

本章先编写简单且简短的程序来创建基本几何图形。Python 提 供了易学易用的 turtle 库,可以帮我们方便地绘制图形。turtle 库又 称为海龟作图,是能够进行绘图操作的一个标准库,包含许多用来 绘制图形的函数。利用海龟作图,我们可以编写代码,让一只海龟 在屏幕上移动,并留下它的运动轨迹,从而画出图形。还可以跟随 海龟看看每句代码如何影响到它的移动,这能帮助我们理解代码的 逻辑。







小时候,我们用七巧板创作出各种各样的图案,那时,我 们就知道简单的几何图形也可以拼凑出精美的图案。几何体绘 图可以帮助我们有效地刻画错综复杂的世界。生活中到处都充 满着几何图形,都是由点、线、面等基本几何图形组成的。无 穷尽的丰富变化使几何图案本身拥有无穷魅力。任何具象的图 形都可以被简化概括成几何体,而几何体也因为不同的排列和 组合变得丰富多彩。几何图形虽然看似简单,却能生出无限的 可能。turtle 库提供了绘制各种几何图形的函数,我们可以利用 这些函数实现几何体绘图。



利用 turtle 库的函数, 绘制一颗可爱的开心蛋, 如图 1-1 所示。







## 1.2.1.1 Python 的安装

这是我们第一次编写 Python 程序,首先要在计算机 里安装 Python。Python 目前已支持所有主流操作系统,在 Linux,UNIX,Mac 系统上自带 Python 环境,在 Windows 系统上 需要自行安装,步骤很简单。

首先,打开 Python 官网的下载中心 https://www.python.org/ downloads/,找到适合自己计算机的安装版本下载,按照安装 向导完成安装,如图 1-2 所示。



图 1-2 Python 安装界面

安装后,可以查看"开始"菜单验证是否安装成功,如果 安装成功,有如图 1-3 所示的菜单。





### 1.2.1.2 绘制几何图形

首先要导入 turtle 库,才能调用库里的函数绘制图形。 turtle 库是 Python 的标准库,导入之前无须安装库。常用的导 入库的方式有以下两种。

(1) import turtle

(2) from turtle import 函数名或子库名

第1种方式,用 import 语句导入整个库内的所有成员(包括变量、函数、类等);第2种方式,用 import 语句只导入库内的指定函数或指定的子库,除非使用 from turtle import \*,可以导入所有成员。

使用第1种方式导入 turtle 库,调用函数时必须在前面添加 "turtle.",例如:

import turtle

```
turtle.goto(0,100)
```

使用第2种方式导入,无须任何前缀,例如: from turtle import \*

goto(0,100)

导入 turtle 库后,我们就能开始海龟绘图了。下面是几种 简单几何图形的绘制举例。

#### 1. 绘制直线

例 1.1 让我们使用海龟绘图来编写第一个程序。在 IDLE 窗口中输入如下代码,并将其保存为 StraightLine.py。

(1) from turtle import\*

(2)pencolor("red")

4



(3)pensize(3)

(4)goto(100,100)

Python Turtle Graphics
 - 
 ×

运行代码后,会在画布中画出一条直线(如图 1-4 所示)。



让我们一行一行地解析程序,看看直线是如何绘制的。

代码1,作用是导入turtle库,这样我们可以使用很多绘 图函数。库(Library)就是可以重复使用的代码的集合。一些 优秀的程序员创建了各种库,有了这些库,我们编写程序时可 以很方便地使用已有的代码,大大减轻了工作量。

代码 2,作用是调用 turtle 库的函数 pencolor 将画笔颜色 设置为红色,默认的画笔颜色是黑色。

代码3,作用是设置画笔宽度,也可以调用函数 width (3) 来设置,数字值越大,线越粗。

代码 4 是画出线段的关键,表示海龟从坐标系原点走到坐标(100,100)的位置。坐标系原点在画布中心,这是海龟的起始位置,所以绘制出一条斜线。

Python 画图中, 画布的坐标为: 从中心起始点出发(即O点),



上值和右值为正值,下值和左值为负值。

## 2. 绘制正方形

例 1.2 让我们来绘制一个红色正方形,正方形位于第一 象限,左下角的顶点就是原点。在 IDLE 窗口中输入如下代码, 并将其保存为 Square1.py。

(1) from turtle import\*

(2) color("red")

(3)pensize(3)

(4)begin\_fill( )

- (5) goto (0, 200)
- (6)goto(200,200)
- (7)goto(200,0)
- (8) goto (0, 0)
- (9)end fill( )



运行代码后,会在画布中画出一个红色正方形(如图 1-5 所示)。



第1章 绘制基本几何图形

通过例 1.1 的解析,我们知道 goto(x,y) 函数可以让海龟移 动到坐标 (x,y) 的位置,所以不难想出可以让海龟移动 4 次, 从而画出正方形。但是,如何用红色填充正方形呢?我们通 过对第 2 行、第 4 行和第 9 行这 3 行代码的解析来回答这个 问题。

代码2的作用是将画笔色和填充色都设置为红色,也可以 用下面两句代码来实现同样的功能:

pencolor("red") # 设置画笔色

```
fillcolor("red") # 设置填充色
```

当画笔色和填充色不同时,可以用两个参数来表示,如 color("red","blue"),可将画笔色设置为红色,填充色设置为蓝色, 相当于如下两句代码:

```
pencolor("red")
```

```
fillcolor("blue")
```

代码 4 和代码 9 的功能是实现对图形的颜色填充。在绘制 需要填充颜色的图形之前调用函数 begin\_fill(),在图形绘制完 毕后调用函数 end fill(),则可以完成填充。

例 1.3 我们继续绘制一个正方形,与例 1.2 不同的是,这个正方形位于画布正中。在 IDLE 窗口中输入如下代码,并将 其保存为 Square2.py。

```
from turtle import*
color("blue","yellow")
pensize(3)
```

```
begin fill( )
```



penup() # 抬起画笔

goto(-100,100)

pendown() # 放下画笔

goto(100,100)

goto(100,-100)

```
goto(-100,-100)
```

```
goto(-100,100)
```

end\_fill( )

运行代码后,会在画布中画出如下正方形(如图1-6所示)。



图 1-6 画布正中的正方形

例 1.3 和例 1.2 的主要区别在于正方形的位置。海龟的起始位置在画布正中心,不论走到正方形哪个顶点,都会留下一条斜线,但是图 1-6 中并没有出现这条线,原因是第 5 行和第 7 行这两句代码 penup()、pendown()。函数 penup()让海龟腾空,腾空期间海龟移动不会留下痕迹,函数 pendown()让海龟降落,降落后又可以继续画图。这两个函数也可以简写为 up()和 down()。



#### 3. 绘制三角形

例 1.4 绘制边长为 200 的等边三角形。在 IDLE 窗口中输入如下代码,并将其保存为 triangle.py。

from turtle import\*

pensize(3)

pencolor("red")

goto(200,0)

left(120) #海龟前进的方向左转120度

pencolor("green")

forward(200) # 向前移动 200 像素点

left(120) #海龟前进的方向左转120度

pencolor("blue")

forward(200) # 向前移动 200 像素点

运行代码后,会在画布中画出如下等边三角形(如图 1-7 所示)。



图 1-7 等边三角形

本题的已知条件是边长为200,可以计算出三角形3个顶



点的坐标,然后用函数 goto(x,y) 画出这个等边三角形,然而 用以上代码画出三角形,并不需要计算顶点坐标,关键在于 left()和 forward()这两个函数。

在解释第5行代码 left(120) 之前,我们要先了解海龟的移动方向。海龟默认的正前方是 x 轴正方向,如果让海龟向前走,海龟会朝着 x 轴正方向前进。可以调用 left()或者 right()函数 来改变海龟的移动方向,第5行代码 left(120)表示海龟前进的 方向左转 120 度,第7行代码 forward(200)表示让海龟向前移 动 200 像素点,所以无须计算顶点坐标就可以画出等边三角形。

#### 4. 绘制圆形

例 1.5 绘制圆、圆弧和多边形。在 IDLE 窗口中输入如下 代码,并将其保存为 circle.py。

from turtle import\*

circle(100) # 绘制半径为 100 的圆

up()

goto(-220,-220)

down()

circle(100,180) # 绘制半径为 100 的 180 度圆弧

up()

goto(180,0)

down()

circle(100,360,6) # 绘制正六边形

运行代码后,会在画布中画出如下圆、圆弧和多边形(本 例为正六边形,如图 1-8 所示)。





图 1-8 圆、圆弧和多边形

函数 circle(radius,extent,steps) 有 3 个参数: 第 1 个参数是 圆半径,不可缺省; 第 2 个参数是圆弧角度,如果缺省则默认 为 360 度; 第 3 个参数是正多边形的边数,如果缺省则默认 画圆。

### 5. 绘制圆点

例 1.6 绘制两个同心的圆点。在 IDLE 窗口中输入如下代码,并将其保存为 dot.py。

from turtle import\*

hideturtle() # 隐藏海龟

dot (200) # 绘制直径为 200 的黑色圆点

dot(100,"white") # 绘制直径为 100 的白色圆点

运行代码后,会在画布中画出如下同心圆点(如图1-9 所示)。







图 1-9 同心圆点

函数 dot(diameter,color) 有两个参数:第1个参数是圆点直径,如果缺省则默认为1;第2个参数是圆点的颜色,如果缺省则默认为黑色。

1.2.1.3 顺序结构

我们知道了几何图形的绘制方法,但是,在编写程序前, 还须了解程序的基本结构,才能正确地写出代码。程序有3种 基本结构:①顺序结构,程序的运行顺序和代码的排列顺序一 样,自上而下,依次执行;②选择结构,根据某个特定的条件 进行判断后,选择某部分代码执行;③循环结构,在程序中需 要反复执行某段代码,直到条件为假时停止循环。

本章,先学习顺序结构。顺序结构是最简单的程序结构, 程序中所有代码都会按照排列顺序依次执行,不会跳过任何一 句代码,每句代码执行一次。上文中的6个例题都是顺序结构 的程序。

顺序结构流程图如图 1-10 所示。





图 1-10 顺序结构流程图

美 1.2.2 算法设计

绘制开心蛋程序流程图如图 1-11 所示。



图 1-11 绘制开心蛋程序流程图





接下来综合运用前面介绍的函数,实现开心蛋的绘制。



from turtle import\* # 导入 turtle 库的所有函数 setup(800, 600, 0, 0) # 设置画布大小和位置 speed(0) # 设置绘画速度 hideturtle() # 隐藏海龟 #绘制身体,身体由4段圆弧构成 up() # 抬起画笔 qoto(-135, 0)right(90) #海龟方向右转90度 down() # 落下画笔 fillcolor("lemon chiffon") # 设置填充色 begin fill() #开始填充 circle(135, 180) # 绘制半径为 135、角度为 180 的圆弧 circle(250, 35) circle(110, 110) circle(250, 35) end fill() # 结束填充 left(90) #海龟方向左转90度



#绘制眉毛, 左右眉毛都由两段圆弧和一条线段构成

# 左眉毛

up()

goto(-20,130)

down()

fillcolor("black")

begin\_fill()# 眉毛用黑色填充

left(100)

fd(12) # 向前移动 12

left(90)

circle(22,80)

up()

goto(-20, 130)

down()

right(80)

circle(40,35)

end\_fill()

#右眉毛

up()

goto(45,125)

down()

begin\_fill()

left(230)

circle(20,85)

up()





goto(45,125)

down()

right(35)

circle(40,35)

right(98)

fd(12)

end fill()

# 绘制眼睛, 眼眶由圆构成, 瞳孔由一个黑色的大圆点和一个白色的小圆点构成

# 左眼

width(2)

up()

```
goto(8,80)
```

down()

circle(32) # 绘制眼眶

up()

```
goto(-20,80)
```

down()

dot(16) # 绘制直径为 16 的黑色圆点

up()

goto (-22,82)

down()

dot(8,"white") # 在瞳孔中绘制直径为 8 的白色圆点

#右眼

up()

goto(4,68)



```
down()
right(140)
circle(28,295)
up()
goto(20,80)
down()
dot(16)
up()
goto(20,82)
down()
dot(8,"white")
#绘制嘴唇,嘴唇由3段圆弧构成
width(3)
up()
goto(-50,50)
down()
right(80)
circle(25,70) # 绘制左嘴角
up()
goto(-66,49)
down()
left(85)
circle(90,100) # 绘制上唇
up()
goto(-50,36)
```





down() right(130) circle(70,115) # 绘制下唇 #绘制牙齿,牙齿由4条线段构成 up() goto (-25,23) down() goto (-27,0) up() goto (-2,20) down() goto(-2,-15) up() goto(21,20) down() goto (24, -18) up() goto(44,29) down() goto(52,-10) #绘制手,手臂由圆弧构成,手掌由圆点构成 up() goto(135,40) down() width(6)



```
right(45)
circle(-40, 80) # 绘制右手臂
dot(14) #绘制手掌
dot(9,"white")
up()
goto(-135,40)
down()
right(110)
circle(-40, 80) # 绘制左手臂
dot(14) #绘制手掌
dot(9,"white")
#绘制腿脚,腿由线段构成,脚由圆点构成
left(12)
up()
goto (25, -152)
down()
dot(16) #绘制右脚
dot(11,"white")
up()
fd(3)
down()
fd(15)
up()
goto(-25,-152)
down()
```





dot(16) # 绘制左脚 dot(11,"white") up() fd(3) down() fd(15)

1.3.2 运行程序

程序会按照顺序依次执行,先绘制身体轮廓,然后依次绘制眉毛、眼睛、嘴唇、牙齿、手和腿脚。如果顺序结构的程序 运行时出现问题,很可能是语法错误,需要认真检查,所以要 求同学们在编写程序的时候要严谨心细。最终绘图效果如图 1-1 所示。



绘制边长为 200 的正五角星, 如图 1-12 所示。





# 2. 绘制奥运五环

绘制如图 1-13 所示的奥运五环。



