

Greetings from



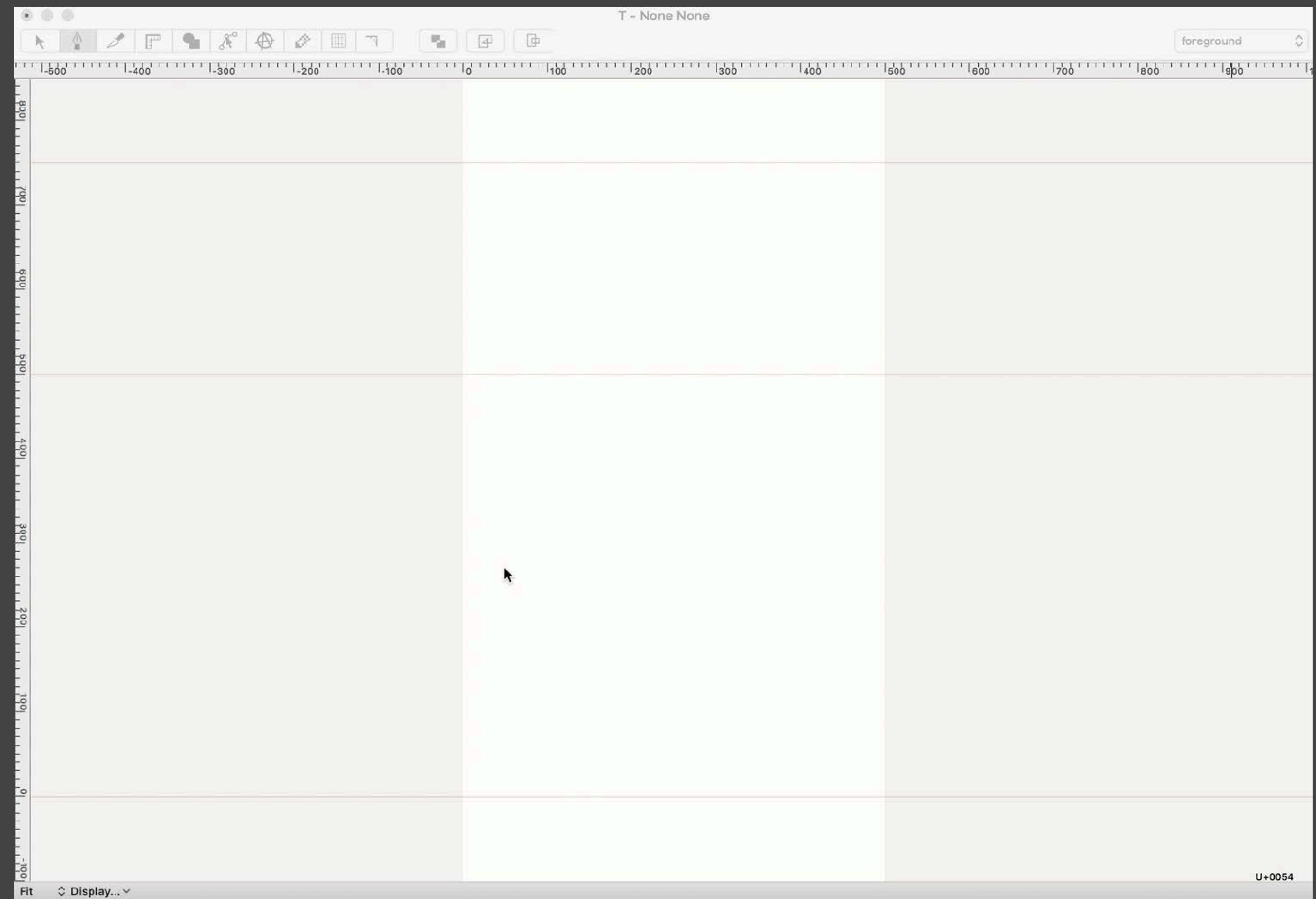
16199

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 blue grid background is visible behind the main content area. On the left side, there is a white rectangular box containing the "FONT OF THE MONTH CLUB" logo, which consists of the words "FONT", "OF THE", "MONTH", and "CLUB" stacked vertically in a bold, sans-serif font, separated by thin horizontal lines. To the right of this logo, a call-to-action box contains the text: "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 the main content area, a dark blue section features the text "GIMLET X-RAY > Discover the inner-workings of a variable font". This section contains a complex diagram titled "GIMLET X-RAY" showing the internal structure of a variable font. It consists of several vertical columns of letters, likely 'O's and 'U's, each represented as a series of points connected by lines, illustrating the font's variable nature through different stroke weights or outlines.

The screenshot shows a web browser window for the URL fontofthemonth.club/. The page features a black header bar with the text "DAVID JONATHAN ROSS" on the left and "TYPEFACES + CLUB + ABOUT NOTES Sign In Cart" on the right. Below the header is a large, bold, black title "FONT OF THE MONTH CLUB" centered on the page. To the right of the title is a red rectangular button containing the text "JOIN THE CLUB!". Below this button, a paragraph of text explains the service: "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." Further down, another red button says "START YOUR MEMBERSHIP TODAY!" followed by the text "Or give a gift subscription to your favorite font lover!" at the bottom. At the very bottom of the page, there's a red footer bar with the text "THIS MONTH'S FONT" on the left, "Klooster Thin" and "Taking the uncial in new directions" in the center, and two buttons on the right: "JOIN TODAY! \$6" and "MORE".

Font of the Month Club | David

fontofthemonth.club/

DAVID JONATHAN ROSS

TYPEFACES + CLUB + ABOUT NOTES Sign In Cart

FONT OF THE MONTH CLUB

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.

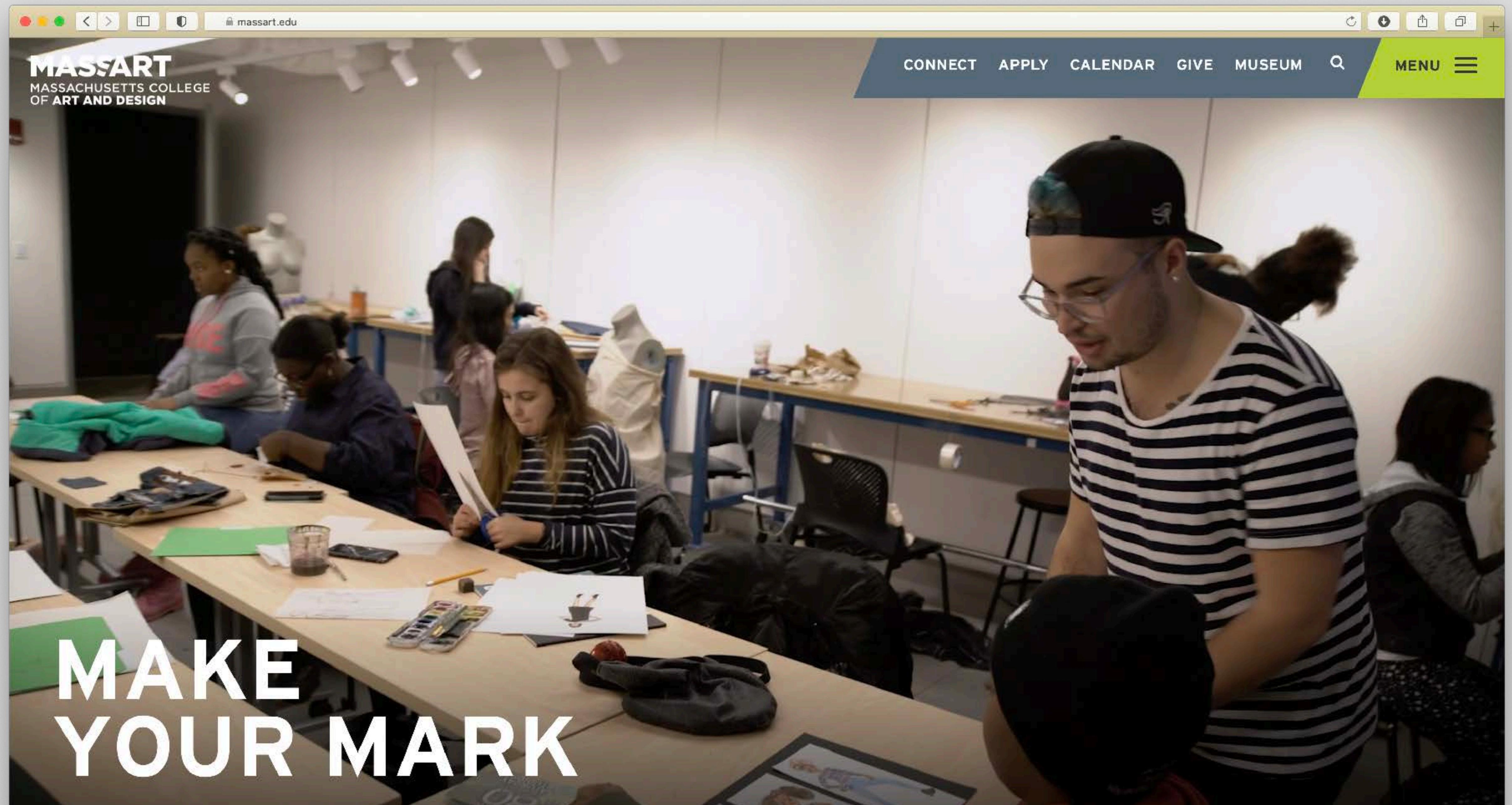
START YOUR MEMBERSHIP TODAY!

Or give a gift subscription to your favorite font lover!

THIS MONTH'S FONT

Klooster Thin
Taking the uncial in new directions

JOIN TODAY! \$6 MORE



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". Below the title bar, there's a navigation menu with links to "Home", "Extended Program", "Condensed Program", "Lecture Series", "Public Workshops", "Instructors", and "FAQ". The main content area features a large, stylized logo for "Type@Cooper" where the '@' symbol is represented by a blue circle. To the left of the logo is a graphic composed of several red circles and lines, resembling a complex typeface or a network diagram. Below the logo, a blue text block reads: "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 geometric pattern of colored squares.

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.

GOUPAR
GOUPAR

PEABAN
PEABAN

GYMPE
GYMPE

BUCKEYE MEADOW

CRESCENT HEIGHTS

WESTBUENA VISTA

GATEWAY TOWERS

PASCO 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 it.

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

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...atures 1/4

Colon Sidebar

colon

58 → 61

= → = ← =

→ ← 2 ← →

Increment

Preview

Layers

+ - Layer Name Color

Braceleft Braceright Hyphen Underscore Endash Endash.salt_en Emdash Emdas...alt_em Quotaleft Quotaright Quotedblleft Quotedblright Quot...g

Guilsingright Brokenbar Slash Backslash Bar Asterisk Dagger

Braceleft.uc Braceright.uc Hyphen.uc Endash.uc Endash...salt_en Emdash.uc Emdas...

Adieresis Aring Amacron Abreve Aogonek AE Ccedilla Ca

Edieresis Emacron Ebreve Edotaccent Eogonek Ecaron Gcircumflex Gb

Imacron Ibreve Iogonek Idotaccent IJ Jcircumflex Kcom...accent La

Ograve Oacute Ocircumflex Otilde Odieresis Omacron Obreve Ohung

Inspector - colon

a - Turnip Black

The screenshot shows the RoboFont application interface. The main window displays a grid of glyphs for the font 'Turnip-Black.ufo'. The 'colon' character is selected and highlighted in blue. The right side of the interface features the 'Inspector' panel, which provides detailed information about the selected glyph, including its name ('colon'), Unicode value (003A), width (337), and bounding box (Left: 58, Right: 61). The 'Sidebar' section contains various keyboard shortcut definitions for the colon character. Below the grid, a smaller window shows a detailed view of the 'colon' glyph's structure, including its stroke order and control points. The bottom of the screen shows the RoboFont toolbar and a status bar indicating the current character code (U+0061).

MERIT BADGE

IS A VARIABLE
COLOR FONT.



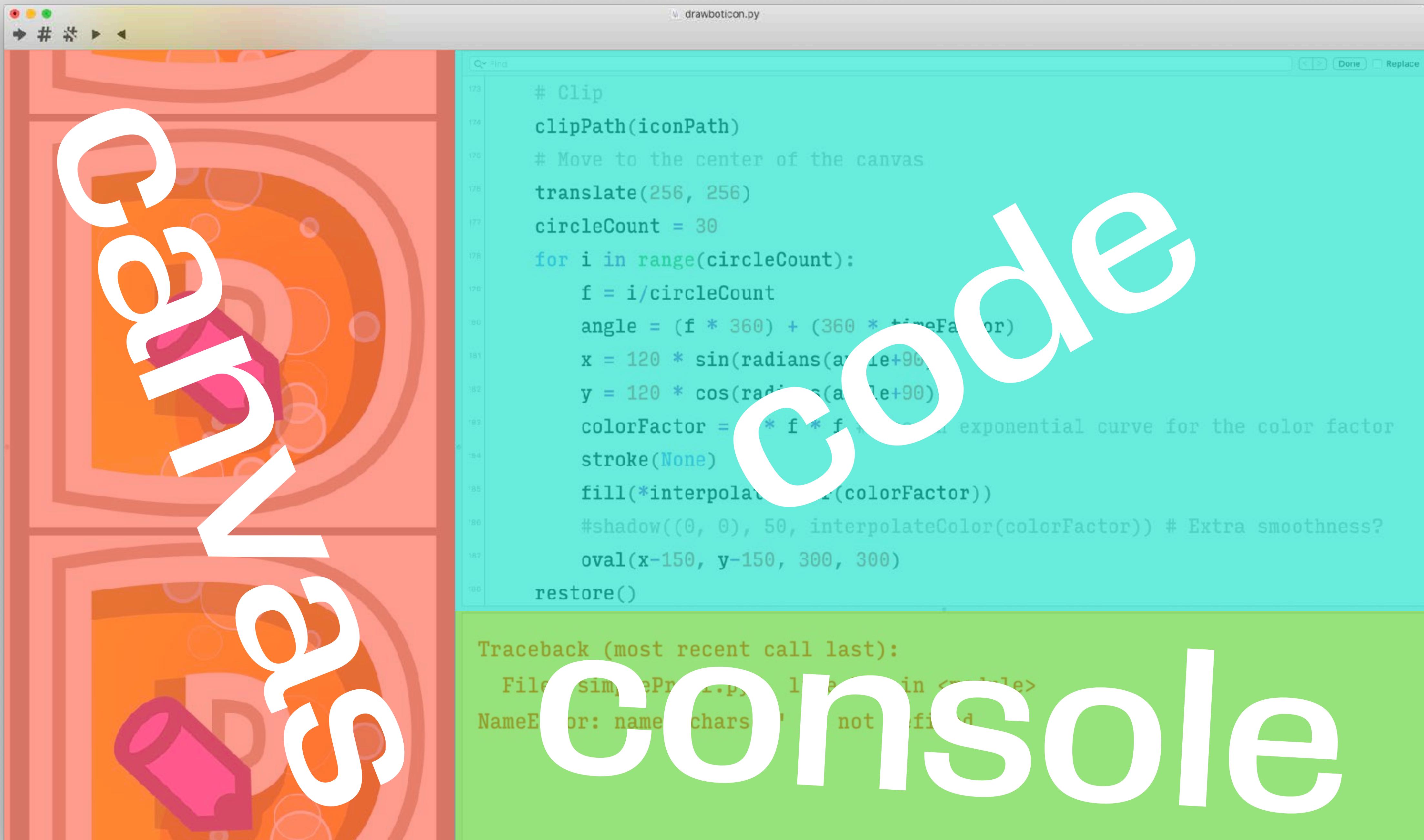
Just van Rossum

Frederik Berlaen

DrawBot

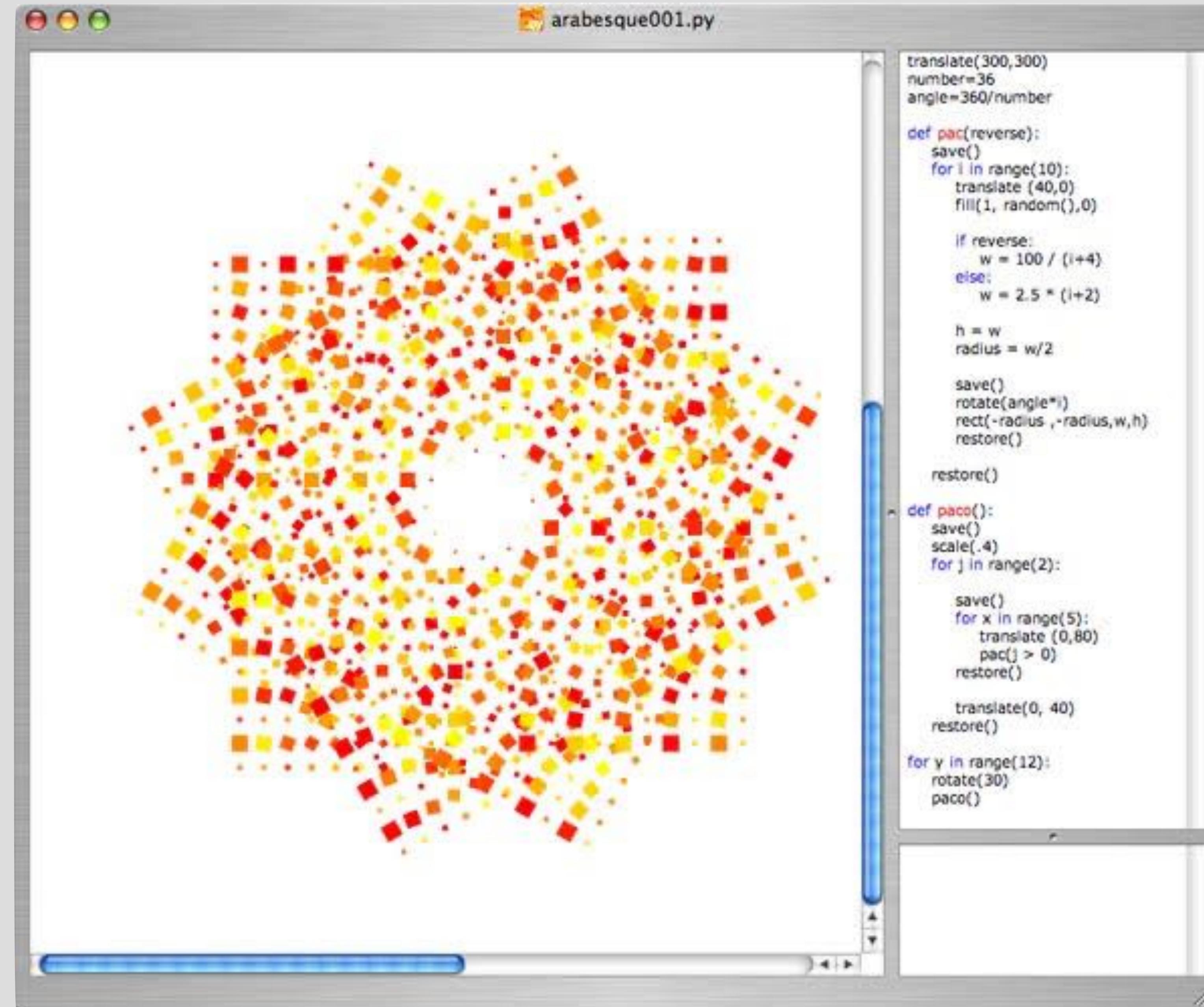


drawbot.com
[@drawbotapp](https://twitter.com/drawbotapp)



Rotation exercise, 2006

Pascal Zoghbi



The screenshot shows a Python script window titled "arabesque001.py". The script contains code for generating a fractal pattern of red and yellow squares. The pattern is centered around a point at (300, 300) and consists of 36 iterations of a recursive function "pac". The "pac" function rotates and scales the pattern. The "paco" function calls "pac" for each iteration. The script uses the "translate", "number", "angle", "def", "pac", "reverse", "for", "if", "else", "w", "h", "radius", "save", "rotate", "rect", "restore", "def", "paco", "scale", "for", "x", "y", and "translate" functions.

```
translate(300,300)
number=36
angle=360/number

def pac(reverse):
    save()
    for i in range(10):
        translate (40,0)
        fill(1, random(),0)

    if reverse:
        w = 100 / (i+4)
    else:
        w = 2.5 * (i+2)

    h = w
    radius = w/2

    save()
    rotate(angle*i)
    rect(-radius ,-radius,w,h)
    restore()

    restore()

def paco():
    save()
    scale(.4)
    for j in range(2):

        save()
        for x in range(5):
            translate (0,80)
            pac(j > 0)
            restore()

        translate(0, 40)
        restore()

    for y in range(12):
        rotate(30)
        paco()
```

Screenshot courtesy
of Just van Rossum

**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.

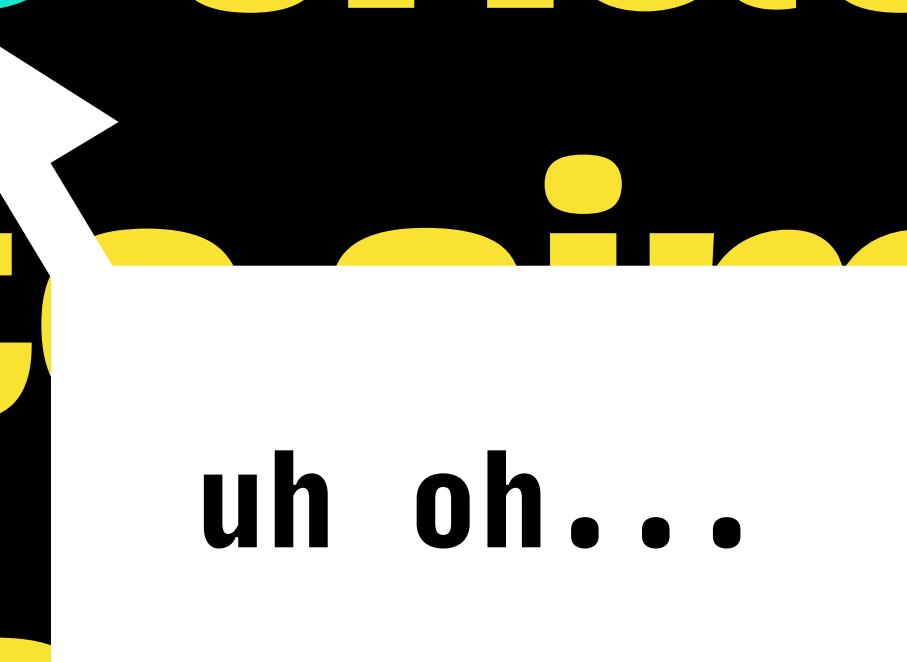
**DrawBot is a powerful,
free application for
MacOS that invites you
to write simple Python
scripts to generate two-
dimensional graphics.**



**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.**

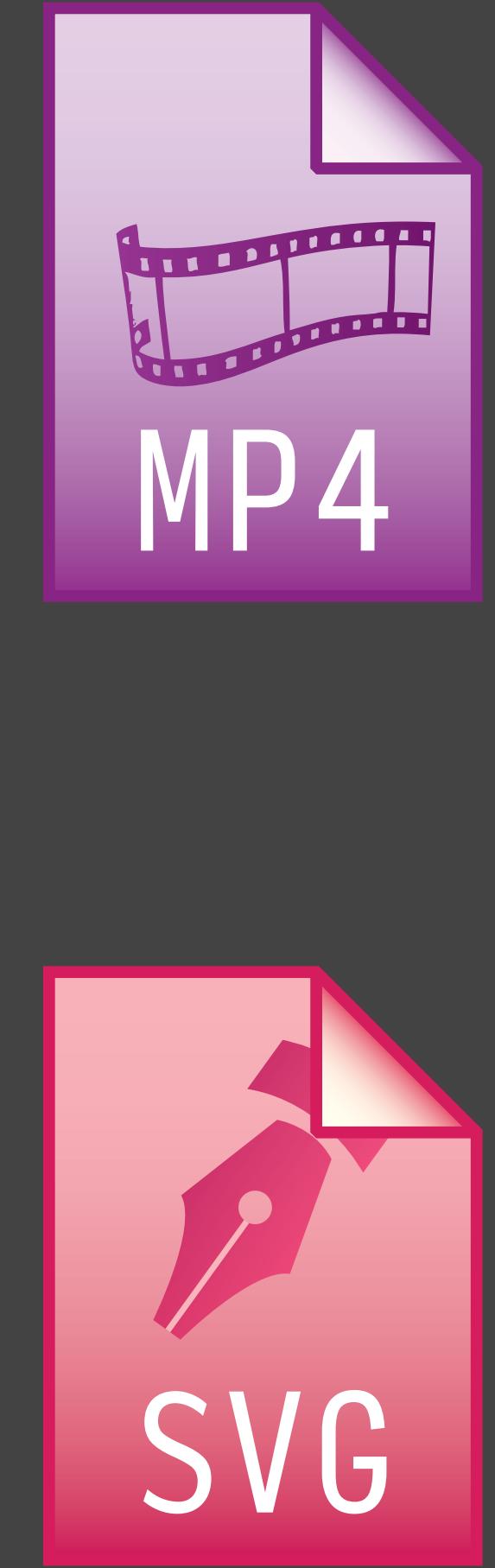


uh oh...

**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 description" (The ReportLab Toolkit), "Project links" (Homepage, Download), and "Statistics".

The screenshot shows the PyPI project page for pyPdf 1.13. The title "pyPdf 1.13" is displayed at the top. Below it is a "pip install pyPdf" button. The main content area includes sections for "Navigation" (Project description, Release history, Download files), "Project description" (A Pure-Python library built as a PDF toolkit. It is capable of extracting document information (title, author, ...), splitting documents page by page, merging documents page by page, cropping pages, merging multiple pages into a single page, encrypting and decrypting PDF files.), and "Project links" (Homepage, Download).

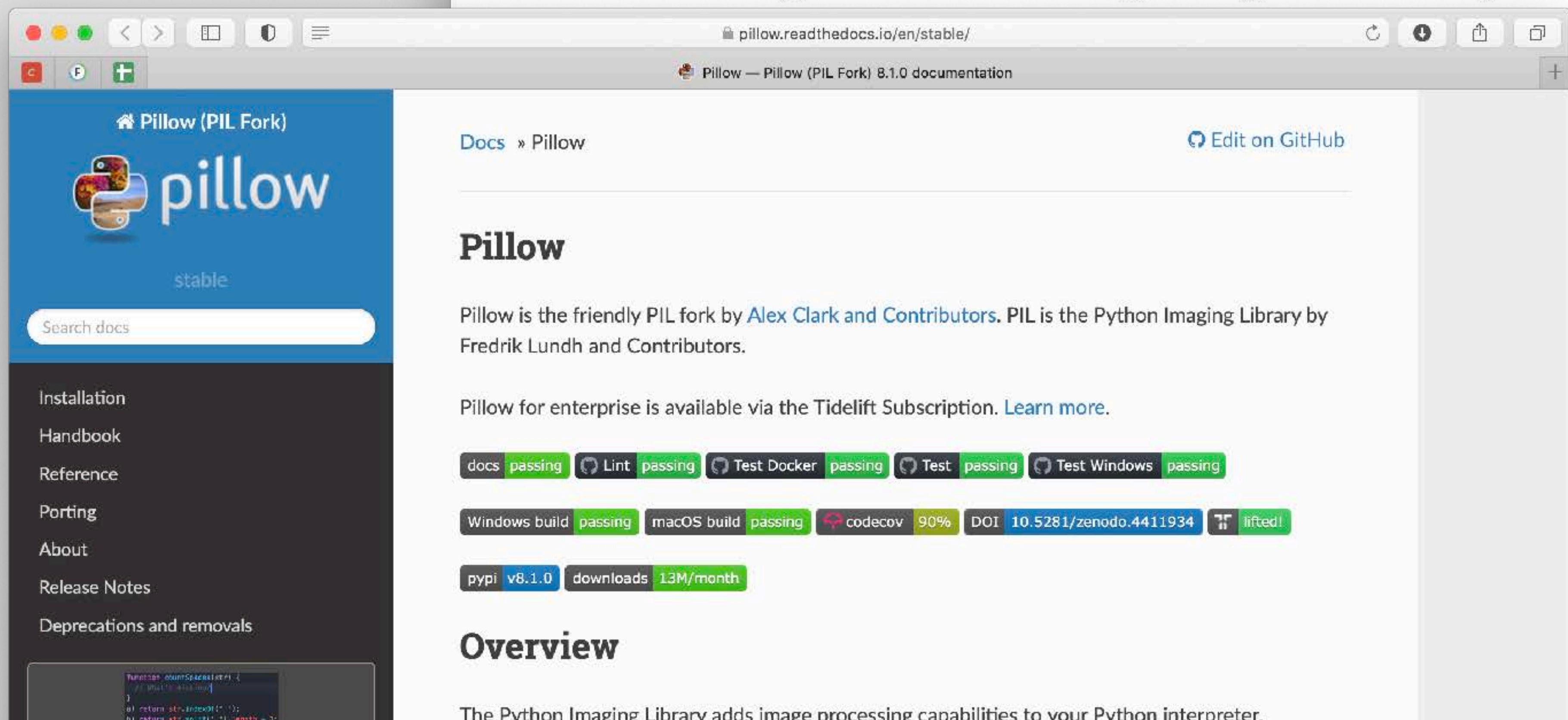
The screenshot shows the GitHub repository page for JazzCore/python-pdfkit. The repository name "JazzCore / python-pdfkit" is at the top. It shows 83 issues, 19 pull requests, and 35 actions. The code tab is selected, showing the master branch with 1 branch and 0 tags. A list of commits is shown, starting with a merge pull request from alanhamlett on Oct 22, 2018.

Commit	Description	Date	Author
c15ed32	Merge pull request #119 from ala...	Oct 22, 2018	alanhamlett
pdfkit	support unicode keyword on Python 3	2 years ago	
tests	patch unittest.TestCase.assertRegex for py27	3 years ago	
travis	fix wkhtml2pdf install on travis	3 years ago	
.gitignore	Init tox config for multi-Python version testing	4 years ago	
.travis.yml	More Travis fixes	4 years ago	
AUTHORS.rst	bump verison to 0.6.1	4 years ago	
HISTORY.rst	bump verison to 0.6.1	4 years ago	
LICENSE	dist preparations	8 years ago	
MANIFEST.in	update setup.py	8 years ago	
README.rst	Merge pull request #92 from fbataill/patch-1	3 years ago	
setup.py	version bump	8 years ago	

pycairo
PIL/pillow

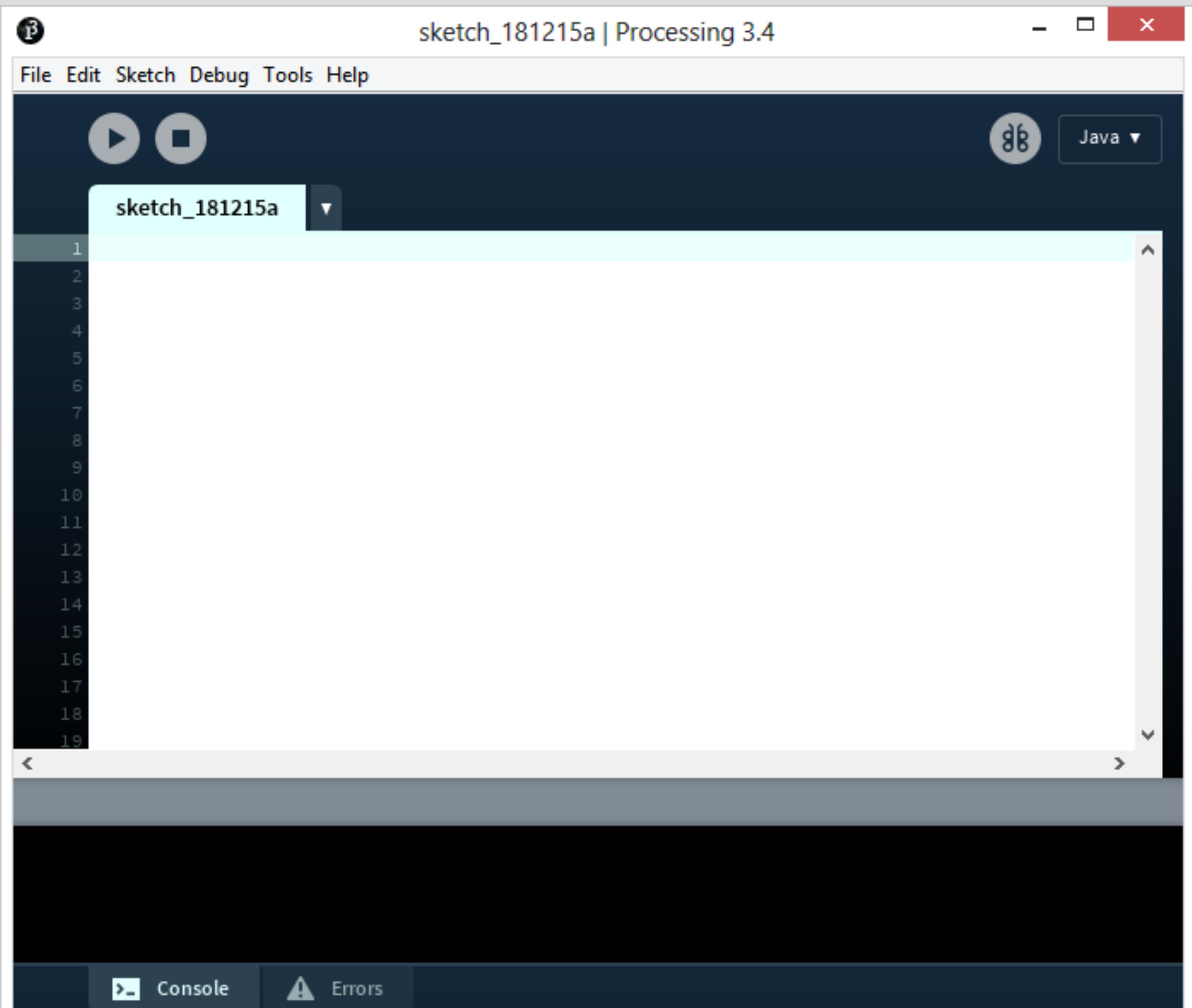


The screenshot shows a web browser window with the URL www.cairographics.org/pycairo/. The page features a logo with two orange beetles carrying blue squares on their backs, with the word "cairo" in large blue letters. A navigation bar at the top includes links for News, Download, Documentation, Contact, and Examples. Below the navigation bar, a section titled "Pycairo" describes it as a set of Python 2 & 3 bindings for the Cairo graphics library. It also notes that 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 is titled "Pillow — Pillow (PIL Fork) 8.1.0 documentation". It features a sidebar with links for Installation, Handbook, Reference, Porting, About, Release Notes, and Deprecations and removals. The main content area includes a "Pillow" heading, a paragraph about Pillow being the friendly PIL fork, and sections on enterprise availability via Tidelift Subscription and build status indicators for various platforms. The footer contains links to releases, snapshots, and a GitHub edit button.

Processing p5.js



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

BERTRAM GOODHUE
AND IN PARTICULAR
IS 18 RING
ORT R OF
IE K RANT

Character panel settings: Pomfret, Regular, TT 108 pt, VA 76 pt, Metrics, 100%, 100%, 0 pt, 0°, English: USA.

BERTRAM GOODHUE
AND IN PARTICULAR
HIS 1892 LETTERING
FOR THE COVER OF
THE KNIGHT ERRANT

A-Master

1

2-3

4-5

6-7

8-9

13 Pages in 7 Spreads

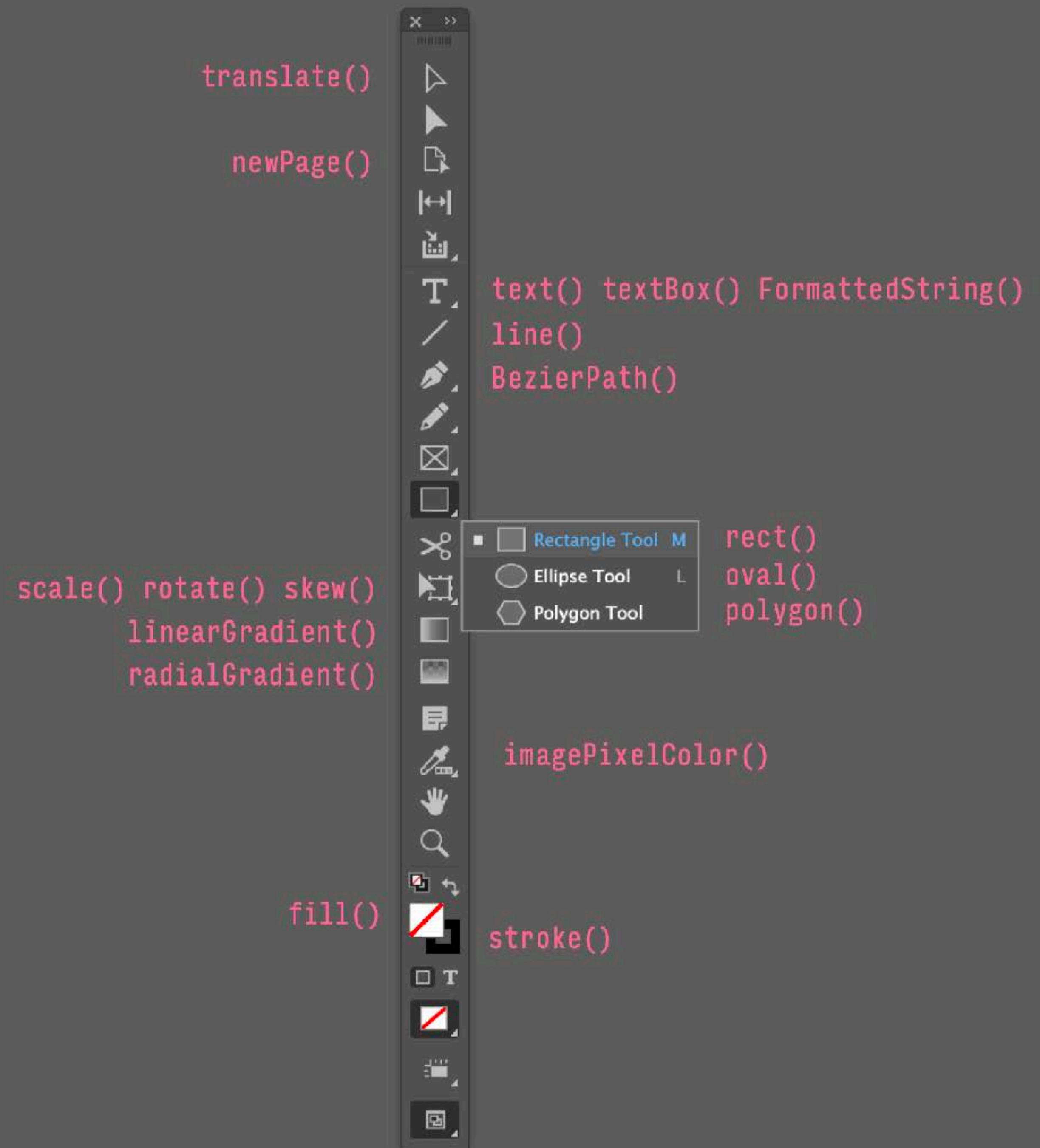
POMFRET Specimens djr.com | 2

BARRANCA
STOPBACK

POMFRET Specimens djr.com | 3

EXPLAINER
RECREANT

The screenshot shows the Adobe InDesign 2020 interface. The main workspace displays a document spread with text "BERTRAM GOODHUE" and "AND IN PARTICULAR" in blue, and "IS 18 RING", "ORT R OF", and "IE K RANT" in dark blue. A character panel is open, showing settings for the selected text: font "Pomfret", style "Regular", size "TT 108 pt", leading "VA 76 pt", metrics "Metrics", tracking "100%", kerning "100%", baseline offset "0 pt", rotation "0°", and language "English: USA". To the right, a sidebar lists page numbers 1 through 9 and spread numbers 2-3, 4-5, 6-7, and 8-9. Below the sidebar, it says "13 Pages in 7 Spreads". At the bottom, two specimen panels are visible: "BARRANCA STOPBACK" in black and red, and "EXPLAINER RECREANT" in red and black. The bottom status bar shows "53.45%" zoom, page 1, "Basic (working)" document status, and "1 error".



Lowlands 2018

Hansje van Halem, Just van Rossum



Lowlands Festival 2018

Hansje van
Halem, Just
van Rossum

hansje.net



The image shows a Mac OS X desktop with two windows open. On the left is a PDF viewer showing a black and white invoice. On the right is a code editor showing the Python script used to generate the invoice.

invoice.py

```
*digitDiff + price
if price == 'None':
    productLine = ''
productLine += '\r'
productInfo.append(productLine,
font=textFont)

if data.get('discount'):
    productInfo.append('Discount: %s\tUS $ - %s\r' %(str(data['discount']['name']),
str(data['discount']['amount-saved'])),
font=inputFont, lineHeight=textLineHeight*2)

if paid:
    OutputSans-Regular
    OutputSans-Italic
    OutputSans-Medium
    OutputSans-MediumItalic
    OutputSans-Bold
    OutputSans-BoldItalic
```

invoice.pdf

The invoice document contains the following information:

From: DAVID JONATHAN ROSS
P.O. Box 461
Conway, MA 01341 USA
+1 339-224-7687
david@djr.com

To: LICENSEE MCLICENSESON
david@djr.com
P.O. Box 461,
Conway, MA

Date: 2020-09-03

ORDER: 35290320
INVOICE: 35290320-20201013-1

ITEMS **AMOUNT**

OutputSans Regular (Small license)	US \$ 50
OutputSans Italic (Small license)	US \$ 50
OutputSans Medium (Small license)	US \$ 50
OutputSans MediumItalic (Small license)	US \$ 50
OutputSans Bold (Small license)	US \$ 50
OutputSans BoldItalic (Small license)	US \$ 50
TOTAL PAID	US \$ 300

Your SMALL license entitles you to use the font on up to 100 desktop workstations, 10 million monthly unique web visitors, 36 e-books, and 18 apps. Please consult your full license for more information on permitted uses and contact <david@djr.com> to upgrade your license at any time.

Typographics 2015





Typographics 2016

2016.typographics.com

HANSJES
VAN HALEM

KARRA
HAUPT

Typographics 2018





Font Trading Cards

2 S

> w

H +

b c

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

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

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

2

Adidas N3XT L3V3L

Grand Army

Andy Clymer

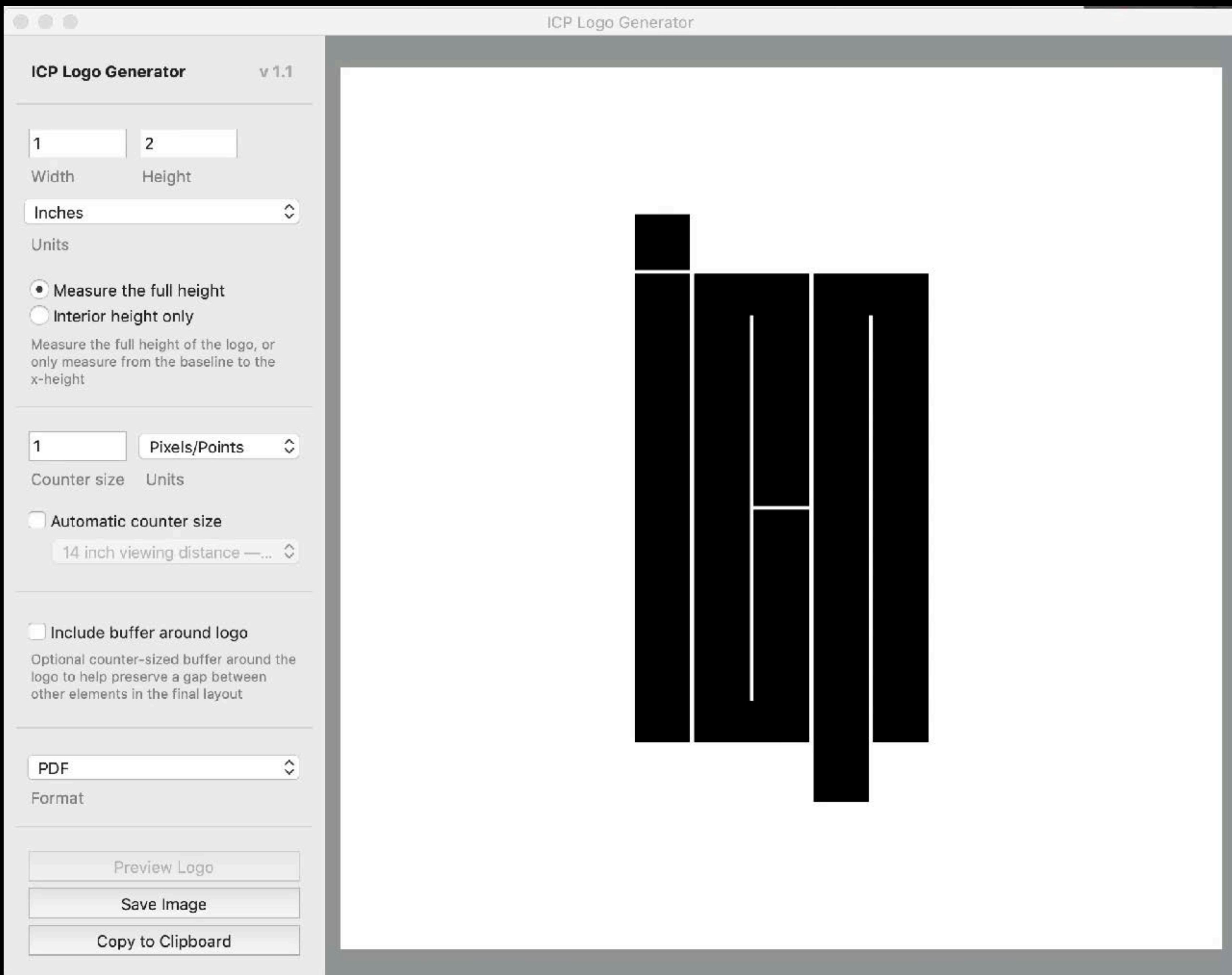


grandarmy.com/projects/n3xt-l3v3l

International Center of Photography, Philadelphia

Pentagram

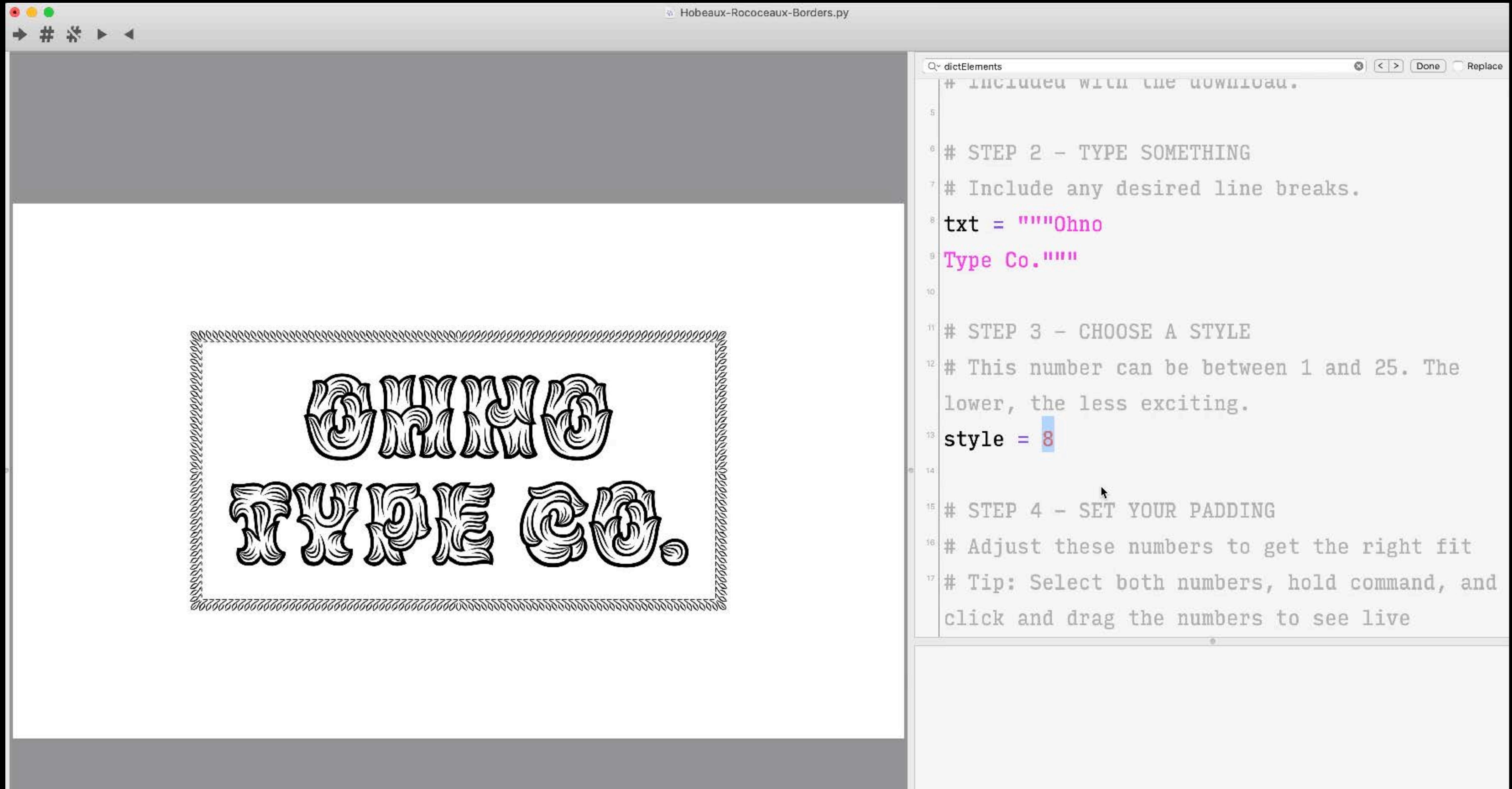
Andy Clymer, Yours Truly



Hobeaux Rococeaux Borders

ohnotype.co/fonts/hobeaux-rococeaux

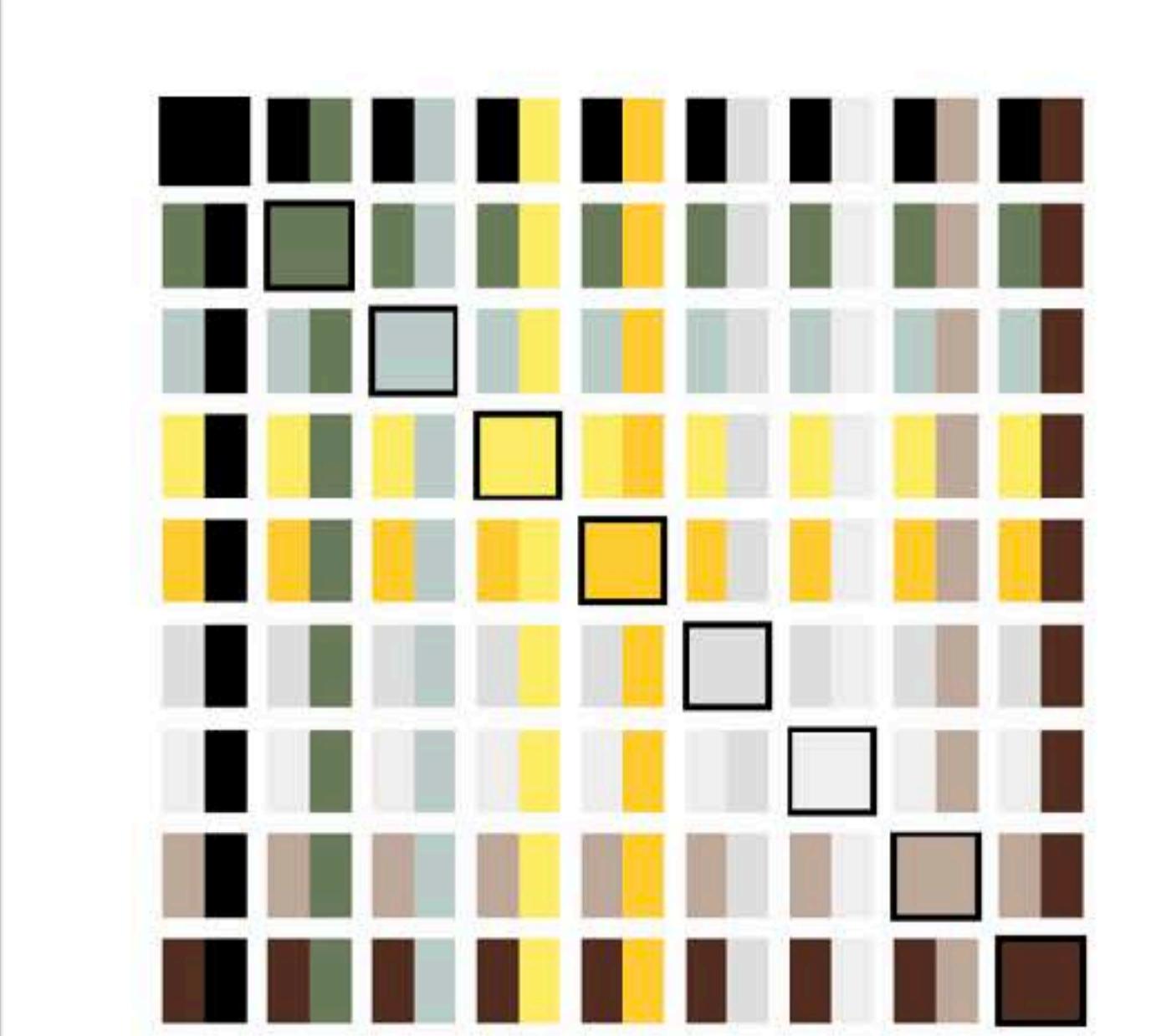
James Edmondson



The image shows a Mac OS X terminal window titled "Hobeaux-Rococeaux-Borders.py". The window contains a Python script with code comments and some placeholder text. The code is as follows:

```
# 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
```

The terminal window also displays two lines of decorative text generated by the script: "Ohno" and "Type Co.", both rendered in a highly ornate, decorative font with intricate scrollwork patterns.



```
Untitled 3

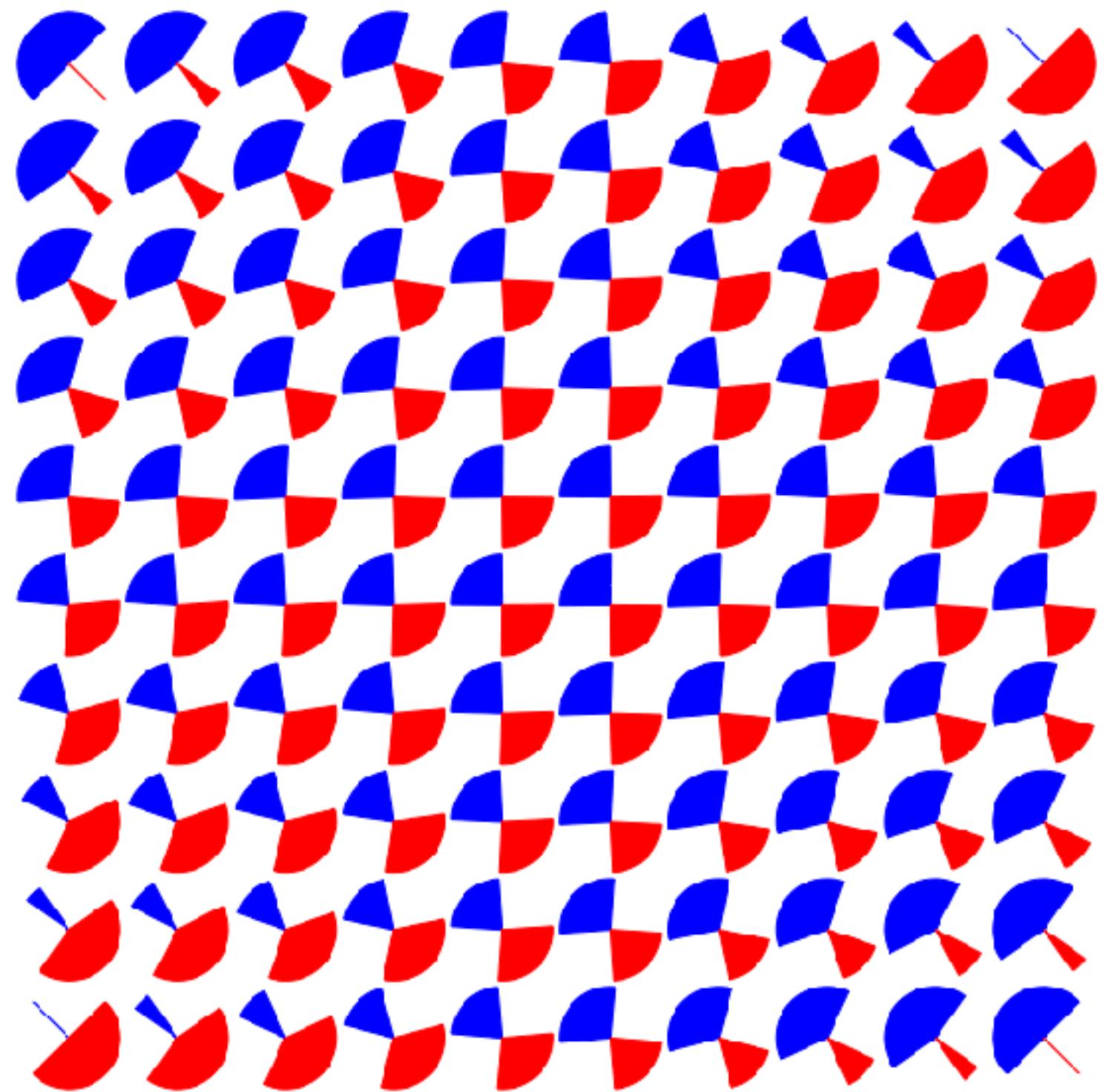
colorSpace(colorspace)
translate(startX,startY)

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)
```

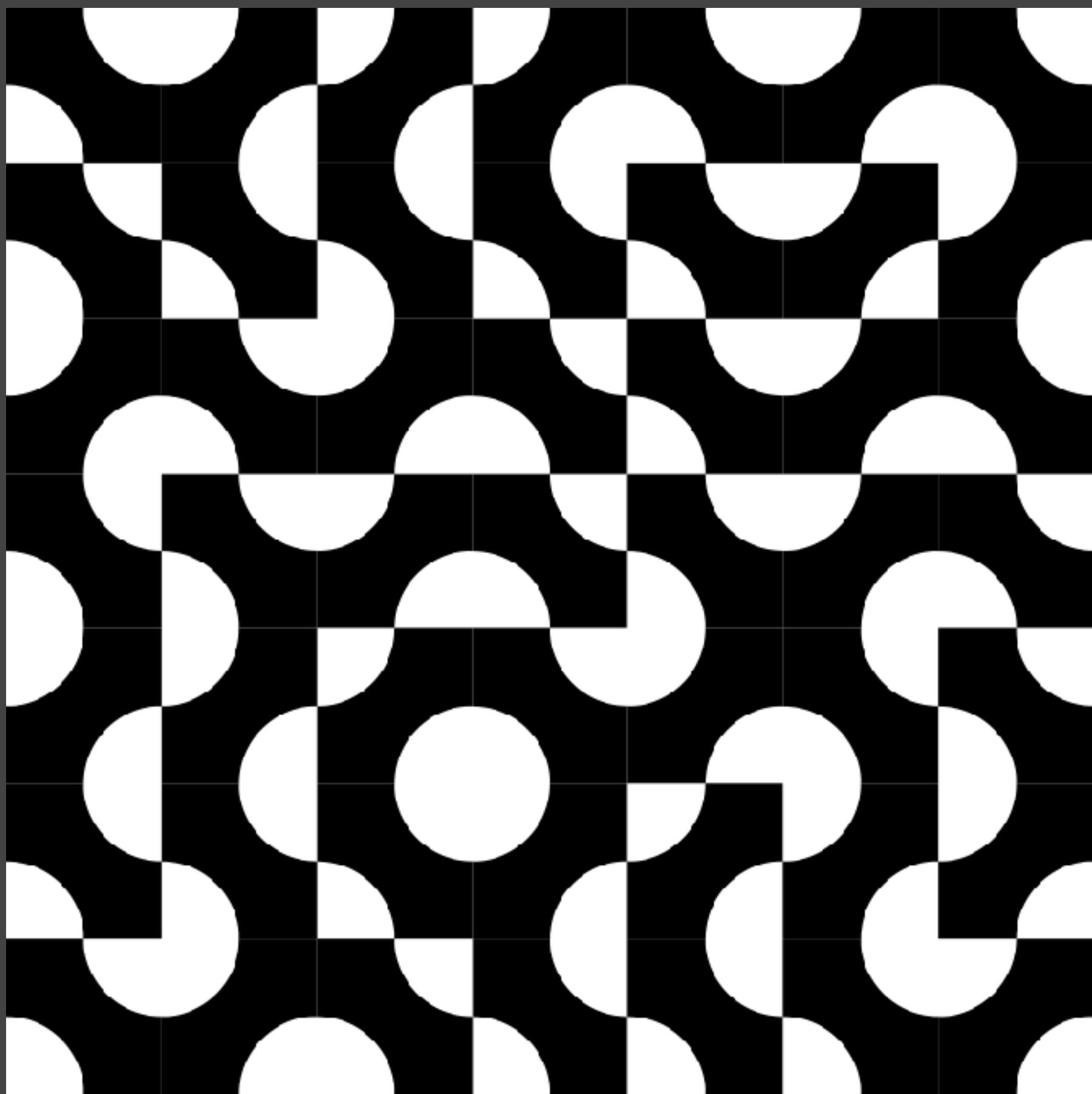
Drawbot Animations

Maurice Meilleur

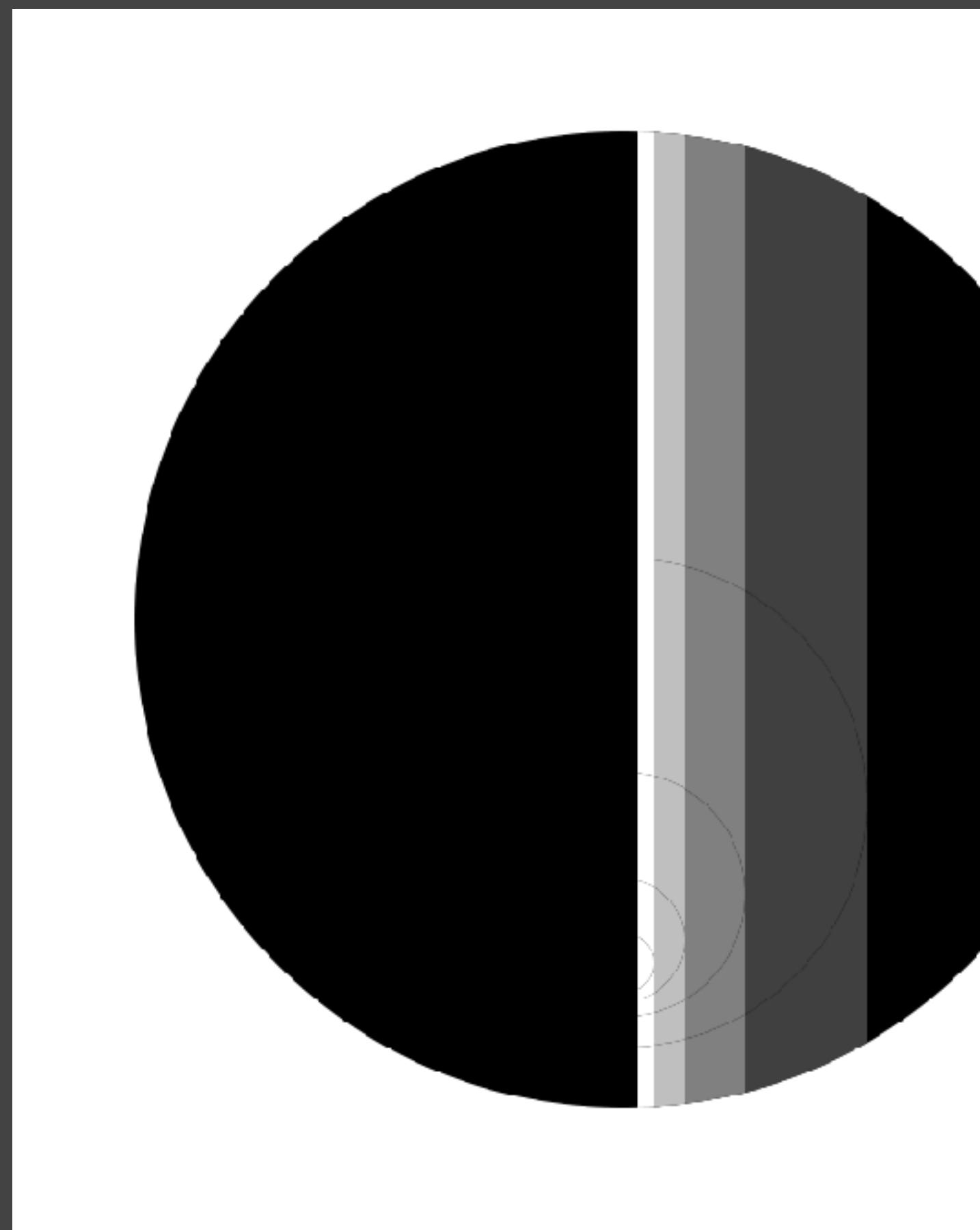
Le Parc transformations



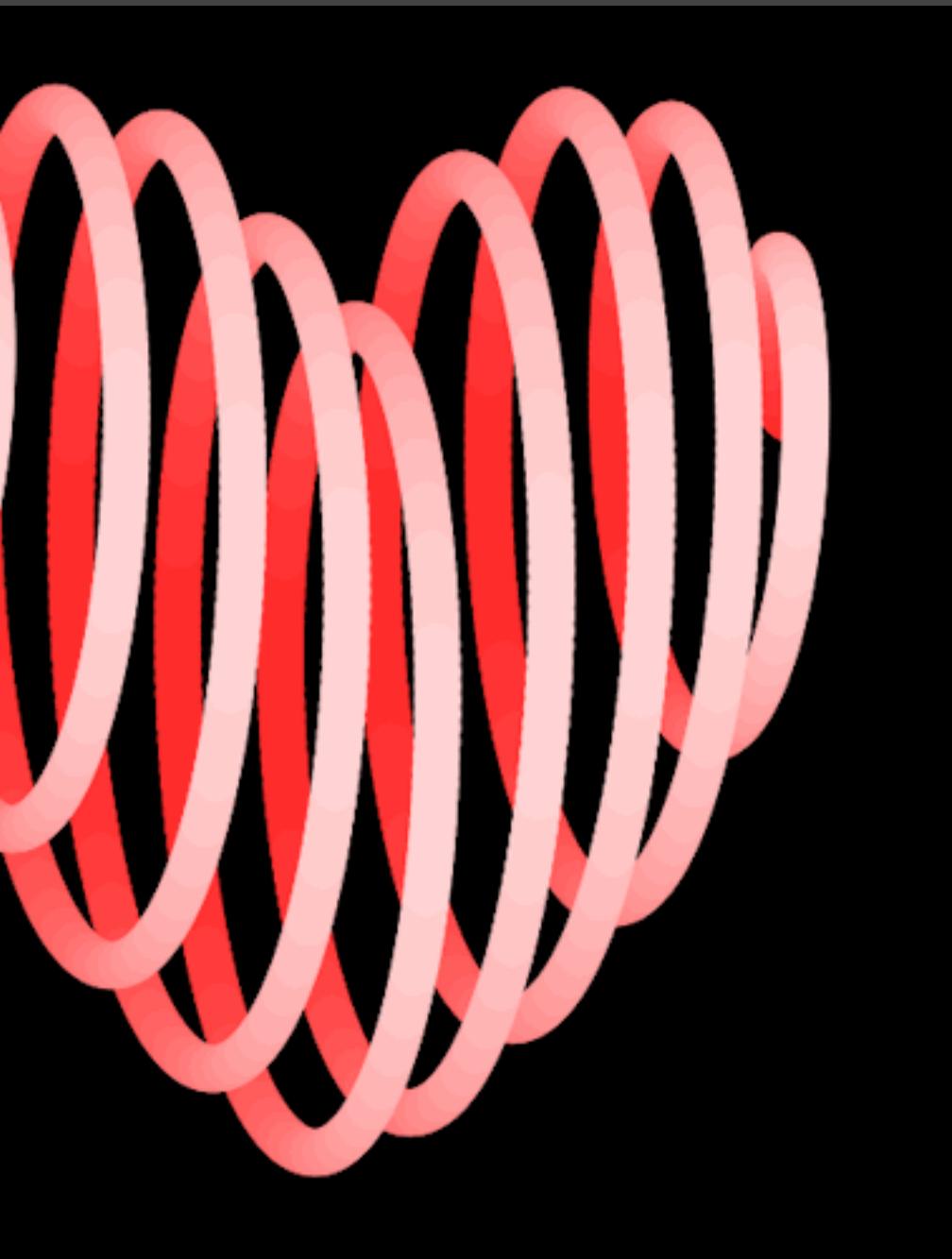
Truchet tiles



Gerstner's Tangential Eccentric

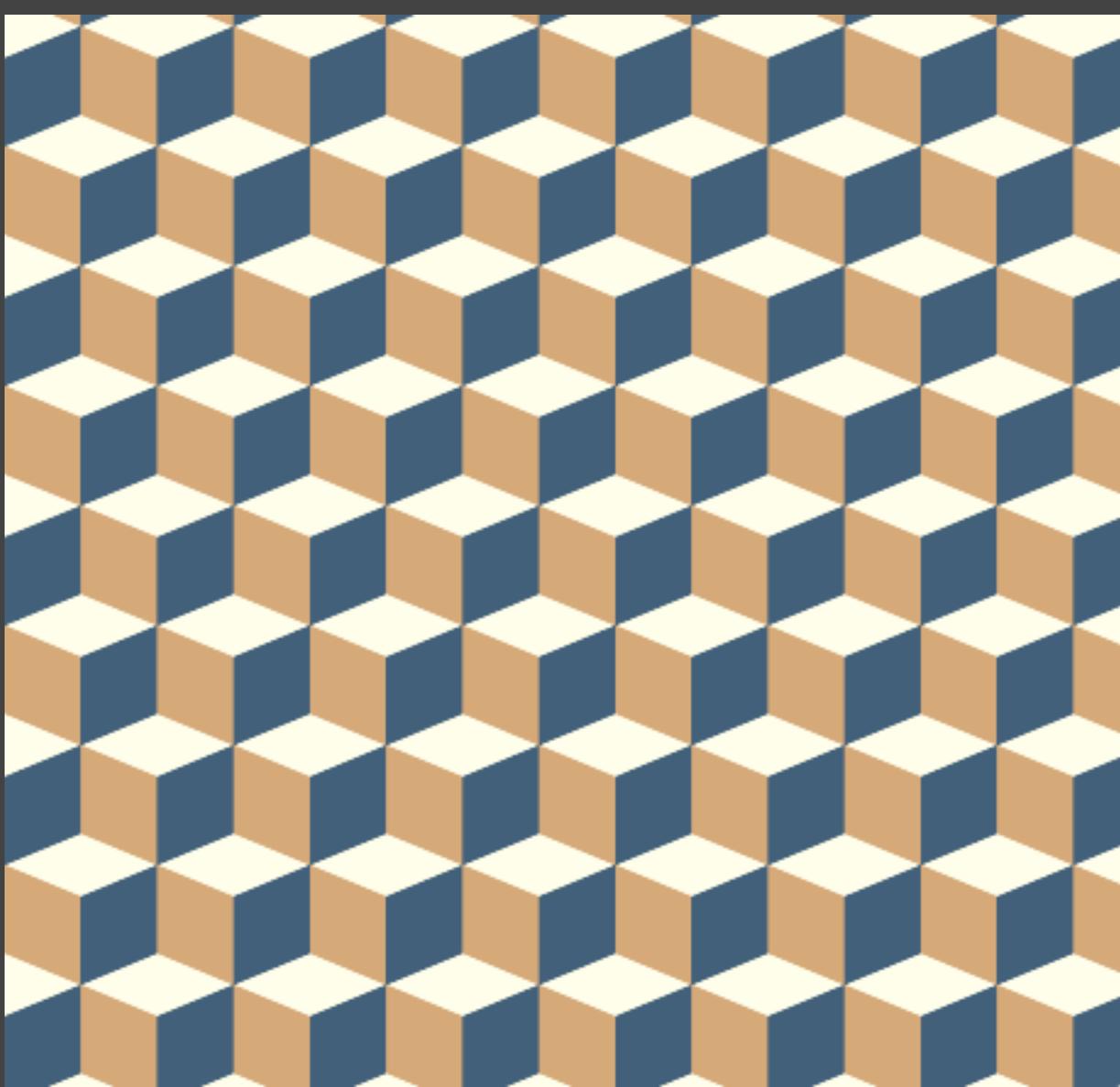
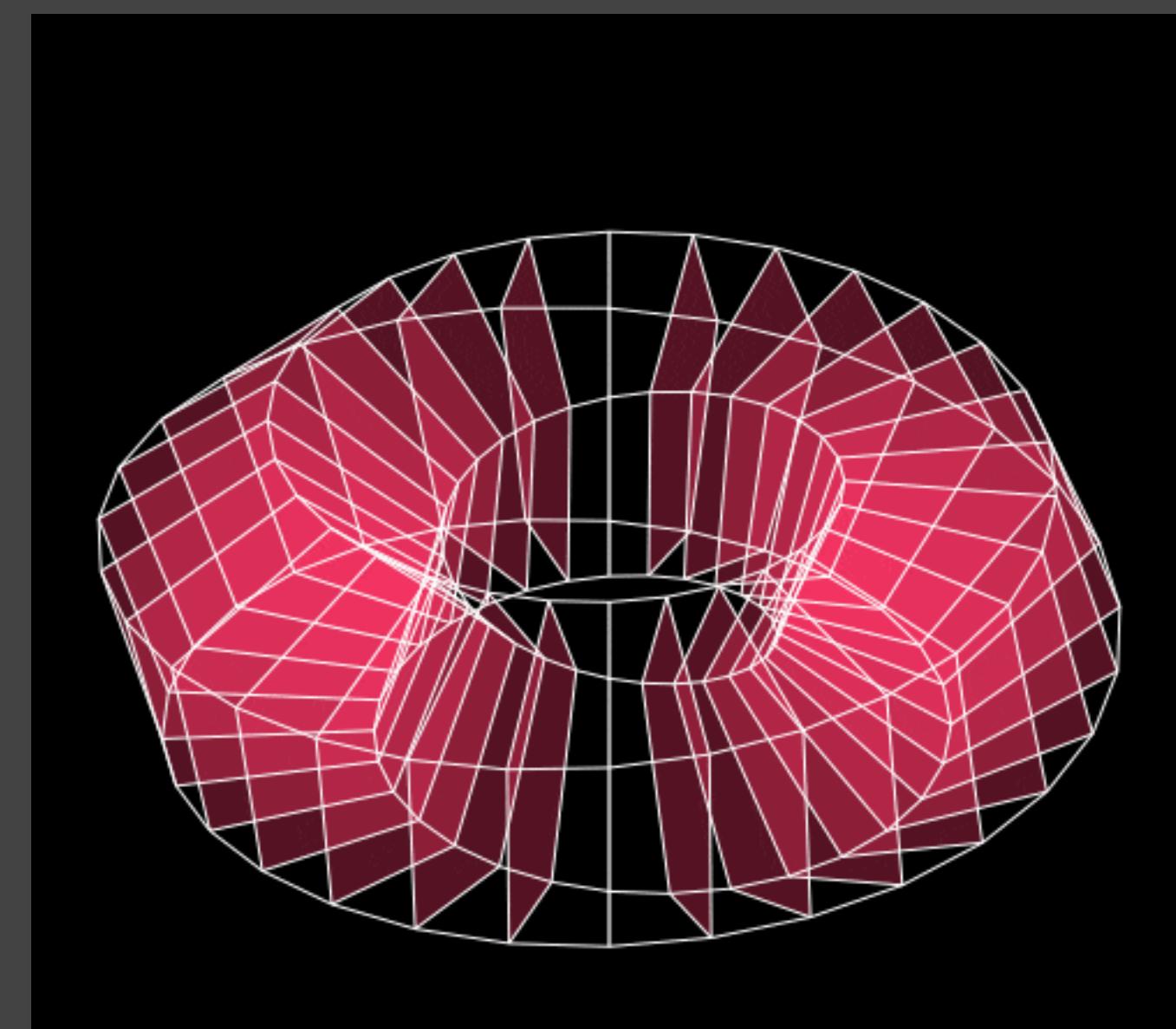
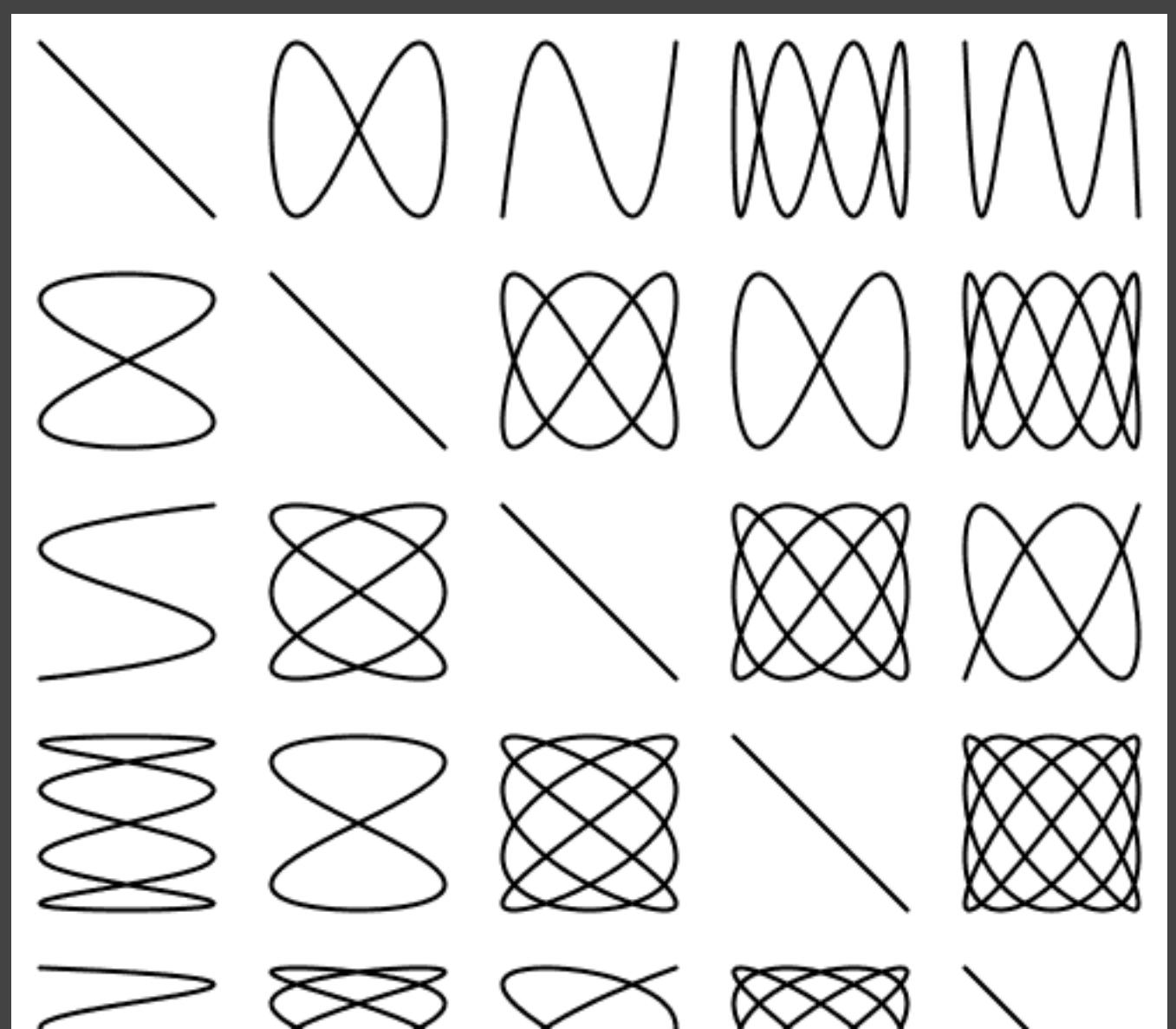
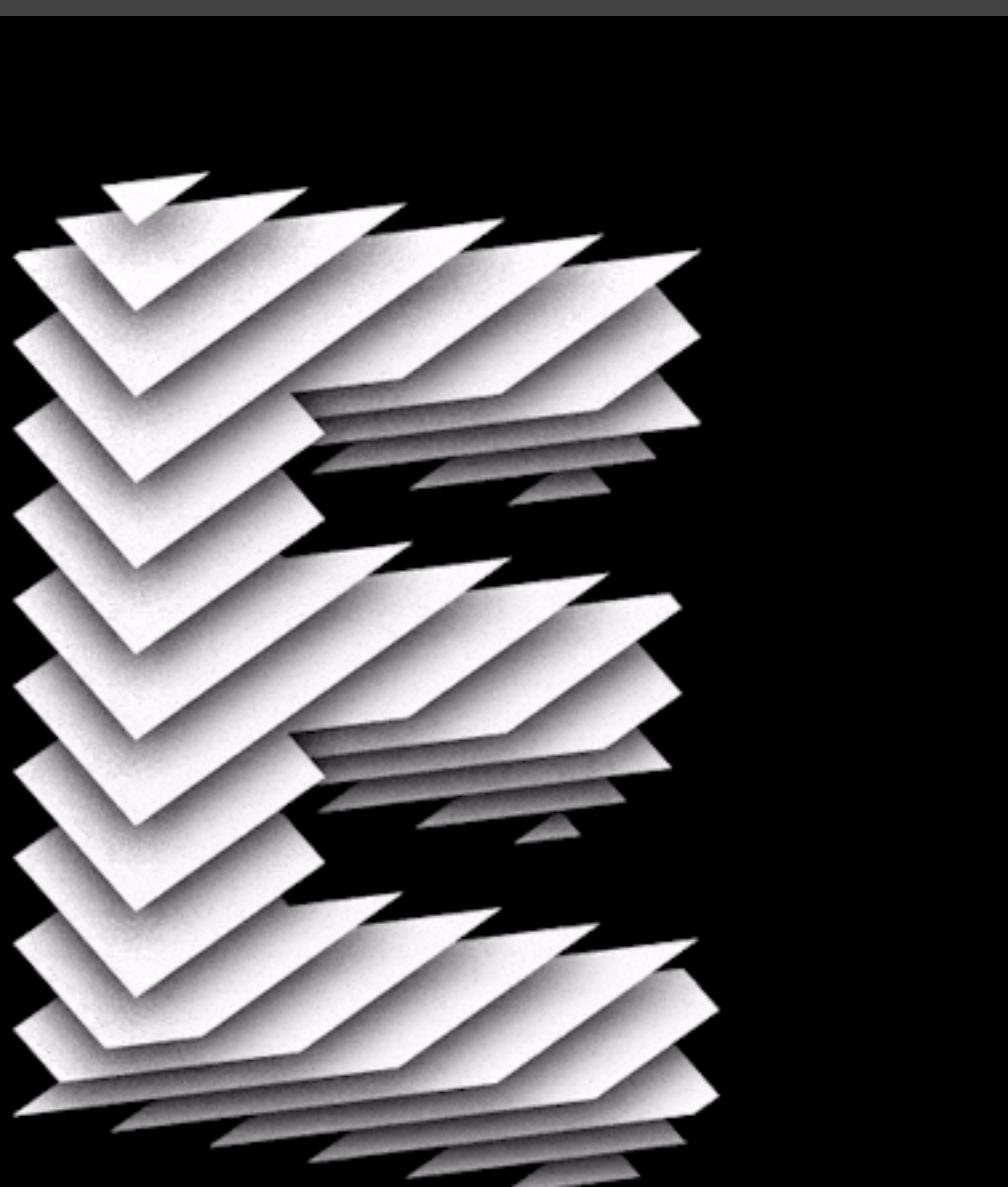
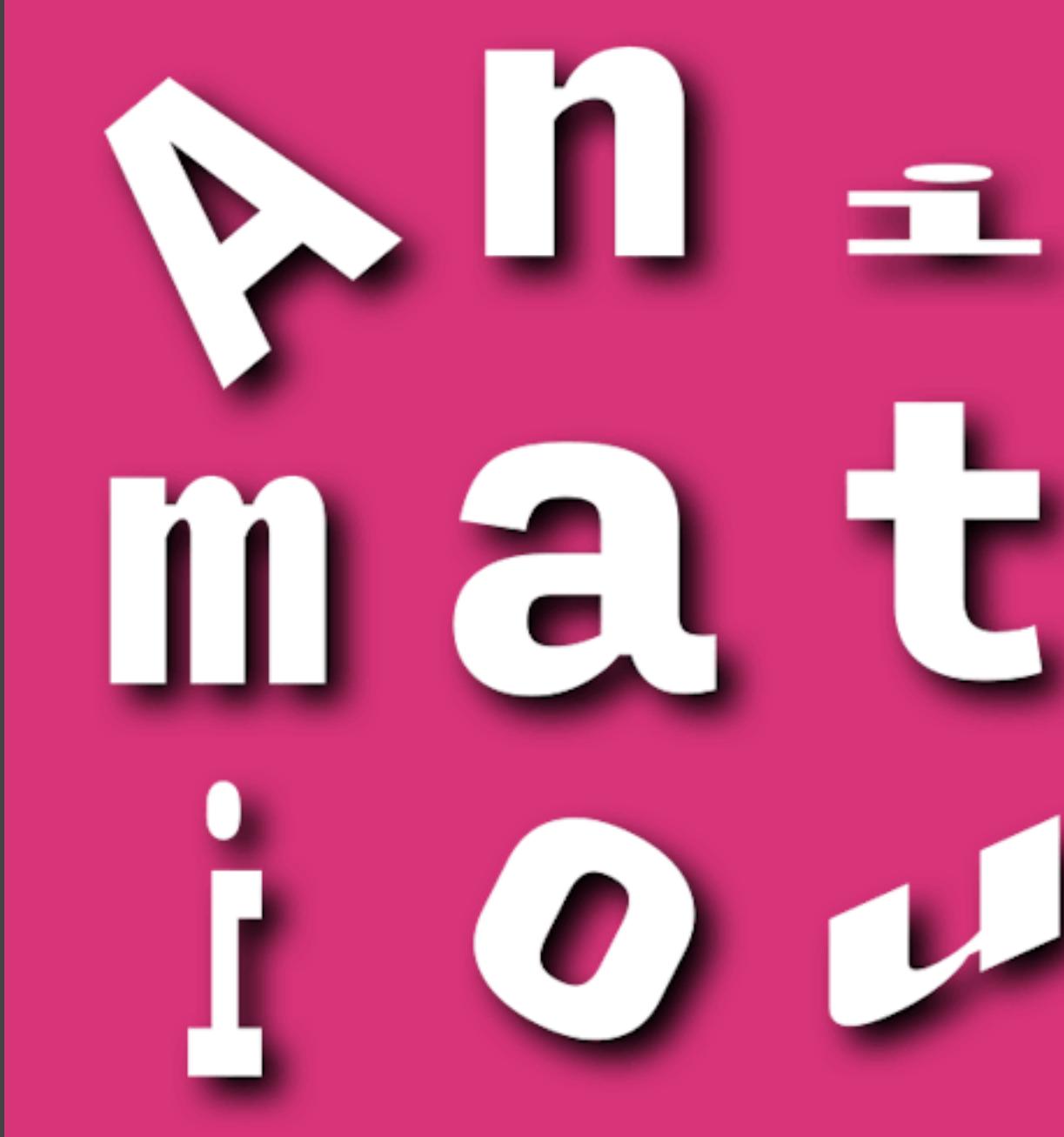


Daily Drawbot



dailydrawbot.tumblr.com

Just van Rossum



Related Projects

Drawbot Skia

Just van Rossum

WIP!

`github.com/justvanrossum/
drawbot-skia`

skia.org

WIP!

github.com/justvanrossum/drawbot-skia

justvanrossum/drawbot-skia

Search or jump to... Pull requests Issues Marketplace Explore

Watch 17 Star 48 Fork 6

Code Issues 3 Pull Requests 0 Issues 3

master 3 branches

justvanrossum Implemented 1 commit

- .github/workflows
- src/drawbot_skia
- tests
- .gitignore
- .pyup.yml
- CHANGELOG.md
- LICENSE.txt
- README.md
- requirements-dev.txt
- requirements.txt
- setup.cfg
- setup.py

README.md

skia.org

Skia Graphics Library

Community Issues Reference Playground ENHANCED

SKIA

Skia Graphics Library

Skia is an open source 2D graphics library which provides common APIs that work across a variety of hardware and software platforms. The graphics engine for Google Chrome and Chrome OS, Android, Flutter, Mozilla Firefox and Firefox OS, and many other products.

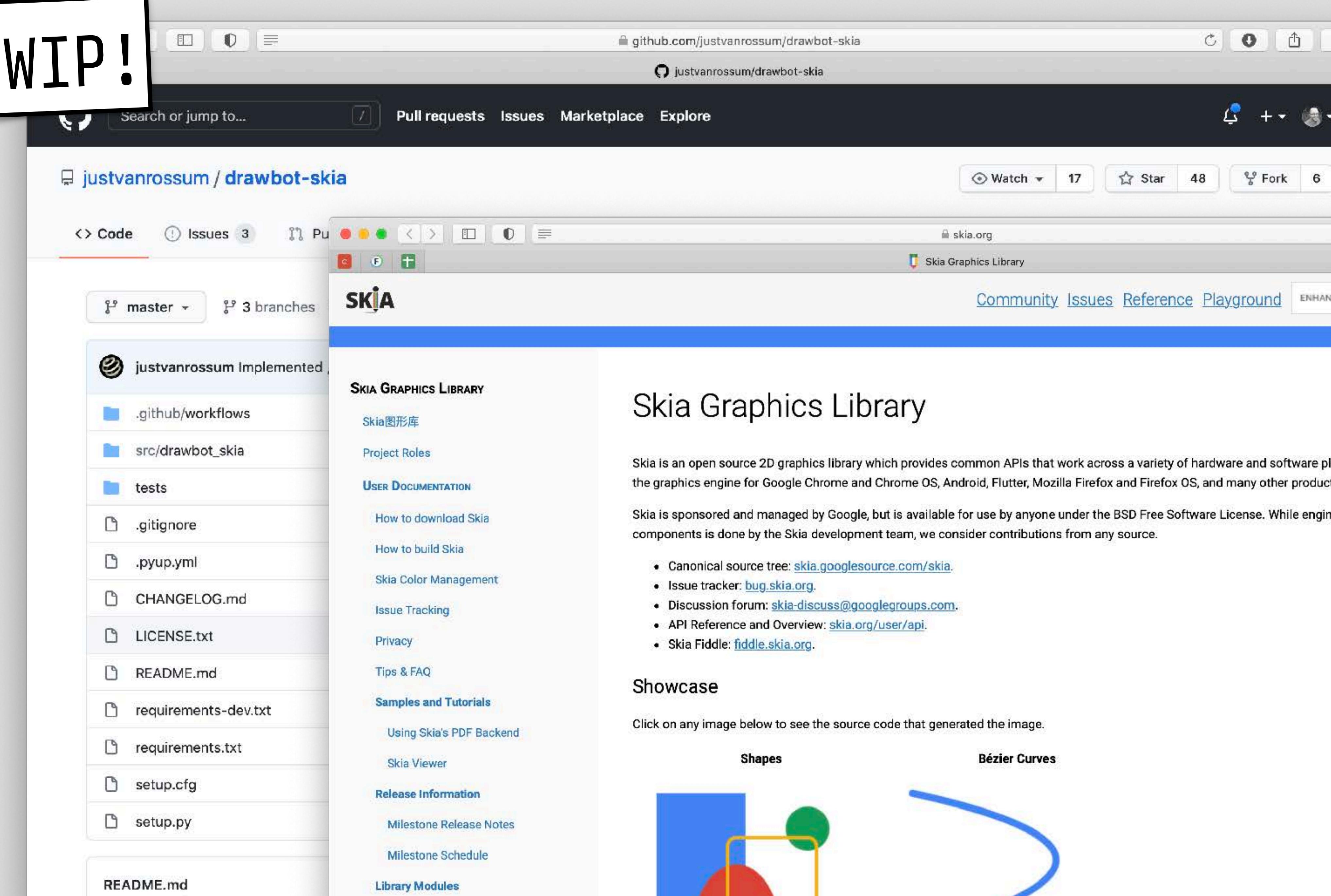
Skia is sponsored and managed by Google, but is available for use by anyone under the BSD Free Software License. While engineering components is done by the Skia development team, we consider contributions from any source.

- Canonical source tree: skia.googlesource.com/skia.
- Issue tracker: bug.skia.org.
- Discussion forum: skia-discuss@googlegroups.com.
- API Reference and Overview: skia.org/user/api.
- Skia Fiddle: fiddle.skia.org.

Showcase

Click on any image below to see the source code that generated the image.

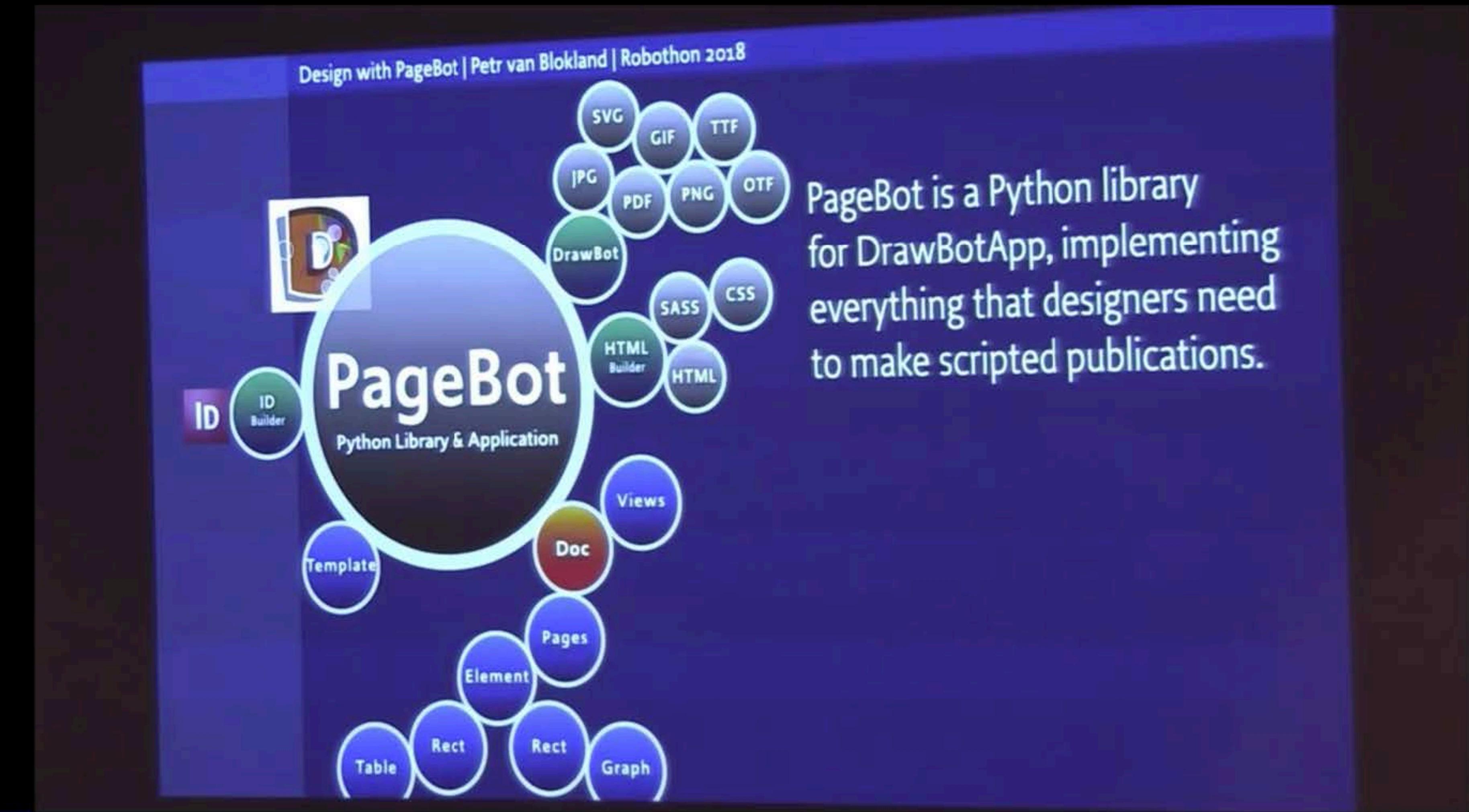
Shapes Bézier Curves



PageBot

Buro Petr van Blokland
+ Claudia Mens

Robothon
2018



Flat

Juraj Sukop

The image shows a web browser window displaying the Flat library documentation at xxxyz.org/flat. The page features a navigation bar with tabs for 'Flat' (which is active) and 'Even'. Below the tabs is a large image of several curved lines in various colors (blue, red, green, yellow, orange) 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 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'. To the right of the documentation, there is a separate code editor window titled 'Tutorial' with the URL xxxyz.org/flat. It contains Python code demonstrating how to use the Flat library to create a document with a circle, text, and an image.

```
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')
```

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

**FOOD
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.