

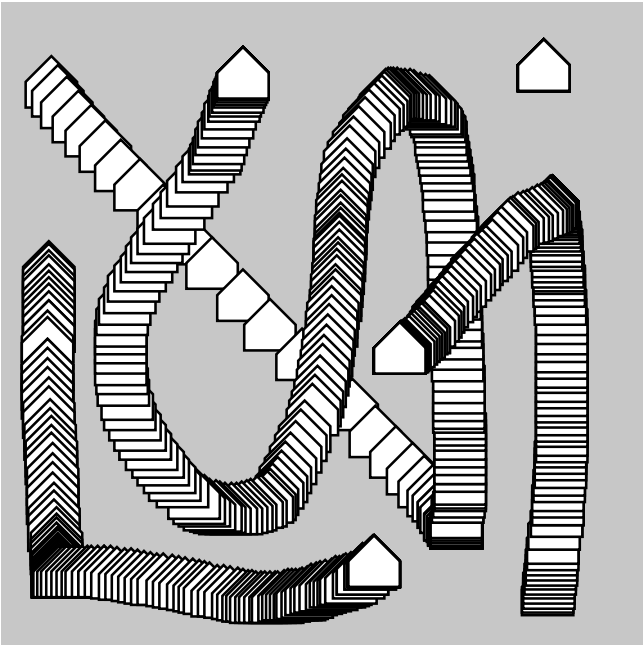
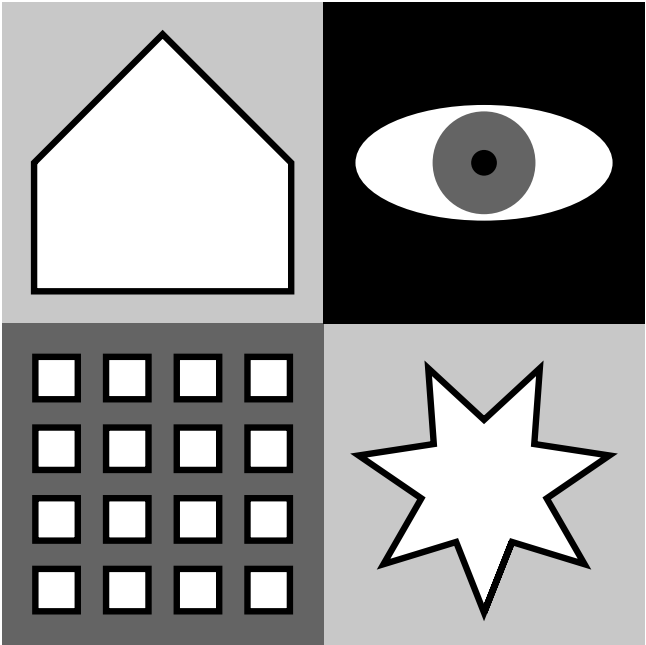
desenho()

```
"""
Exemplo de desenho interativo com setup() e draw().
"""

def setup(): # função executada uma vez no começo
    size(500, 500) # define área de desenho
    background(100) # limpa a área com fundo cinza escuro
    strokeWeight() # ajusta espessura do traço

def draw(): # função executada cerca de 60 vezes por segundo
    if mousePressed: # se o mouse estiver pressionado
        casinha(mouseX, mouseY, 40) # desenha no x,y do mouse

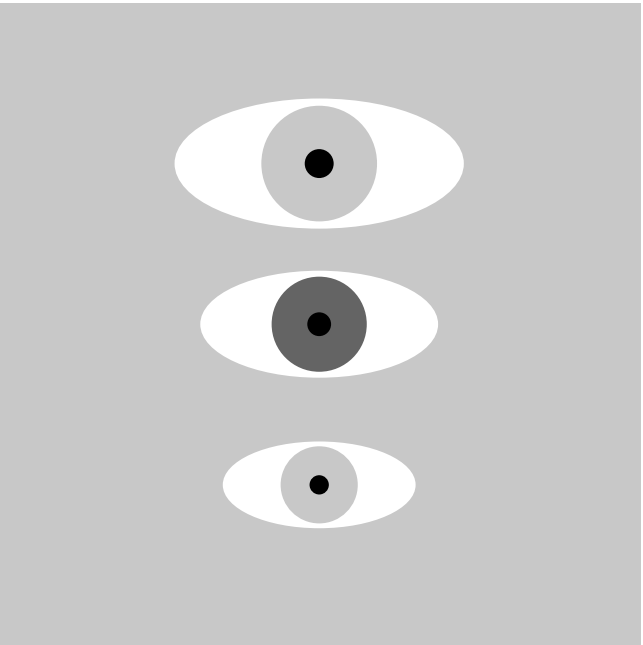
def casinha(x, y, tamanho):
    """
    Desenha casinha em x, y com largura e altura 'tamanho'.
    """
    metade = tamanho / 2
    pushMatrix() # preserva o sistema de coordenadas atual
    translate(x, y) # translada a origem das coordenadas
    beginShape() # começa a desenhar a forma/poligono
    vertex(0, -metade)
    vertex(-metade, 0)
    vertex(-metade, metade)
    vertex(metade, metade)
    vertex(metade, 0)
    endShape(CLOSE) # encerra fechando no 1o vértice
    popMatrix() # retorna o sistema de coordenadas anterior
```



```
"""
Exemplo de uso de uma função que desenha um olho.
"""

def setup():
    size(500, 500)
    background(0) # fundo preto
    cinza_escuro = color(100) # cria um valor de cor
    olho(width / 2, width / 2, width * .8, cinza_escuro)

def olho(x, y, largura, cor):
    """Desenhe olho na posição x, y com largura e cor."""
    pushStyle() # preserva os atributos gráficos atuais
    noStroke() # desliga o traço
    fill(255) # preenchimento branco
    ellipse(x, y, largura, largura * .45) # desenha branco
    fill(cor) # cor de preenchimento do parâmetro
    circle(x, y, largura * .4) # desenha iris
    fill(0) # preenchimento preto
    circle(x, y, largura * .1) # desenha pupila
    popStyle() # retorna aos atributos gráficos anteriores
```



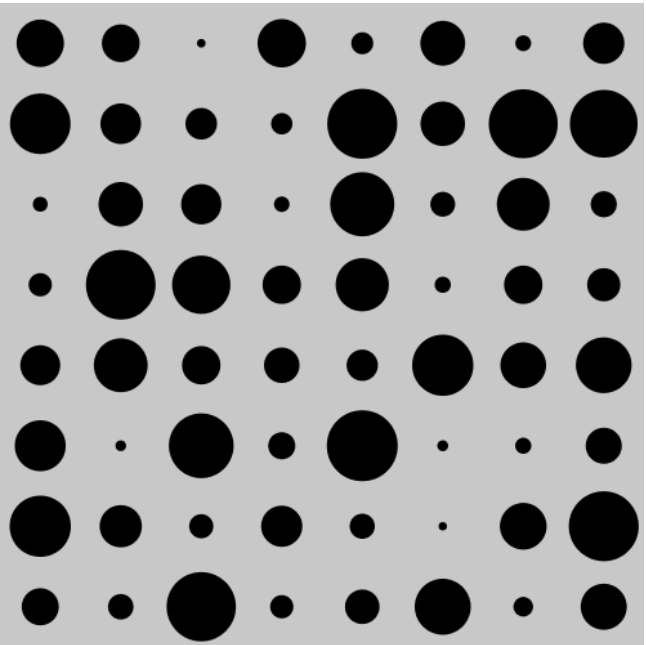
```
"""
Exemplo de uso de uma função que desenha uma grade.
"""

def setup():
    size(500, 500)
    noStroke()
    fill(0)
    noLoop()

def draw():
    background(200)
    grade(250, 250, 8, width, rnd_circ=True)

def grade(x_centro, y_centro, n, tam_total, rnd_circ=False):
    tam = tam_total / n
    desloc = (tam - tam_total) / 2.
    for i in range(n):
        x = x_centro + desloc + tam * i
        for j in range(n):
            y = y_centro + desloc + tam * j
            if rnd_circ:
                # no Processing random(inicio, final_ni)
                circle(x, y, random(tam * .1, tam * .9))
            else:
                square(x, y, tam * .75)

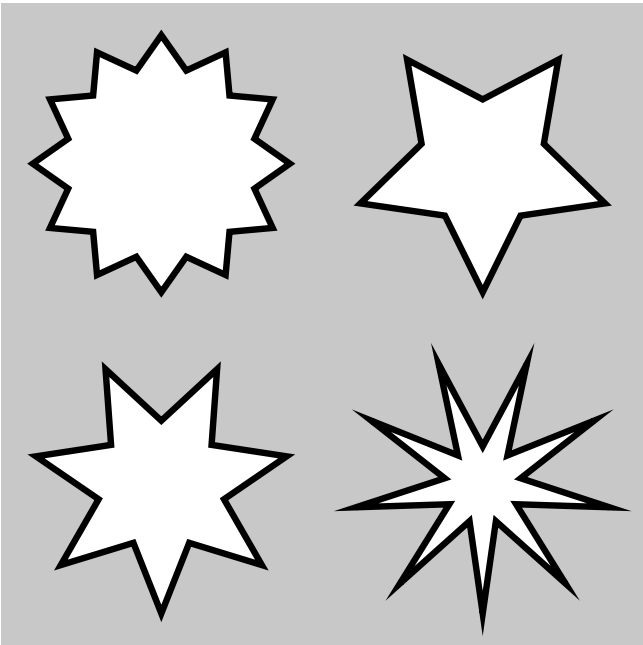
def keyPressed():
    saveFrame("grade.png")
    redraw()
```



```
"""
Desenhe quatro estrelas diferentes
"""

def setup():
    size(500, 500) # área de desenho
    strokeWeight(5) # espessura do traço
    estrela(125, 125, 12, 100, 75) # estrela de 12 pontas
    estrela(375, 125, 5, 100, 50) # estrela de 5 pontas
    estrela(125, 375, 7, 100, 50) # estrela de 7 pontas
    estrela(375, 375, 9, 100, 30) # estrela de 9 pontas

def estrela(x_centro, y_centro, num_pontas, raio_a, raio_b):
    """
    Desenha uma estrela em x_centro, y_centro com num_pontas
    raio_a e raio_b são raios dos pontos internos e pontas.
    """
    n = num_pontas * 2 # pontos totais são dobro de pontas
    ang = radians(360. / n) # 360 graus / n em radianos
    beginShape() # começa a desenhar a forma
    for i in range(n):
        r = raio_a if i % 2 == 0 else raio_b
        x = x_centro + sin(ang * i) * r
        y = y_centro + cos(ang * i) * r
        vertex(x, y)
    endShape(CLOSE) # encerra fechando no primeiro ponto
```



```
"""
Para executar o código instale Processing Modo Python.
Saiba mais em https://desenho.lugaralgun.com

Contribuíram para a impressão inicial:
Adri Patomima, Advan Shumitski, André Burnier, Bernardo Fontes,
Fábio C. Barriounevo da Luz, Juan Lopes, Lucia Dossin, Monica Rizzolli,
Otavio Carneiro, Rodolfo Viana, Thais Fassina, Yorik van Hevre.
(até agora... confira no site!)

Agradecimentos especiais:
Décio Ottoni de Almeida, Bernardo Fontes, Monica Rizzolli

Copyright (c) 2019 Alexandre B A Villares
Texto e imagens: CC-BY-SA-NC v4.0 / Código: GPL v3.0
"""

def capa():
    # width é a largura da área de desenho
    metade, quarto = width / 2, width / 4
    noStroke() # desliga o traço
    fill(100) # preenchimento cinza escuro
    rect(0, metade, metade, metade) # fundo para casinha
    fill(0) # preenchimento preto
    rect(metade, 0, metade, metade) # fundo preto para o olho
    olho(metade + quarto, quarto, quarto, 200)
    fill(255) # preenchimento branco
    stroke(0) # traço preto
    strokeWeight(5) # espessura do traço
    casinha(quarto, quarto, 200)
    grade(quarto, metade + quarto, 4, 220)
    estrela(metade + quarto, metade + quarto, 7, 100, 50)
```

#0_out_2019 desenho()