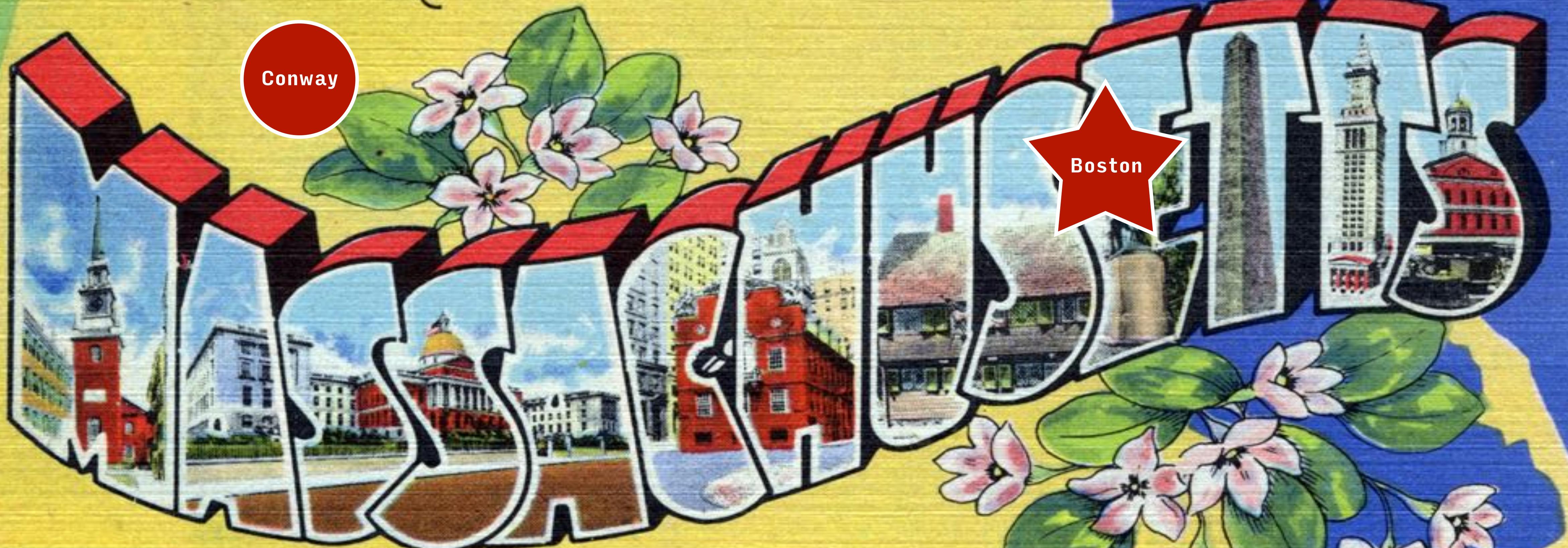


Greetings from

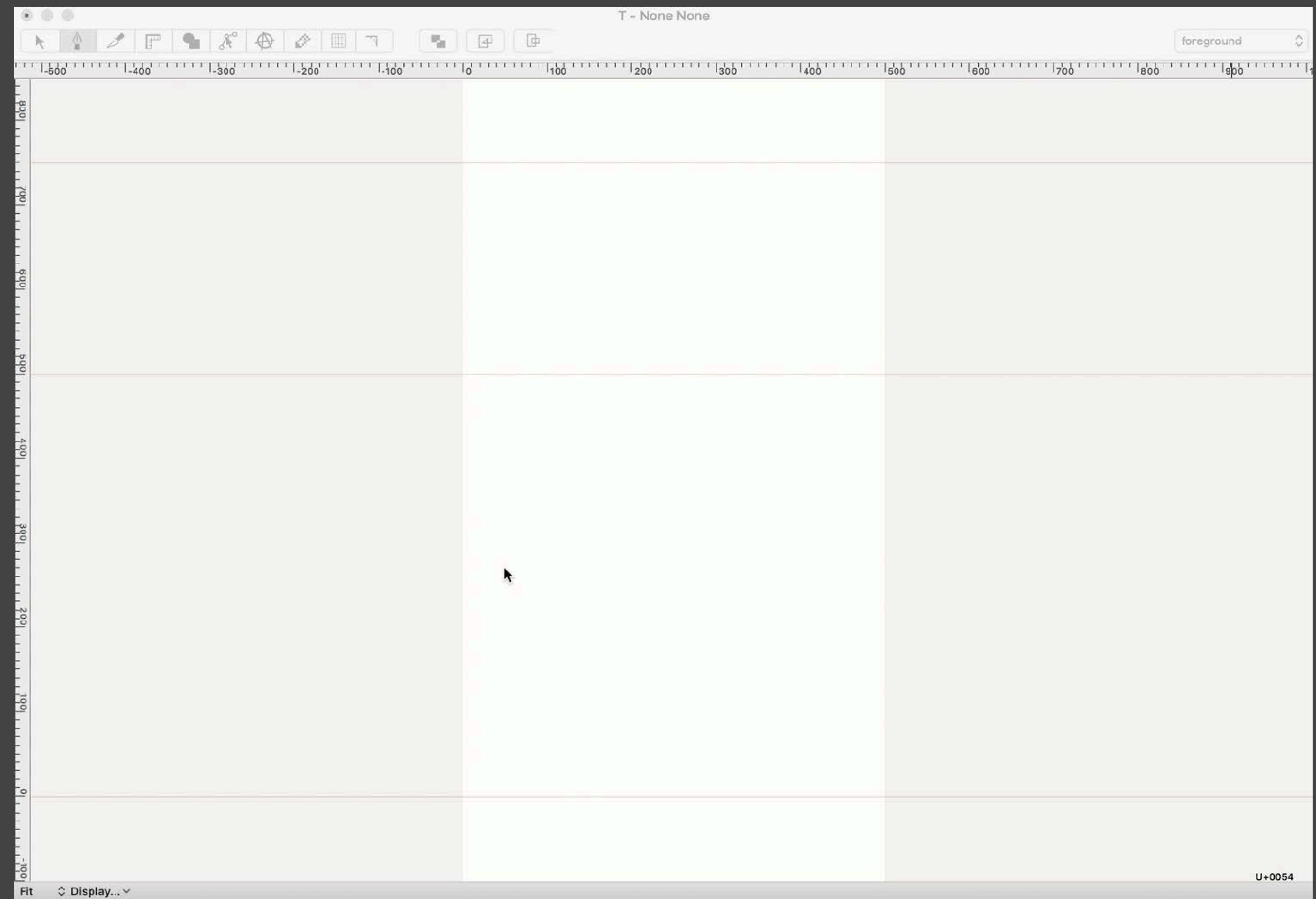


16199

B-3



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". 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." Another red button below the text says "START YOUR MEMBERSHIP TODAY!". Underneath it, a smaller text box says "Or give a gift subscription to your favorite font lover!" with a small gift icon. At the bottom of the page, there's a red banner with the text "THIS MONTH'S FONT" in white. On the left side of this banner, there's a section for "Klooster Thin" with the subtitle "Taking the uncial in new directions". On the right side, there are two buttons: a red one that says "JOIN TODAY! \$6" and a black one that says "MORE". The background of the page has a light blue and white grid pattern.

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!

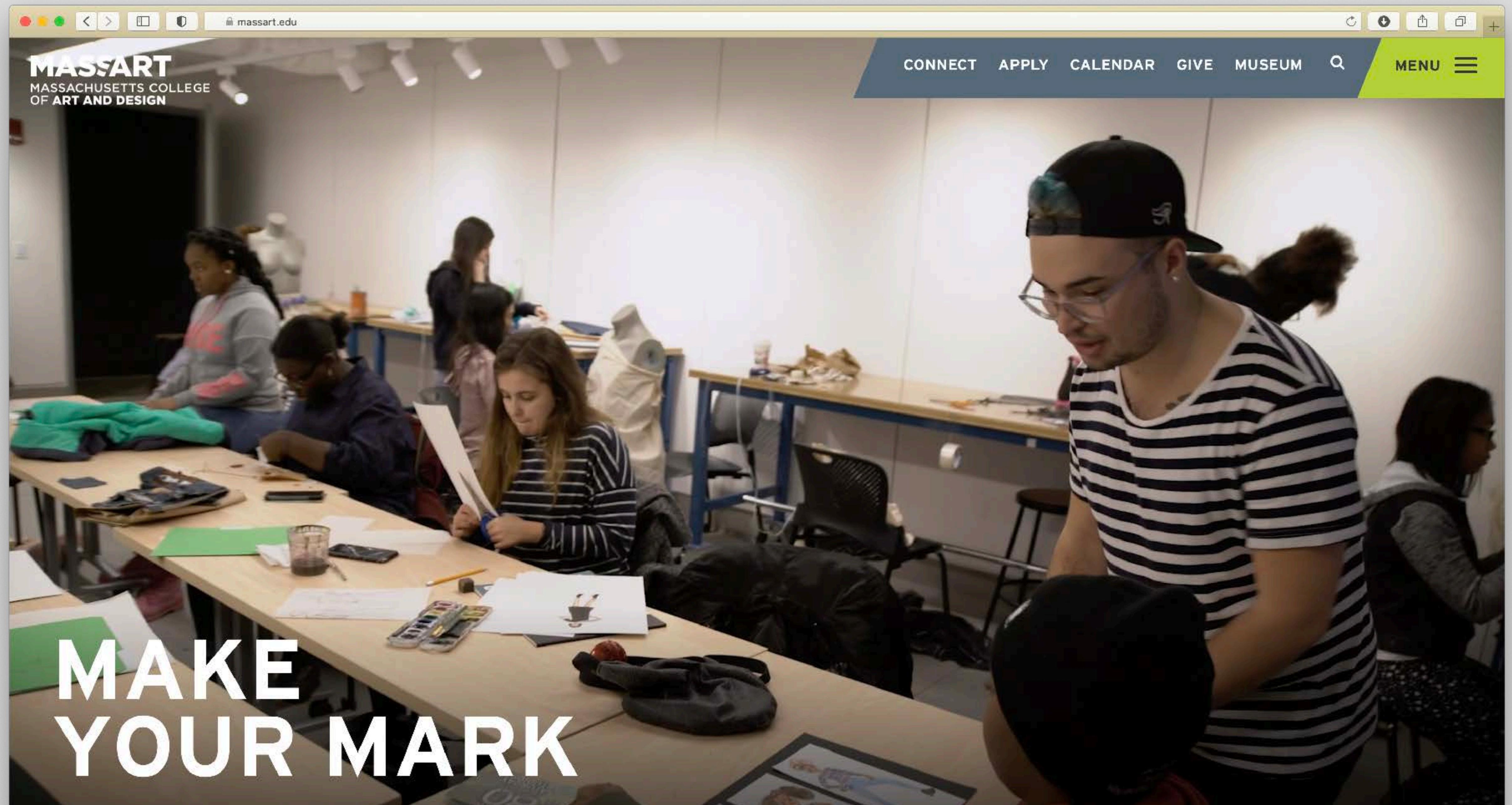
Or give a gift subscription to your favorite font lover!

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.

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 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")

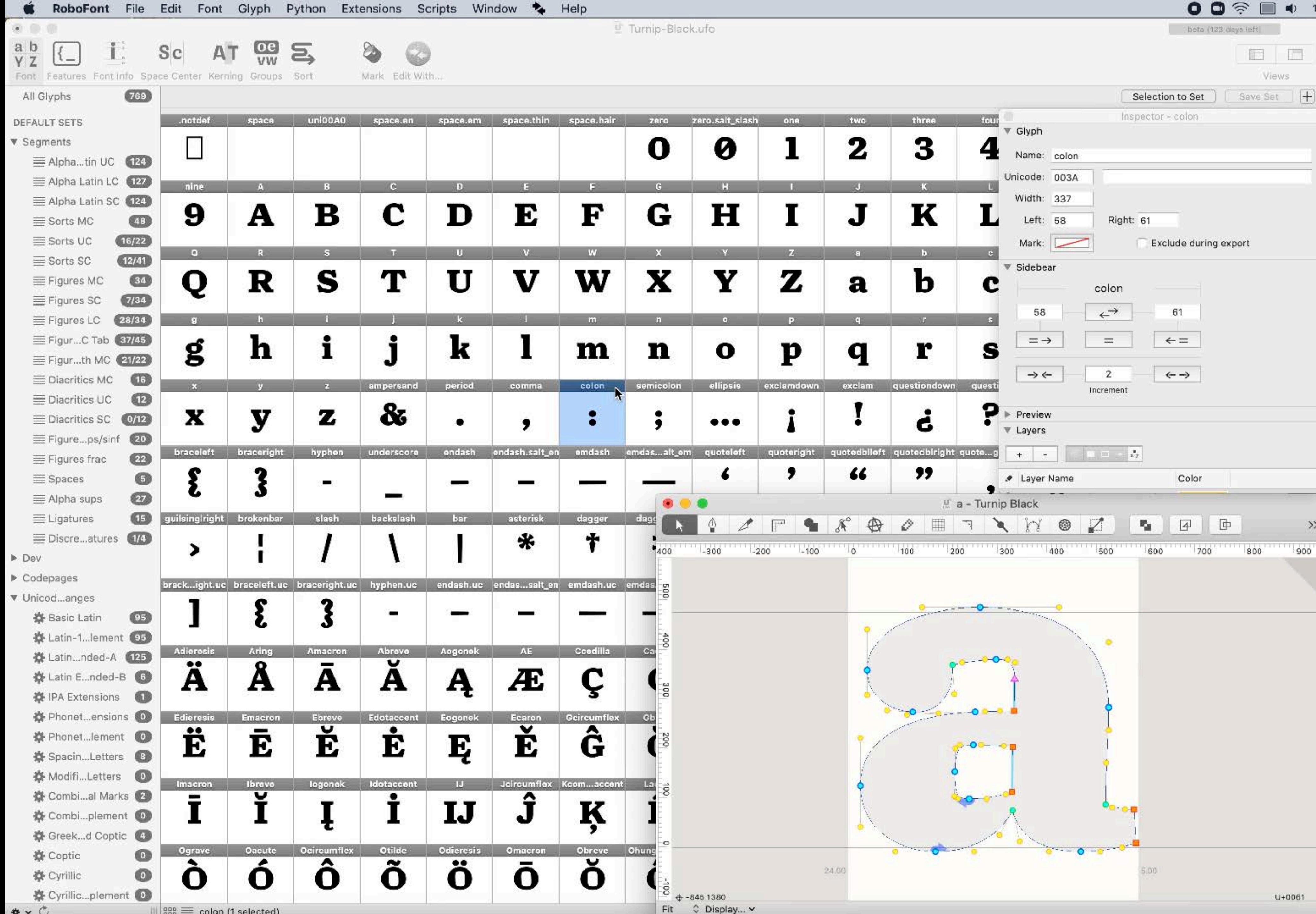
    def mouseDragged(self, info):
        point = info["point"]
        positionSymbol = unichr(8982)
        deltaPoint = info["delta"]
        angle = math.degrees(math.atan2(deltaPoint.y, deltaPoint.x))
        distance = math.hypot(deltaPoint.x, deltaPoint.y)
```

Input
input.fontbureau.com

free download!

RoboFont

Frederik Berlaen



MERIT BADGE

IS A VARIABLE
COLOR FONT.



Just van Rossum

Frederik Berlaen

DrawBot

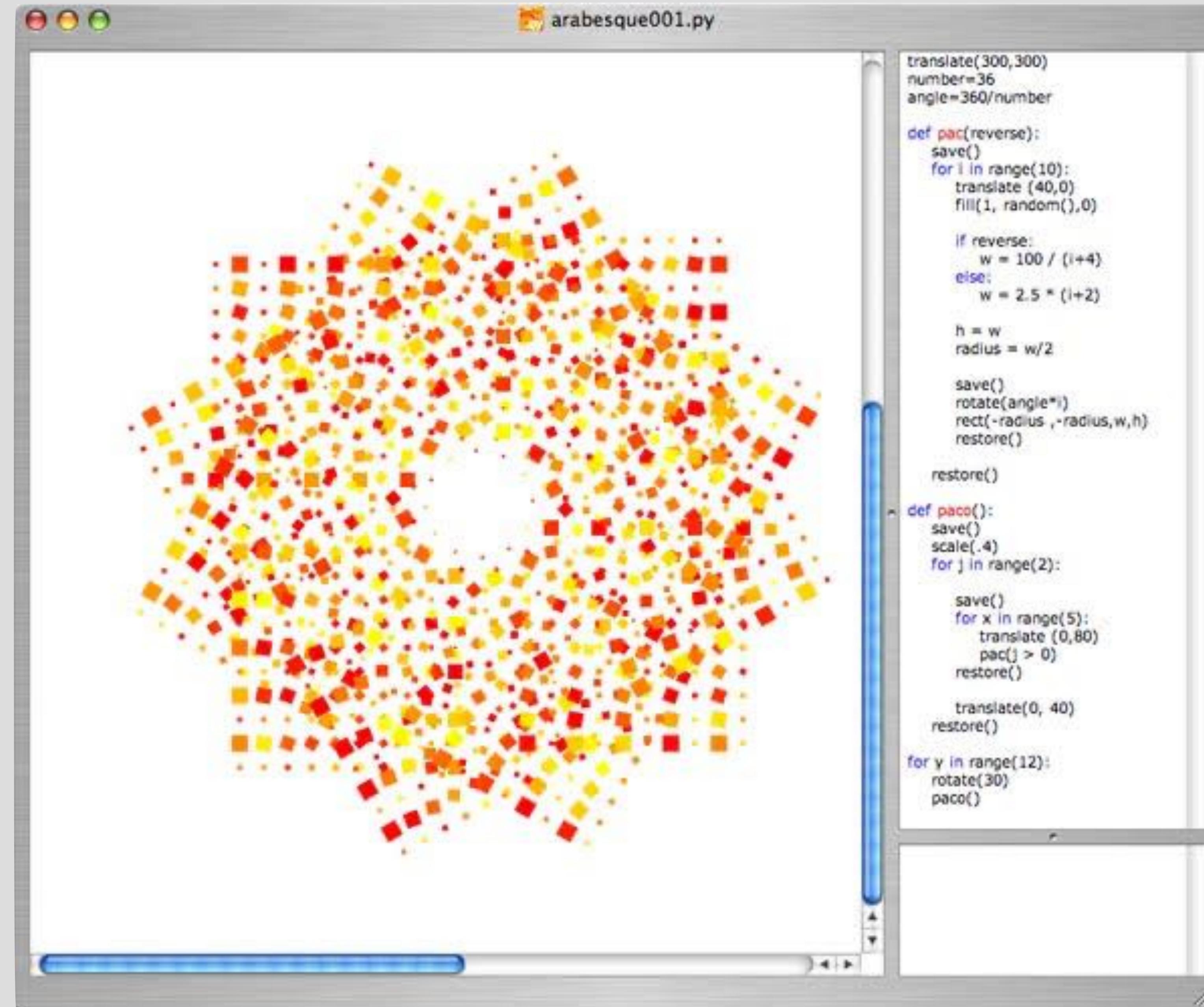


drawbot.com
[@drawbotapp](https://twitter.com/drawbotapp)



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.**

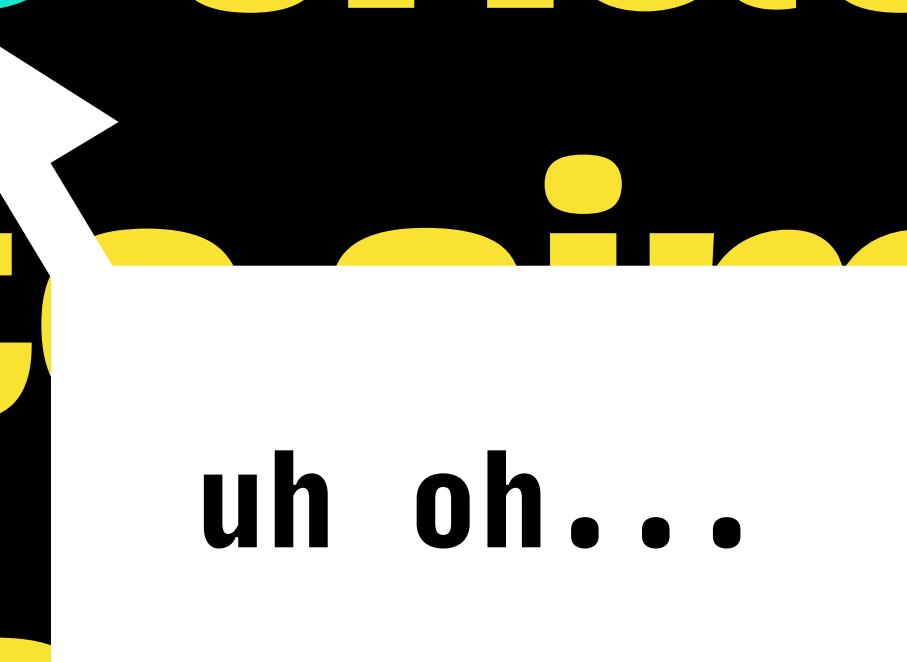


BSD license

**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.**

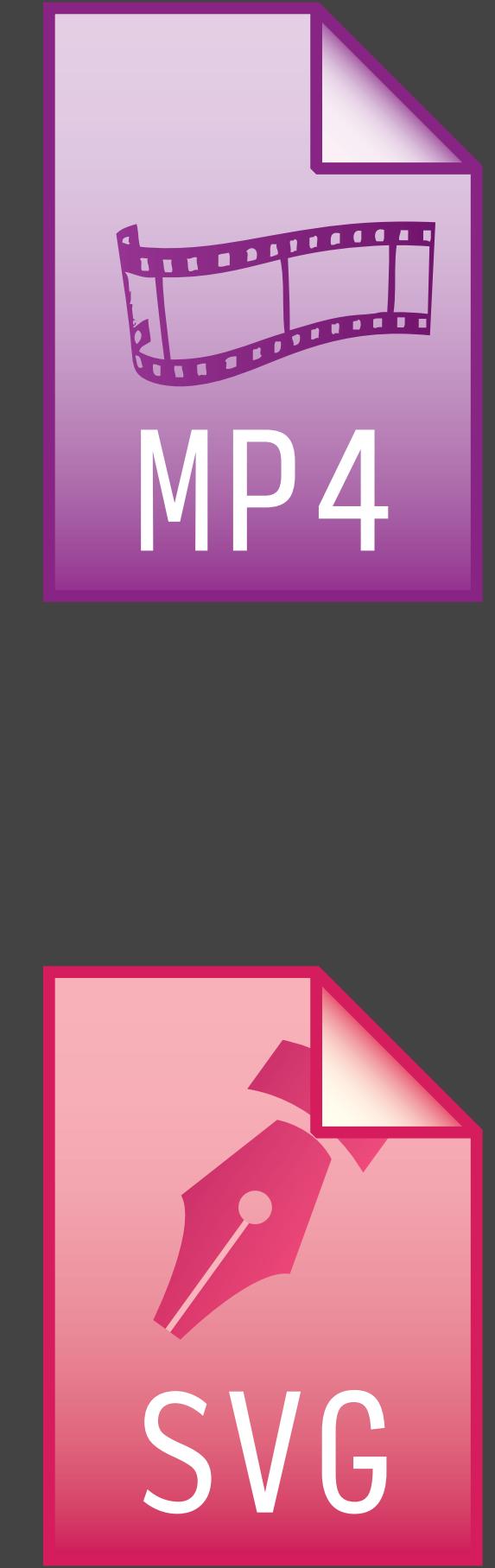


uh oh...

**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 displayed prominently. 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 prominently. 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).

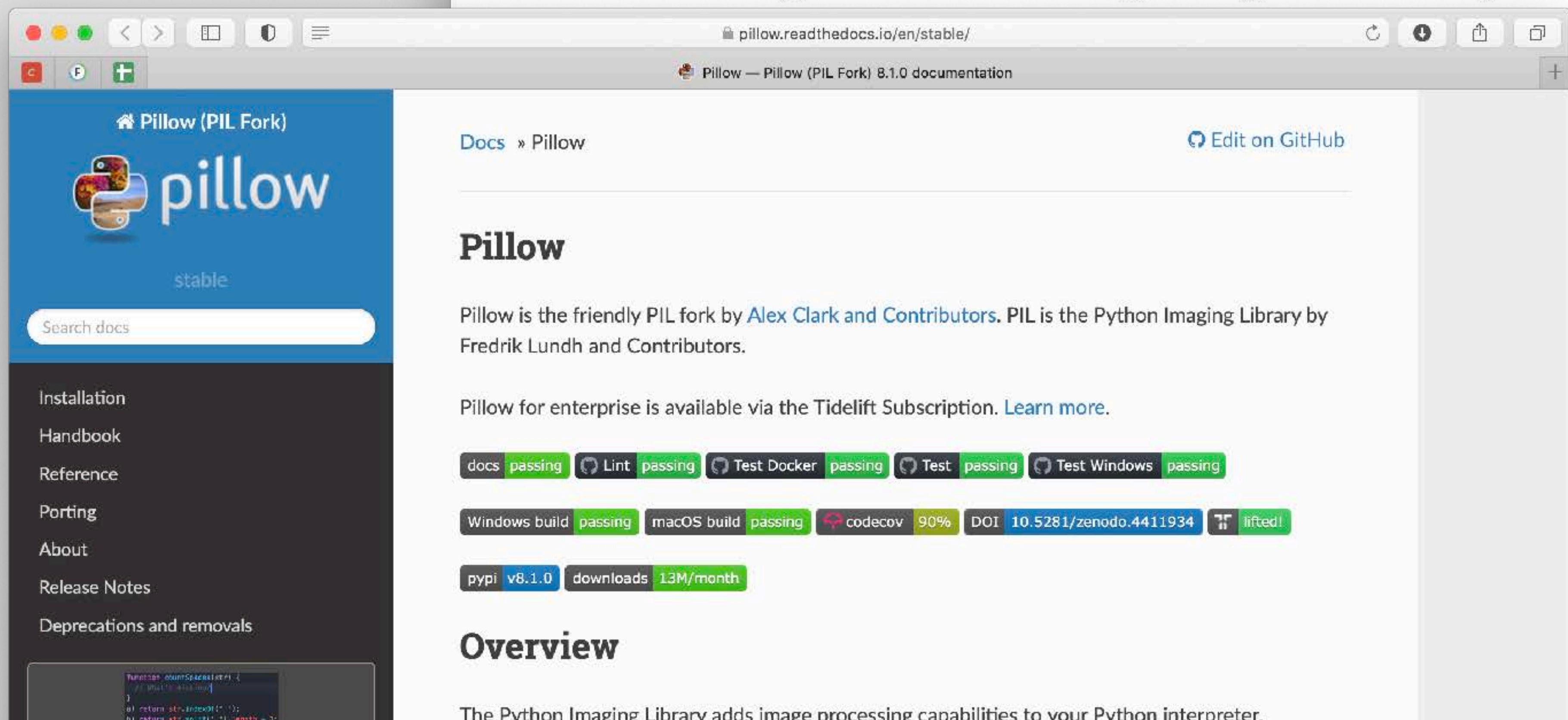
The screenshot shows the GitHub repository page for python-pdfkit. The repository name is "JazzCore/python-pdfkit". The page includes tabs for Code, Issues (83), Pull requests (19), Actions, Projects, and Wiki. The "Code" tab is selected, showing the master branch (1 branch, 0 tags). A list of commits is shown, starting with a merge pull request from alanhamlett. Other commits include support for Python 3, fixes for unittest.TestCase.assertRegex, and updates to Travis CI configuration.

Commit	Description	Date
alanhamlett Merge pull request #119 from ala... c15ed32	Merge pull request #119 from ala... support unicode keyword on Python 3	on Oct 22, 2018
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" written in large blue letters between them. Below the logo is a navigation bar with links for News, Download, Documentation, Contact, and Examples. The main content area is titled "Pycairo" in orange and contains text explaining that Pycairo is 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". On the left, there is a sidebar with links for Installation, Handbook, Reference, Porting, About, Release Notes, and Deprecations and removals. The main content area starts with a heading "Pillow" and a brief description: "Pillow is the friendly PIL fork by [Alex Clark and Contributors](#). PIL is the Python Imaging Library by [Fredrik Lundh and Contributors](#)". It also mentions that Pillow for enterprise is available via the Tidelift Subscription and provides links for documentation status (docs passing), Lint (passing), Test Docker (passing), Test (passing), and Test Windows (passing). Below this are links for Windows build (passing), macOS build (passing), codecov (90%), DOI (10.5281/zenodo.4411934), and Tidelift (lifted). At the bottom, it shows pypi v8.1.0 and downloads 13M/month. The footer includes links for releases and snapshots, and a note about bindings for cairo-ocaml and cookbook/convertcopyandback. The page was last edited on Fri May 5 00:10:48 2017.

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. The bottom navigation bar includes "Console" and "Errors" tabs.

```
sketch_181215a
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19



Processing/p5.js



DrawBot



- Cross-platform
- MacOS

P Processing/p5.js

D DrawBot

- Cross-platform
- 2D+3D (WebGL)
- MacOS
- 2D

P Processing/p5.js

D DrawBot

- Cross-platform
- 2D+3D (WebGL)
- Interactive
- MacOS
- 2D
- Hardly interactive

P Processing/p5.js

D DrawBot

- Cross-platform
- 2D+3D (WebGL)
- Interactive
- Big community
- MacOS
- 2D
- Hardly interactive
- Small community

P Processing/p5.js

D DrawBot

- Cross-platform
- 2D+3D (WebGL)
- Interactive
- Big community
- Web integration

- MacOS
- 2D
- Hardly interactive
- Small community
- Print-ready output

P Processing/p5.js

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

D DrawBot

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

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

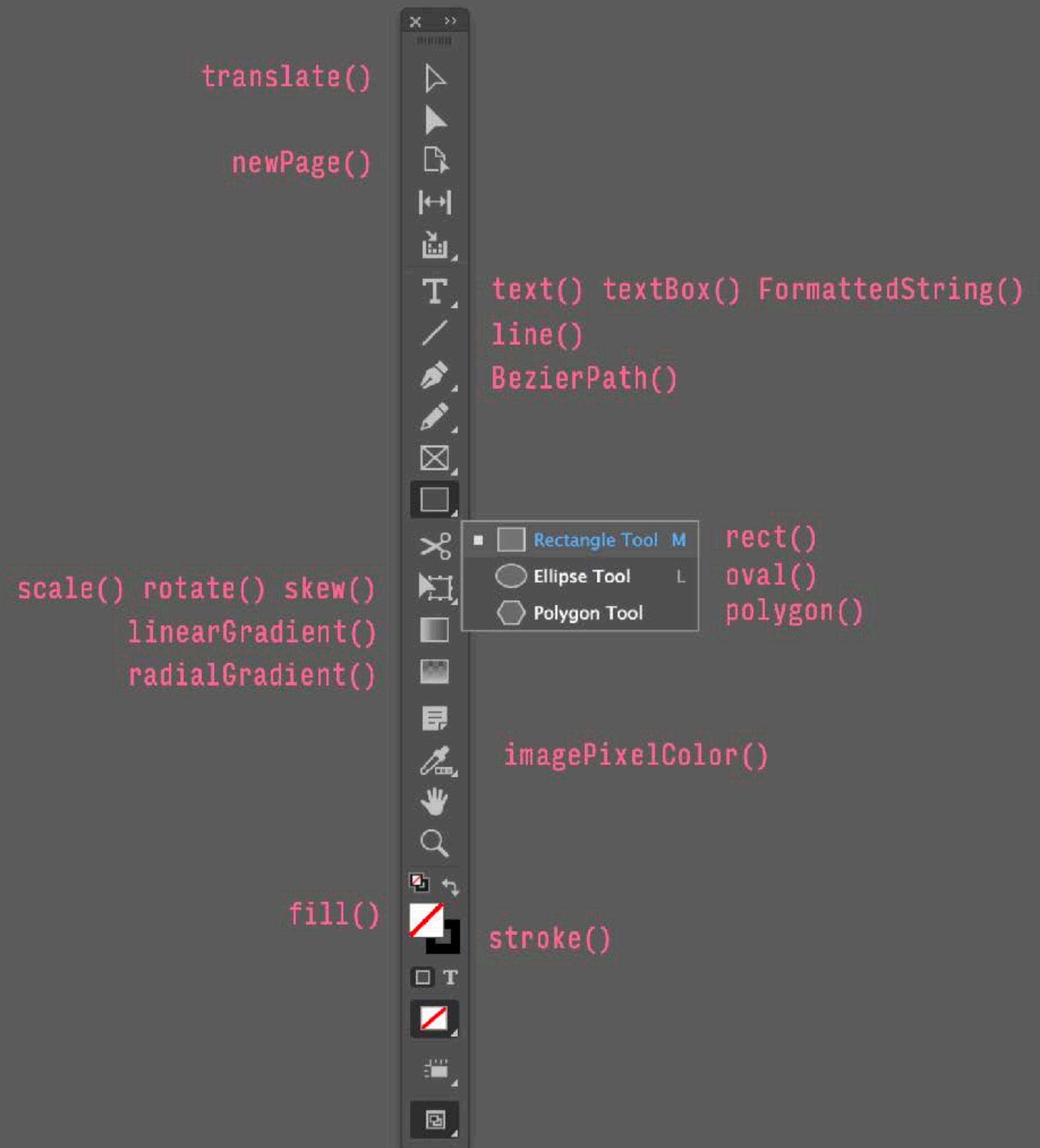
POMFRET Specimens djr.com | 2

BARRANCA
STOPBACK

POMFRET Specimens djr.com | 3

EXPLAINER
RECREANT

This screenshot shows the Adobe InDesign 2020 interface. The main workspace displays a document spread with text in the 'Pomfret' font. The left side of the spread contains blue text, while the right side contains red text. A character panel is open, showing settings for the selected font and style. The right margin of the spread features a vertical grid of lines. The bottom of the workspace contains two specimen panels for the 'POMFRET' font, labeled 'Specimens' and showing examples of the font's range, including 'BARRANCA STOPBACK' in black and red. The top menu bar includes options like 'File', 'Edit', 'Object', 'Type', 'Layout', 'View', 'Color', 'Script', and 'Help'. The bottom status bar shows page numbers, a working color profile, and an error count.



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
hhahohohoo
hbhbhoboboo
hhchohocoo
hhdhohodoo
hhehohoeoo
hhFhohofoo
hhchohocoo
hhhhohohoo
hhihohoiioo

hbjhohojoo
hbkhohokoo
hbilhoholoo
hbphohophoo
hbphohondoo
hbphohohoooo
hbphohopoo
hbphohohoooh
hbphohoroo
hbshohosoo

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

Adidas N3XT L3V3L

Grand Army

Andy Clymer

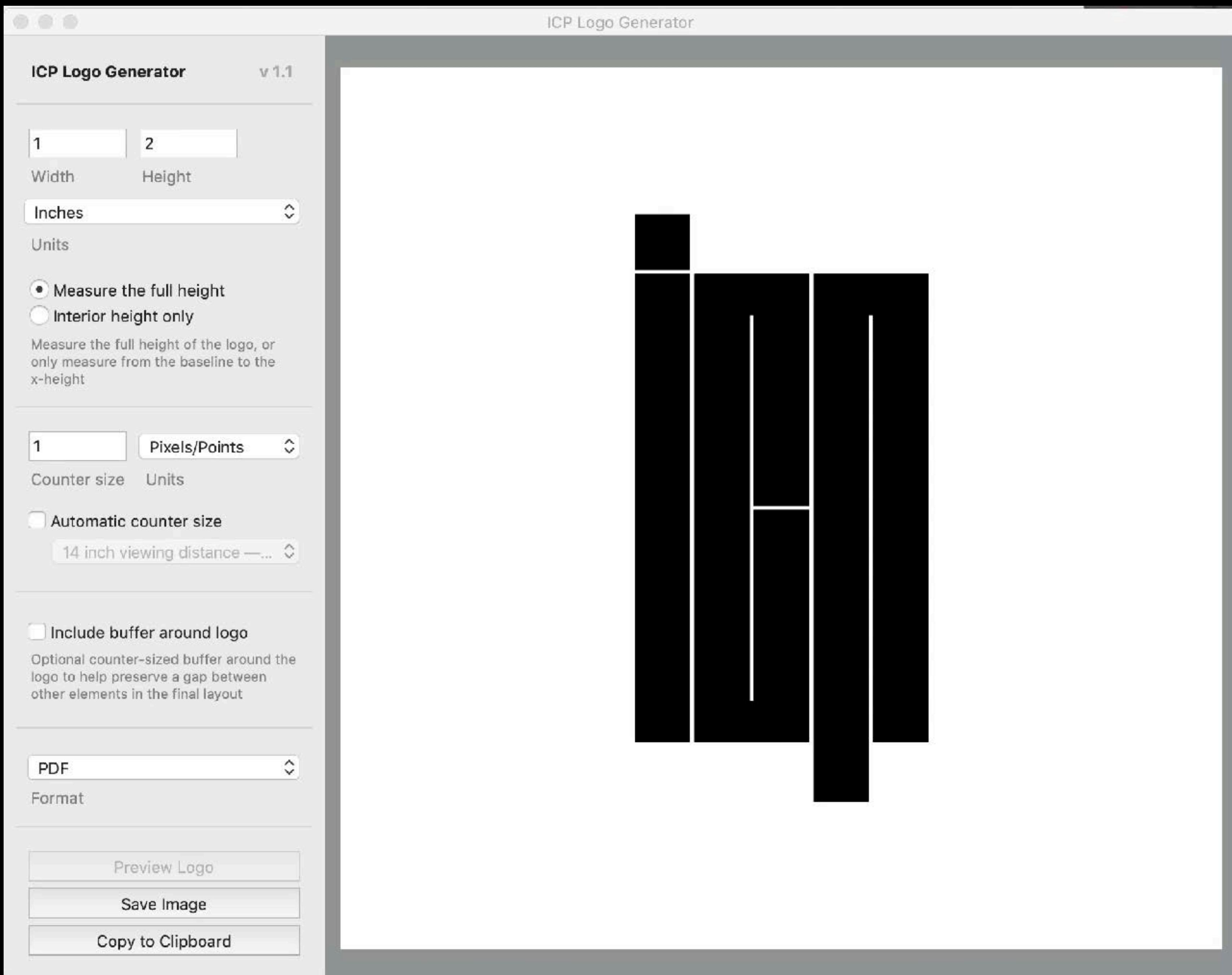


grandarmy.com/projects/n3xt-l3v3l

International Center of Photography, Philadelphia

Pentagram

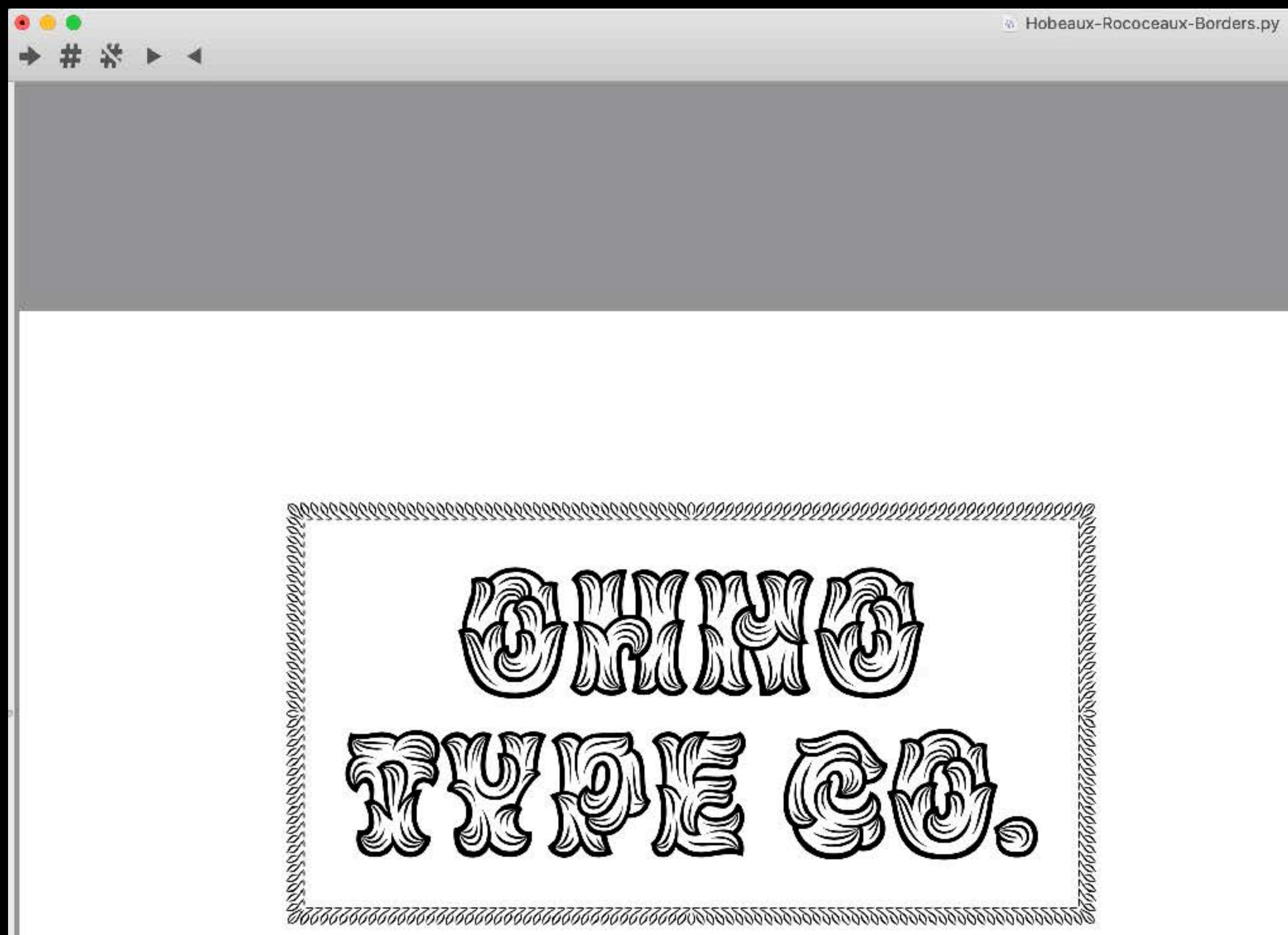
Andy Clymer, Yours Truly



Hobeaux Rococeaux Borders

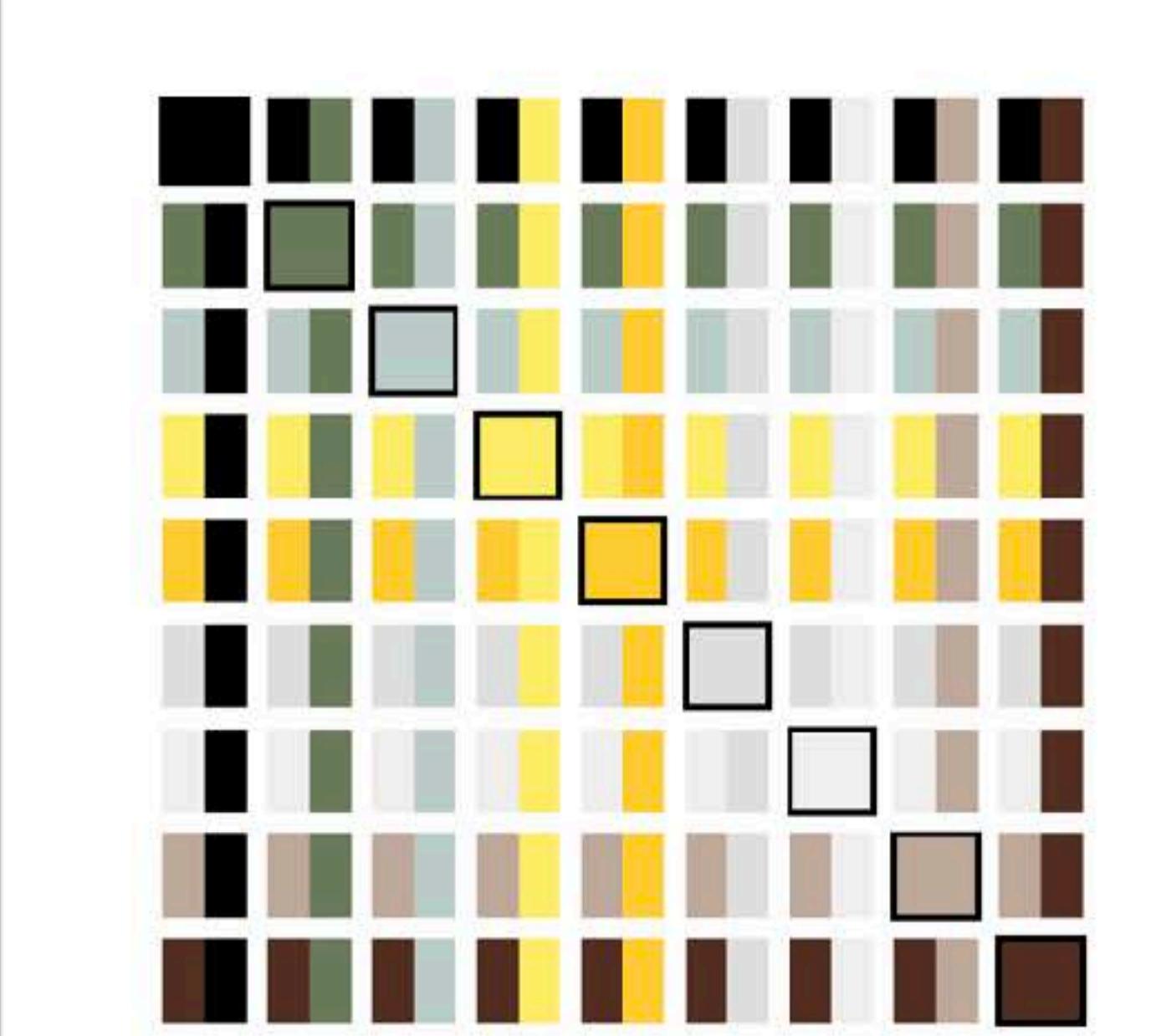
ohnotype.co/fonts/hobeaux-rococeaux

James Edmondson



The screenshot shows a Mac OS X terminal window titled "Hobeaux-Rococeaux-Borders.py". The window contains a code editor with a file named "dictElements". The code is a Python script that generates decorative borders using the "Hobeaux Rococeaux" font.

```
# 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
```



```
Untitled 3

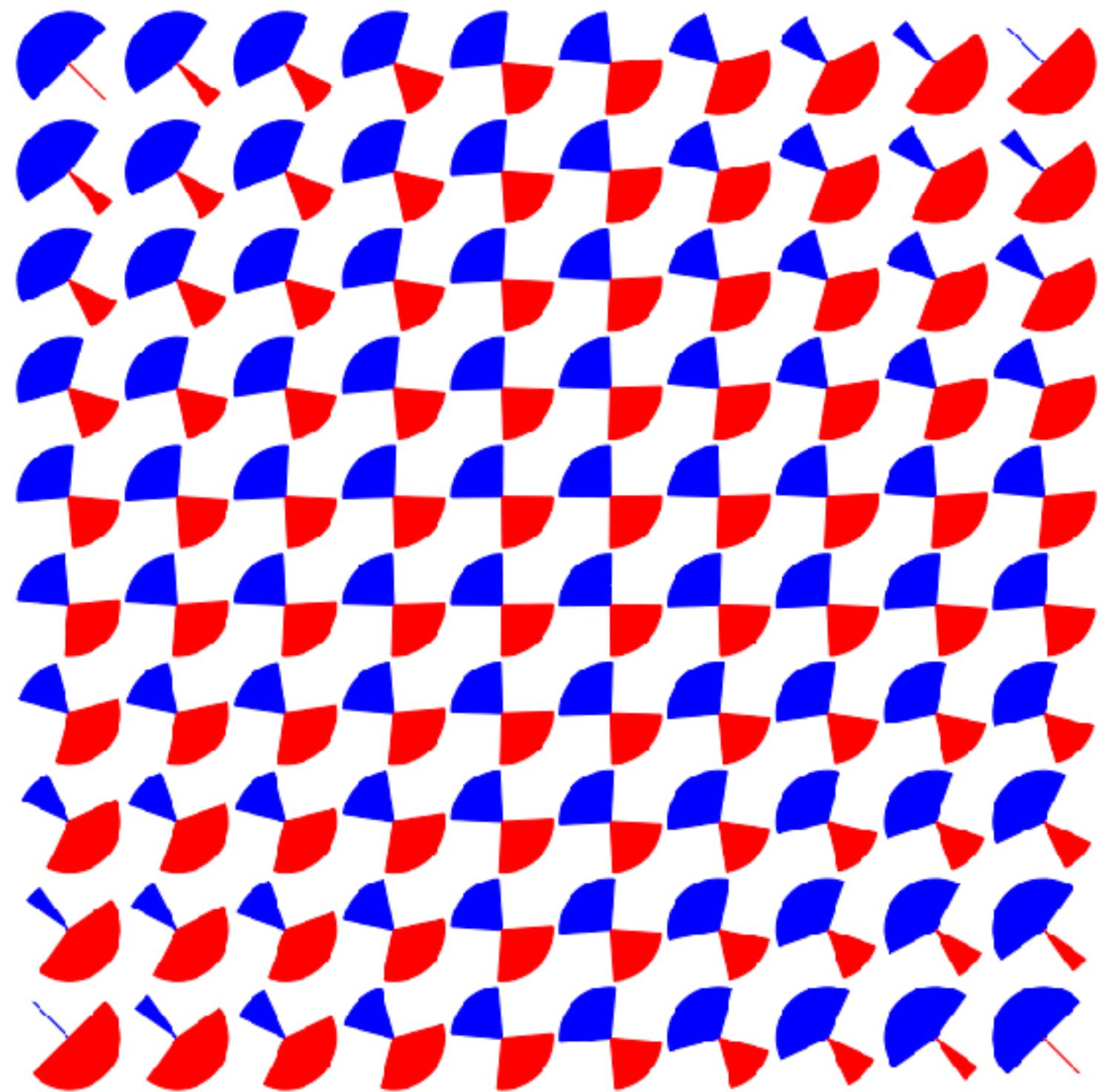
colorSpace(colorspace)
translate(startX,startY)

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)
```

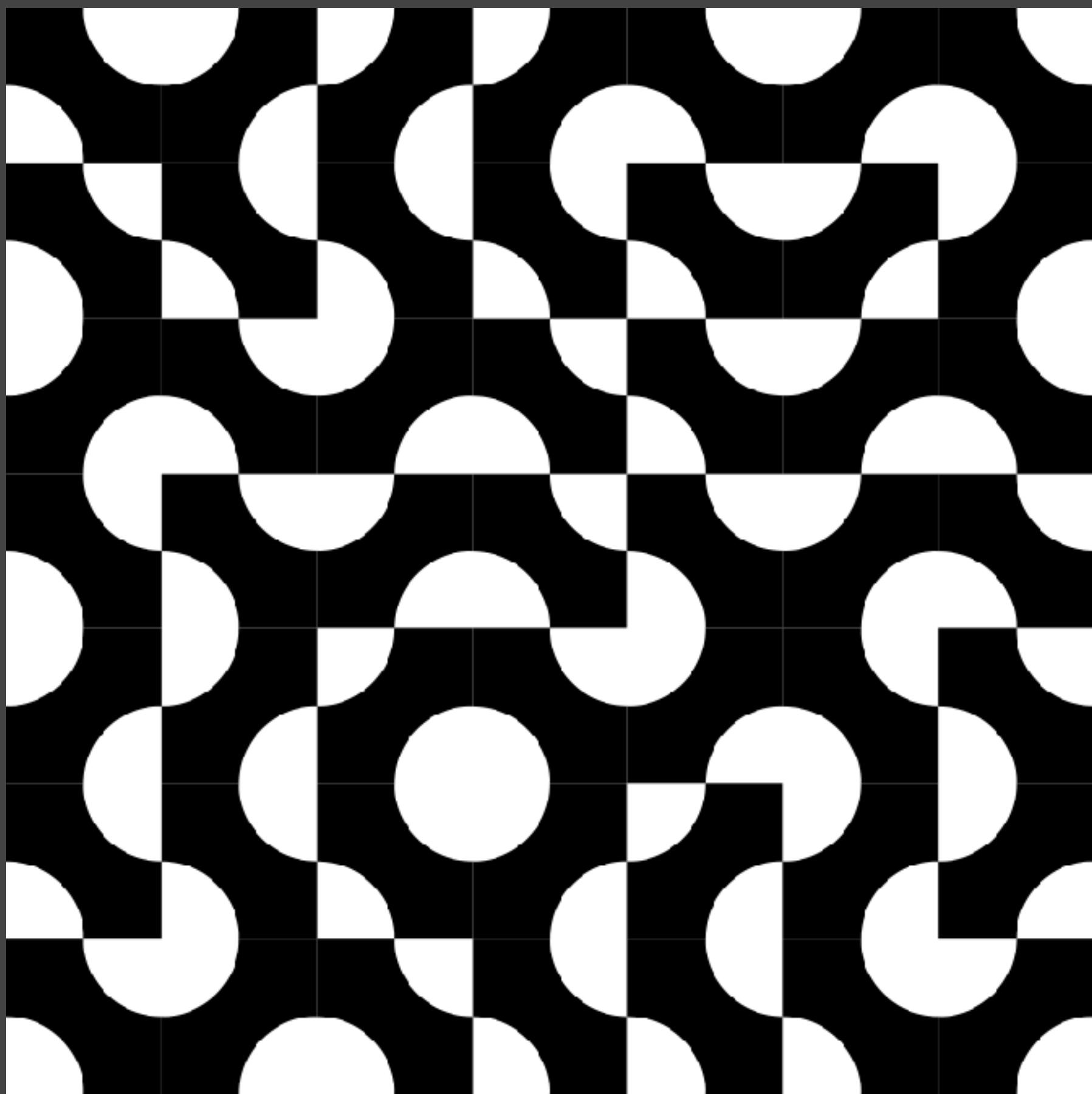
Drawbot Animations

Maurice Meilleur

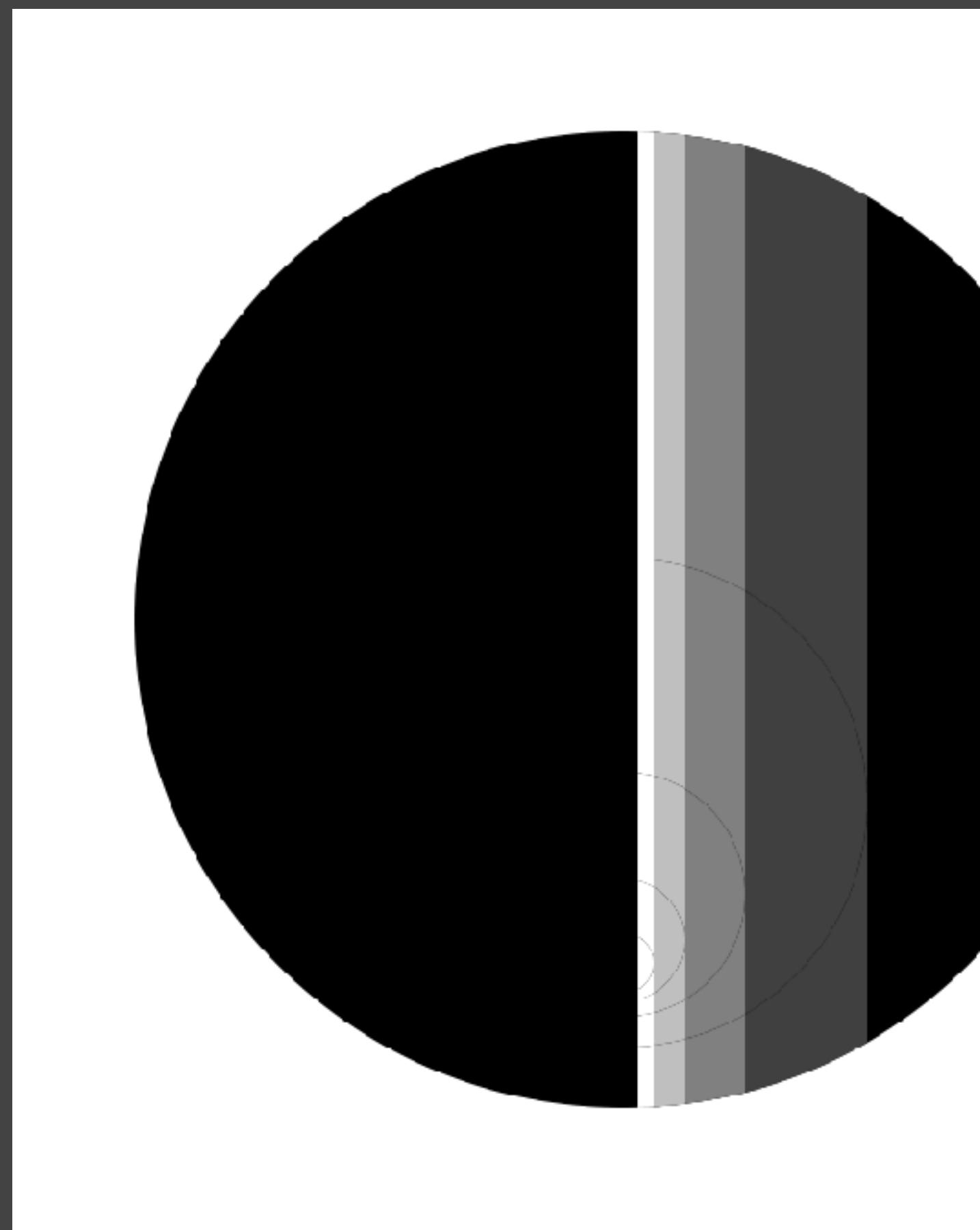
Le Parc transformations



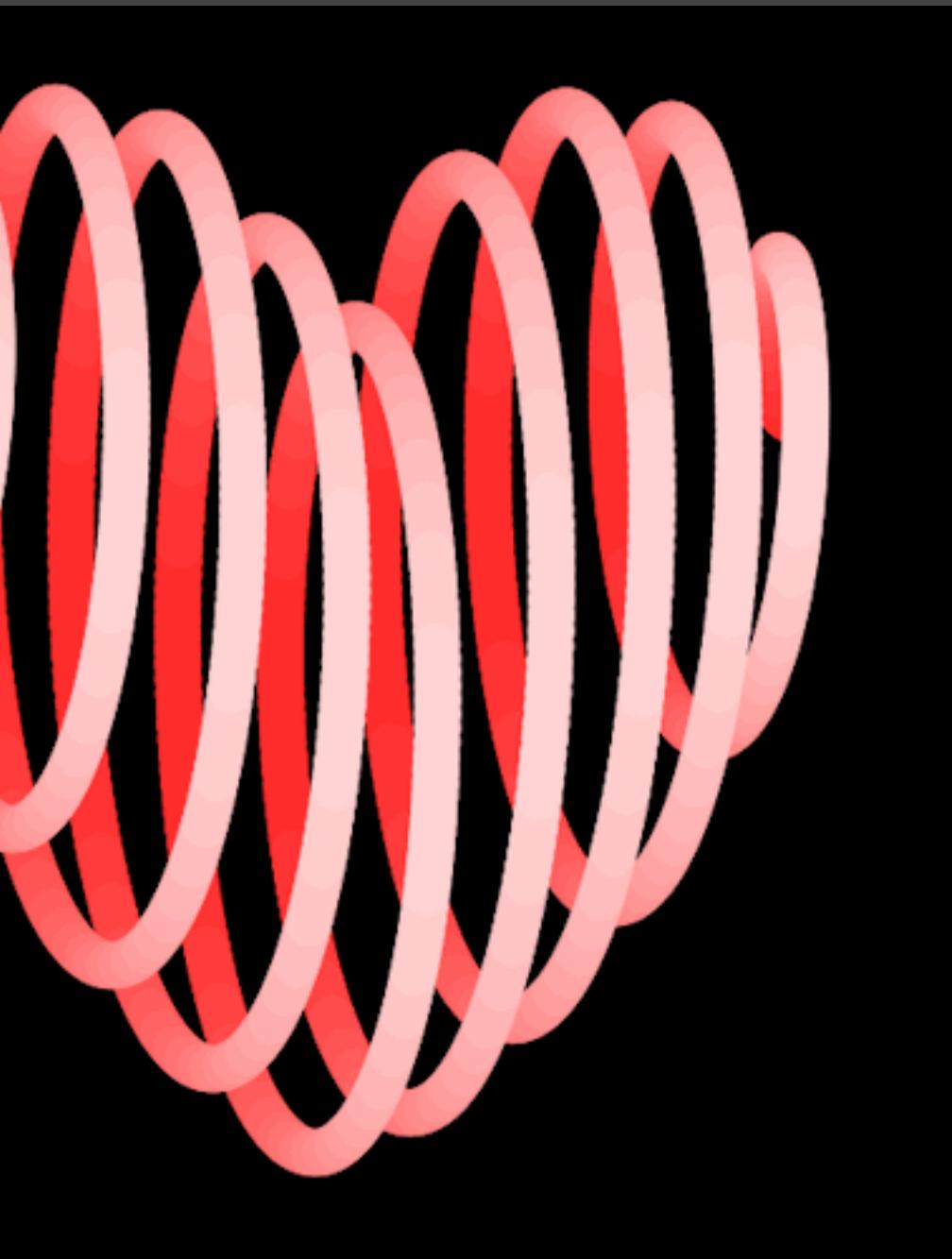
Truchet tiles



Gerstner's Tangential Eccentric

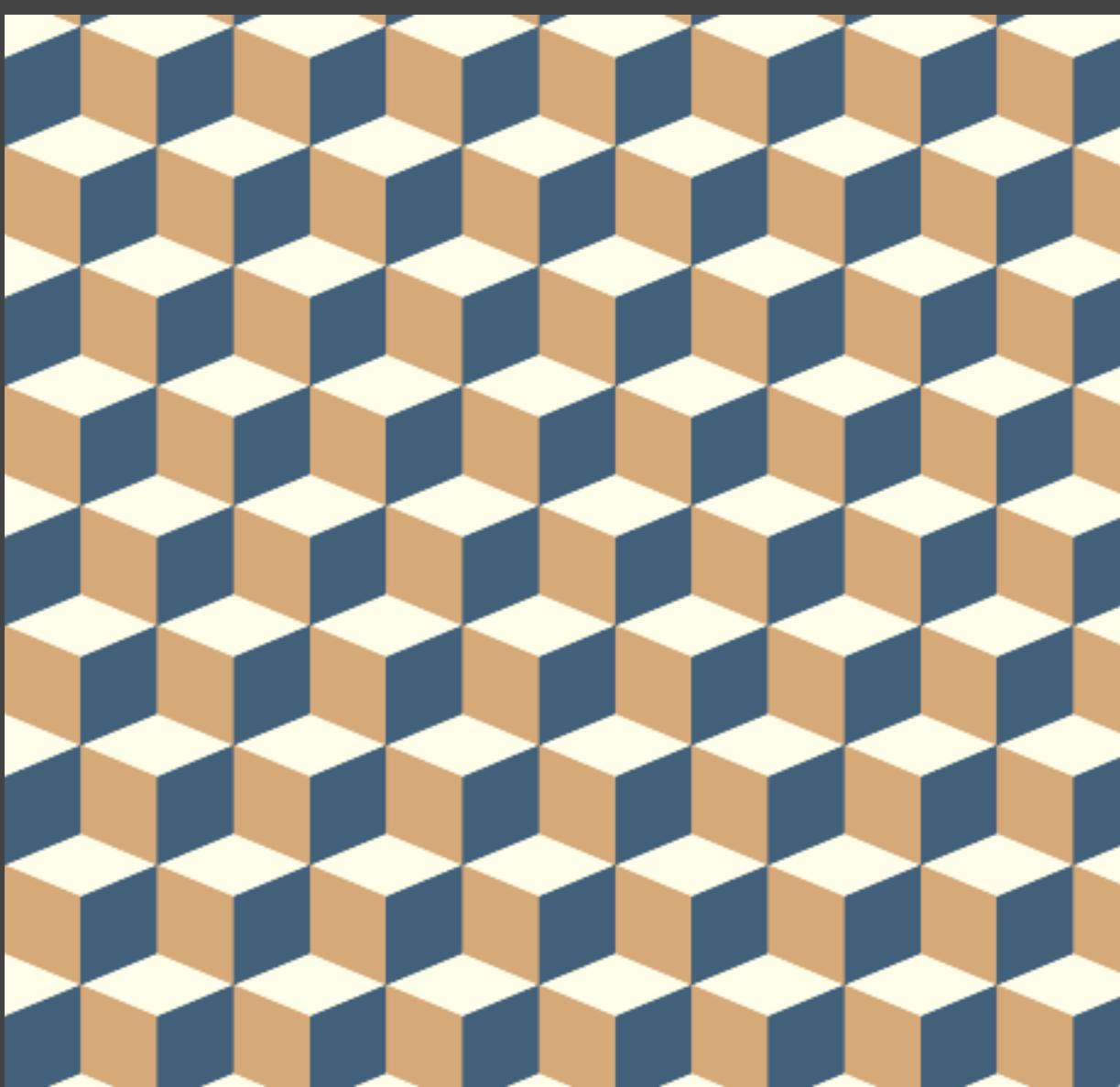
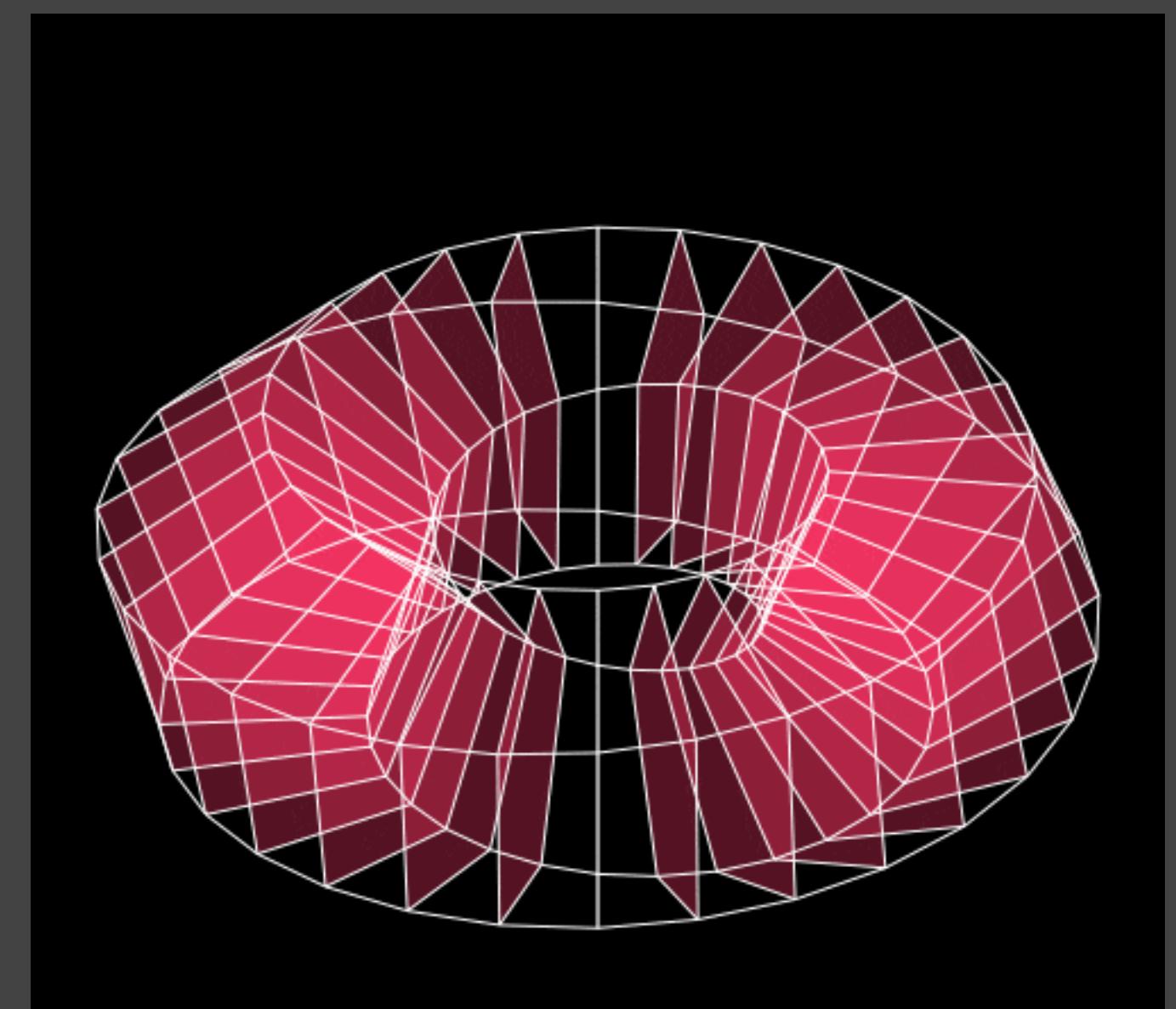
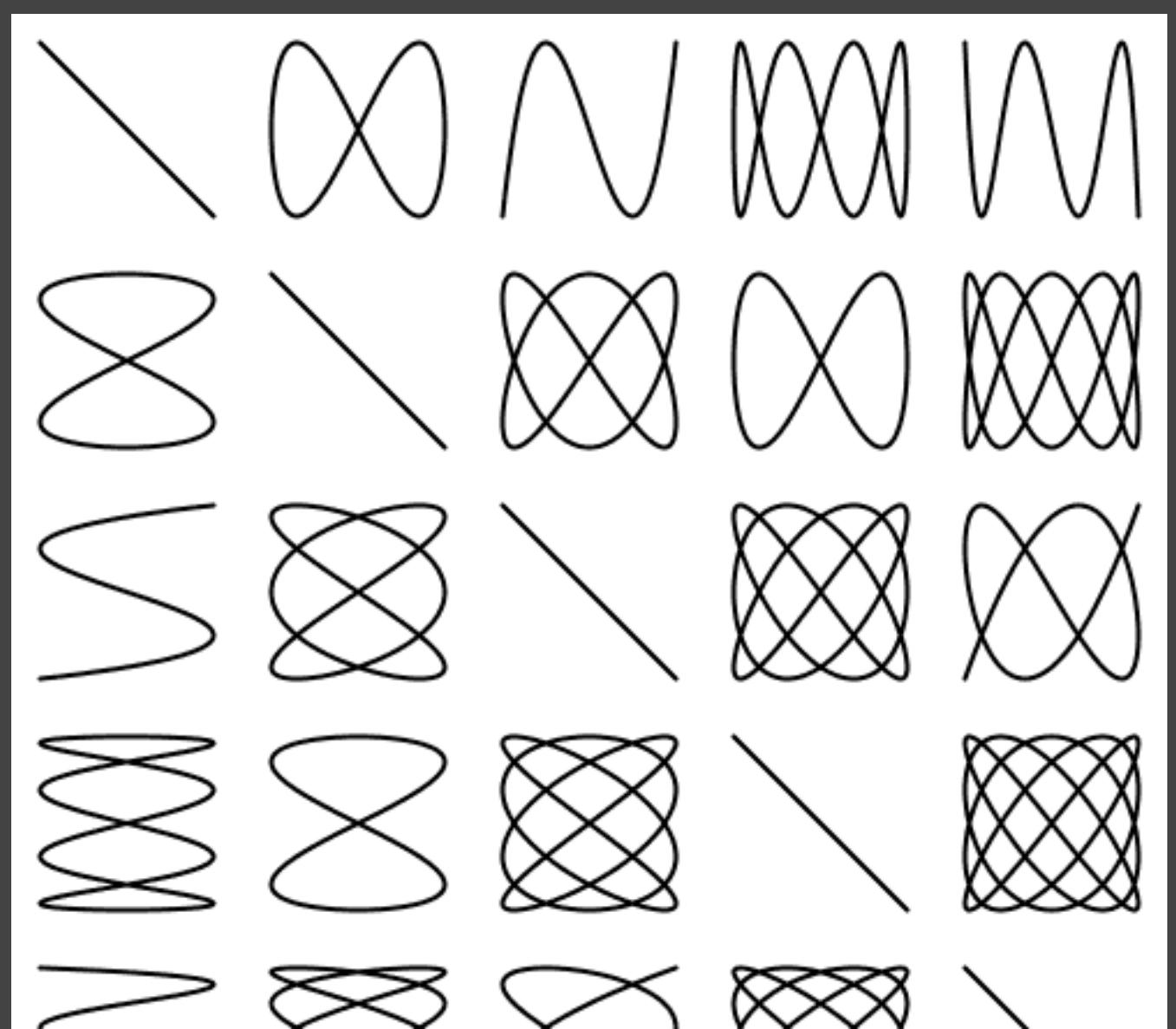
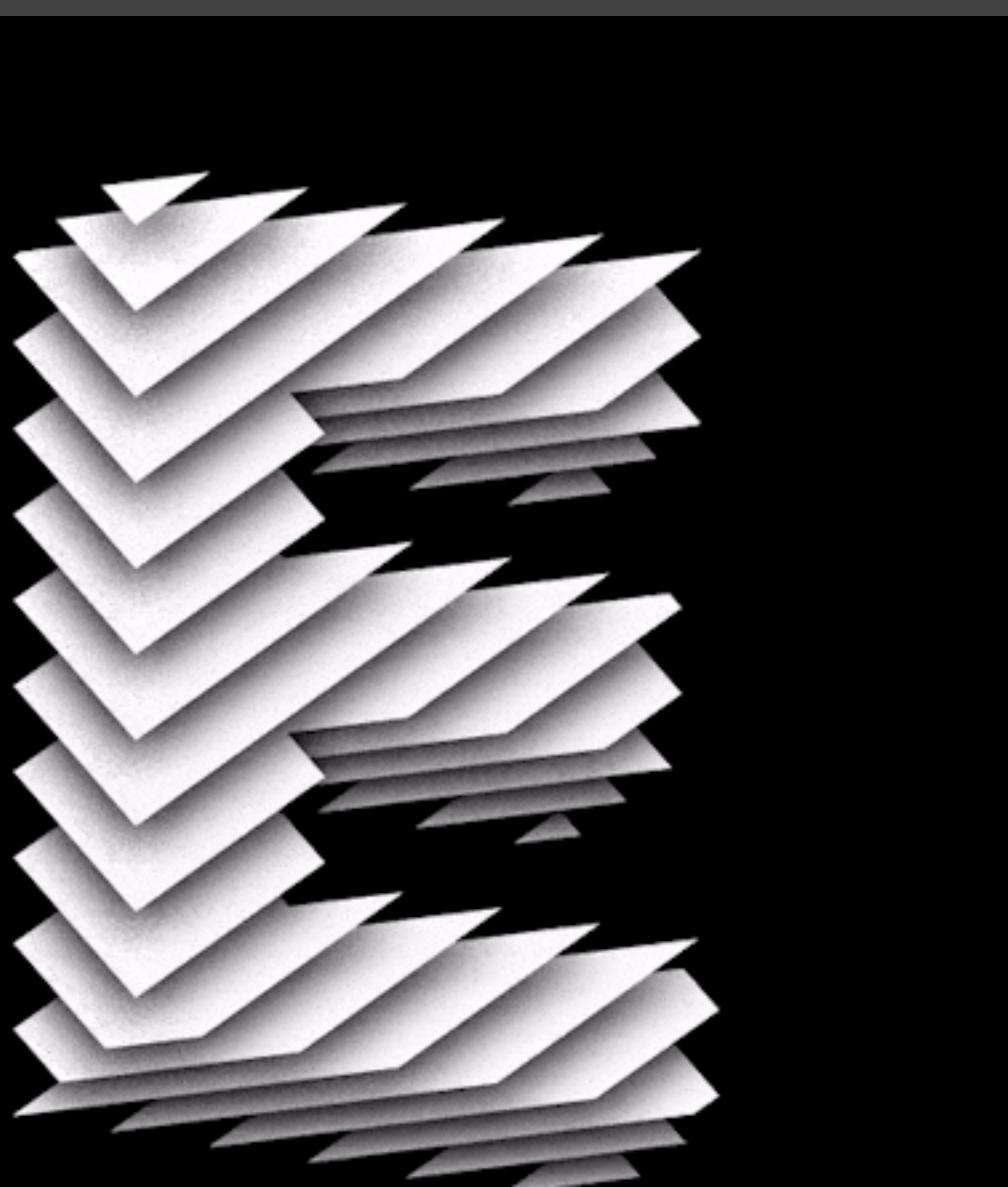
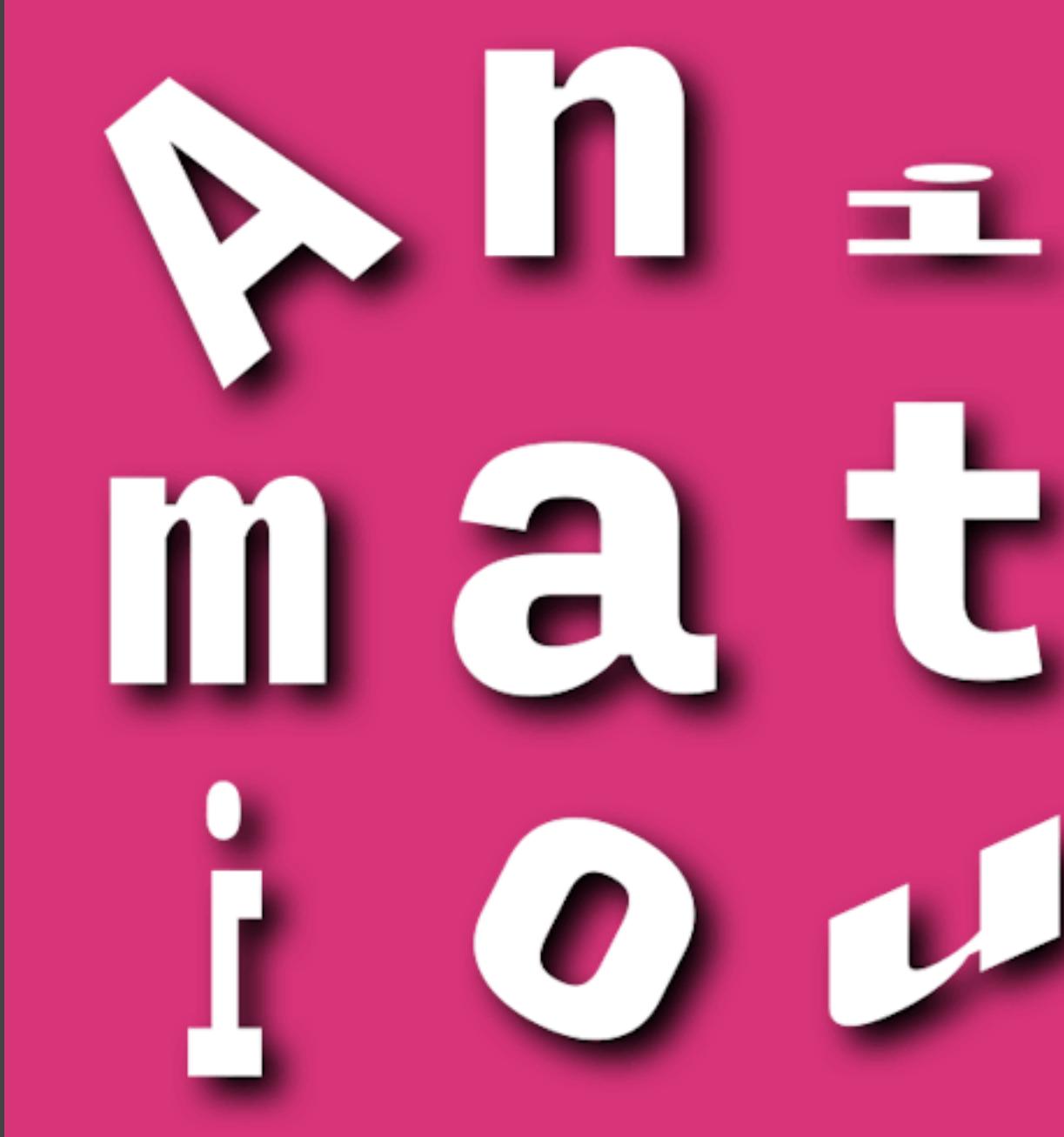


Daily Drawbot



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图形库
- Project Roles

USER DOCUMENTATION

- How to download Skia
- How to build Skia
- Skia Color Management
- Issue Tracking
- Privacy
- Tips & FAQ

Samples and Tutorials

- Using Skia's PDF Backend
- Skia Viewer

Release Information

- Milestone Release Notes
- Milestone Schedule

Library Modules

Skottie - Lottie Animation Player

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.

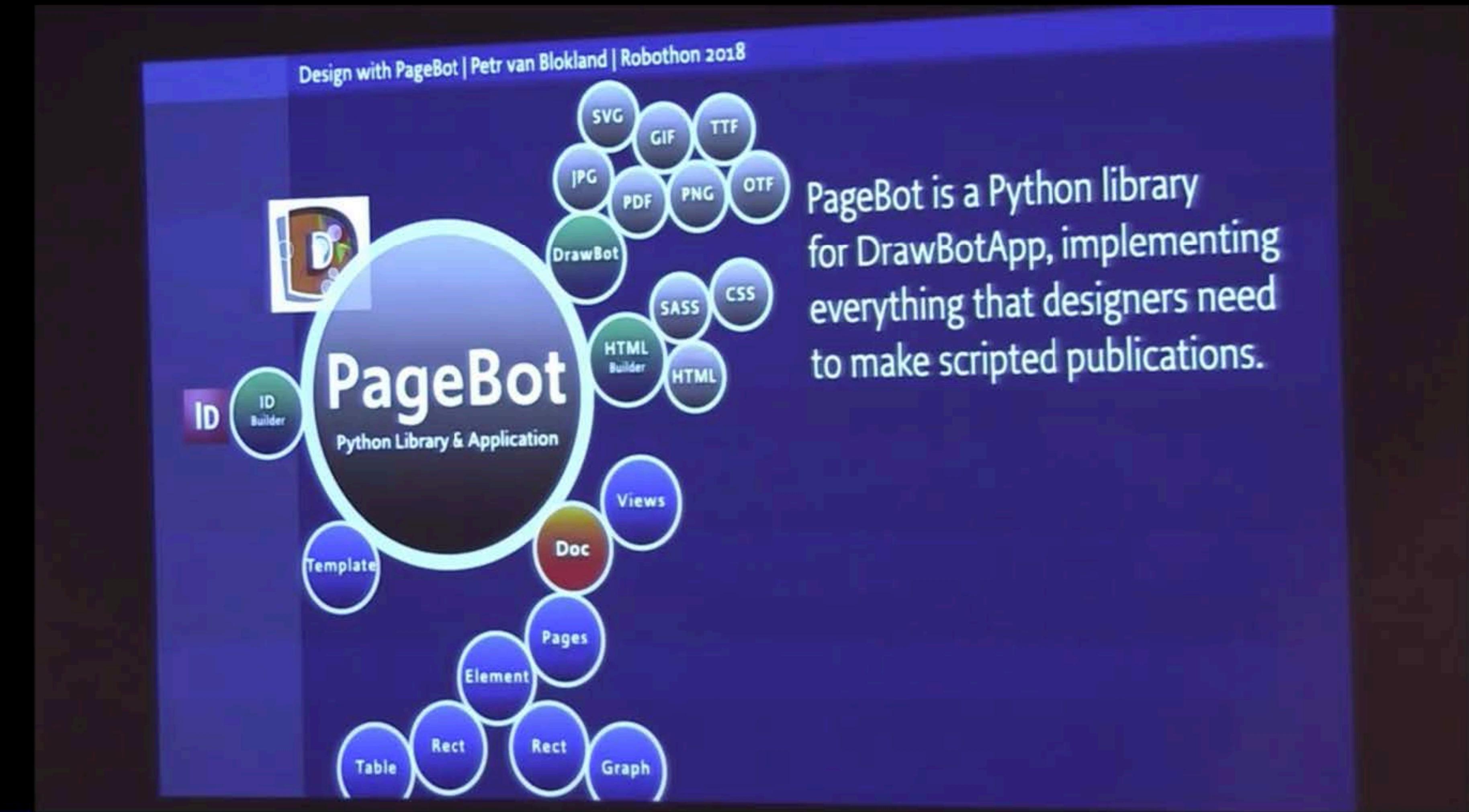
Shapes Bézier Curves



PageBot

Buro Petr van Blokland
+ Claudia Mens

Robothon
2018



Flat

Juraj Sukop

The image shows two overlapping browser windows. The top window displays the Flat library documentation at xxxyz.org/flat. It 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). A section titled 'Flat — Generative infrastructure for Python' follows, with a detailed description of the library's purpose and its Python API. A 'Content' section lists links to 'Features', 'Core concepts', 'Tutorial', 'Examples', and 'API reference'. The bottom window is a code editor titled 'Tutorial' at xxxyz.org/flat. It contains a Python script demonstrating how to use the Flat library to create a document with a circle, text, and an image.

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 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 be 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

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.