

多彩图形争霸赛 (一)



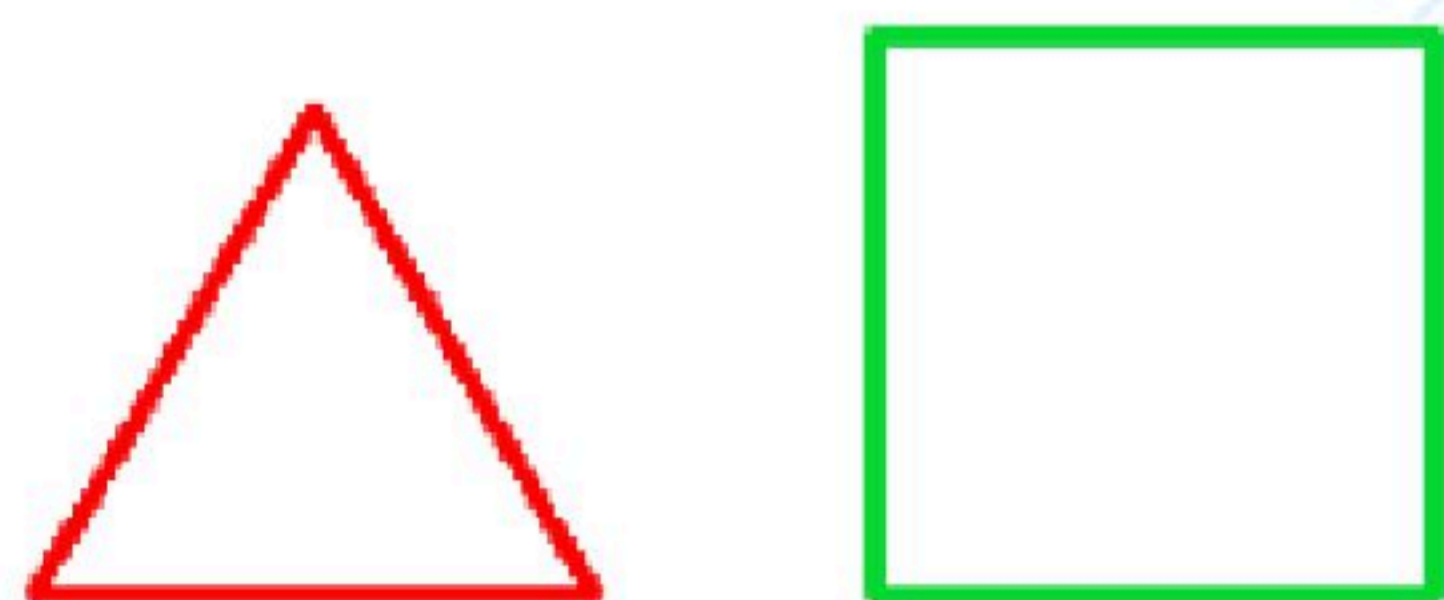


1. 探索新知

1.1

什么是函数

用scratch代码画出该图形，代码如右图所示



这这这这这---么长的代码，吓坏我老黑了！



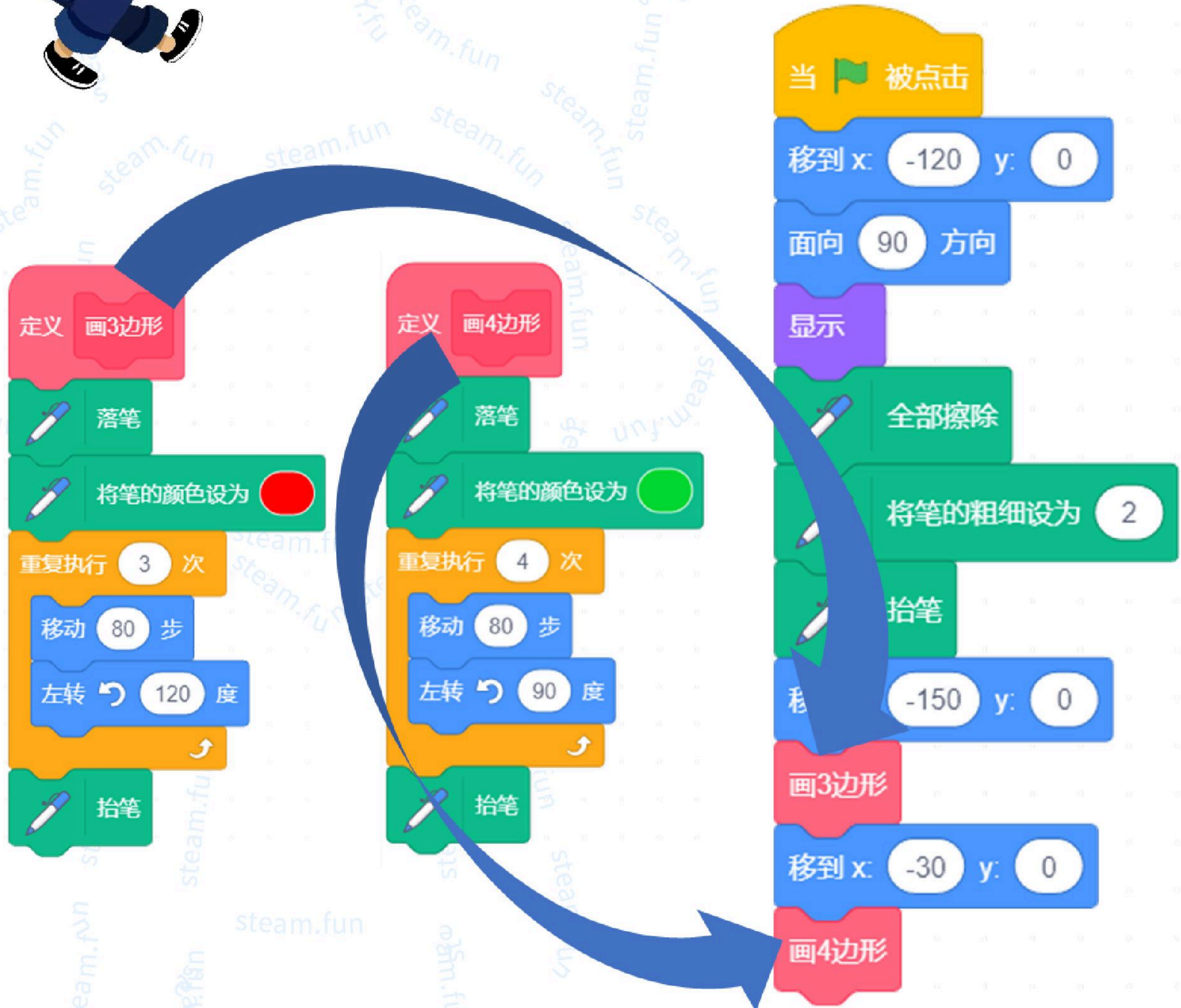
```
当 被点击
移到 x: -120 y: 0
面向 90 方向
显示
全部擦除
将笔的粗细设为 2
抬笔
移到 x: -150 y: 0
落笔
将笔的颜色设为 红色
重复执行 3 次
  移动 80 步
  左转 120 度
抬笔
移到 x: -30 y: 0
落笔
将笔的颜色设为 绿色
重复执行 4 次
  移动 80 步
  左转 90 度
抬笔
```

1.1

什么是函数



这段代码的确又长又不好理解，我来把它简化一下吧！



上面积木程序，把绘制三角形、正方形的积木块，打包起来，定义为一个函数，取名为“画3边形”、“画4边形”。如果还需要再画一个三角形或者正方形，只需要调用相应的函数，就可以了

这种方式可以让代码被重复利用，编写代码更快了，而且更简洁，容易让人理解

1.1

什么是函数



举了这么久的例子，现在我来总结一下函数的概念吧！
函数其实是组织好的、可以重复使用的、用来实现单一功能的代码



大概知道函数是个什么东西了，
那Python中的函数是个什么样子呢？

函数其实是我们的老熟人了，我们在先前的课堂中已经接触过一些Python自带的函数（内置函数）。

比如说我们想要实现“打印”这个功能，大家知道该用哪个函数实现么？

1.2

函数的定义



除了python自带的函数，我们能不能根据自己编写程序的需要，自己定义一个独一无二的函数呢？

首先，我们需要先定义一个函数，想象这个函数的名字、功能是什么

定义函数的语法

```
def 函数名():
```

```
    函数体
```

1.2

函数的定义

➤ 分析

1. def 的意思是定义 (define)
2. 函数名应该有意义, 能够清晰表达函数功能
3. 函数名后必须有(), 里面可以什么都不写, 也可以写参数 (下节课讲函数的参数)
4. ()后一定要写冒号 :, 下一行开始缩进, 表示所属关系, 缩进的代码就是函数要实现的功能, 叫做【函数体】

自己写个函数试试吧!



```
def MyFirstFun():  
    print('这是我创建的第一个函数')  
    print('我感到无比的激动')  
    print('在这里, 我要感谢CCTV, 感谢老师, 感谢同学')
```

现在你可以运行代码, 看看会发生什么?

1.3

函数的调用



截至目前, 咱们已经完成了【定义函数】。但定义函数只是将函数的内部功能封装起来, 安静地待在那里, 等待你的召唤出场!

1.3

函数的调用



那怎么调用函数呢？
很简单，喊出函数的名字即可！

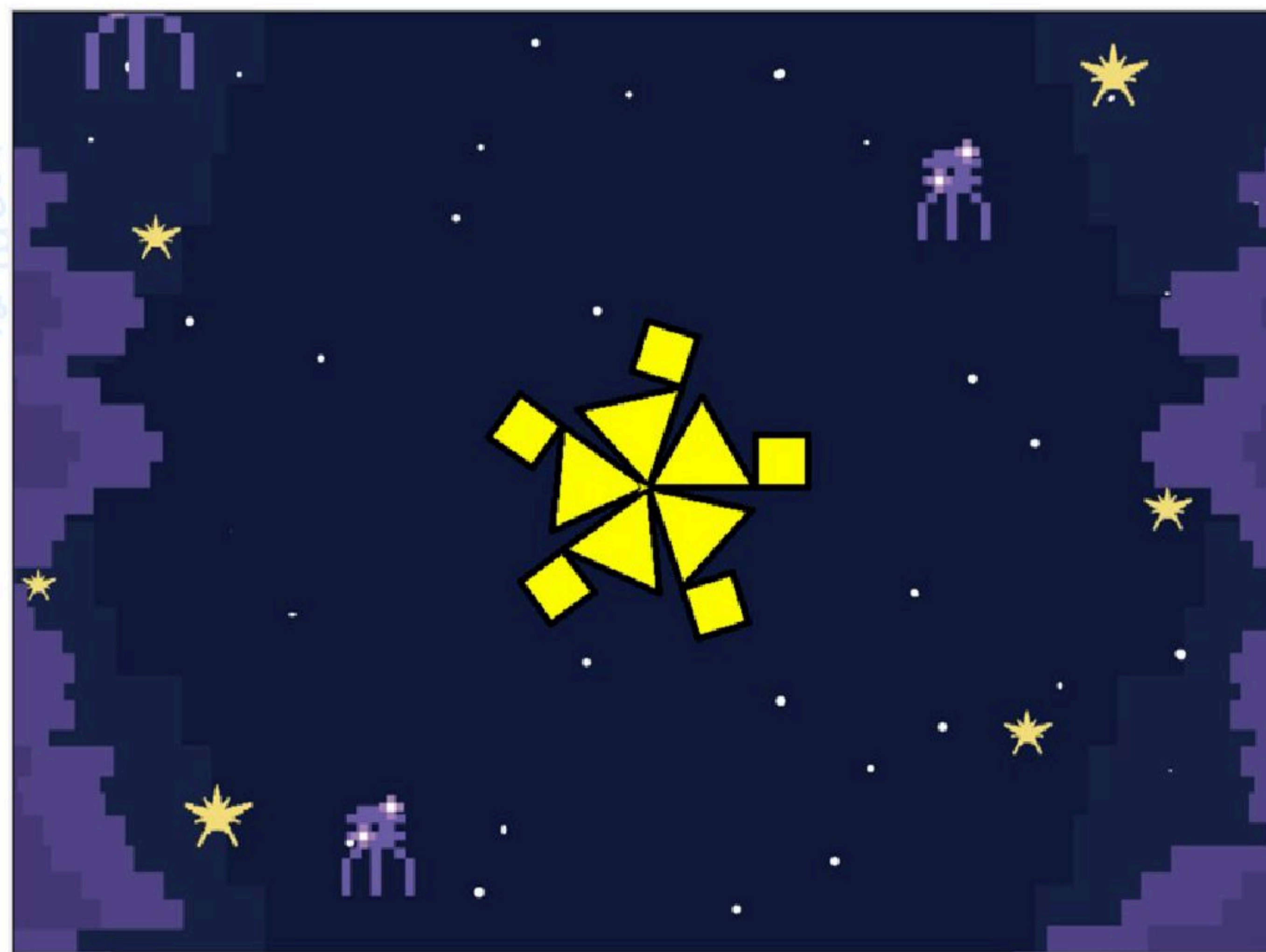
```
def MyFirstFun():
    print('这是我创建的第一个函数')
    print('我感到无比的激动')
    print('在这里，我要感谢CCTV，感谢老师，感谢同学')
MyFirstFun()
```

千万别漏了 ()

1.4

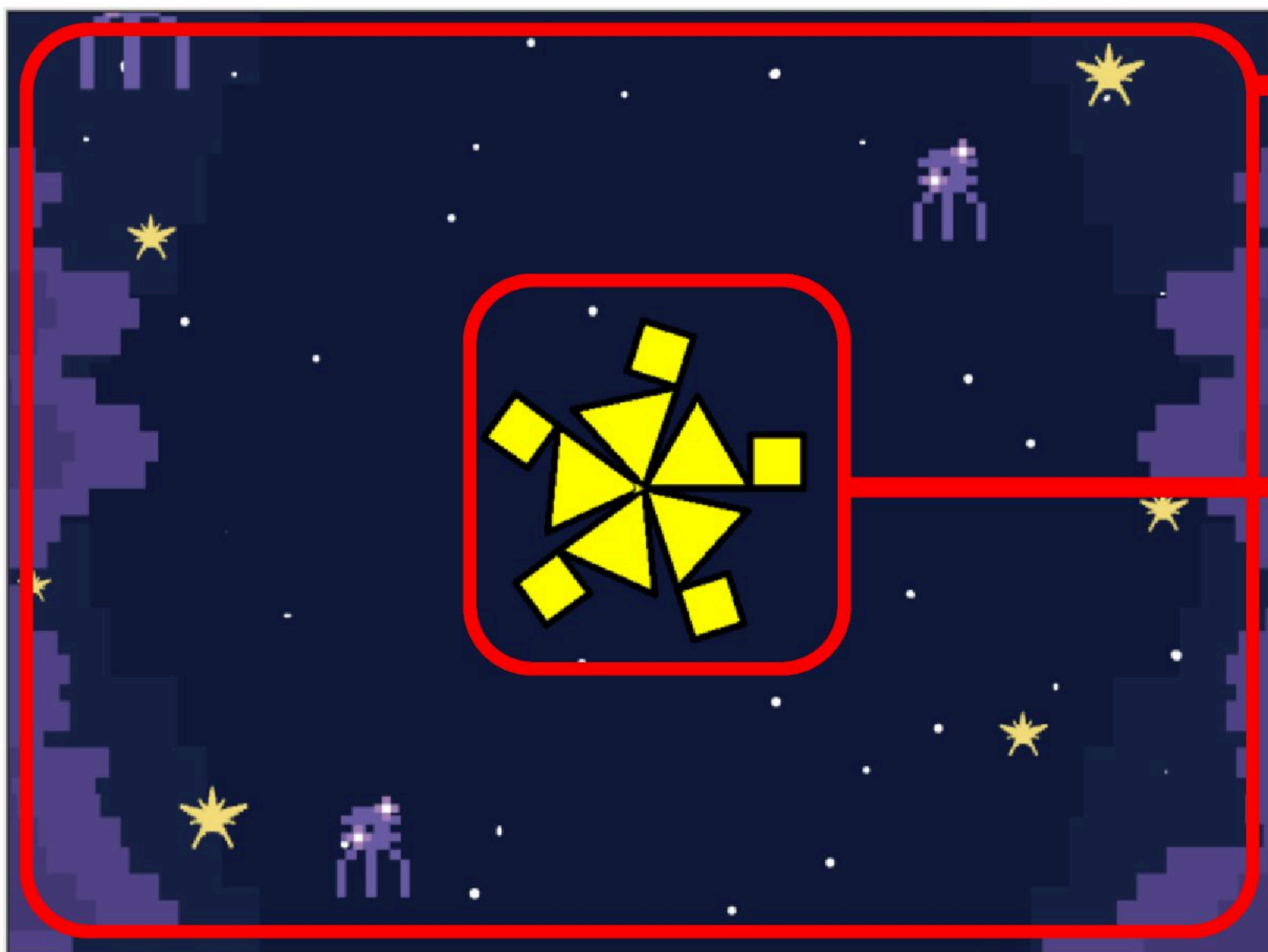
便捷的函数

多彩图形争霸赛进行到白热化的阶段，P仔的朋友球球正在台上比赛，P仔请来函数家族给球球助威。请定义函数并灵活调用，帮助球球快速完成右边的作品吧！



1.4

便捷的函数

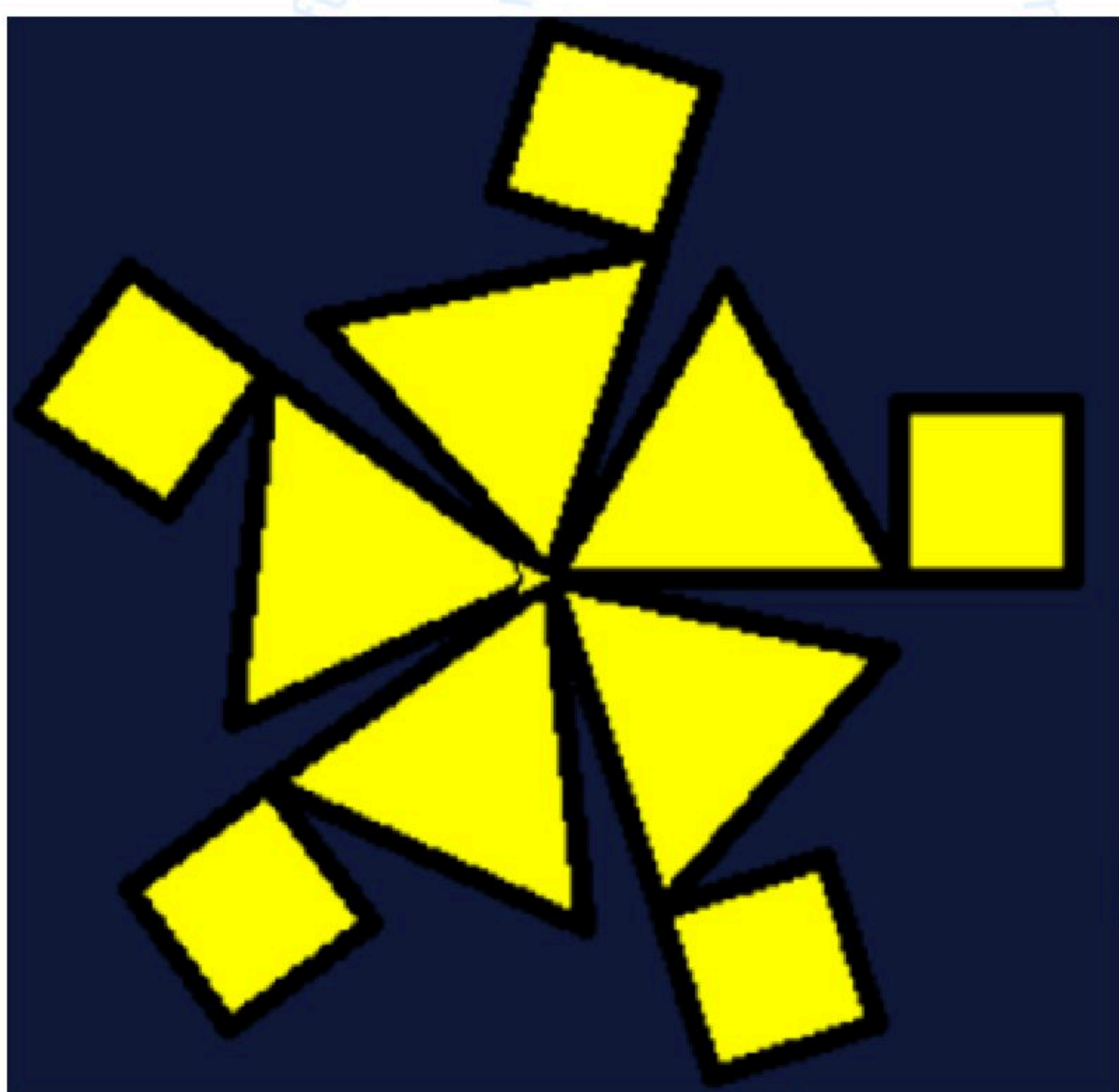


加载背景图片

绘制复杂图形



这个复杂图形是由哪些基础图形组成的呢？





我们可以把画三角形和正方形的代码打包成两个函数，方便理解和调用

```
import turtle as t

def side3():
    for i in range(3):
        t.forward(80)
        t.left(120)

def side4():
    for i in range(4):
        t.forward(40)
        t.left(90)
```

1. 创建绘图窗口



```
t.setup(960,720)
```

还记得用哪个函数创建绘图窗口吗？

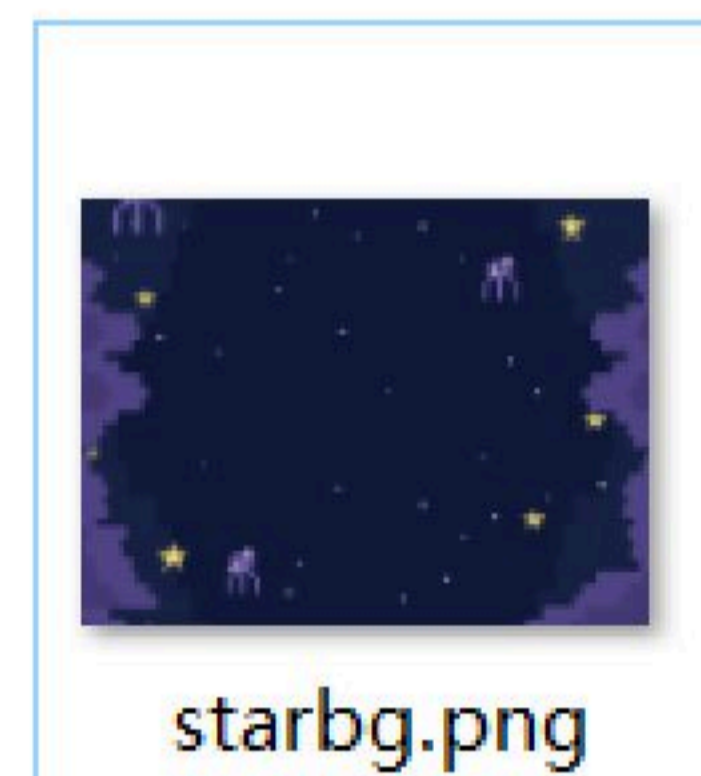


2. 加载背景图片

```
t.bgpic('starbg.png')
```



注意：
确保图片文件跟python程序
在同一个目录下



3. 画笔设置

```
t.pensize(5)           # 设置画笔粗细
t.fillcolor('yellow') # 设置填充色
t.begin_fill()        # 开始填充
<绘制过程>
t.end_fill()          # 结束填充
```

4. 绘制过程

```
for i in range(5):    # 重复执行5次
    side3()           # 画三角形
    t.forward(80)    # 移动到三角形外侧点
    side4()           # 画正方形
    t.goto(0,0)      # 移到出发点(0,0)
    t.left(72)       # 左转360° /5
```

5. 停止画笔绘制，绘图窗口不关闭

```
t.done()
```

完整代码

```
import turtle as t

def side3():
    for i in range(3):
        t.forward(80)
        t.left(120)
def side4():
    for i in range(4):
        t.forward(40)
        t.left(90)

t.setup(960,720)
t.bgpic('starbg.png')
t.pensize(5)
t.fillcolor('yellow')
t.begin_fill()
for i in range(5):
    side3()
    t.forward(80)
    side4()
    t.goto(0,0)
    t.left(72)
t.end_fill()
t.done()
```



2. 强化练习

1. 下列有关函数的描述中，正确的是？（ ）

- A. 函数中必须有参数
- B. 函数体可以缩进也可以不缩进
- C. 函数能提高应用的模块化程度和代码的重复利用率
- D. 函数内容以大括号起始，并且缩进

2. 下列哪个关键字可用来创建自定义函数？（ ）

- A. function
- B. def
- C. class
- D. procedure

3. 下列有关自定义函数的描述，正确的是？（ ）

- A. 函数的定义必须在程序的开头
- B. 函数定义后，其中的程序可以自动执行
- C. 函数定义后需要调用才会执行
- D. 函数体与关键字必须左对齐

4. 下列选项中不能作为自定义函数名的是？（ ）

- A. Int
- B. _int2
- C. str2int
- D. 2_int



2. 强化练习

5. turtle中，下面哪个函数可以加载背景图片？（ ）

A. t.bgpic()

B. turtle.bgpic

C. turtle.bgpic()

D. turtle中没有能够加载背景图片的函数

3. 术语箱

define 定义

side 边

4. 课后挑战

补全下面的代码，绘制出如图所示的图案

```
_____①_____
def side6():
    for i in range(6):
        t.forward(100)
        _____②_____
    for i in range(6):
        _____③_____
        t.right(60)
    t.done()
```

