

# 冠军的即兴show





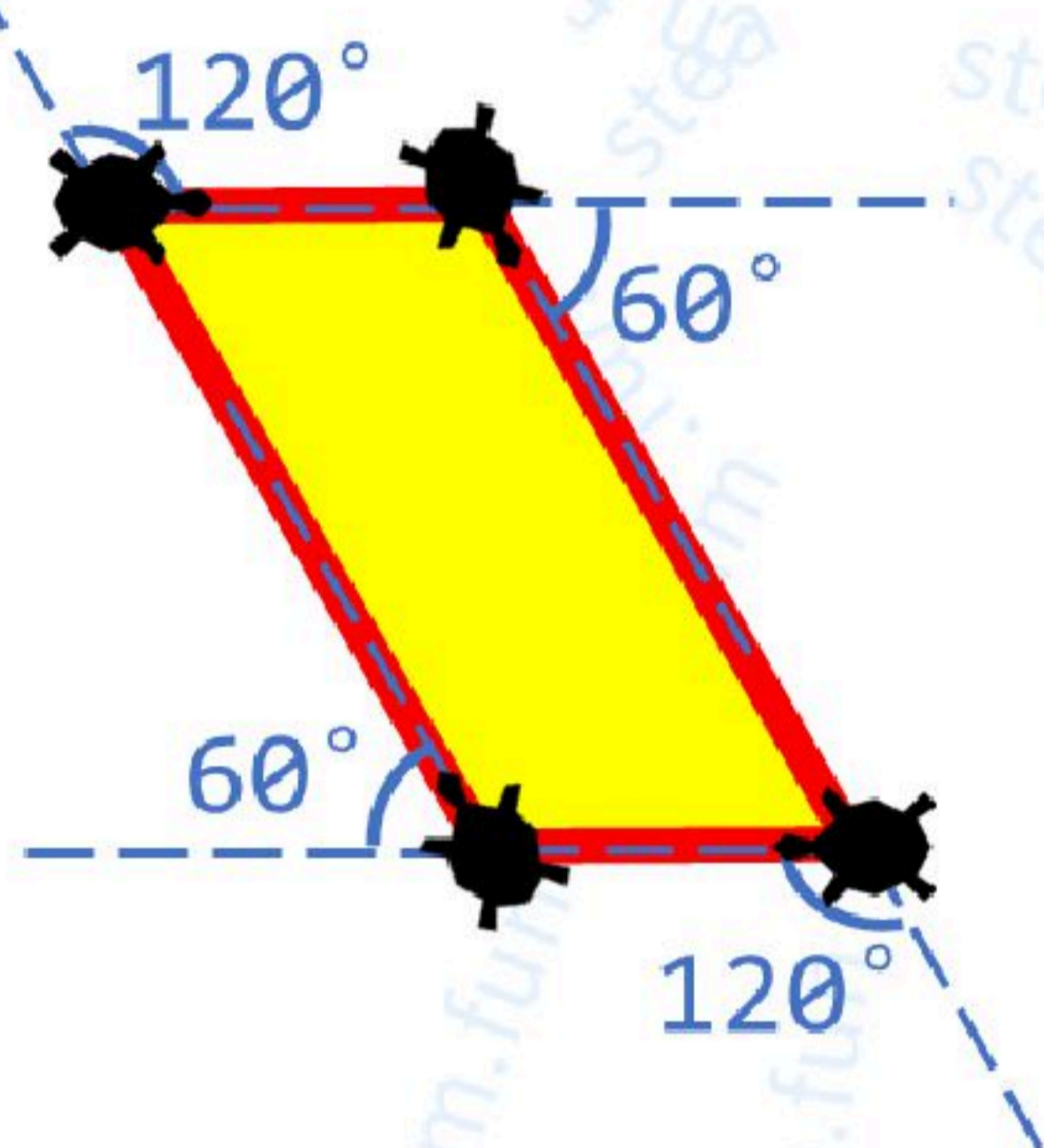
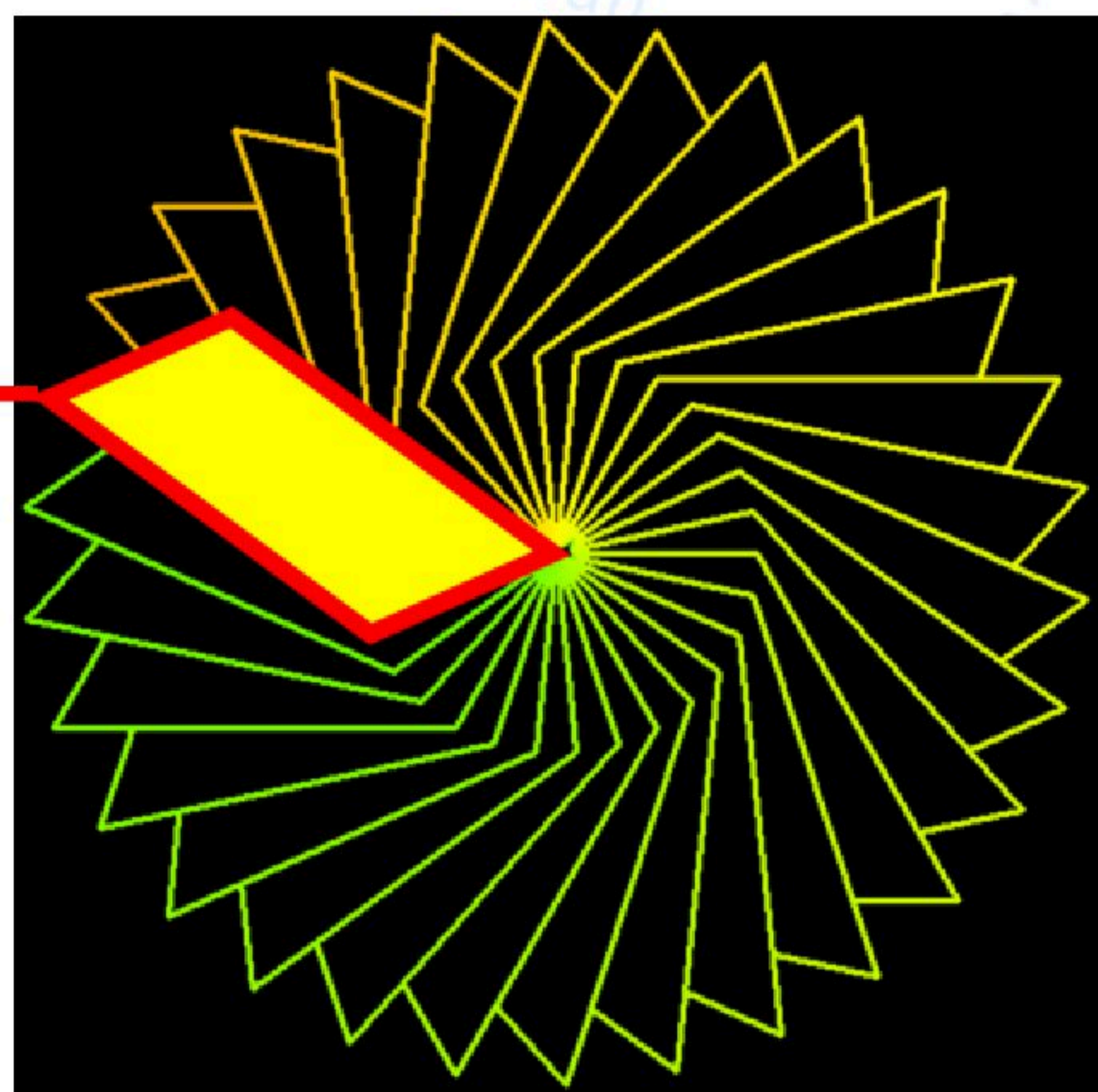
# 1. 探索新知

## 1.1

### 基础图形

基础图形找到啦！快来定义绘制平行四边形的函数吧！

基础图形



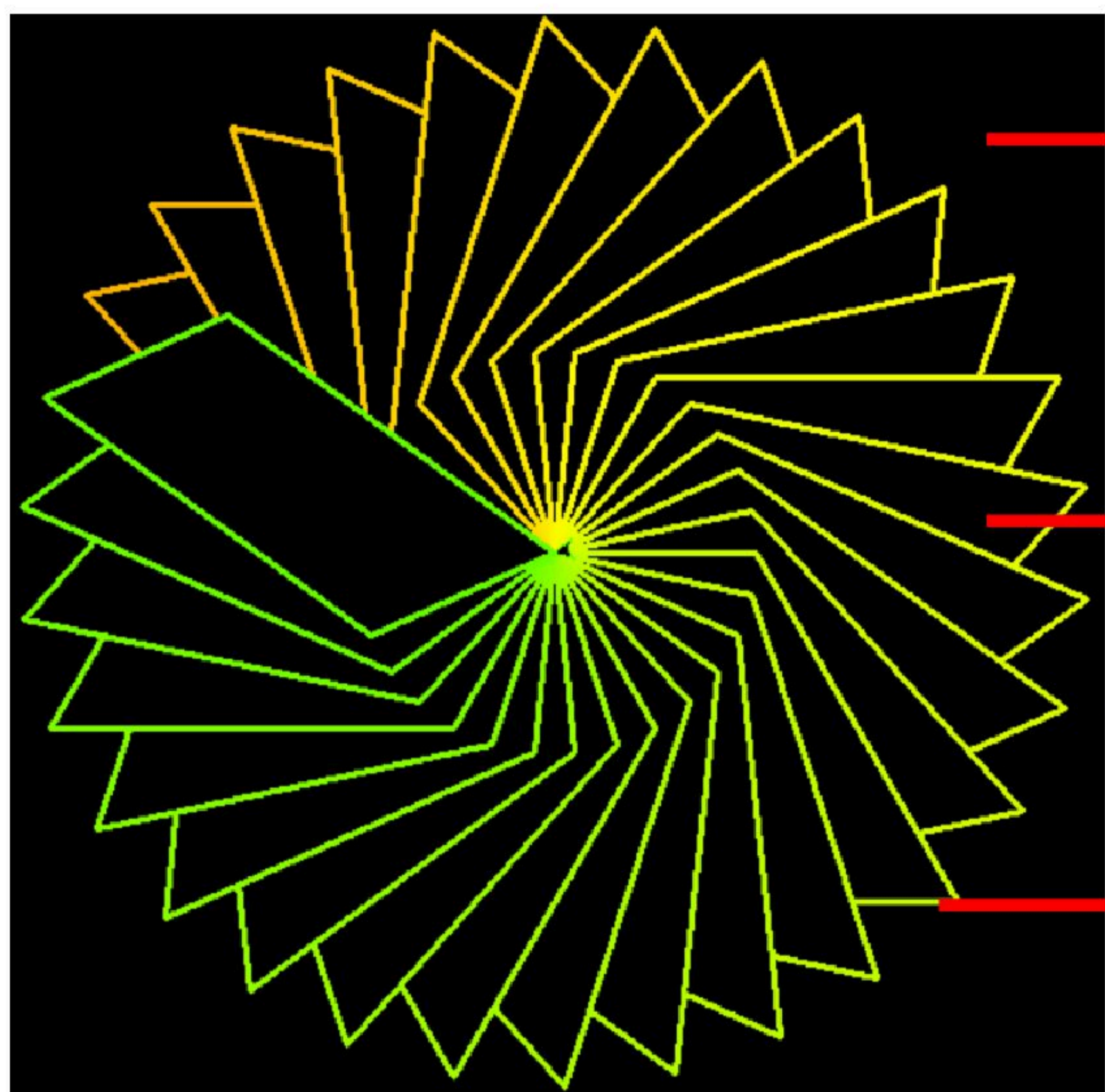
```
def pingxing():  
    t.forward(100)  
    t.right(60)  
    t.forward(200)  
    t.right(120)  
    t.forward(100)  
    t.right(60)  
    t.forward(200)  
    t.right(120)
```

化简为：

```
def pingxing():
```

## 1.2

## 渐变色



背景色：黑色

填充色：黑色

画笔色：渐变色



颜色渐变的效果非常酷炫，如何实现呢？

### t.pencolor(参数)

参数：颜色名称字符串或 RGB 元组

我们平时用的 RGB 色彩模式不能方便地实现颜色渐变效果，这里我们用一个新的 HSV 色彩模式，不用深入理解，我们学习实现方法即可~

colorsys模块

简单了解一下



### 1. 导入colorsys模块

```
import colorsys
```

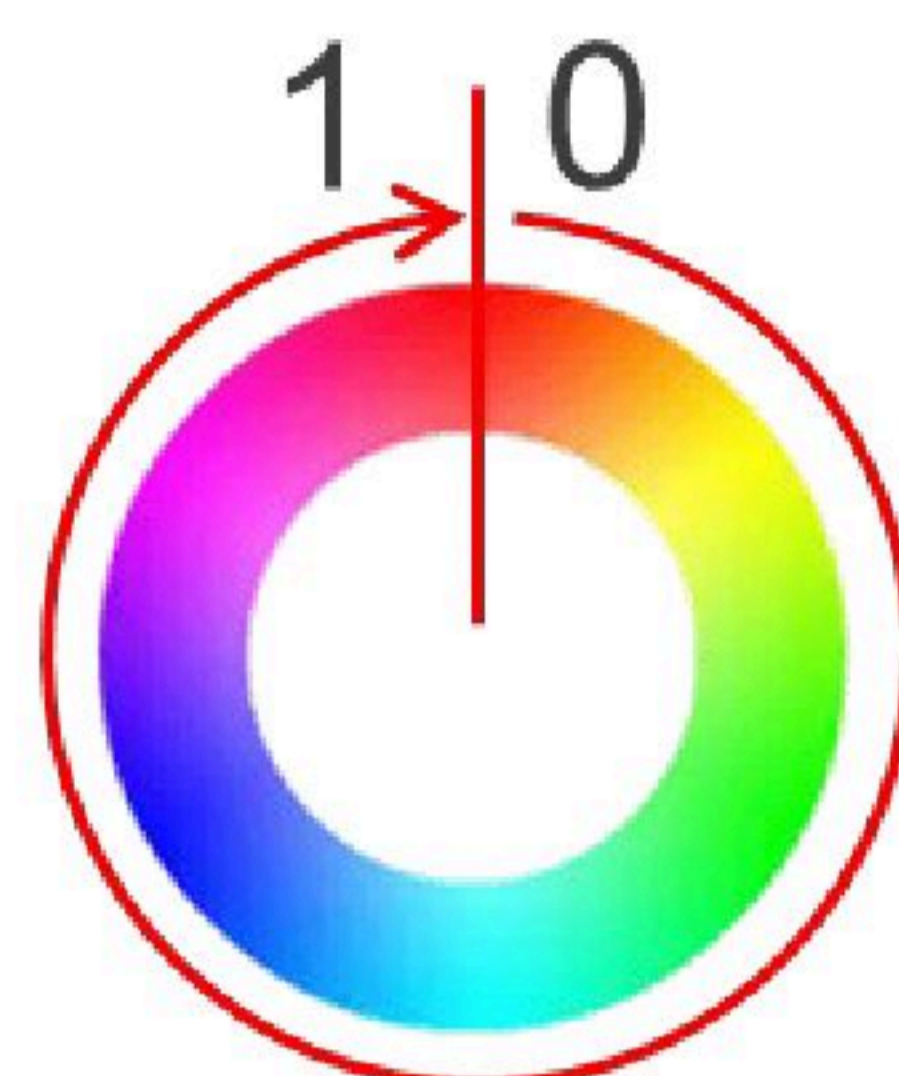
## 1.2

## 渐变色

### 2. 生成某个颜色

```
c = colorsys.hsv_to_rgb(h,1,1)
```

通过改变h的值，改变颜色。h的值在0~1之间。



h值从0→1，对应到上面的色环中，就是跟着箭头指的方向颜色渐变，所以我们限制h的范围从0到1



那如何编程实现 h 从 0→1 的递增呢？

#### ① 设置h的初始值

```
h = 0
```

#### ② 设置不断渐变的颜色

```
while True:  
    c = colorsys.hsv_to_rgb(h,1,1)  
    h += 0.005  
    if h > 1 :  
        h = 0
```

# 1.3

## 炫彩动图

### 1. 导入所需模块

```
import turtle as t
import colorsys
```

### 2. 定义绘制平行四边形的函数

```
def pingxing():
    for j in range(2):
        t.forward(100)
        t.right(60)
        t.forward(200)
        t.right(120)
```

### 3. 初始化设置

```
_____ # 设置背景颜色
_____ # 设置画笔粗细
_____ # 设置h初始值
```

### 4. 循环部分

```
while True:
    c = colorsys.hsv_to_rgb(h,1,1) # 将渐变色依次存在c中
    h += 0.005 # h递增
    if h > 1 :
        h = 0
    _____ # 设置画笔颜色
    _____ # 设置填充颜色
    _____ # 开始填充
    _____ # 绘制平行四边形
    _____ # 停止填充
    _____ # 旋转一定角度，准备下一轮绘制
```

## 5. 停止画笔绘制，绘图窗口不关闭

```
t.done()
```



运行看看效果吧！会发现作品的运行速度太慢了，没有出现炫彩动图的效果。如何完善作品呢？`speed()`方法可以吗？



用`speed()`方法加快了画笔的绘制速度，的确效果好了一些，那怎样才能实现演示视频中的效果呢？

### tracer()方法

- 接收一个整数作为参数时，可以用于设置绘图的速度
- 如果参数设置为大于0的整数，例如 `turtle.tracer(10)`，设置绘制的速度。

## 完整代码

```
import turtle as t
import colorsys

# 绘制平行四边形
def pingxing():
    for j in range(2):
        t.forward(100)
        t.right(60)
        t.forward(200)
        t.right(120)

t.tracer(10)
t.bgcolor('black')
t.pensize(3)
h = 0

while True:
    c = colorsys.hsv_to_rgb(h,1,1)
    h += 0.005
    if h > 1 :
        h = 0
    t.pencolor(c)
    t.fillcolor('black')
    t.begin_fill()
    pingxing()
    t.end_fill()
    t.right(12)

t.done()
```



## 2. 强化练习

1. 为使下面程序代码正常执行，横线处应填写的代码是（ ）

A. import random

B. import Random

C. from random import random

D. import random as Random

```
# 随机函数的使用
```

```
print(random.random())
```

2. 以下选项中，哪一个不属于函数的作用？（ ）

A. 提高代码的执行速度

B. 提高代码的重复利用率

C. 增强代码的可读性

D. 降低编程的复杂度

3. 以下用于绘制弧形的函数是？（ ）

A. turtle.seth( )

B. turtle.right( )

C. turtle.circle( )

D. turtle.fd( )

4. 小海龟正在学习有趣的绘图，小海龟正在使用 `turtle.speed(a)` 指令（`a` 的当前值为 3），小海龟想将画笔的运动速度设置为最快，让它立刻呈现绘制的结果，应将 `a` 的值更改为？（ ）

A. 0

B. 4

C. 5

D. 10



## 2. 强化练习

5. 如图所示，运行下列程序，请问输出结果是？（ ）

- A. 12
- B. 21
- C. 32
- D. 36

```
1.py - C:/Users/Administrator/Desktop/1.py (3.8.3rc1)
File Edit Format Run Options Window Help
a=1
b=2
a, b=b+1, a+a
print(a, b)
```

### 3. 术语箱

tracer

追踪

RGB

RGB 色彩模式

HSV (了解即可)

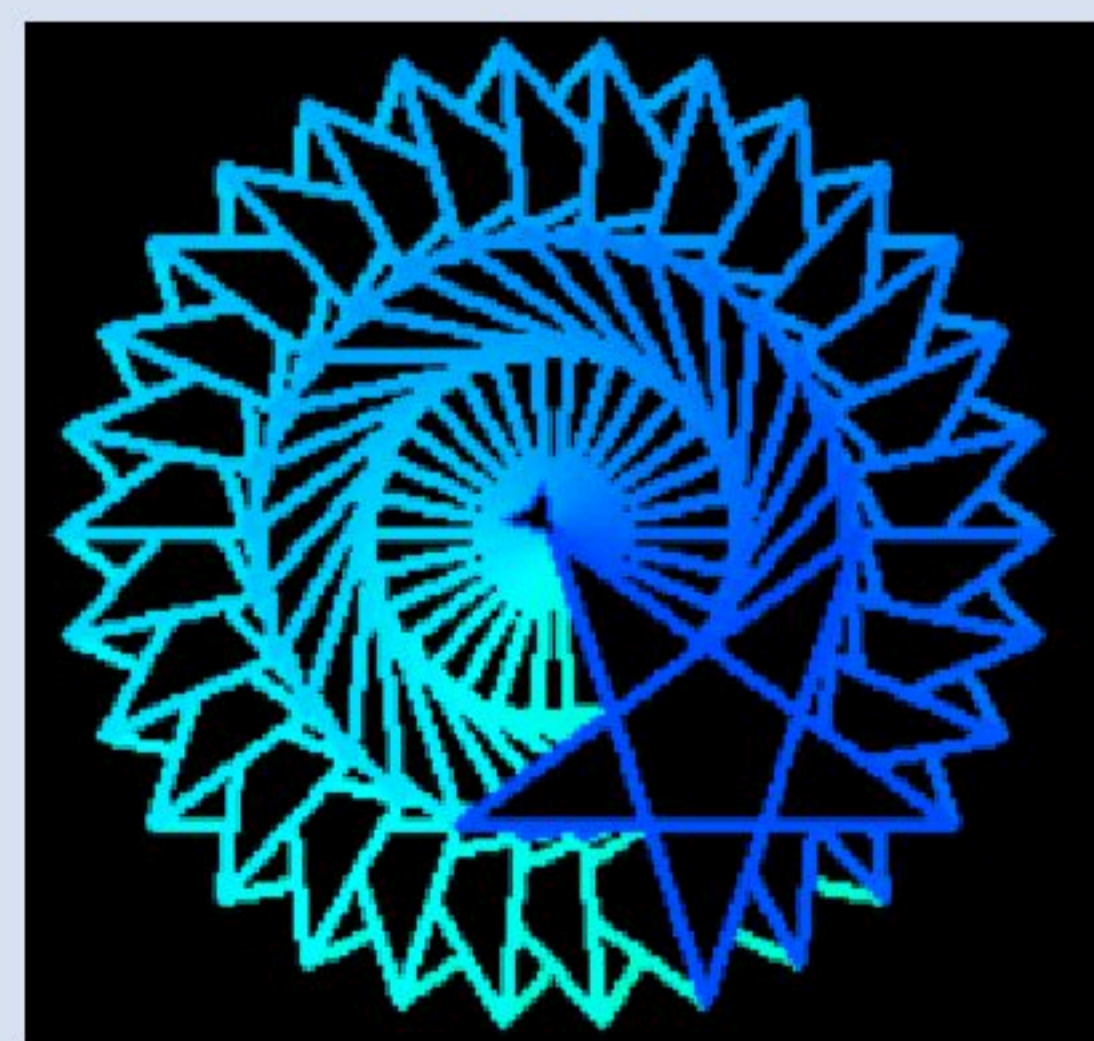
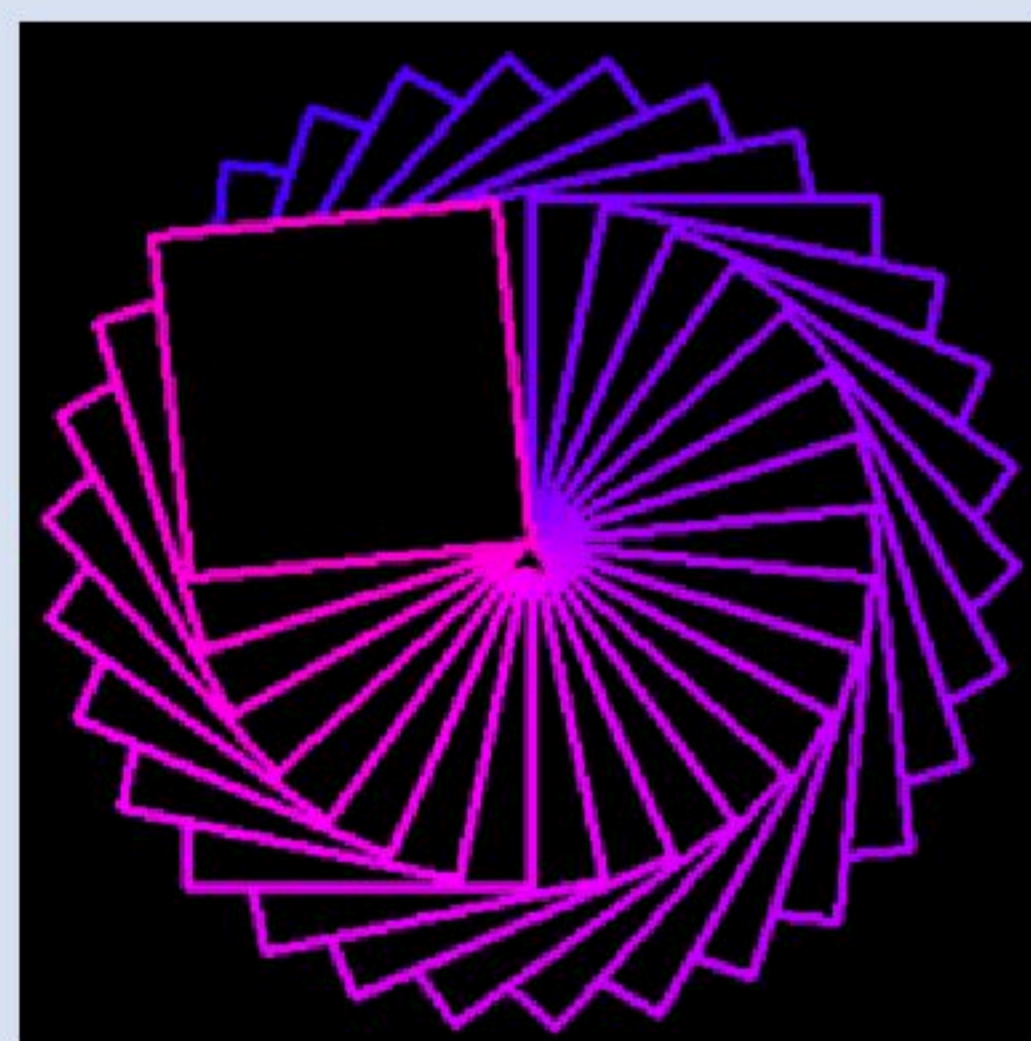
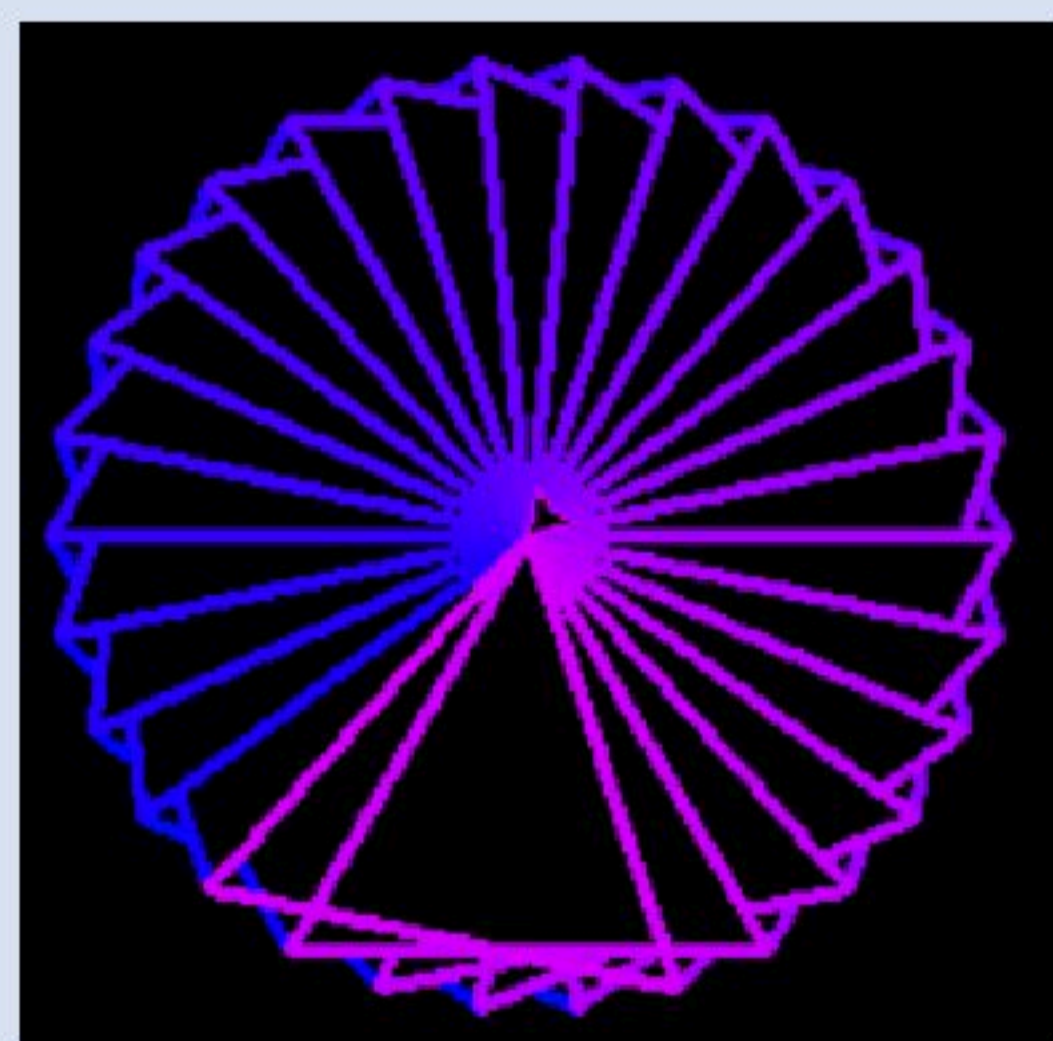
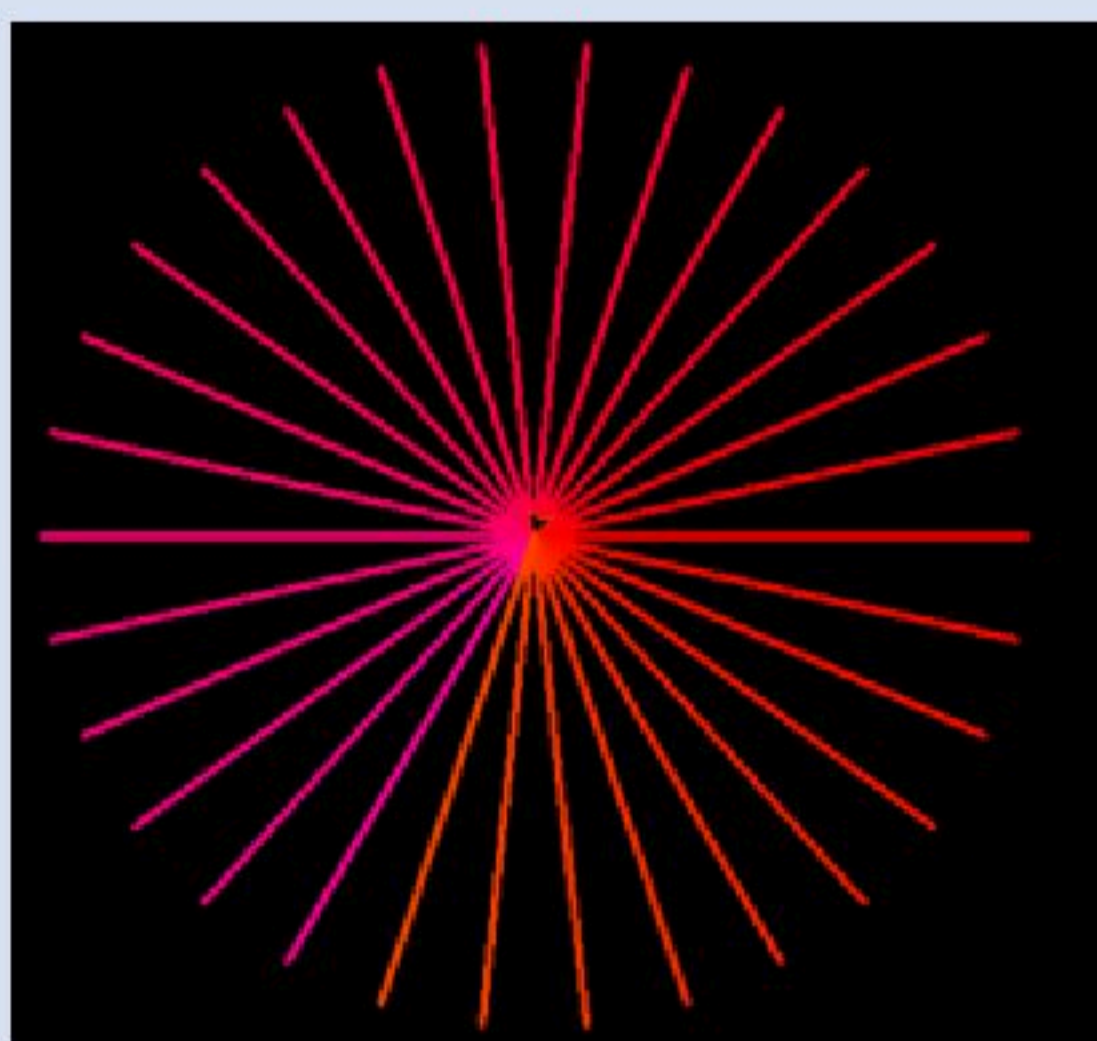
HSV 色彩模式

### 4. 课后挑战

#### 一、多样炫彩动图

选择绘制五角星、三角形、正方形或者线条等函数，用相同的方法完成自己的炫彩动图。

参考如下（答案不统一）：



#### 二、升级版视觉漩涡

将上节课的作品视觉漩涡进行升级，添加渐变色，加快绘制速度。

效果如下：

