





目录 Contents

- 第一部分  Python的内建函数和自定义函数
- 第二部分  Python的绘制图形库turtle
- 第三部分  画国旗的基本原理
- 第四部分  画国旗的实例讲解



第一部分

Python的内建函数和自定义函数

■ 函数

- 函数可以说是任何程序的基石。程序中通常会有一些要重复使用的片段，可以先把它们搜集起来，以比较严谨的定义设置其名称、传递的参数和返回值，然后在程序其它地方就以调用的方式来使用这些片段，这样可以提升编程的效率。
- Python 默认带有的“别人”写好的函数就是内建函数。
- 自己写给自己用的函数就是自定义函数。

内建函数

- 许多特定的功能（例如把整数类型变量转化为字符串的`str()`、计算变量内元素个数的`len()`）都是Python 内建的函数，也就是不用导入外部模块即可使用的函数。下表列出了几个在程序中比较常用的内建函数。

函数	描述	举例
<code>abs(x)</code>	返回 <code>x</code> 的绝对值	<code>abs(-2)=2</code>
<code>max(x1,x2,...)</code>	返回 <code>x1,x2,...</code> 的最大值	<code>max(1,5,2)=5</code>
<code>min(x1,x2,...)</code>	返回 <code>x1,x2,...</code> 的最小值	<code>min(1,5,2)=1</code>
<code>pow(a,b)</code>	返回 <code>a^b</code> 的值，类似 <code>a ** b</code>	<code>pow(2,3)=8</code>
<code>round(x)</code>	返回与 <code>x</code> 最接近的整数，如果 <code>x</code> 与两个整数接近程度相同，则返回偶数值	<code>round(5.4)=5</code> <code>round(5.5)=6</code> <code>round(4.5)=4</code>
<code>round(x,n)</code>	保留小数点后 <code>n</code> 位小数的浮点值	<code>round(5.466,2)=5.47</code> <code>round(5.463,2)=5.46</code>

自定义函数

- 自定义函数是 `def` 开头，再加上自定义函数名称，如果需要参数就加上小括号接在函数名称的后面，最后别忘了加上“:”。
- 定义完成之后的函数并不会主动被执行，直到在程序中调用才会被执行，请看下面的例子：

```
>>> def add2number(a, b):  
...     return a + b  
...  
>>> add2number(10, 5)  
15  
>>>
```

- 如上，我们定义了一个叫做`add2number`的自定义函数，有两个参数`a`和`b`，实现两数相加。



第二部分

Python的绘制图形库turtle

■ 海龟绘图

- Python 内置了一个简单的绘图工具：海龟绘图(Turtle Graphics)，我们借助它可以画出各种各样的图形。
- turtle库是Python的内部库, 使用导入即可：`import turtle`
(注：在MicroPython中是uturtle，使用导入：`import uturtle`)



■ 画布

- 画布是**turtle**为我们展开用于绘图的区域，我们可以设置它的大小和初始位置。

设置画布大小：

```
turtle.screensize(canvwidth=None, canvheight=None, bg=None)
```

参数分别为画布的宽(单位像素), 高, 背景颜色

例如：

```
turtle.screensize(800, 600, "green")
```

```
turtle.screensize() #返回默认大小(400, 300)
```

■ 画笔

- 画笔的状态

- 在画布上，默认有一个坐标原点为画布中心的坐标轴，坐标原点上有一只面朝x轴正方向小乌龟。
- 描述小乌龟时使用两个词语：坐标原点(位置)、面朝x轴正方向(方向)。turtle绘图中，就是使用位置方向描述小乌龟(画笔)的状态

- 画笔的属性

- `turtle.pensize()` #设置画笔的宽度;
- `turtle.pencolor()` #没有参数传入，返回当前画笔颜色；传入参数设置画笔颜色，可以是字符串如"green", "red", 也可以是RGB 3元组;
- `turtle.speed(speed)` #设置画笔移动速度，画笔绘制的速度范围是[0,10]整数，越大越快。

■ 画笔

- 绘图命令

命令	说明
<code>turtle.forward(distance)</code>	向当前画笔方向移动distance像素长
<code>turtle.backward(distance)</code>	向当前画笔相反方向移动distance像素长度
<code>turtle.right(degree)</code>	顺时针移动degree°
<code>turtle.left(degree)</code>	逆时针移动degree°
<code>turtle.pendown()</code>	移动时绘制图形,缺省时也为绘制
<code>turtle.goto(x,y)</code>	将画笔移动到坐标为x,y的位置
<code>turtle.penup()</code>	移动时不绘制图形,提起笔,用于另起一个地方绘制时用
<code>turtle.speed(speed)</code>	画笔绘制的速度范围[0,10]整数
<code>turtle.circle()</code>	画圆,半径为正(负),表示圆心在画笔的左边(右边)画圆

- 绘图命令

命令	说明
<code>turtle.pensize(width)</code>	绘制图形时的宽度
<code>turtle.pencolor()</code>	画笔颜色
<code>turtle.fillcolor(colorstring)</code>	绘制图形的填充颜色
<code>turtle.color(color1, color2)</code>	同时设置 <code>pencolor=color1</code> , <code>fillcolor=color2</code>
<code>turtle.filling()</code>	返回当前是否在填充状态
<code>turtle.begin_fill()</code>	准备开始填充图形
<code>turtle.end_fill()</code>	填充完成;
<code>turtle.hideturtle()</code>	隐藏箭头显示;
<code>turtle.showturtle()</code>	与 <code>hideturtle()</code> 函数对应

- 绘图命令

命令	说明
<code>turtle.clear()</code>	清空turtle窗口，但是turtle的位置和状态不会改变
<code>turtle.reset()</code>	清空窗口，重置turtle状态为起始状态
<code>turtle.undo()</code>	撤销上一个turtle动作
<code>turtle.isvisible()</code>	返回当前turtle是否可见
<code>stamp()</code>	复制当前图形
<code>turtle.write(s[,font=("font-name",font_size,"font_type")])</code>	写文本，s为文本内容，font是字体的参数，里面分别为字体名称，大小和类型；font为可选项，font的参数也是可选项

- 绘圆命令详解

```
turtle.circle(radius, extent=None, steps=None)
```

描述：以给定半径画圆

参数：

radius(半径)：半径为正(负),表示圆心在画笔的左边(右边)画圆；

extent(弧度) (optional)；

steps(optional)：做半径为radius的圆的内切正多边形，多边形边数为steps。

举例：

```
turtle.circle(50) #整圆
```

```
turtle.circle(50, steps=3) #三角形
```

```
turtle.circle(120, 180) #半圆
```




第三部分 画国旗的基本原理

■ 要点

- 搜索各国国旗图片，了解其设计理念和设计规则；
- 导入turtle库，使用turtle的各种绘图指令根据国旗规则进行绘制；
- skids开发板板载屏幕大小为240x320，根据缩放比例关系将绘制出的国旗在屏幕上完美地显示出来。



■ 原理

- `import uturtle`
- `turtle = uturtle.Turtle()`
- 演示画线段
 - `turtle.forward(100)`
 - 前进100步
- 角度
 - `turtle.right(90)`
 - 右转90°
 - `turtle.left(-30)`
 - 左转-30° = 右转30°
- 来回走两步

- 绘制原理(德国国旗)

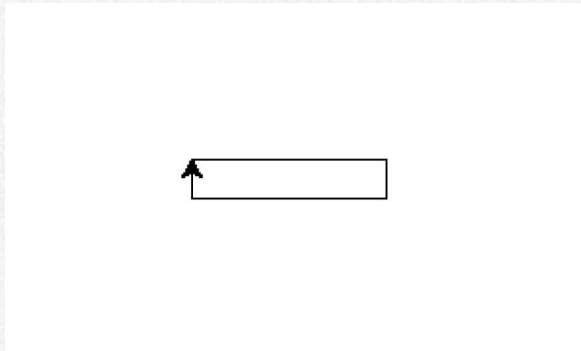
- 国旗规则

- 比例是为 $100:60=5:3$
 - 三个等高矩形，颜色是黑红金三色。黑色象征勤勉与力量，红色象征国民的热情，金色代表重视荣誉



- 画一个矩形

- 前进100，转 90° ，再前进20.....得到一个黑色的矩形框
 - `turtle.home()` #回到原点
 - `turtle.forward(100)`
 - `turtle.right(90)`
 - `turtle.forward(20)`
 - `turtle.right(90)`
 - `turtle.forward(100)`
 - `turtle.right(90)`
 - `turtle.forward(20)`



- 绘制原理(德国国旗)

- 填充颜色

- `turtle.color('red', 'red')` #设置边框颜色和填充颜色
 - `turtle.begin_fill()` #开始填充
 - #画矩形
 - `turtle.end_fill()` #结束填充

- 画三个矩形并填充

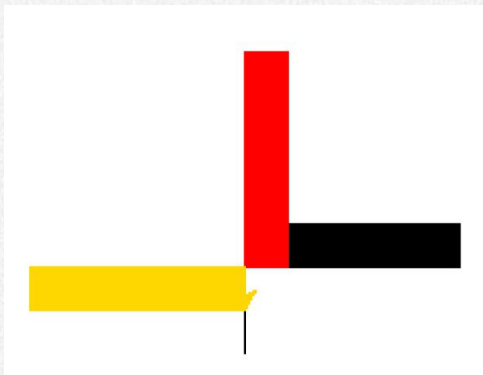
- 有什么不同?
 - 大小都相同
 - 位置、颜色不同
 - 做一个函数(封装相似的功能)
 - 函数(`def rect`)
 - 参数(位置、左上角坐标、颜色)

```
def rect(x, y, color):  
    turtle.goto(x,y)  
    turtle.color(color, color)  
    turtle.begin_fill()  
    turtle.forward(100)  
    turtle.right(90)  
    turtle.forward(20)  
    turtle.right(90)  
    turtle.forward(100)  
    turtle.right(90)  
    turtle.forward(20)  
    turtle.end_fill()
```

原理

- 绘制原理(德国国旗)

```
turtle.home()  
rect(0,60,'black')  
rect(0,40,'red')  
rect(0,20,'gold')
```



```
turtle.setheading(0)
```

#永远成0度角

```
def rect(x, y, color):  
    turtle.goto(x,y)  
    turtle.setheading(0)  
    turtle.color(color, color)
```




第四部分 画国旗的实例讲解

法国国旗

- 比例2:3
- 蓝白红三色纵旗。
- 代表法国的国家铭言「自由、平等、博爱」



- 学生做好以后提问：能不能改进rect函数，加入右下角的坐标作为参数
- `def rect(x, y, color, x2, y2):`
 - `width = x2 - x` # 横坐标，右比左大
 - `height = y - y2` # 纵坐标，上比下大
 - `width = abs(x2 - x)` #直接取绝对值
 - `height = abs(y2 - y)`

法国国旗

```
import uturtle
turtle = uturtle.Turtle()
def rect(x, y, color, x2, y2):
    width = abs(x2 - x)
    height = abs(y2 - y)
    turtle.pencolor(color)
    turtle.penup()
    turtle.goto(x,y)
    turtle.pendown()
    turtle.setheading(0)
    turtle.fillcolor(color)
    turtle.begin_fill()
    turtle.fd(width)
    turtle.right(90)
    turtle.forward(height)
    turtle.right(90)
    turtle.forward(width)
    turtle.right(90)
    turtle.forward(height)
    turtle.end_fill()
```

```
def france():
    rect(-120, 160, 'black', 120, -160)
    rect(-75, 50, 'blue', -25, -50)
    rect(-25, 50, 'white', 25, -50)
    rect(25, 50, 'red', 75, -50)

turtle.clear()
turtle.speed(0)
#turtle.delay(0)
france()
```


捷克国旗

- 国旗特征
 - 比例2:3
 - 蓝色三角及白红两色横旗，顶角至旗帜一半
 - 白色代表摩拉维亚，红色代表波希米亚，蓝色代表斯洛伐克，三角形象征喀尔巴阡山
- 代码实现

```
rect(-120, 160, 'gray', 120, -160)
rect(-75, 50, 'white', 75, 0)
rect(-75, 0, 'red', 75, -50)
turtle.home()
turtle.begin_fill()
turtle.color('blue', 'blue')
turtle.goto(-75, 50)
turtle.goto(-75, -50)
turtle.goto(0, 0)
turtle.end_fill()
```



■ 巴勒斯坦国旗

- 国旗特征

- 比例1:2
- 左边红色三角形的黑白绿三色横旗。原为巴勒斯坦解放组织于1964年采用的会旗。由于约旦境内有60%的人口为巴勒斯坦人，故两国国旗十分相似。
- 红色代表巴勒斯坦革命，白色代表革命的纯洁和坚定，黑色象征奋斗建国的历史与死亡，绿色代表伊斯兰教的信仰，也代表和平。由于该国尚未完全独立，所以将黑色置上。

- 代码实现

```
turtle.reset()
turtle.speed(0)
#turtle.delay(0)
rect(-120, 160, 'gray', 120, -160)
rect(-72, 36, 'black', 72, 12)
rect(-72, 12, 'white', 72, -12)
rect(-72, -12, 'green', 72, -36)
```

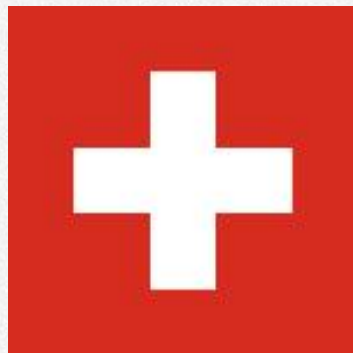
```
turtle.penup()
turtle.goto(-27,0)
turtle.pendown()
turtle.begin_fill()
turtle.color('red', 'red')
turtle.goto(-72, 36)
turtle.goto(-72, -36)
turtle.goto(-27, 0)
turtle.end_fill()
```



■ 瑞士国旗

- 国旗特征
 - 比例1:1
 - 红底正方形，中央有白色正十字。设计源自施维茨州的标志。
 - 自从1240年施维茨州人民使用该旗为战旗来反抗哈布斯堡王室统治时后，瑞士各势力在战争时常常以此旗为军旗。该面旗帜逐渐成为瑞士民族自由解放的象征，最后成为瑞士国旗。
- 代码实现

```
turtle.reset()
turtle.speed(0)
#turtle.delay(0)
rect(-120, 160, 'gray', 120, -160)
rect(-75, 75, 'red', 75, -75)
rect(-50, 12, 'white', 50, -12)
rect(-12, 50, 'white', 12, -50)
```



瑞典国旗

- 国旗特征
 - 比例5:8
 - 蓝底黄十字的旗帜，金十字代表瑞典福尔孔王朝中的金色十字架。
 - 1157年瑞典国王埃里克九世于远征芬兰前向神祷告，突然看到如同金色十字架的光芒横越青空，后来成为现在的国旗。
- 代码实现

```
turtle.reset()
turtle.speed(0)
#turtle.delay(0)
rect(-120, 160, 'gray', 120, -160)
rect(-100, 62, 'blue', 100, -62)
rect(-100, 12, 'yellow', 100, -12)
rect(-36, 62, 'yellow', -12, -62)
```



日本国旗

- 国旗特征

- 比例2:3
- 白底，中央紅日
- 圆的直径占宽度的60%
- 先画白色矩形 100:150 — `rect(0, 100, 'white', 150, 0)`
- 但是不明显，所以先把整个屏幕涂成灰色 — `rect(0, 320, 'gray', 240, 0)`



- 代码实现

```
turtle.reset()
turtle.speed(0)
#turtle.delay(0)
height = 100
width = 100 * 3 / 2
rect(-120, 160, 'gray', 120, -160)
rect(-width/2, height/2, 'white', width/2, -height/2)
```

```
r = height*0.6/2
turtle.penup()
turtle.home()
turtle.forward(r)
turtle.pendown()
turtle.begin_fill()
turtle.setheading(90)
turtle.color('red', 'red')
turtle.circle(r)
turtle.end_fill()
```

越南国旗

- 国旗特征

- 比例2:3
- 为红底，中央黄星的旗帜。红色表示革命和胜利。金星代表越南共产党，五个角分别表示工人、农民、士兵、知识分子和青年。
- 五角星，顶角 36° ，钝角 108° （五星永远差 72° ）

- 代码实现

```
turtle.reset()
turtle.speed(0)
#turtle.delay(0)
rect(-120, 160, 'gray', 120, -160)
rect(-75, 50, 'red', 75, -50)
```

```
turtle.penup()
turtle.goto(-25, 10)
turtle.pendown()
turtle.setheading(0)
turtle.pencolor("yellow")
turtle.fillcolor("yellow")
turtle.begin_fill()
for i in range(5):
    turtle.forward(50)
    turtle.right(144)
turtle.end_fill()
```



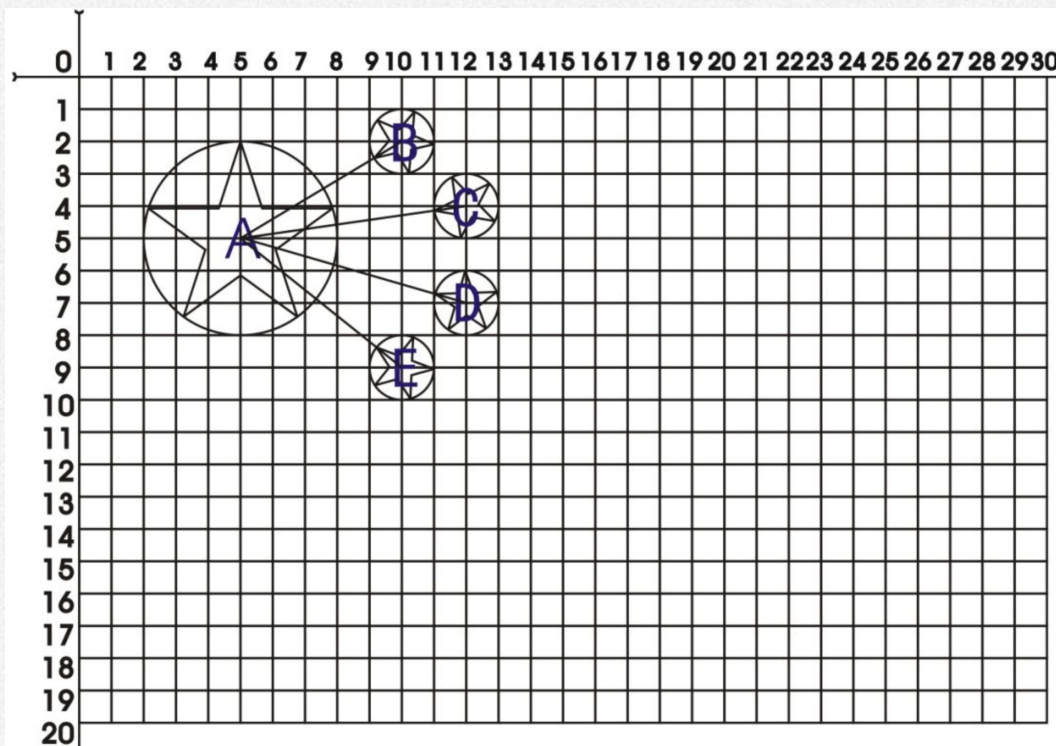
中国国旗

- 国旗特征
 - 比例2:3
 - 红底，左上角有大五角黄星与四颗小五角黄星的旗帜。红色代表革命(血)。黄色是为着在红地上显出光明。
 - 大五角黄星代表中国共产党，四颗小五角黄星代表中国人民的四个阶级：工人阶级、农民阶级、小资产阶级和民族资产阶级。四星环绕大星象征中国共产党领导下的革命人民大团结



中国国旗

- 国旗特征



- 代码实现

```
import uturtle
turtle = uturtle.Turtle()

def draw_rect(x1, y1, color, x2, y2):
    width = abs(x2 - x1)
    height = abs(y2 - y1)
    turtle.penup()
    turtle.goto(x1,y1)
    turtle.pendown()
    turtle.setheading(0)
    turtle.color(color, color)
    turtle.begin_fill()
    for i in range(2):
        turtle.forward(width)
        turtle.right(90)
        turtle.forward(height)
        turtle.right(90)
    turtle.end_fill()
```

```
def draw_star(center_x, center_y, radius, color):
    turtle.penup()
    pt1=turtle.pos()
    turtle.circle(-radius, 72)
    pt2=turtle.pos()
    turtle.circle(-radius, 72)
    pt3=turtle.pos()
    turtle.circle(-radius, 72)
    pt4=turtle.pos()
    turtle.circle(-radius, 72)
    pt5=turtle.pos()

    turtle.pendown()
    turtle.color(color, color)
    turtle.begin_fill()
    turtle.goto(pt3)
    turtle.goto(pt1)
    turtle.goto(pt4)
    turtle.goto(pt2)
    turtle.goto(pt5)
    turtle.end_fill()
```


中国国旗

```
def star(center_x, center_y, radius, big_center_x, big_center_y):  
    turtle.penup()  
    turtle.goto(center_x, center_y)  
    turtle.pendown()  
    turtle.left(turtle.towards(big_center_x, big_center_y) - turtle.heading())  
    turtle.forward(radius)  
    turtle.right(90)  
    draw_star(turtle.pos().x, turtle.pos().y, radius, 'yellow')  
  
turtle.reset()  
turtle.speed(0)  
width = 180  
height = 120  
draw_rect(-width/2, height/2, 'red', width/2, -height/2)  
  
pice = width/30  
big_center_x = -width/3  
big_center_y = height/4  
star(big_center_x, big_center_y-1, pice*3, big_center_x, big_center_y)  
star(-width/6, height*2/5, pice, big_center_x, big_center_y)  
star(-width/10, height*3/10, pice, big_center_x, big_center_y)  
star(-width/10, height*3/20, pice, big_center_x, big_center_y)  
star(-width/6, height/20, pice, big_center_x, big_center_y)
```



THANK YOU

牛艾科技