展开式和圆周率

# -\*- coding: cp936 -\*-  
import turtle  
import math  
  
vertex\_list = []  
  
  
def p\_line(t, n, length, angle):  
 *"""Draws n line segments."""* # tt.pen(fillcolor="blue", pencolor="#fe6023", pensize=5 )  
 for i in range(n):  
 t.fd(length)  
 vertex\_list.append(t.pos())  
 t.right(angle)  
  
  
def polygon(t, n, length):  
 *"""Draws a polygon with n sides."""* angle = 360.0 / n  
 p\_line(t, n, length, angle)  
  
  
def draw(tt, n, L):  
 global X, Y  
 tt.seth(0)  
 tt.pen(fillcolor="blue", pencolor="black", pensize=1)  
 start\_pos = tt.pos() # start\_pos  
 polygon(tt, n, L)  
  
 R = L / 2 / math.sin(math.pi / n) # 半径  
 r = R \* math.cos(math.pi / n) # 边心距  
 # 求中心点坐标(x,y)  
 x = L / 2 + start\_pos[0]  
 y = -r + start\_pos[1]  
 tt.goto(start\_pos) # start\_pos  
 for pos in vertex\_list:  
 tt.goto(x, y)  
 tt.pendown()  
 tt.goto(pos)  
 tt.penup()  
 print(x, y)  
  
  
# main  
n = 24  
L = 24  
  
window = turtle.Screen() # create a screen  
window.bgcolor("white")  
tt = turtle.Turtle()  
tt.hideturtle()  
tt.pencolor("black")  
tt.fillcolor("violet")  
tt.width(1)  
tt.speed(10)  
tt.pu()  
S = L / 10 # 几何放大系数  
X = 50 \* S  
Y = 4 \* S # 轮胎径长为4cm  
print('X=', X, ', Y=', Y)  
tt.goto(0, -Y)  
tt.pd()  
draw(tt, n, L)  
tt.pu()  
window.exitonclick()

