

# Processing ならびに Processing による作品の話

@KojiSaito



Processing って、どんな言語？



# Welcome to Processing!

Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. There are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning and prototyping.

[Download](#)[Reference](#)[Donate](#)[Open Editor](#)

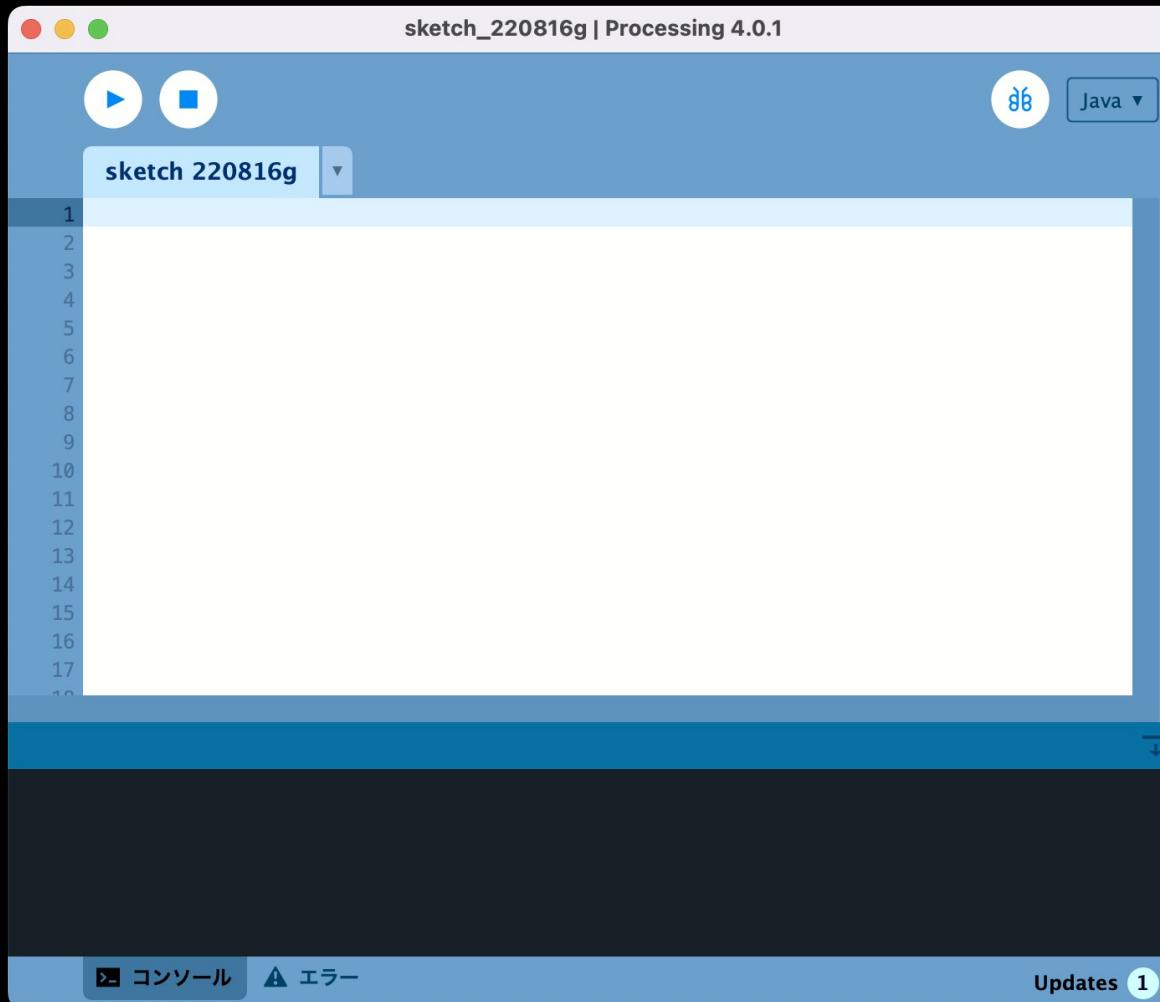


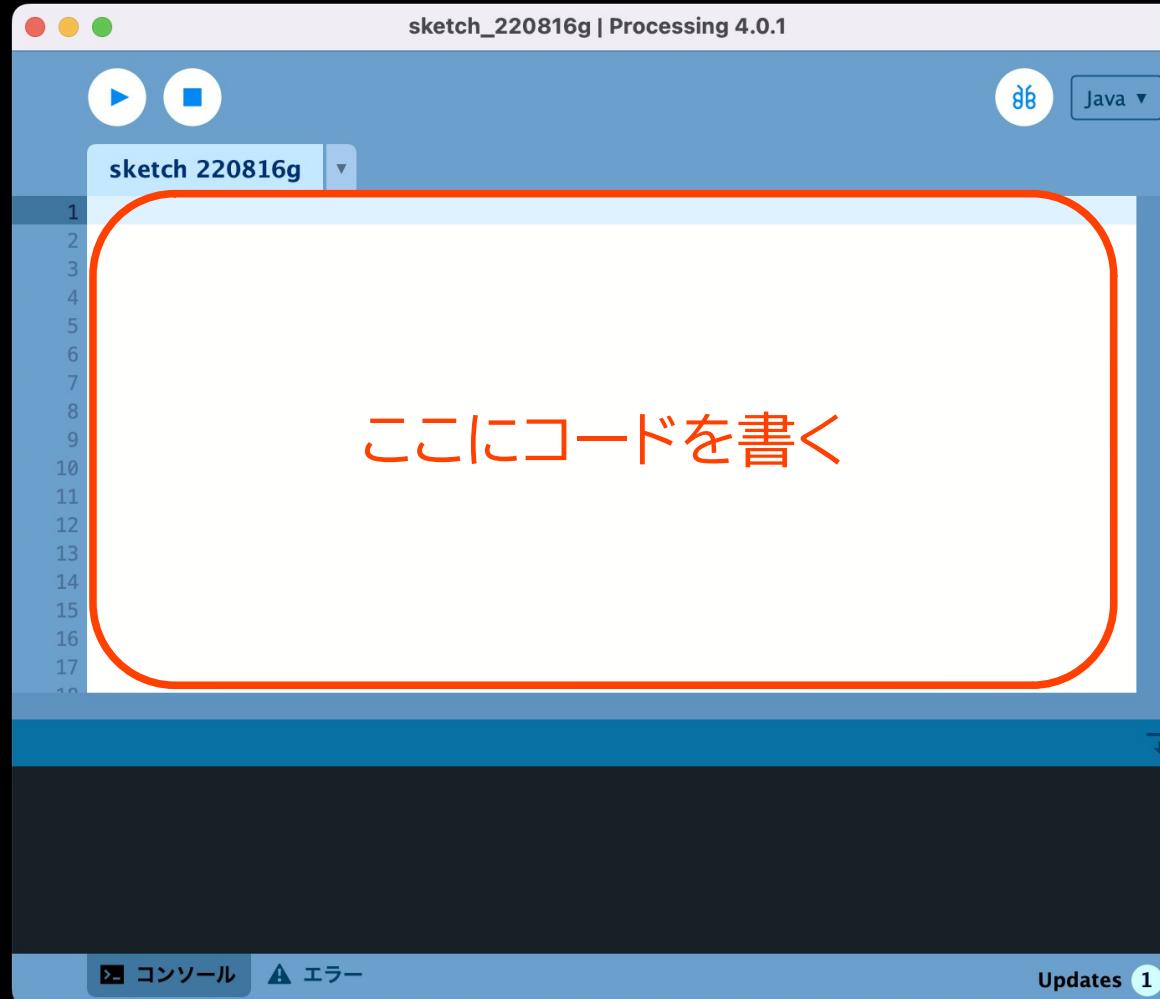
# Welcome to Processing!

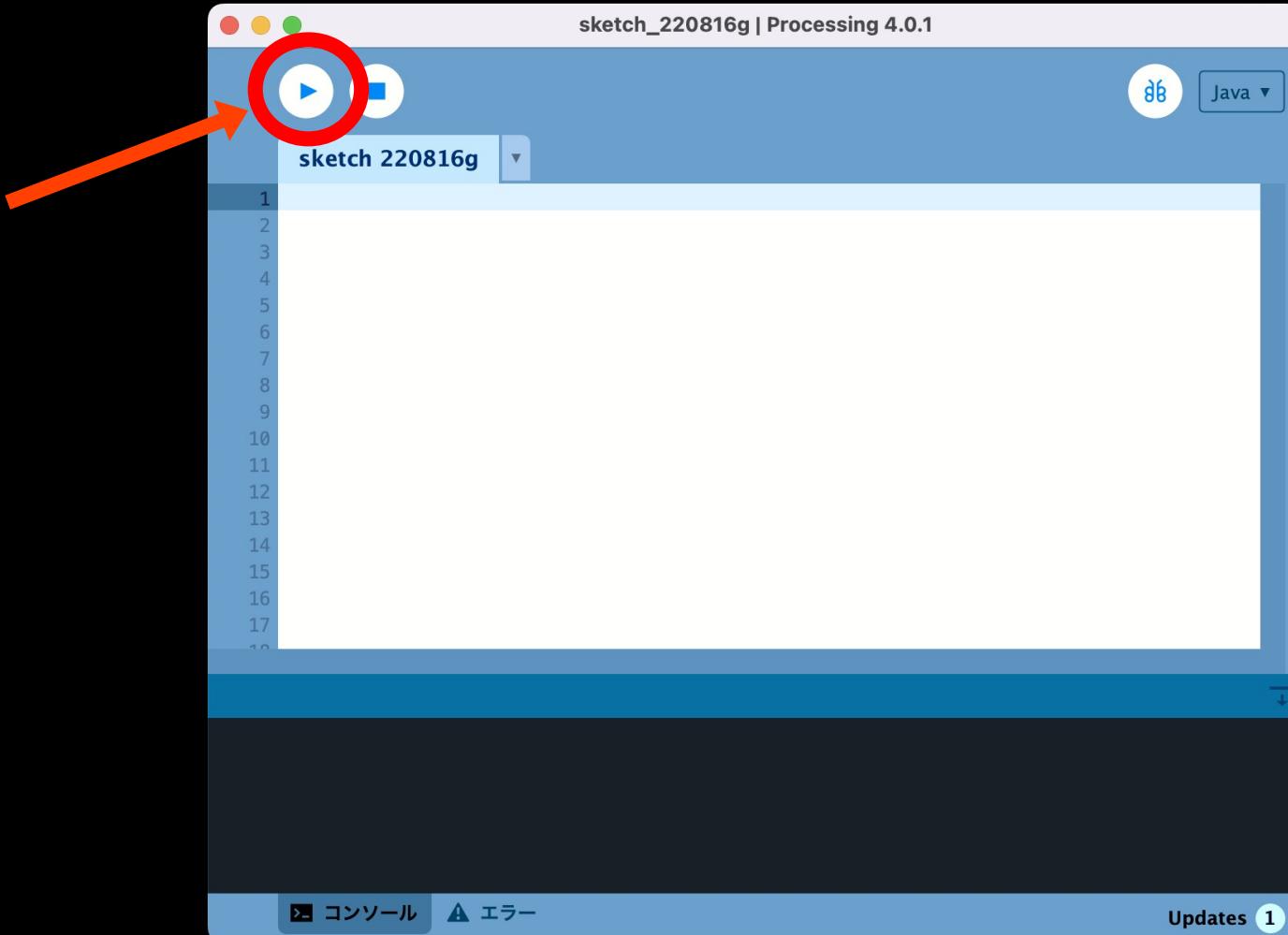
Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. There are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning and prototyping.

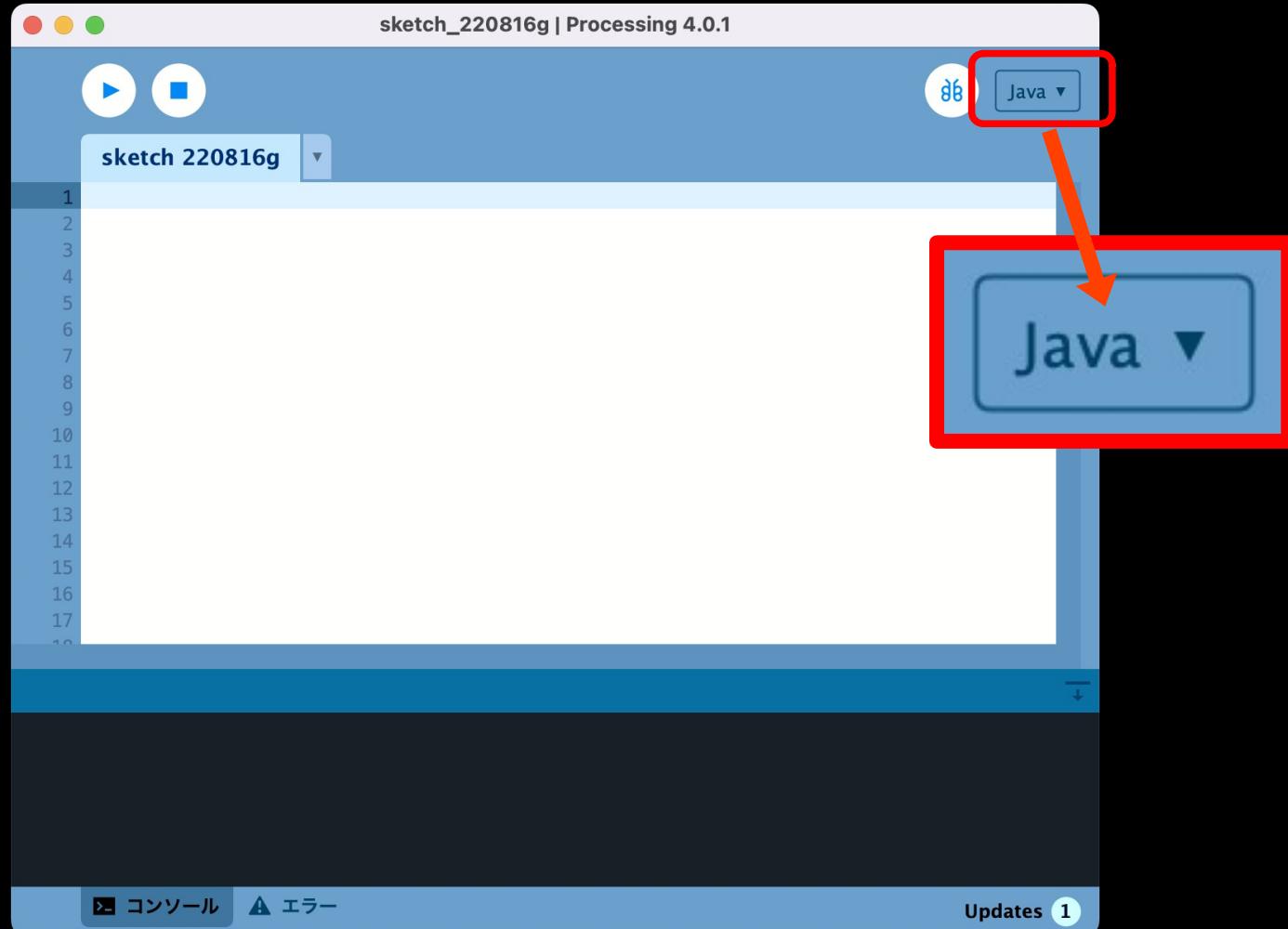
Processing is a flexible software sketchbook and a **language** for learning how to code within the context of the visual arts.

[Download](#)[Reference](#)[Donate](#)[Open Editor](#)









sketch\_220816g | Processing 4.0.1

The image shows a screenshot of the Processing 4.0.1 software interface. At the top, there are three window control buttons (red, yellow, green) and the title "sketch\_220816g | Processing 4.0.1". Below the title is a toolbar with a play button, a stop button, a refresh icon, and a "Java" dropdown menu. The main area is divided into two sections: a code editor on the left and a preview window on the right. The code editor contains the following Java code:

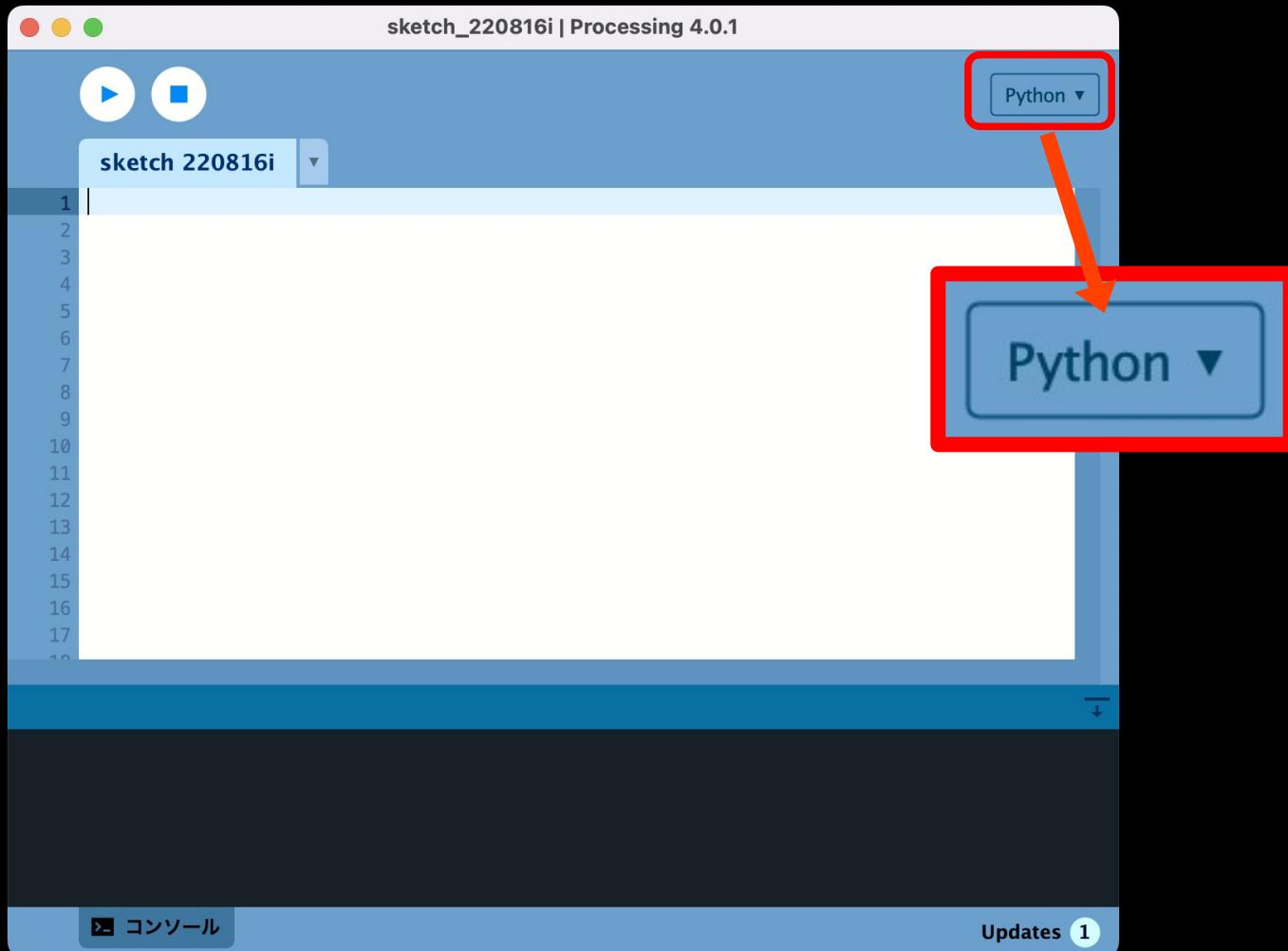
```
1 size(800,400);
2 background(20,120,220);
3 noFill();
4 stroke(255,255,255);
5 for(int i=0; i<200; i++) {
6     float x=random(width);
7     float y=random(height);
8     float r=random(100);
9     float t=random(2)+1;
10    strokeWeight(t);
11    circle(x,y,r);
12 }
13
14
15
16
17
18
```

The preview window shows a white canvas with a single black circle at the center. The bottom of the interface features a dark blue footer bar with several icons: a magnifying glass for search, a downward arrow for updates, and tabs for "コンソール" (Console), "エラー" (Errors), and "Updates 1".

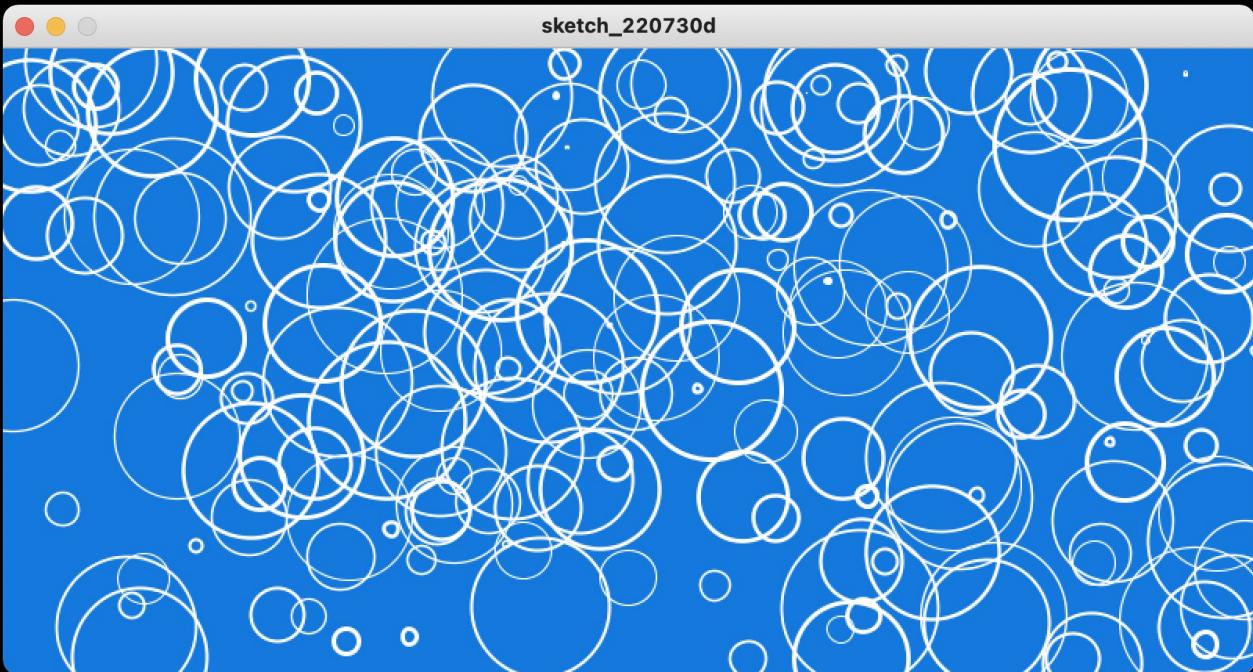


```
size(800,400);           // スクリーンサイズを 800 x 400 に
background(20,120,220);   // 背景色 (R,G,B)=(20,120,220) で
                         // 塗りつぶし
noFill();                // 描画する図形は塗りつぶし無し
stroke(255,255,255);    // 輪郭線の色は白(R=G=B=255)
for(int i=0; i<200; i++) {
  float x=random(width); // x に 0~800 までの乱数を格納
  float y=random(height); // y に 0~400 までの乱数を格納
  float r=random(100);    // r に 0~100 までの乱数を格納
  float t=random(2)+1;    // t に 1~3 までの乱数を格納
  strokeWeight(t);        // 輪郭線の太さを指定
  circle(x,y,r);         // 亂数を使って円を描く
}
```

```
size(800,400);           // スクリーンサイズを 800 x 400 に
background(20,120,220);   // 背景色 (R,G,B)=(20,120,220) で
                         // 塗りつぶし
noFill();                // 描画する図形は塗りつぶし無し
stroke(255,255,255);    // 輪郭線の色は白(R=G=B=255)
for(int i=0; i<200; i++) {
    float x=random(width); // x に 0~800 までの乱数を格納
    float y=random(height); // y に 0~400 までの乱数を格納
    float r=random(100);    // r に 0~100 までの乱数を格納
    float t=random(2)+1;    // t に 1~3 までの乱数を格納
    strokeWeight(t);        // 輪郭線の太さを指定
    circle(x,y,r);         // 亂数を使って円を描く
}
```



```
size(800,400)
background(20,120,220)
noFill()
stroke(255,255,255)
for i in range(200):
    x=random(width)
    y=random(height)
    r=random(100)
    t=random(2)+1
    strokeWeight(t)
    circle(x,y,r)
```



```
size(800,400)
background(20,120,220)
noFill()
stroke(255,255,255)
for i in range(200):
    x=random(width)
    y=random(height)
    r=random(100)
    t=random(2)+1
    strokeWeight(t)
    circle(x,y,r)
```

Python mode

```
size(800,400);
background(20,120,220);
noFill();
stroke(255,255,255);
for(int i=0; i<200; i++) {
    float x=random(width);
    float y=random(height);
    float r=random(100);
    float t=random(2)+1;
    strokeWeight(t);
    circle(x,y,r);
}
```

Java mode

Contribution Manager

Libraries Modes Tools Examples Updates

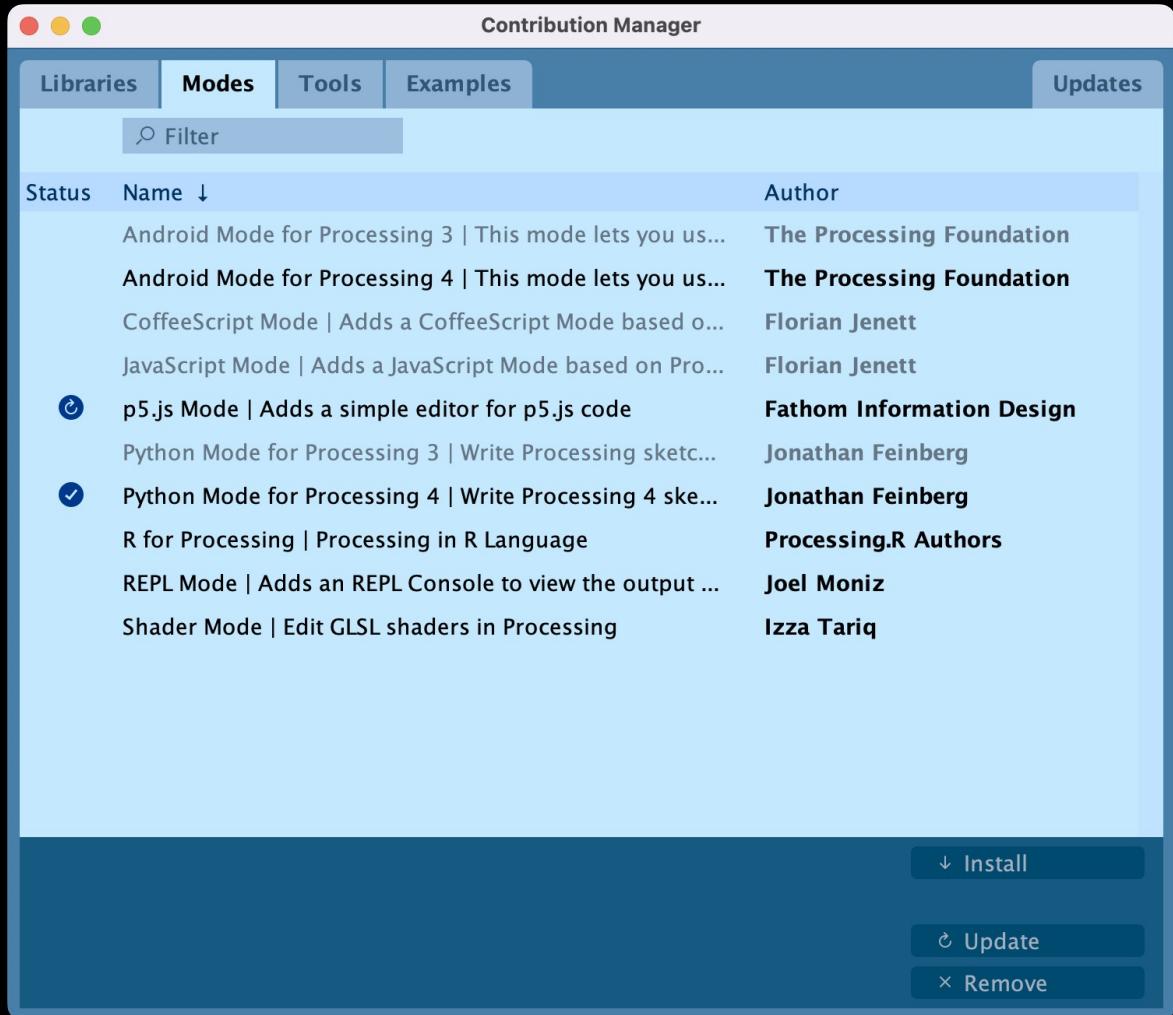
Filter

Status	Name ↓	Author
	Android Mode for Processing 3   This mode lets you us...	The Processing Foundation
	Android Mode for Processing 4   This mode lets you us...	The Processing Foundation
	CoffeeScript Mode   Adds a CoffeeScript Mode based o...	Florian Jenett
	JavaScript Mode   Adds a JavaScript Mode based on Pro...	Florian Jenett
⌚	p5.js Mode   Adds a simple editor for p5.js code	Fathom Information Design
	Python Mode for Processing 3   Write Processing sketc...	Jonathan Feinberg
✓	Python Mode for Processing 4   Write Processing 4 ske...	Jonathan Feinberg
	R for Processing   Processing in R Language	Processing.R Authors
	REPL Mode   Adds an REPL Console to view the output ...	Joel Moniz
	Shader Mode   Edit GLSL shaders in Processing	Izza Tariq

Install

Update

Remove



p5js = JavaScript mode

proce<sup>55</sup>ing.org



processing.org

```
void setup() {  
    // 最初に一度だけ呼ばれる関数  
    // size 関数の実行 = 画面サイズの指定はここに書く  
}  
  
void draw() {  
    // 描画のために何度も呼ばれる関数  
    // この関数内に記述された処理が画面を作る  
}
```

```
void setup() {  
    size(800,500);  
    colorMode(HSB);      // (H,S,B)=(色相, 彩度, 輝度)  
}  
  
float x=250, y=250; // ボールの位置  
float dx=-10, dy=-5; // ボールの変位量  
void draw() {  
    clear();                // 画面クリア(黒で塗る)  
    x += dx;   y += dy;    // 位置の更新  
    if(x<0) { x=0; dx=-dx; }  
    if(width<=x) { x=width; dx=-dx; }  
    if(y<0) { y=0; dy=-dy; }  
    if(height<=y) { y=height; dy=-dy; }  
    fill(frameCount%256, 255, 255); // H,S,B で指定  
    circle(x,y,40);  
}
```

```
void setup() {  
    size(800,500);  
    colorMode(HSB);      // (H,S,B)=(色相, 彩度, 輝度)  
}  
}
```

```
float x=250, y=250; // ボールの位置  
float dx=-10, dy=-5; // ボールの変位量  
void draw() {  
    clear();           // 画面クリア(黒で塗る)  
    x += dx; y += dy; // 位置の更新  
    if(x<0) { x=0; dx=-dx; }  
    if(width<=x) { x=width; dx=-dx; }  
    if(y<0) { y=0; dy=-dy; }  
    if(height<=y) { y=height; dy=-dy; }  
    fill(frameCount%256, 255, 255); // H,S,B で指定  
    circle(x,y,40);  
}
```

```
void setup() {  
    size(800,500);  
    colorMode(HSB);      // (H,S,B)=(色相, 彩度, 輝度)  
}  
} // グローバル変数
```

```
float x=250, y=250; // ボールの位置  
float dx=-10, dy=-5; // ボールの変位量
```

```
void draw() {  
    clear();                  // 画面クリア(黒で塗る)  
    x += dx; y += dy;        // 位置の更新  
    if(x<0) { x=0; dx=-dx; }  
    if(width<=x) { x=width; dx=-dx; }  
    if(y<0) { y=0; dy=-dy; }  
    if(height<=y) { y=height; dy=-dy; }  
    fill(frameCount%256, 255, 255); // H,S,B で指定  
    circle(x,y,40);  
}
```

```
void setup() {  
    size(800,500);  
    colorMode(HSB);      // (H,S,B)=(色相, 彩度, 輝度)  
}  
} // グローバル変数
```

```
float x=250, y=250; // ボールの位置  
float dx=-10, dy=-5; // ボールの変位量  
void draw() {
```

```
    clear();           // 画面クリア(黒で塗る)
```

```
    x += dx; y += dy; // 位置の更新  
    if(x<0) { x=0; dx=-dx; }  
    if(width<=x) { x=width; dx=-dx; }  
    if(y<0) { y=0; dy=-dy; }  
    if(height<=y) { y=height; dy=-dy; }  
    fill(frameCount%256, 255, 255); // H,S,B で指定  
    circle(x,y,40);  
}
```

```
void setup() {  
    size(800,500);  
    colorMode(HSB);      // (H,S,B)=(色相, 彩度, 輝度)  
}  
} // グローバル変数
```

```
float x=250, y=250; // ボールの位置  
float dx=-10, dy=-5; // ボールの変位量  
void draw() {  
    clear();           // 画面クリア(黒で塗る)  
    x += dx; y += dy; // 位置の更新  
    if(x<0) { x=0; dx=-dx; }  
    if(width<=x) { x=width; dx=-dx; }  
    if(y<0) { y=0; dy=-dy; }  
    if(height<=y) { y=height; dy=-dy; }  
    fill(frameCount%256, 255, 255); // H,S,B を指定  
    circle(x,y,40);  
}
```

```
void setup() {  
    size(800,500);  
    colorMode(HSB);      // (H,S,B)=(色相, 彩度, 輝度)  
}  
} // グローバル変数
```

```
float x=250, y=250; // ボールの位置  
float dx=-10, dy=-5; // ボールの変位量  
void draw() {  
    clear();           // 画面クリア(黒で塗る)  
    x += dx; y += dy; // 位置の更新  
    if(x<0) { x=0; dx=-dx; }  
    if(width<=x) { x=width; dx=-dx; }  
    if(y<0) { y=0; dy=-dy; }  
    if(height<=y) { y=height; dy=-dy; }  
    fill(frameCount%256, 255, 255); // H,S,B で指定  
    circle(x,y,40);  
}
```

```
void setup() {  
    size(800,500);  
    colorMode(HSB);      // (H,S,B)=(色相, 彩度, 輝度)  
}  
} // グローバル変数
```

```
float x=250, y=250; // ボールの位置  
float dx=-10, dy=-5; // ボールの変位量  
void draw() {  
    clear();           // 画面クリア(黒で塗る)  
    x += dx; y += dy; // 位置の更新  
    if(x<0) { x=0; dx=-dx; }  
    if(width<=x) { x=width; dx=-dx; }  
    if(y<0) { y=0; dy=-dy; }  
    if(height<=y) { y=height; dy=-dy; }  
    fill(frameCount%256, 255, 255); // H,S,B で指定  
    circle(x,y,40);  
}
```



**mouseX** // 現在のマウスカーソルの X 座標  
**mouseY** // 現在のマウスカーソルの Y 座標

**pmouseX** // ひとつ前のフレームにおけるマウスカーソルの X 座標  
**pmouseY** // ひとつ前のフレームにおけるマウスカーソルの Y 座標

**mousePressed** // マウスのボタンが押されているか否か

**mouseClicked()** // マウスのボタンが押されたら1度だけ呼ばれる関数  
**mouseMoved()** // マウスが動いたら呼ばれる関数

…等など

作品を制作した後の楽しみ方について

はう君(@Hau\_kun)さん発案の  
#つぶやきProcessing



Koji Saito @KojiSaito · 15s

...

#つぶやき Processing

G=range

L=G(8)

```
def D(x,y,t,q):[t&q/2**i and rect(x+i*4,y,4,4) for i in L]
```

```
R=lambda n:int(random(n))
```

```
P=[129,66,36,24]
```

```
size(500,200)
```

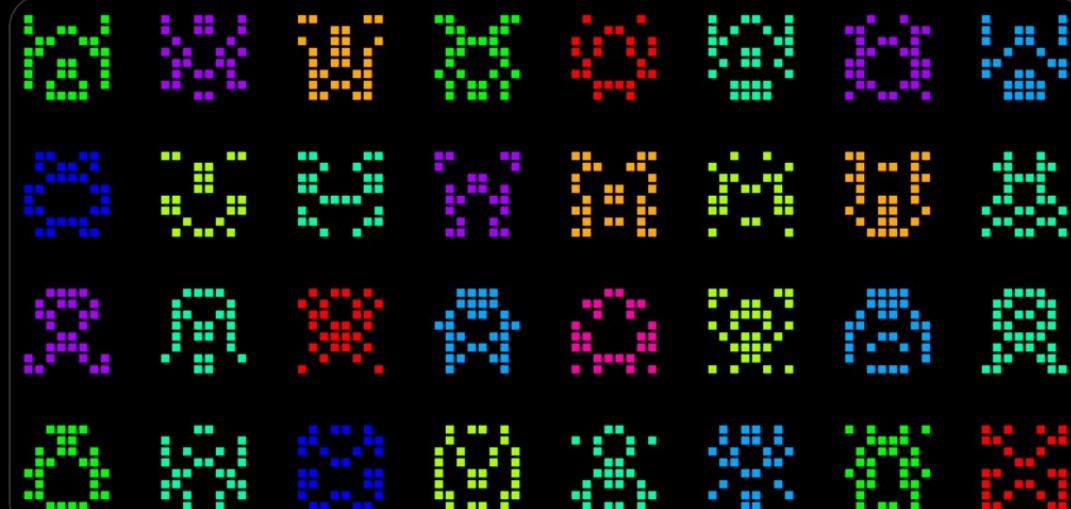
```
colorMode(HSB,9)
```

```
clear()
```

```
for t in G(100):
```

```
x=t%10;y=t/10;fill(R(9),9,9)
```

```
for i in L:D(x*50+8,y*50+i*4+8,P[R(4)]|P[R(4)],128)
```



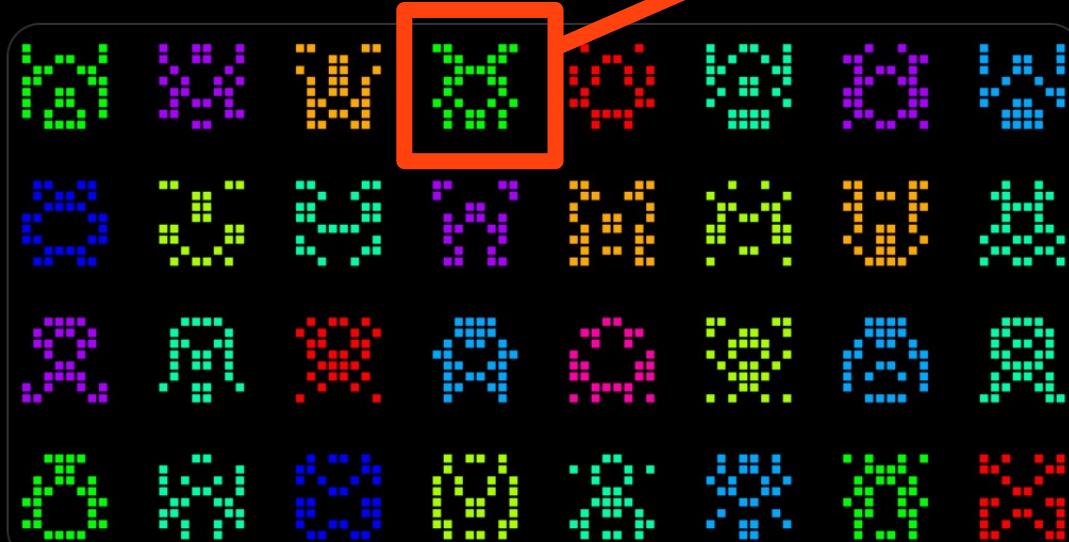
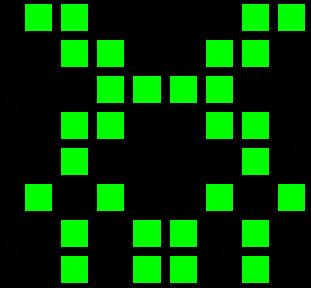


Koji Saito @KojiSaito · 15s

#つぶやき Processing

...

```
G=range  
L=G(8)  
def D(x,y,t,q):[t&q/2**i and rect(x+i*4,y,4,4) for i in L]  
R=lambda n:int(random(n))  
P=[129,66,36,24]  
size(500,200)  
colorMode(HSB,9)  
clear()  
for t in G(100):  
    x=t%10;y=t/10;fill(R(9),9,9)  
    for i in L:D(x*50+8,y*50+i*4+8,P[R(4)]|P[R(4)],128)
```





Koji Saito @KojiSaito · 15s

...

#つぶやき Processing

G=range

L=G(8)

```
def D(x,y,t,q):[t&q/2**i and rect(x+i*4,y,4,4) for i in L]
```

```
R=lambda n:int(random(n))
```

```
P=[129,66,36,24]
```

```
size(500,200)
```

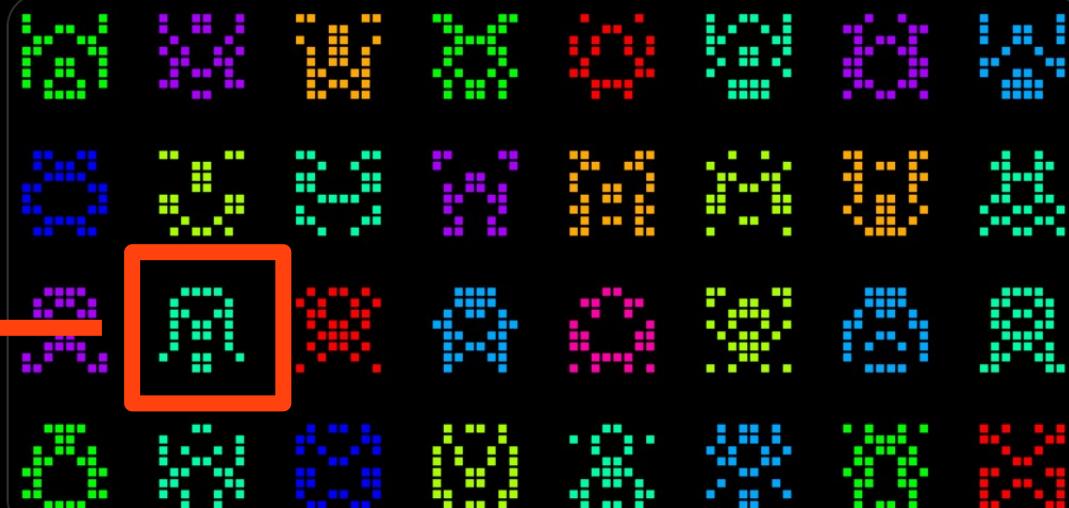
```
colorMode(HSB,9)
```

```
clear()
```

```
for t in G(100):
```

```
    x=t%10;y=t/10;fill(R(9),9,9)
```

```
    for i in L:D(x*50+8,y*50+i*4+8,P[R(4)]|P[R(4)],128)
```



つぶやきProcessing ポータル  
<https://tweetprocessing.projectroom.jp>

2022.05.31 時点

参加者: 350 名以上

作品数: 7,600 以上

8,100 以上となっている(2022.08.15 時点)

30/06/2020: [ONLINE] #つぶやきProcessing, tweets da comunidade japonesa de programação criativa

- Alexandre Villares



[https://garoa.net.br/wiki/Noite\\_de\\_Processing](https://garoa.net.br/wiki/Noite_de_Processing)

Alexandre B A Villares  
(@villares)

`#つぶやきProcessing` é a hashtag criada por @Hau\_kun e utilizada pela comunidade japonesa de programação criativa, reunindo no Twitter sketches cujo código cabe em apenas um tweet.

Alexandre Villares apresenta alguns dos autores desses tweets e explora com os participantes o código de alguns deles.

Slides: [https://docs.google.com/presentation/d/1-sF\\_Dbwpkz4i5\\_r0xYtitRU\\_nc7d5s4\\_T6GPASvlvik/edit?usp=sharing](https://docs.google.com/presentation/d/1-sF_Dbwpkz4i5_r0xYtitRU_nc7d5s4_T6GPASvlvik/edit?usp=sharing)

背景画像は @ntsutae さんの作品

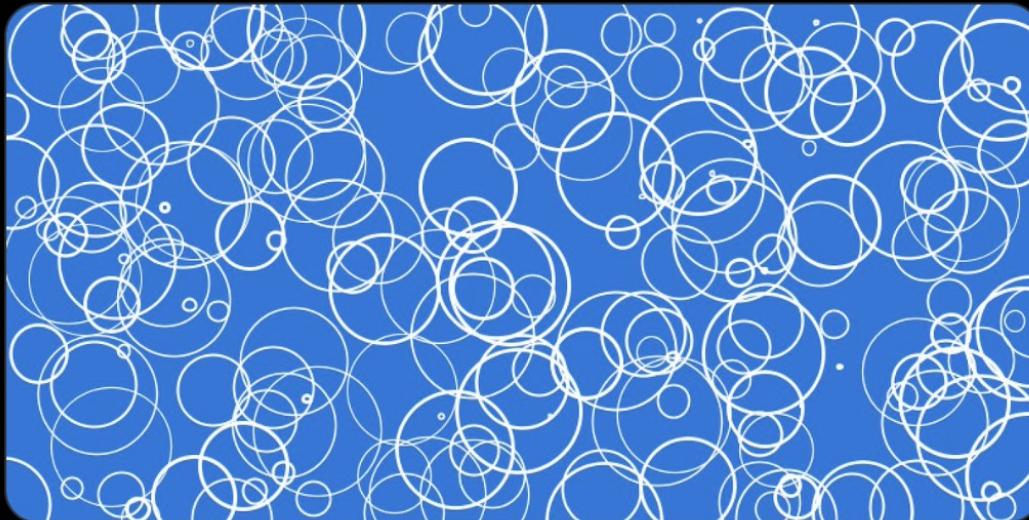


Koji Saito @KojiSaito · Aug 2

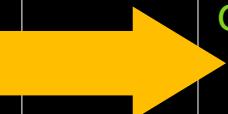
...

```
#つぶやきProcessing  
size(800,400)  
background(20,120,220)  
noFill()  
stroke(255,255,255)  
for i in range(200):  
    x=random(width)  
    y=random(height)  
    r=random(100)  
    t=random(2)+1  
    strokeWeight(t)  
    circle(x,y,r)
```

#LearnLanguages2022



```
void setup() {  
    size(800,500);  
    colorMode(HSB);  
}  
  
float x=250, y=250;  
float dx=-10, dy=-5;  
void draw() {  
    clear();  
    x += dx; y += dy;  
    if(x<0) { x=0; dx=-dx; }  
    if(width<=x) { x=width; dx=-dx; }  
    if(y<0) { y=0; dy=-dy; }  
    if(height<=y) { y=height; dy=-dy; }  
    fill(frameCount%256, 255, 255);  
    circle(x,y,40);  
}
```



```
def setup():  
    size(800,500)  
    colorMode(HSB)  
  
    x=250; y=250  
    dx=-10;dy=-5  
  
def draw():  
    global x,y,dx,dy  
    clear()  
    x+=dx; y+=dy  
    if x<0: x=0; dx=-dx  
    if width<=x: x=width; dx=-dx  
    if y<0: y=0; dy=-dy  
    if height<=y: y=height; dy=-dy  
    fill(frameCount%256,255,255)  
    circle(x,y,40)
```

```
def setup():
    size(800,500)
    colorMode(HSB)
```

```
x=250; y=250
dx=-10;dy=-5
```

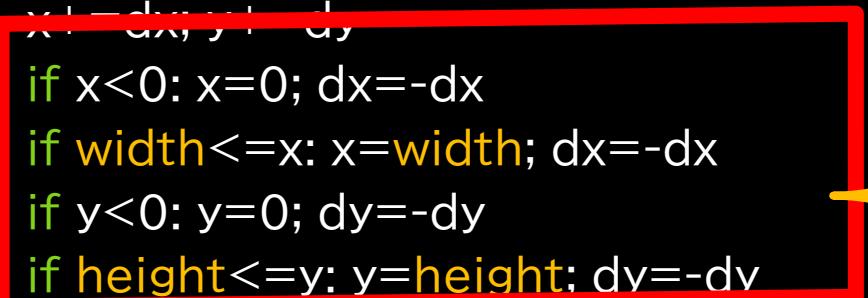
```
def draw():
    global x,y,dx,dy
    clear()
    x+=dx; y+=dy
    if x<0: x=0; dx=-dx
    if width<=x: x=width; dx=-dx
    if y<0: y=0; dy=-dy
    if height<=y: y=height; dy=-dy
    fill(frameCount%256,255,255)
    circle(x,y,40)
```

x を 0 から width の範囲に収める  
y を 0 から height の範囲に収める  
範囲から外れたら方向転換

```
def setup():
    size(800,500)
    colorMode(HSB)
```

```
x=250; y=250
dx=-10;dy=-5
```

```
def draw():
    global x,y,dx,dy
    clear()
    x+=dx; y+=dy
    if x<0: x=0; dx=-dx
    if width<=x: x=width; dx=-dx
    if y<0: y=0; dy=-dy
    if height<=y: y=height; dy=-dy
    fill(frameCount%256,255,255)
    circle(x,y,40)
```



```
newX=min(max(x,0),width)
newY=min(max(y,0),height)
if newX!=x: dx=-dx
if newY!=y: dy=-dy
```



Koji Saito  
@KojiSaito

...

## #つぶやき Processing

```
def setup():size(800,500);colorMode(HSB)
P=[250,250,-10,-5]
def draw():
    clear()
    P[0]+=P[2];P[1]+=P[3];x=P[0];y=P[1]
    P[0]=min(max(x,0),800);P[1]=min(max(y,0),500)
    if P[0]!=x:P[2]*=-1
    if P[1]!=y:P[3]*=-1
    fill(frameCount%256,255,255);circle(x,y,40)
#LL2022
```



0:14 131 views

つぶやきProcessingで  
どの程度の表現ができるのか？



Koji Saito  
@KojiSaito

...

#つぶやき Processing

S=square;F=fill;K=240

```
def setup():size(550,K);noStroke()
```

```
def draw():
```

```
background(99,150,K);f=frameCount*.003
```

```
for i in range(4**7):x=i%150;y=99-
```

```
i/150;t=noise(x*.01+f,y*.02,f);x*=4;y*=3;t=pow(t,8)*8;F  
(20,t*10);S(x,y,15);F(K,t*K);S(x-25,y-30,15)
```

```
filter(BLUR,3)
```



0:49 | 27 views

```
S=square;F=fill;K=240
def setup():size(550,K);noStroke()
def draw():
background(99,150,K);f=frameCount*.002
for i in range(4**7):x=i%150;y=99-i/150;t=noise(x*.01+f,y*.02,f)*4;y*=3;t=pow(t,8)*8;F(20,t*10);S(x,y,15);F(K,t*K);S(x-25,y-30,15)
filter(BLUR,3)
```

t=noise(x\*.01+f,y\*.02,f);



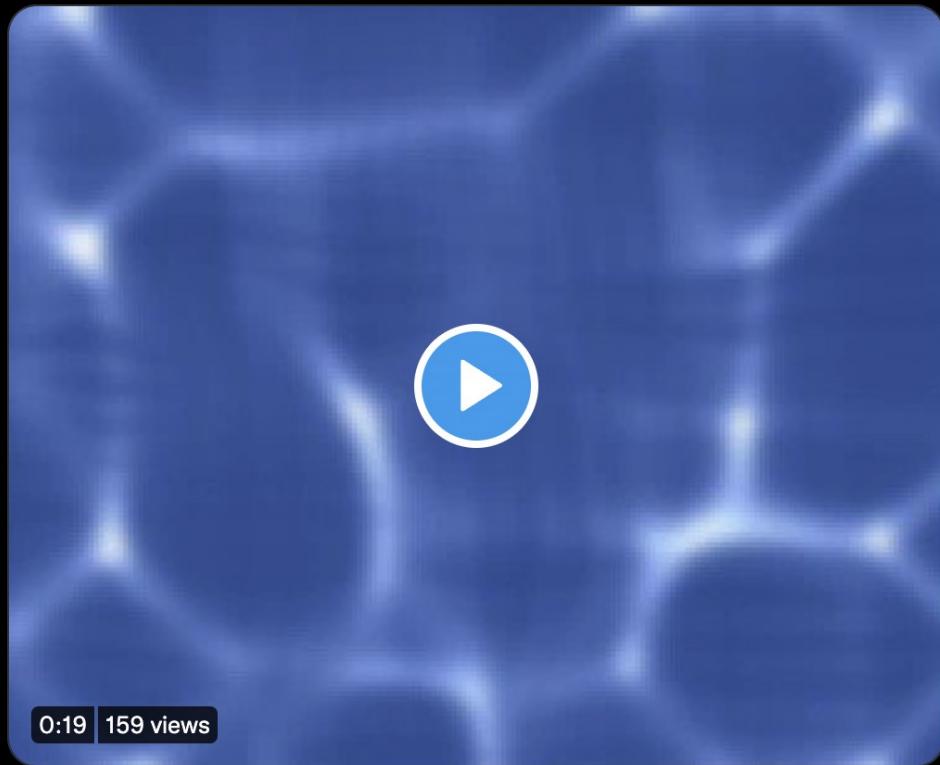


Koji Saito @KojiSaito · 12h

#つぶやき Processing

```
def setup():size(192,157);noStroke();blendMode(ADD)
H=lambda x,y:noise(x/200.,y/200.,frameCount*.1)
def draw():
    background(50,80,150);fill(2)
    for i in range(8**5):
        y=i/200;x=i%200;h=H(x,y)
        square(x+(H(x+1,y)-h)*5000,y+(H(x,y+1)-h)*5000,4)
        filter(BLUR,2)
```

...

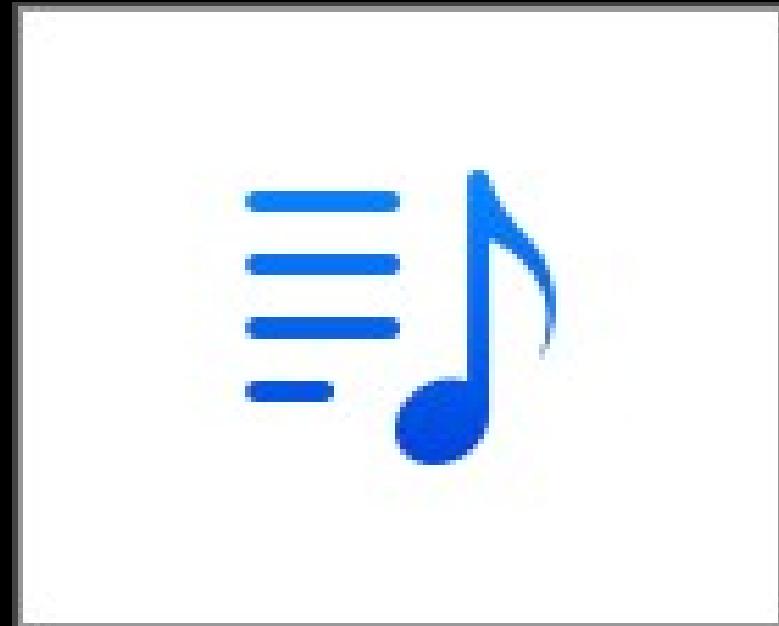


0:19 | 159 views

```
def setup():size(192,157);noStroke();blendMode(ADD)
H=lambda x,y:noise(x/200.,y/200.,frameCount*.1)
def draw():
background(50,80,150);fill(2)
for i in range(8**5):
y=i/200;x=i%200;h=H(x,y)
square(x+(H(x+1,y)-h)*5000,y+(H(x,y+1)-h)*5000,4)
```

filter(BLUR,2)

ぼかし





Koji Saito @KojiSaito · Aug 15

...

#つぶやき Processing

W=500

```
def setup():size(W,250)
```

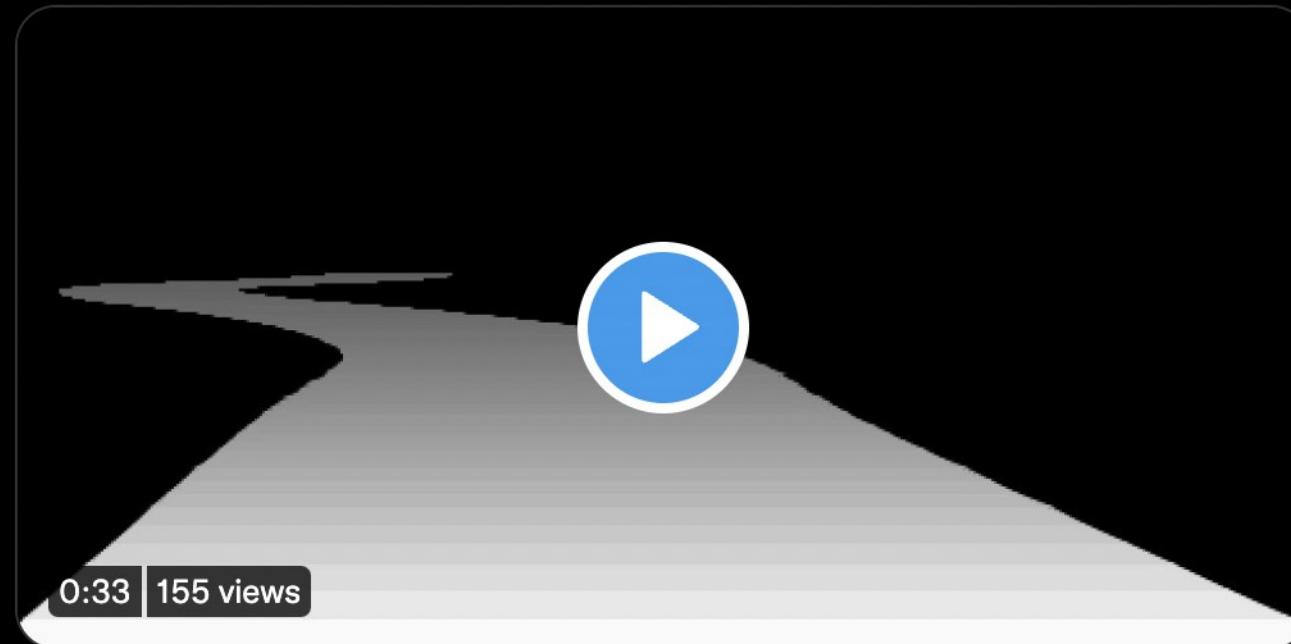
```
def draw():
```

```
    clear();f=frameCount;u=0;c=(noise(f*.01)-.5)*.2;p=0;q=W;v=300
```

```
    for t in range(300):u+=
```

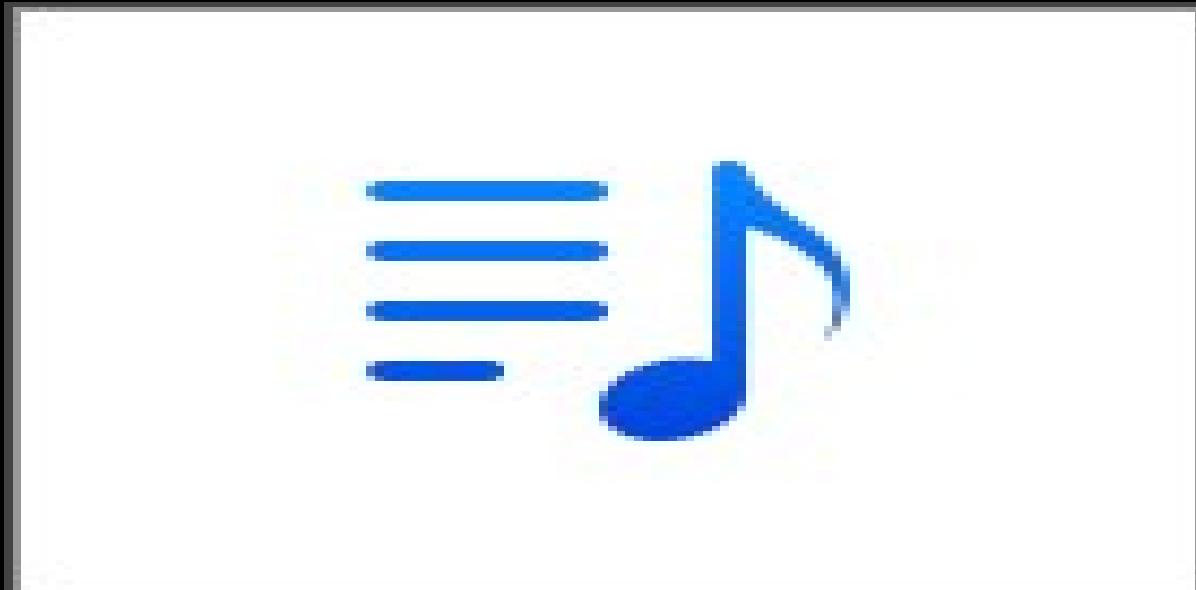
```
(noise((t+f)*.01)-.3)*.2;z=t*.1+1;s=1/z;y=100+150*s;x=W*(u-c)*s-
```

```
30;w=x+W/z+30;stroke(v);fill(v);quad(p,v,x,y,w,y,q,v);p=x;q=w;v=y
```



```
W=500
def setup():size(W,250)
def draw():
    clear();f=frameCount;u=0;c=(noise(f*.01)-.5)*.2;p=0;q=W;v=300
    for t in range(300):u+=(noise((t+f)*.01)-.3)*.2;z=t*.1+1;s=1/z;y=100+150*s;x=W*(u-c)*s-30;w=x+W/z+30;stroke(v);fill(v);quad(p,v,x,y,w,y,q,v);p=x;q=w;v=y
```

```
W=500
def setup():
    size(W,250)
def draw():
    clear();f=frameCount;
    u=0;
    c=(noise(f*.01)-.5)*.2;
    p=0;q=W;v=300
    for t in range(300):
        u+=(noise((t+f)*.01)-.3)*.2;
        z=t*.1+1;
        s=1/z;
        y=100+150*s;
        x=W*(u-c)*s-30;
        w=x+W/z+30;
        stroke(v);
        fill(v);
        quad(p,v,x,y,w,y,q,v);
        p=x;q=w;v=y
```





Koji Saito  
@KojiSaito

...

#つぶやき Processing

R=random;N=noise

F=lambda r,t:[r\*cos(t)+250,r\*sin(t)+125]

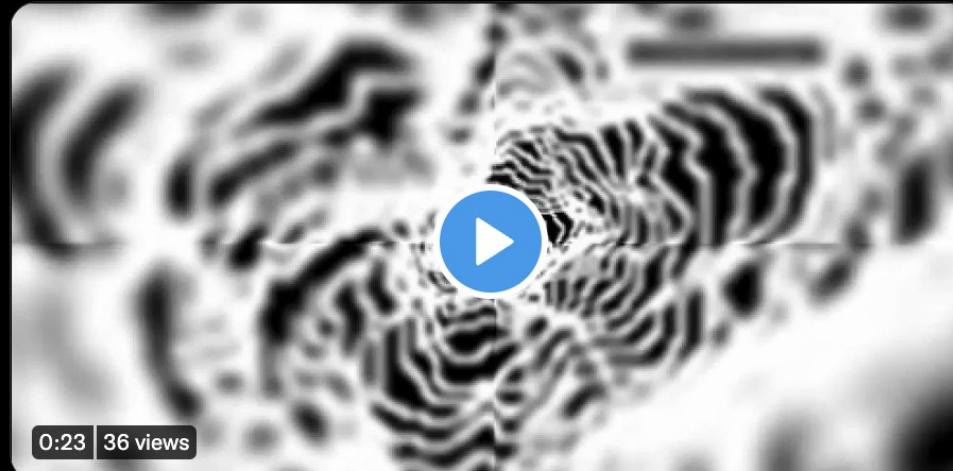
def setup():size(500,250);clear();stroke(-1)

def draw():

```
f=frameCount*.1;x,y=F(N(f,f)*50,f);copy(get(),20,10,46  
0,230,0,0,500,250)
```

```
for i in
```

```
range(628):t=i*.01+f;u,v=F(N(t,f)*40,t);line(x,y,u,v);x,y  
=u,v
```



0:23 | 36 views

```
R=random;N=noise  
F=lambda r,t:[r*cos(t)+250,r*sin(t)+125]  
def setup():size(500,250);clear();stroke(-1)  
def draw():  
f=frameCount*.1;x,y=F(N(f,f)*50,f);copy(get(),20,10,460,230,0,0,500,250)  
for i in range(628):t=i*.01+f;u,v=F(N(t,f)*40,t);line(x,y,u,v);x,y=u,v
```

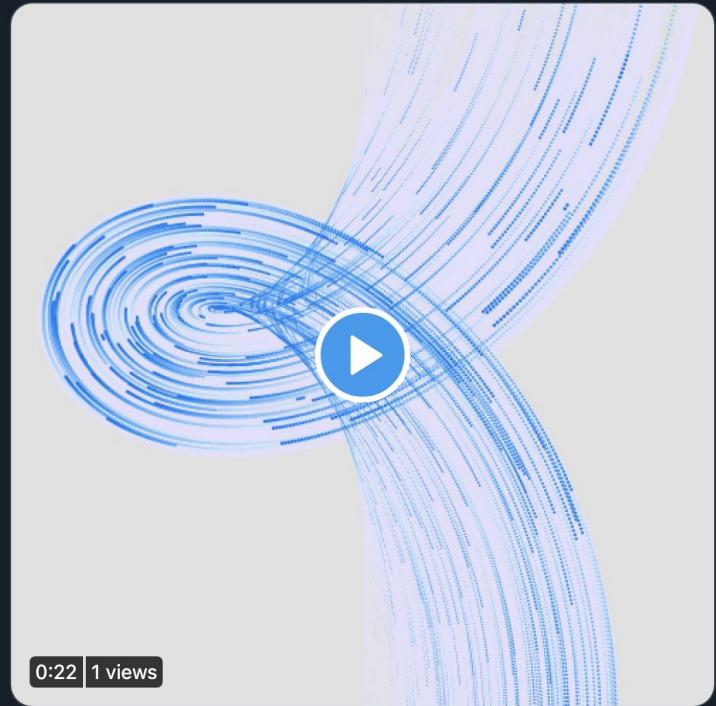




Koji Saito @KojiSaito · 46s

...

#つぶやきProcessing  
R=random  
C=Circle  
def setup():size(500,700);noStroke();  
F=lambda:[R(3),R(.1)\*.01+.01,R(250)]  
P=[]  
def draw():  
  fill(-1,5);C(0,0,3000);fill(10,120,255)  
  for e in P:  
    e[0]+=e[1];t=e[0];r=e[2]/(1+t\*t\*.01)  
    C(r\*cos(t)+250,r\*sin(t)+t\*100,.2+r\*.01)  
  P[:]+=[F()]



0:22 | 1 views

```
R=random  
C=circle  
def setup():size(500,700);noStroke();  
F=lambda:[R(3),R(.1)*.01+.01,R(250)]  
P=[]  
def draw():  
    fill(-1,5);C(0,0,3000);fill(10,120,255)  
    for e in P:  
        e[0]+=e[1];t=e[0];r=e[2]/(1+t*t*.01)  
        C(r*cos(t)+250,r*sin(t)+t*100,.2+r*.01)  
    P[:]+=[F()]
```



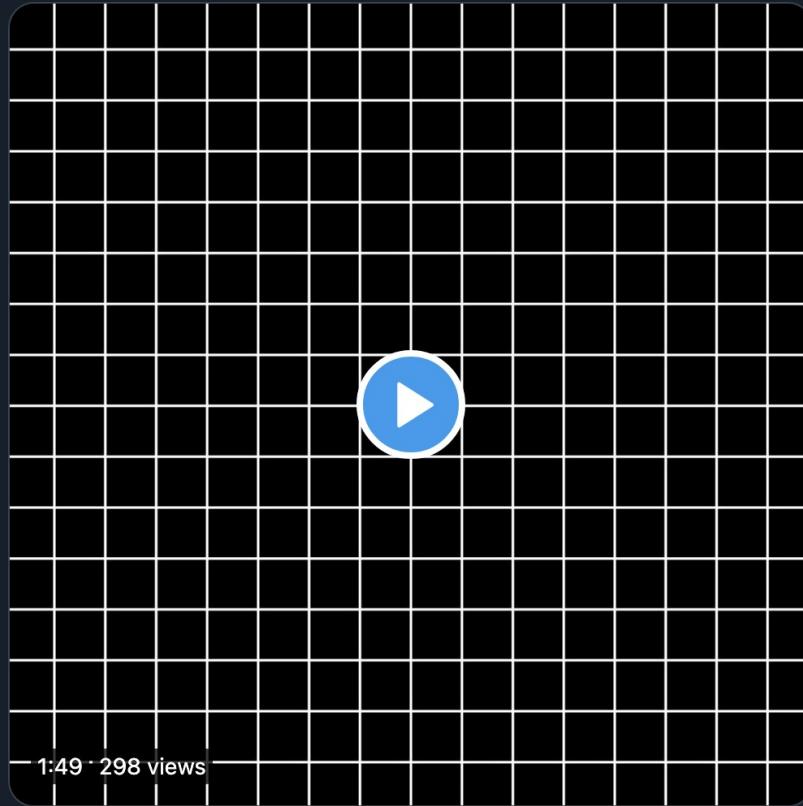


Koji Saito @KojiSaito · Aug 15

...

#つぶやき Processing

```
P=[0]*256;C=[0];K=20
def setup():size(321,321);stroke(-1);colorMode(HSB,8)
def draw():
    for i in range(256):x=i%16;y=i/16;fill(P[i]);square(x*K,y*K,K)
def mouseClicked(e):i=e.x/K+e.y/K*16;P[i]=C[0]
def keyTyped():t=ord(key);C[0]=color(t-48,(t!=57)*8,8)*(t!=48)
```



```
P=[0]*256;C=[0];K=20
def setup():size(321,321);stroke(-1);colorMode(HSB,8)
def draw():
    for i in range(256):x=i%16;y=i/16;fill(P[i]);square(x*K,y*K,K)
def mouseClicked(e):i=e.x/K+e.y/K*16;P[i]=C[0]
def keyTyped():t=ord(key);C[0]=color(t-48,(t!=57)*8,8)*(t!=48)
```



# まとめ

## 前半:Processing の紹介

- ・Java モード、Python モード、p5js
- ・関数名などは同じ
- ・静止画の生成例
- ・動画の生成方法

## 後半:Processing による作品の話

- ・#つぶやきProcessing について
- ・作品例

その他:質問などあれば @KojiSaito まで連絡を下さい



# まとめ

## 前半:Processing の紹介

- ・Java モード、Python モード、p5js
- ・関数名などは同じ
- ・静止画の生成例
- ・動画の生成方法

## 後半:Processing による作品の話

- ・#つぶやき Processing について
- ・作品例

その他:質問などあれば @KojiSaito まで連絡を下さい



終