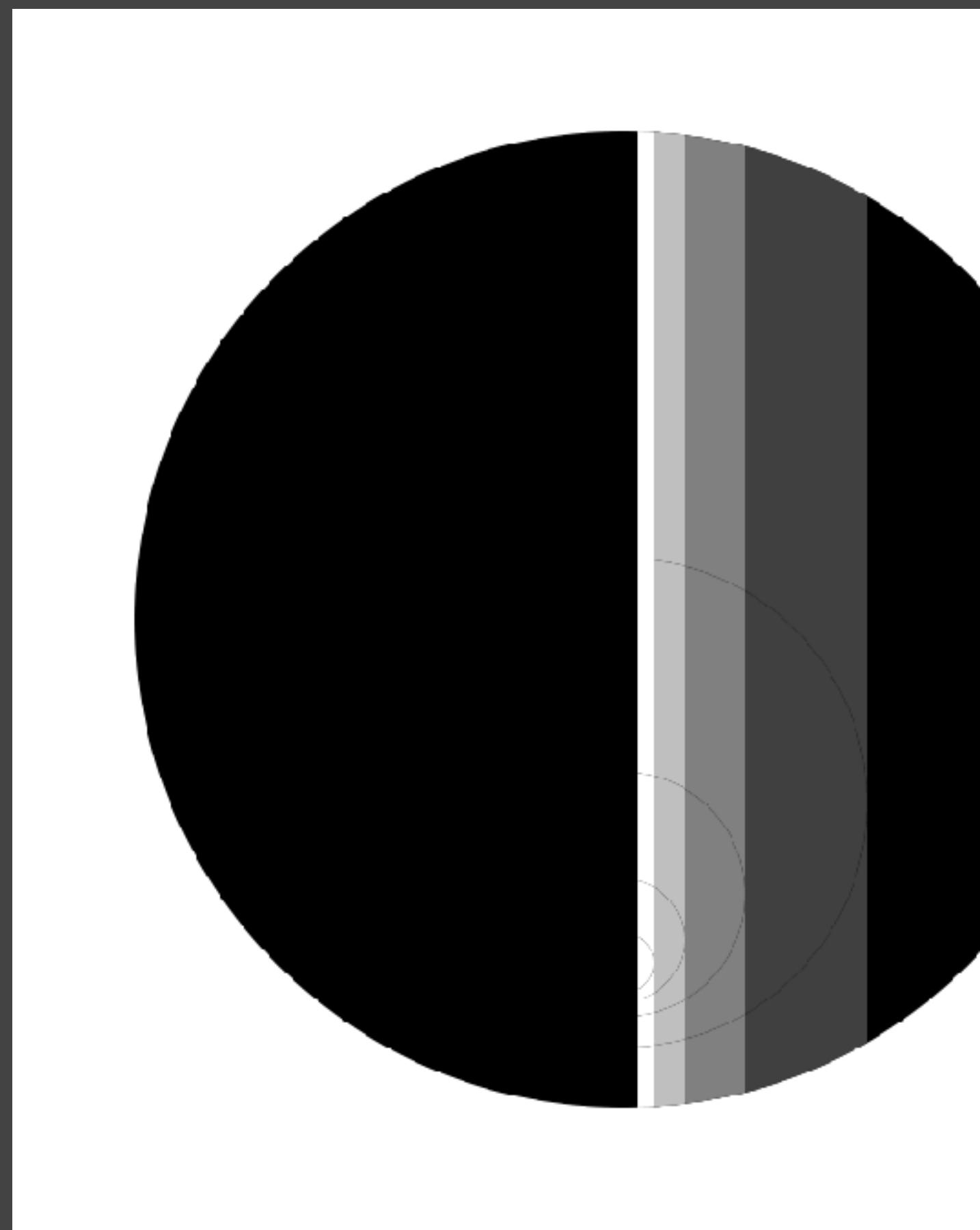
Le Parc transformations



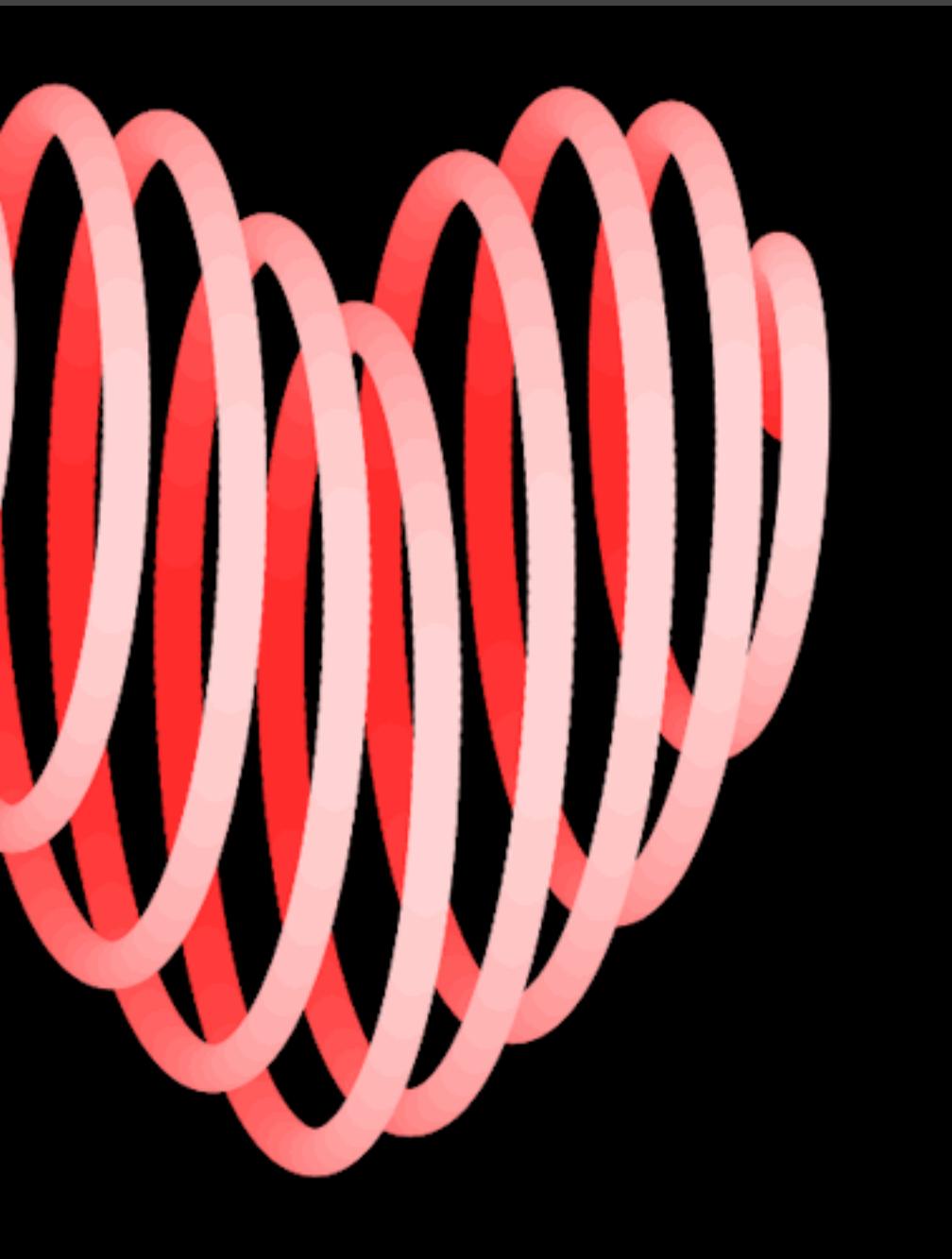
Truchet tiles



Gerstner's Tangential Eccentric

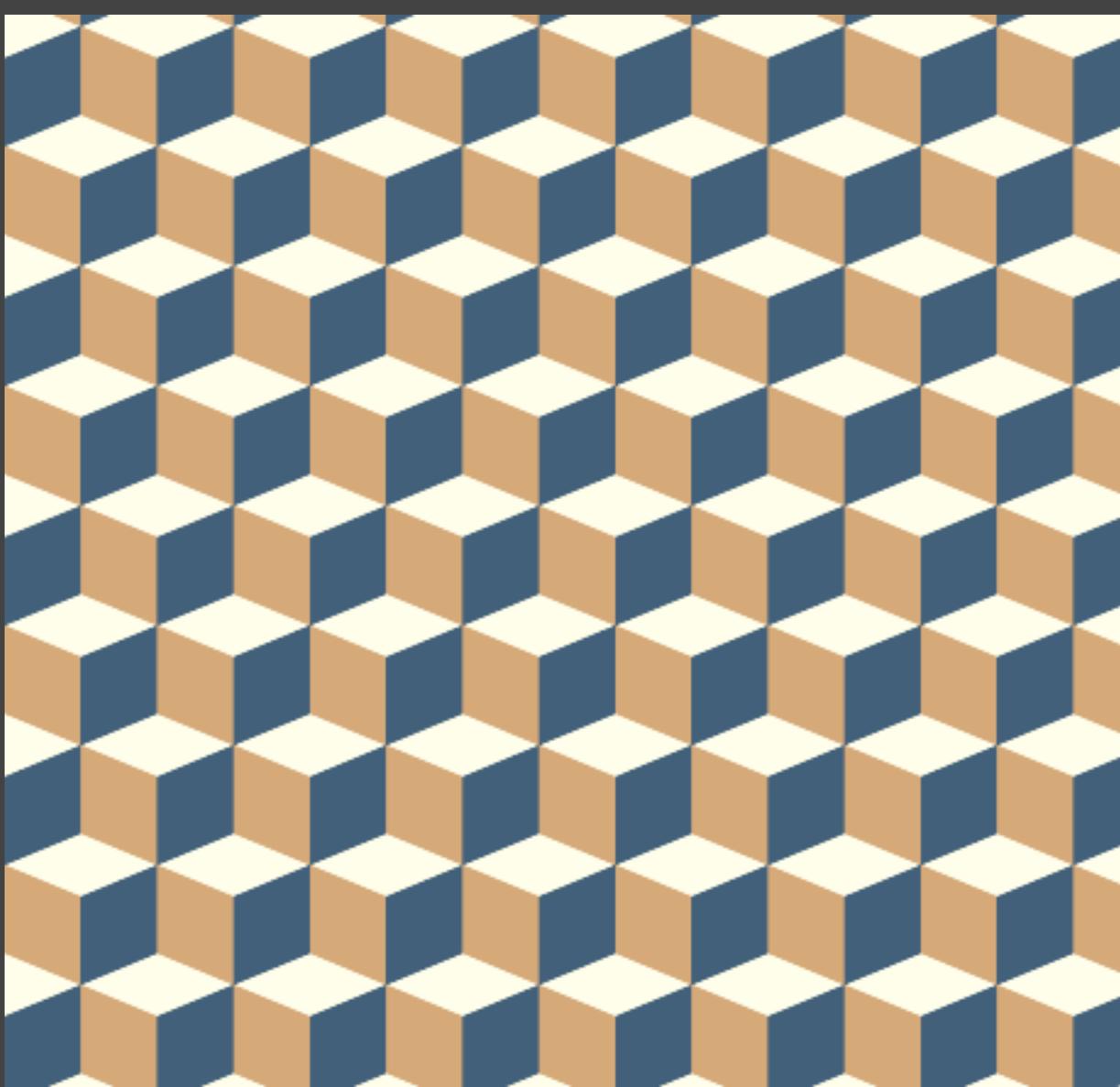
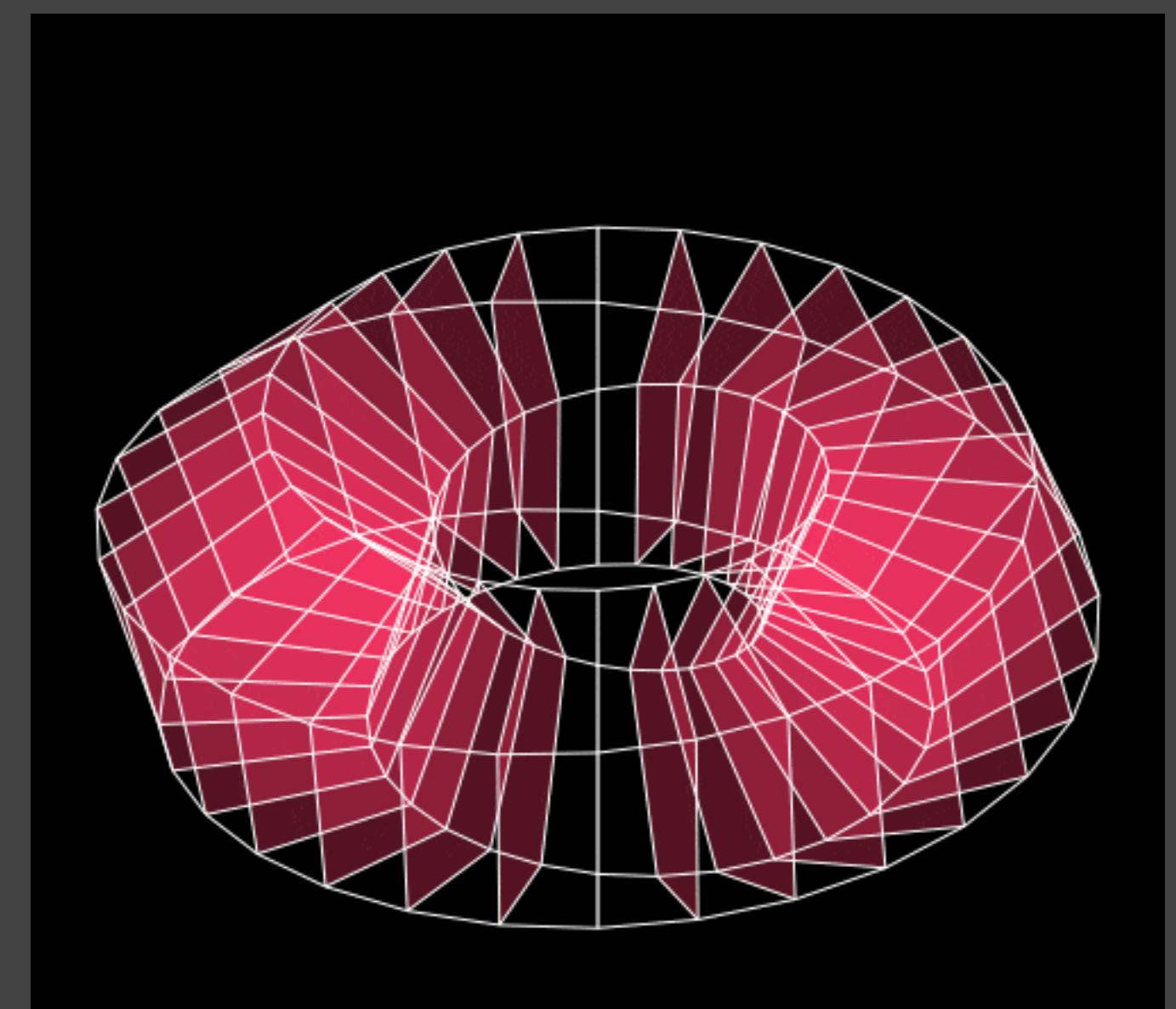
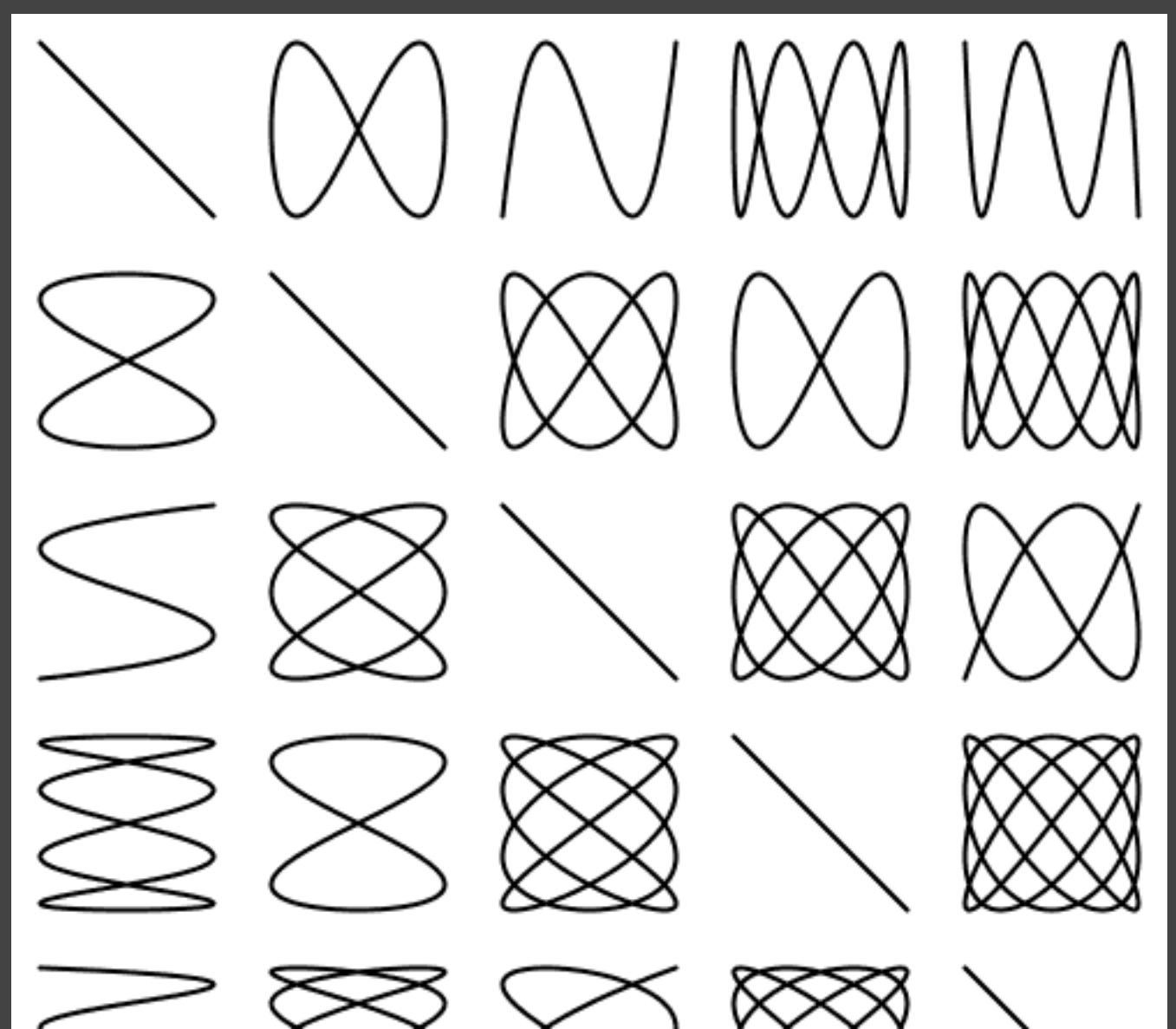
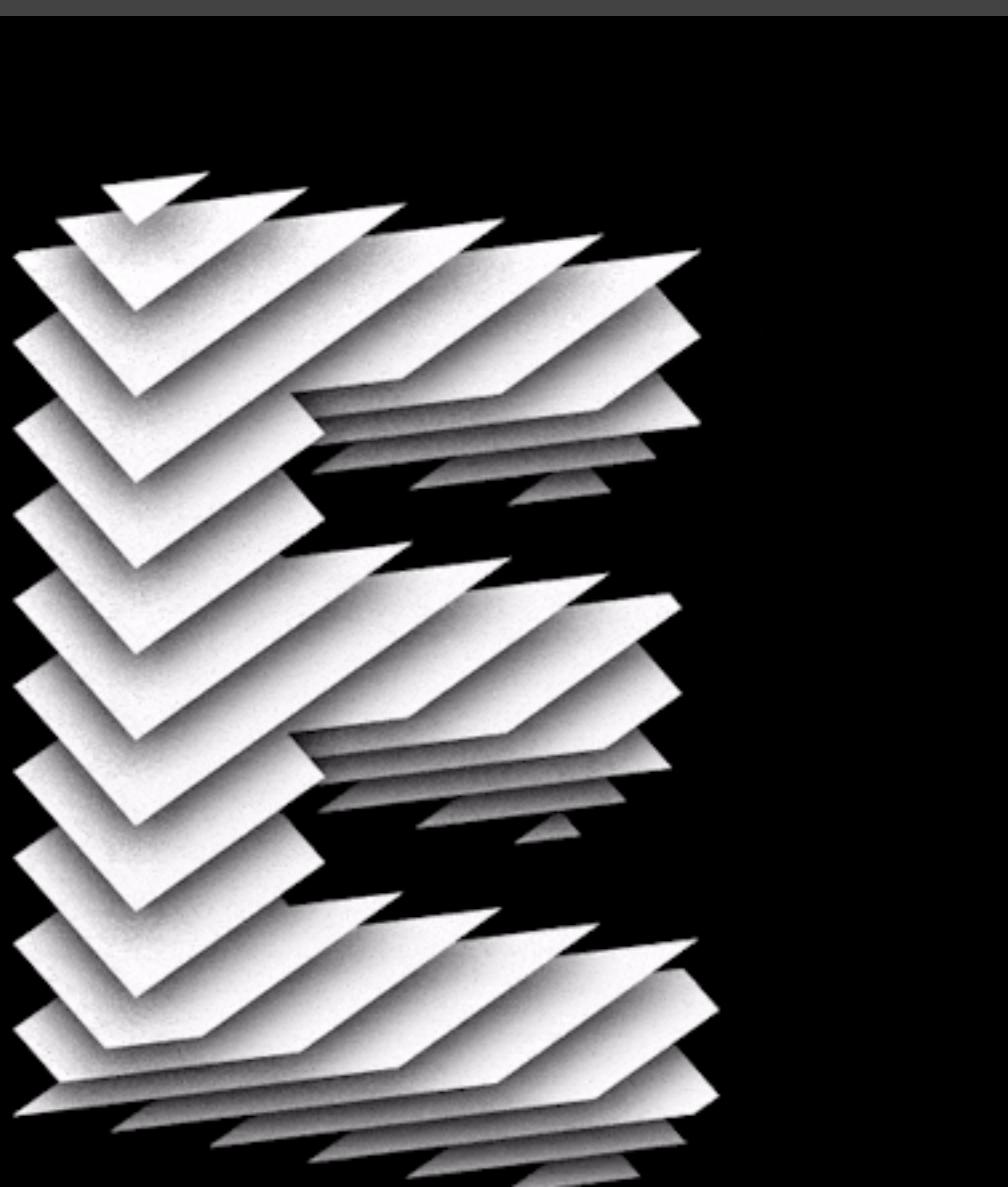
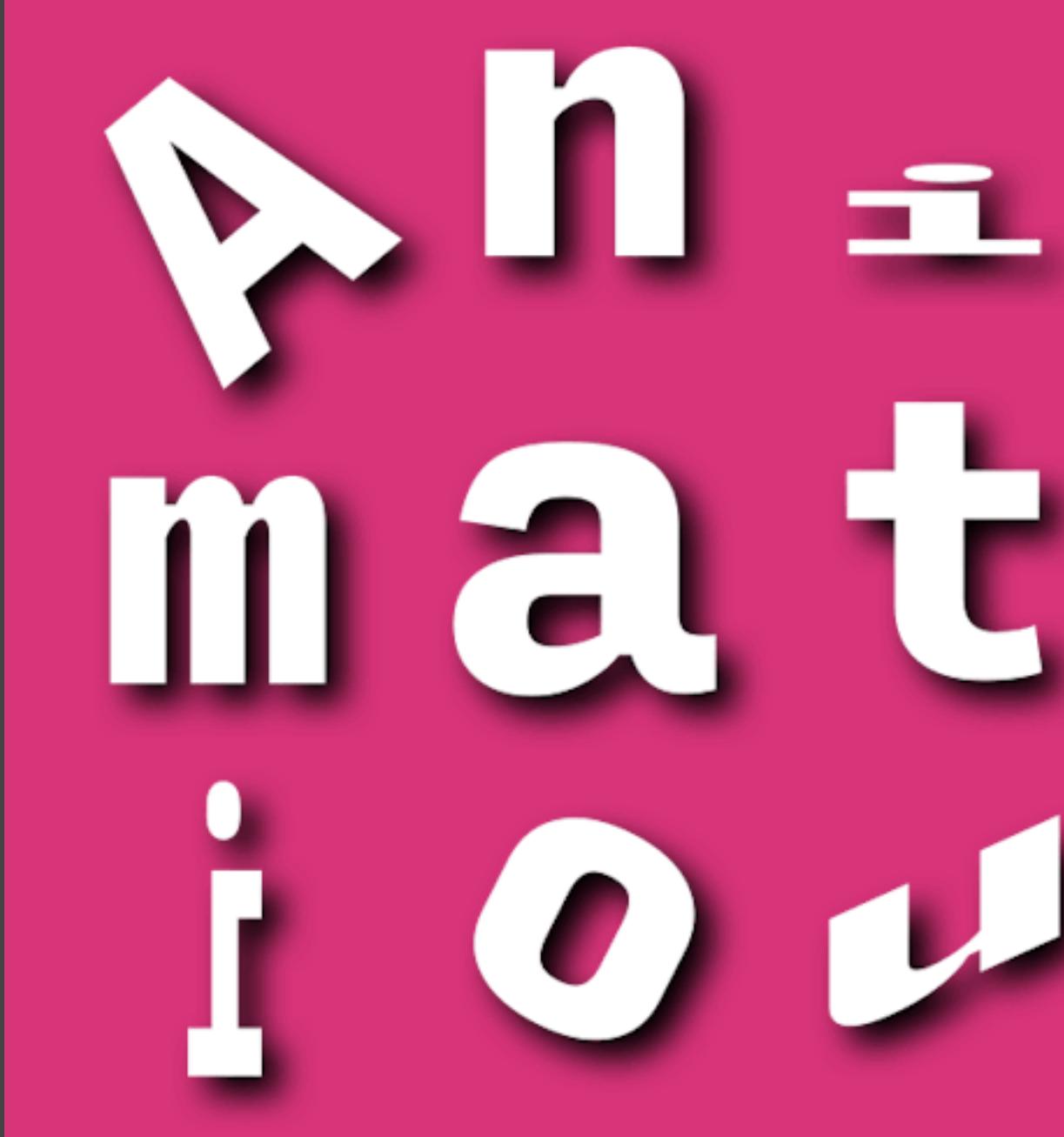


Daily Drawbot



dailydrawbot.tumblr.com

Just van Rossum



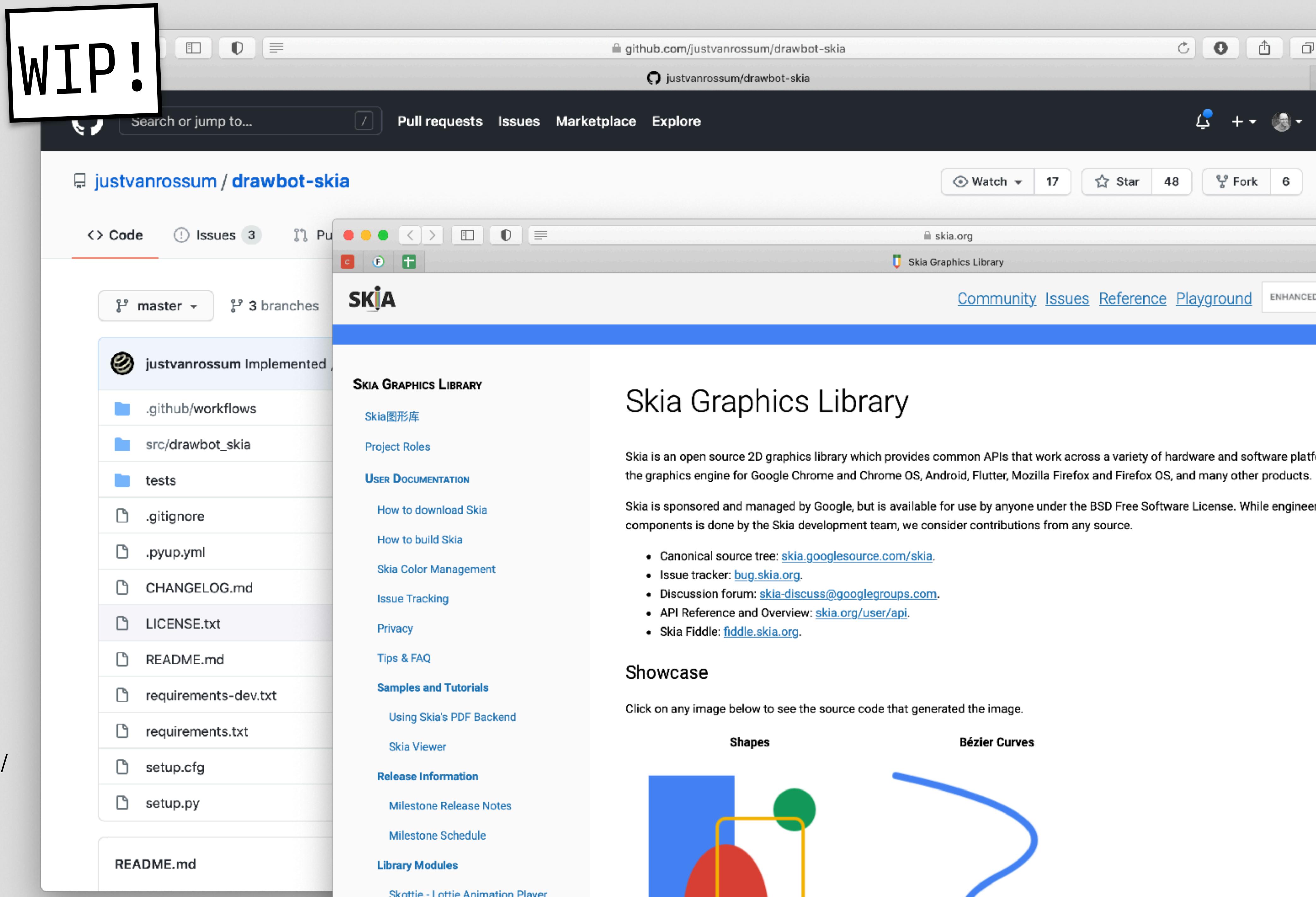
Related Projects

Drawbot Skia

Just van Rossum

[github.com/justvanrossum/
drawbot-skia](https://github.com/justvanrossum/drawbot-skia)

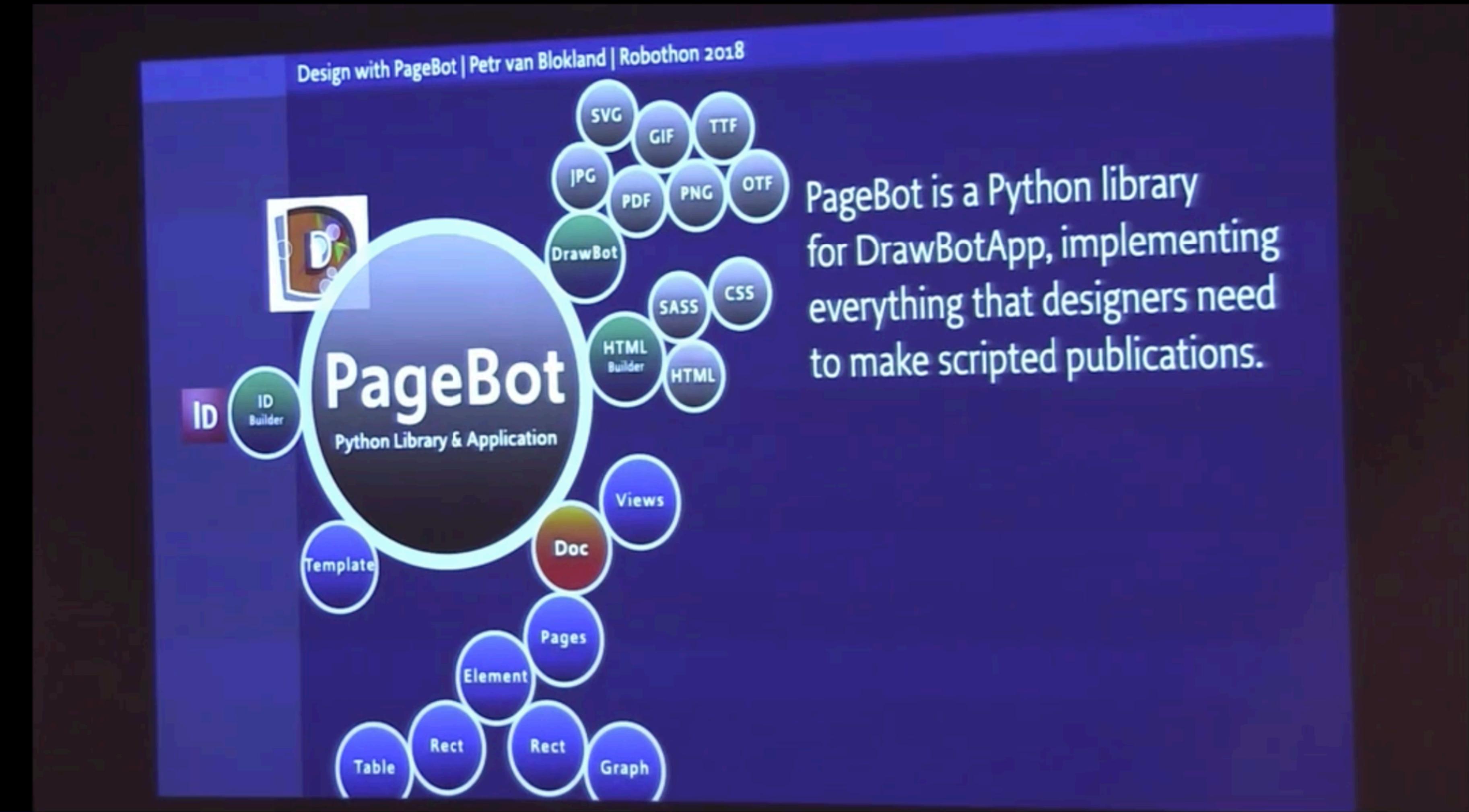
skia.org



PageBot

Buro Petr van Blokland
+ Claudia Mens

Robothon
2018



Flat

Juraj Sukop

The image shows two overlapping windows. The top window is a web browser displaying the Flat library documentation at xxxyz.org/flat. It features a navigation bar with tabs for 'Flat' and 'Even'. Below the tabs is a large image of several curved lines in various colors (blue, red, green, yellow) on a white background. The main content area has a heading 'Flat — Generative infrastructure for Python'. It includes a paragraph about the library's purpose, mentioning it grew from generative design needs and is now a library for creating and manipulating documents. It also notes it is written in pure Python and distributed under the MIT license. A 'Content' section lists links to 'Features', 'Core concepts', 'Tutorial', 'Examples', and 'API reference'. The bottom window is a code editor showing a Python script titled 'Tutorial'. The script uses the Flat library to create a document with a red circle, a red headline with the text 'Hello world!', and an image from a file named 'hello.png'. It also generates an SVG file 'hello.svg' and a PDF file 'hello.pdf'. A 'Short commentary' section below the code provides context for the Flat library's design and its relationship to generative design.

Tutorial

```
from flat import rgb, font, shape, strike, document

red = rgb(255, 0, 0)
lato = font.open('Lato-Reg.otf')
figure = shape().stroke(red).width(2.5)
headline = strike(lato).color(red).size(20, 24)

d = document(100, 100, 'mm')
p = d.addpage()
p.place(figure.circle(50, 50, 20))
p.place(headline.text('Hello world!')).frame(10, 10, 80, 80)
p.image(kind='rgb').png('hello.png')
p.svg('hello.svg')
d.pdf('hello.pdf')
```

Short commentary:

We first prepared some invariants which we are going to use later, like the body color, typeface, some RGB color or a typeface we opened from a font file. One can think of `shape` and `strike` as of customizable factories which produce more concrete objects, for example lines or spans of text.

Next is the basic document hierarchy with just one page that can have items being placed into. The origin of coordinate system (0, 0) is at the top left corner and most of the time the default unit is "points" (1 inch = 72 points). A placed item may have some additional properties as position or frame. The latter is used to define the boundaries inside whose the text may run. As Flat currently lacks any kind of layouting, the frame is just a bounding box.

ColdType

Rob Stenson

**fool
type**

Goodhertz

LAX

Vulfpeck



**A quick
overview!**

**TEXT
SHAPE
IMAGE
CANVAS**

`text()`

`textBox()`

`FormattedString`

`rect()`

`polygon()`

`line()`

`BezierPath`

`oval()`

`arc()`

`image()`

`imagePixelColor()`

`ImageObject`

`rotate()`

`skew()`

`width()`

`fill()`

`translate()`

`scale()`

`height()`

`stroke()`

Okay then,
let's code.