

用PYTHON实现国旗的绘制，中国加油，武汉加油。

标签：`python`

之前了解过用python中的turtle模块绘制国旗，但是很多介绍和画法都是错误的，中国国旗有严格的规则 and 标准，本文就是在了解国旗知识的前提下，给出的一个严格准确的绘制方法。

（1）了解一下中国国旗的知识。

• （一）基本介绍

旗面为红色，长方形，其长与高为三与二之比，旗面左上方缀黄色五角星五颗。一星较大，其外接圆直径为旗高十分之三，居左；四星较小，其外接圆直径为旗高十分之一，环拱于大星之右。旗杆套为白色。

• （二）五星之位置与画法如下：

- （1）为便于确定五星之位置，先将旗面对分为四个相等的长方形，将左上方之长方形上下划为十等分，左右划为十五等分。
- （2）大五角星的中心点，在该长方形上五下五、左五右十之处。其画法为：以此点为圆心，以三等分为半径作一圆。
 - 在此圆周上，定出五个等距离的点，其一点须位于圆之正上方。然后将此五点中各相隔的两点相联，使各成一直线。
 - 此五直线所构成之外轮廓线，即为所需之大五角星。五角星之一角尖正向上方。
- （3）四颗小五角星的中心点，第一点在该长方形上二下八、左十右五之处，第二点在上四下六、左十二右三之处，

一等分为半径，

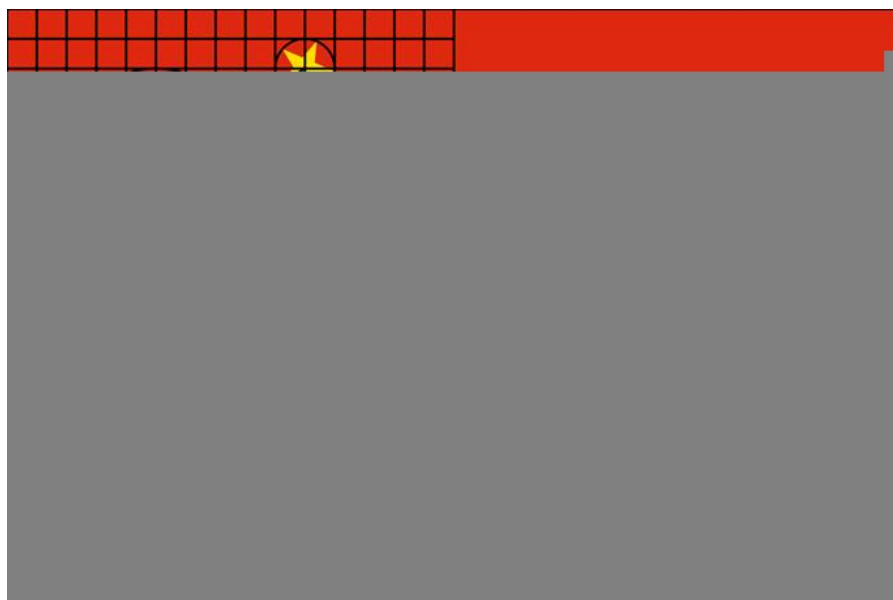
•分别作四个圆。在每个圆上各定出五个等距离的点，其中均须各有一点位于大五角星中心点与以上四个圆心的各联结线上。

•然后用构成大五角星的同样方法，构成小五角星。此四颗小五角星均各有一个角尖正对大五角星的中心点。

•（三）国旗之通用尺度定为如下五种，各界酌情选用：

- （1）长288厘米，高192厘米。
- （2）长240厘米，高160厘米。
- （3）长192厘米，高128厘米。
- （4）长144厘米，高96厘米。
- （5）长96厘米，高64厘米。

（四）具体位置关系如下图：



（五）PYTHON代码实现

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  ''' 利用turtle类画五星红旗。'''
4  __author__ = 'xu xinjian'
```

[首页](#)[联系我们](#)[版权申明](#)[隐私政策](#)

```
8 '''定义一个坐标点的类'''
9 class Point:
10     def __init__(self,x=0,y=0):
11         self.x=x
12         self.y=y
13     def getx(self):
14         return self.x
15     def gety(self):
16         return self.y
17
18 '''定义个获取两点间距离的类'''
19 class Getlen:
20     def __init__(self,p1,p2):
21         self.x=p1.getx()-p2.getx()
22         self.y=p1.gety()-p2.gety()
23         #用math.sqrt () 求平方根
24
25         self.len= math.sqrt((self.x**2)+(self.y**2)
26         #定义得到直线长度的函数 26
27     def getlen(self): 27         return self.len
28
29 '''初始化窗口和画布大小'''
30
31
32 turtle.setup(width=240*2,height=160*2,startx=350,s
33 turtle.bgcolor("red") 33
34 turtle.shape("turtle") 34
35 turtle.color("yellow","yellow") 35
36 '''
37 画大五角星
38 '''
39 turtle.penup()
40 turtle.goto(-160,108)
41 turtle.right(72)
42 turtle.pendown()
43
44 length1=2*48*math.cos(18)#计算五角星的边长
45 # print(length1)
46 turtle.begin_fill()#开始填充
47 #重复画五条边
48 for i in range(5):
49     turtle.forward(length1)
50     turtle.right(144)
51 turtle.end_fill()
```

[首页](#)[联系我们](#)[版权申明](#)[隐私政策](#)

```
56 turtle.goto(-160,80)
57 turtle.setheading(0)#朝向往右
58
59 '''
60 画第一个小五角星
61 '''
62 p0=Point(-160,80)
63 p1=Point(-80,124)
64 d1=Getlen(p0,p1).getlen() #
65
66 angle1=math.degrees(math.atan(3/5))
67 print("angle1:",angle1)
68 turtle.left(angle1)
69 turtle.penup()
70 turtle.forward(d1-16)
71 turtle.pendown()
72 turtle.left(18)
73 length2=2*16*math.cos(18)
74 turtle.begin_fill()
75 for i in range(5):
76     turtle.forward(length2)
77     turtle.right(144)
78 turtle.end_fill()
79
80
81 '''回到大五角星的中心'''
82 turtle.penup()
83 turtle.goto(-160,80)
84 turtle.setheading(0)
85 '''
86 画第二个小五角星
87 '''
88 p0=Point(-160,80)
89 p1=Point(-48,96)
90 d1=Getlen(p0,p1).getlen()
91
92 angle1=math.degrees(math.atan(1/7))
93 print("angle1:",angle1)
94 turtle.left(angle1)
95 turtle.penup()
96 turtle.forward(d1-16)
97 turtle.pendown()
98 turtle.left(18)
99 length2=2*16*math.cos(18)
```

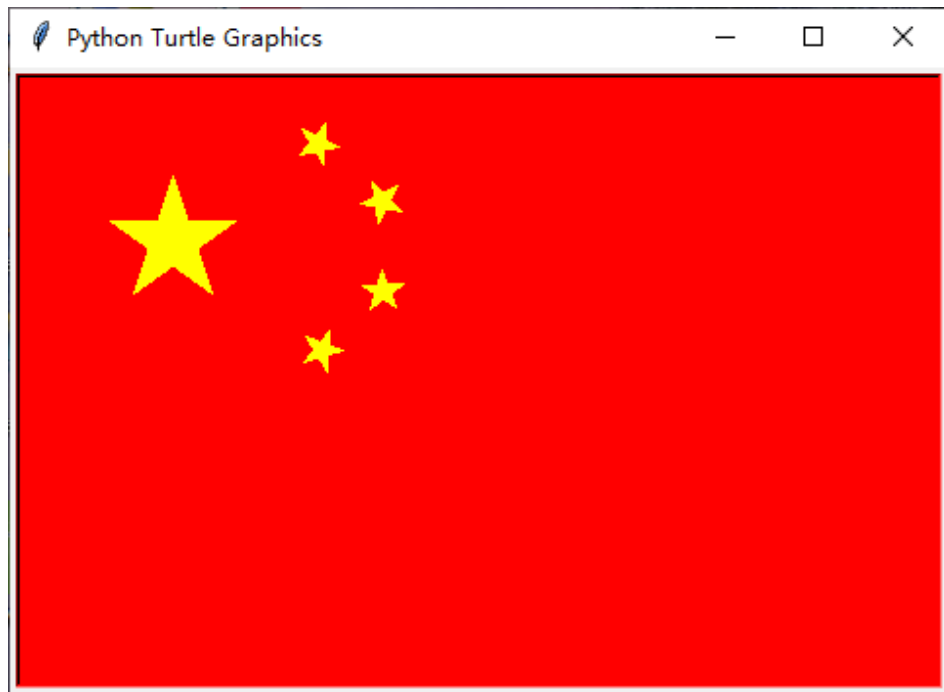
[首页](#)[联系我们](#)[版权申明](#)[隐私政策](#)

```
103     turtle.right(144)
104 turtle.end_fill()
105
106
107 '''回到大五角星的中心'''
108 turtle.penup()
109 turtle.goto(-160,80)
110 turtle.setheading(0)
111 '''
112 画第三个小五角星
113 '''
114 p0=Point(-160,80)
115 p1=Point(-48,48)
116 d1=Getlen(p0,p1).getlen()
117
118 angle1=math.degrees(math.atan(2/7))
119 print("angle1:",angle1)
120 turtle.right(angle1)
121 turtle.penup()
122 turtle.forward(d1-16)
123 turtle.pendown()
124 turtle.left(18)
125 length2=2*16*math.cos(18)
126 turtle.begin_fill()
127 for i in range(5):
128     turtle.forward(length2)
129     turtle.right(144)
130 turtle.end_fill()
131
132
133
134 '''回到大五角星的中心'''
135 turtle.penup()
136 turtle.goto(-160,80)
137 turtle.setheading(0)
138 '''
139 画第四个小五角星
140 '''
141 p0=Point(-160,80)
142 p1=Point(-80,16)
143 d1=Getlen(p0,p1).getlen()
144
145 angle1=math.degrees(math.atan(4/5))
146 print("angle1:",angle1)
147 turtle.right(angle1)
```

[首页](#)[联系我们](#)[版权申明](#)[隐私政策](#)

```
151 turtle.left(18)
152 length2=2*16*math.cos(18)
153 turtle.begin_fill()
154 for i in range(5):
155     turtle.forward(length2)
156     turtle.right(144)
157 turtle.end_fill()
158
159 turtle.hideturtle()
160 turtle.mainloop()
```

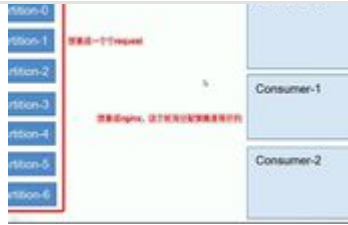
（六）实现效果



注释：五星红旗介绍材料来源于百度百科。

版权声明：本文为xuxin555原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，
转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/xuxin555/article/details/104561678>

[首页](#)[联系我们](#)[版权申明](#)[隐私政策](#)

提升内容 - KAFKA CONSUMER 小结(1)

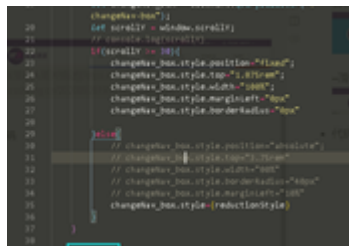
目录 kafka consumer 消费方式 消费分区分配策略 消费过程中offset的维护 - 老版本zk节点维护 1. 消费方式 1.1 broker push 这种消费方式

由broker主动推送消息给消费者，消费者被动接收消息。缺点：consumer 消费能力不强的情况下可能出现拒绝服务、以及因网络问题产生的网络拥塞的情况; 1.2 consumer pull 消费者主动轮询 broke...



SELENIUM+WEBDRIVER+CHROMEDRIVER 爬取百度图片 (一)

环境准备：安装selenium, chromedriver, BeautifulSoup 建议先安装pip，然后可以直接pip来安装很多相关的插件 直接上代码 那么，就能将证件照保存到本地了...



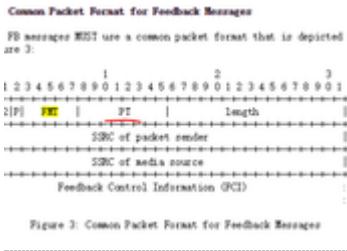
2020/09/28 REACT 函数式组件 USEEFFECT 监听滚动事件 获取滚动高度

目标效果，以圆通速递官网为例 - 滑动一定高度后 代码 - 代码块 - reductionStyle 写的是样式，可以提前在函数之前声明...

PYTHON2.7 报错 UNICODEENCODINGERROR: 'ASCII' CODEC CAN'T ENCODE CHARACTERS IN POSITION 10-29



position 10 25. Ormal not in range(128) 问题发现：用print打印可以输出，要用文件写入 write（）方法发现报错！解决办法：如图：至此问题解决！ ...



WEBRTC FB 类型区分

<https://tools.ietf.org/html/rfc4585#page-31>
janus 代码 RTCP协议的FB分三层不同的msg
Transport layer FB messages
Payload-specific FB messages
Application layer FB messages

Payload type (PT): 8 bits This is the RTC...

猜你喜欢



PYTHON 数据生成正态分布图_PYTHON 数据分析(2):生成NDARRAY

生成ndarray 1.使用array函数接收任意的序列型对象，生成一个新的包含传递数据的Numpy数组 2.使用zeros函数一次性创造全0数组 3.使用ones函数可以一次性创造全1数组 4.使用empty函数可以创建一个没有初始化数值的数组 5.使用python内建函数arange创建 ndarray的属性 1.ndim属性：查看数组的维度 2.hape属性：查看数组每一维度的数目 3.dt...

SSM项目学习之路（二）：CONTROLLER层获取JSON对象报错



threw exception Description The server encountered an unexpected condition that prevented it from fulfilling the reque...

消息类别	描述
EM_MOUSE	鼠标移动消息。
	鼠标滚轮移动消息。
	左键按下消息。
	左键弹起消息。
	右键按下消息。
	右键弹起消息。
	中键按下消息。
	中键弹起消息。
	中键按下消息。
	中键弹起消息。
	右键按下消息。
	右键弹起消息。

关于ESAYX中GETMOUSEMSG() 函数以不可用，只能用GETMESSAGE() 有关GETMESSAGE() 函数的用法。

总之就是GetMouseMsg () 过时了，用不了，该换成getmessage () 用getmessage () 来实现GetMouseMsg ()。首先 但实际上不止有EM_MOUSE,还有其他的消息类型 如剩下的用法已经跟GetMouseMsg () 差不多了 但防止有人像我一样半桶水的不知道GetMouseMsg()的（我是不知道getm...



LINUX下使用JOHN检测用户是否存在弱口令

实验环境 Ubuntu 16.04 &&Centos 6.5 前言 在互联网环境下，工作中避免不了会考虑服务器的安全问题，那么对于简单的口令来说，则是服务器所面临的巨大风险。即便大家都会有这方面的安全意识，设置一个符合长度且复杂性的口令会更加安全，但总是会有一些用户因贪图方便而采用简单且易记的口令。那么对于任何一个承担着安全责任的管理人员来说，通过某种手段来更高效的找出那些弱口令...

PYTHON3 FROM IMPORT 与 IMPORT 区别

【以下说明以tkinter模块为例进行说明】 【下图为安装后在python解释器路径下lib(库)文件夹下的tkinter文件夹下的内容】 1.import ... 【语法】 import tkinter 【说明】 （1）引用包 import引入的是包

