049 使用海龟（turtle）模块绘制动态时钟

* 核心技术

**在实现动态时钟时，需要使用python内置的**datetime**模块与**turtle模块。

1．datetime****模块****

datetime**模块是python的内置模块，所以在使用前需要使用import导入该模块。代码如下：**

import datetime

（1）datetime**模块中datetime类的**today()方法，用于返回一个表示当前本地日期时间的datetime对象。其语法格式如下：

datetime.today()

参数说明：

* 返回值：返回一个表示当前本地日期时间的datetime对象。

实现获取当前本地日期与时间的示例代码如下：

1. **from** datetime **import** datetime # 导入datetime模块中的datetime类
2. dt=datetime.today() #把获取的当前本地日期时间赋给变量dt
3. print(dt) # 打印当前本地日期时间

输出结果为：

2019-10-11 09:18:35.784341

（2）如果已经获取了当前本地日期时间的datetime对象，此时可以通过该对象的变量名称获取对应的年、月、日、时、分、秒及微秒。代码如下：

1. **from** datetime **import** datetime # 导入datetime模块中的datetime类
2. dt=datetime.today() #把获取的当前本地日期时间赋给变量dt
3. print(dt) # 打印当前本地日期时间
4. print(dt.year,'年')
5. print(dt.month,'月')
6. print(dt.day,'日')
7. print(dt.hour,'时')
8. print(dt.minute,'分')
9. print(dt.second,'秒')
10. print(dt.microsecond,'微秒')

输出结果为：

2019-10-11 09:25:35.665185

2019 年

10 月

11 日

9 时

25 分

35 秒

665185 微秒

（3）datetime对象中提供了weekday()方法，用于返回指定日期在一周内的序号，即星期几，其中周一表示为0，周日表示为6。其语法格式如下：

datetime.weekday()

参数说明：

* 返回值：返回指定日期在一周内的序号，即星期几，其中周一表示为0，周日表示为6。

实现获取当前本地日期对应星期的示例代码如下：

1. **from** datetime **import** datetime # 导入datetime模块中的datetime类
2. week = ['星期一', '星期二', '星期三', '星期四', '星期五', '星期六', '星期日']
3. dt=datetime.today() #把获取的当前本地日期时间赋给变量dt
4. print('当前日期时间为：',dt)
5. print(week[dt.weekday()])

输出结果为：

当前日期时间为： 2019-10-11 09:46:34.851536

星期五

2．turtle模块

**turtle模块是python的内置模块，所以在使用前需要使用import导入该模块。代码如下：**

import **turtle**

**（1）创建游戏窗体可以使用turtle模块中的setup()方法来实现。语法格式如下：**

**turtle.setup(width=\_CFG["width"],height=\_CFG["height"],startx=\_CFG["leftright"],starty=\_CFG["topbottom"])**

**参数说明如下：**

* **width**：表示窗体的宽度，如果是整数，大小以像素为单位，如果是浮动，则为屏幕的一小部分；默认值为屏幕的50%。
* **height：表示窗体的高度，**如果是整数，高度以像素为单位，如果是浮动，则为屏幕的一小部分；默认值为屏幕的75%**。**
* **startx：表示窗体显示在屏幕的横向位置，**如果为正，则以像素为单位从屏幕左边缘开始，如果为负，则从右边缘开始，如果没有，则水平居中窗口**。**
* **starty：表示窗体显示在屏幕的纵向位置，**如果为正，则以像素为单位从屏幕上边缘开始，如果为负，则从下边缘开始，如果没有，则垂直居中窗口**。**

**创建游戏窗体的关键代码如下：**

1. **import** turtle # 导入绘图海龟模块
2. turtle.setup(450, 450) # 创建窗体大小
3. turtle.mainloop() # 不关闭窗体

**（2）turtle模块中的forward()方法，用于实现画笔按照指定方向，移动指定的距离。其语法格式如下：**

**turtle.forward(distance)**

**参数说明如下：**

* **distance：表示需要指定的移动距离（整数或浮点数）。**

**说明：turtle模块中的fd()方法与forward()方法相同。**

**实现移动绘制的示例代码如下：**

1. **import** turtle # 导入绘图海龟模块
2. turtle.forward(50) # 默认方向移动50，并绘制直线
3. turtle.mainloop() # 不关闭窗体

运行结果如图1所示。



图1 移动并绘制直线

**说明：如果仅需要让画笔移动并不进行绘制时，可以在forward()方法前添加抬笔方法penup()、pu()或up()。**

**（3）turtle模块中的forward()方法，用于实现绘制指定直径的点，可设置点的颜色。其语法格式如下：**

**turtle.dot(size=None,\*color)**

**参数说明如下：**

* **size：表示设置圆点直径的大小。**
* **color：表示设置圆点的颜色，该值为字符串类型。**

**实现绘制圆点的示例代码如下：**

1. **import** turtle # 导入绘图海龟模块
2. **def** skip(distance): # 移动方法，不留移动痕迹
3. turtle.penup() # 抬笔不绘制
4. turtle.forward(distance) # 移动指定距离
5. turtle.pendown() # 落笔移动绘制
6. **if** \_\_name\_\_ == '\_\_main\_\_':
7. turtle.hideturtle() # 隐藏箭头
8. turtle.dot() # 默认值圆点
9. skip(20) # 向前移动20
10. turtle.dot(10) # 指定大小的圆点
11. skip(20) # 向前移动20
12. turtle.dot(15,'red') # 红色圆点
13. turtle.mainloop() # 不关闭窗体

运行结果如图2所示。



图2 绘制圆点

（4）**turtle模块中的right()方法，用于实现向右旋转画笔的角度。其语法格式如下：**

**turtle.right(angle)**

**参数说明如下：**

* **angle：表示设置需要旋转的角度。**

实现旋转绘制的示例代码如下：

1. **import** turtle # 导入绘图海龟模块
2. **def** skip(distance): # 移动方法，不留移动痕迹
3. turtle.penup() # 抬笔不绘制
4. turtle.forward(distance) # 移动指定距离
5. turtle.pendown() # 落笔移动绘制
6. **if** \_\_name\_\_ == '\_\_main\_\_':
7. turtle.hideturtle() # 隐藏箭头
8. **for** i **in** range(60):
9. skip(160) # 移动160
10. turtle.dot(15,'red') # 红色圆点
11. skip(-160)
12. turtle.right(6)

运行结果如图3所示。



图3 旋转绘制圆点

**（5）turtle模块中的register\_shape()方法，用于注册图形。其语法格式如下：**

**turtle.register\_shape(name,shape=None)**

**参数说明如下：**

* **name：表示注册图形的字符串名称。**
* **shape：表示数据结构的建模形状，可以是多边形的坐标元组、图片以及复合形状。**

实现注册图形的示例代码如下：

1. **import** turtle # 导入绘图海龟模块
2. turtle.begin\_poly() # 开始记录行走位置
3. **for** i **in** range(2):
4. turtle.forward(100) # 向前移动100
5. turtle.left(90) # 向左旋转90度
6. turtle.forward(20) # 向前移动20
7. turtle.left(90) # 向左旋转90度
8. turtle.end\_poly() # 结束记录行走位置
9. turtle.register\_shape("rectangle", turtle.get\_poly())
10. turtle.mainloop() # 不关闭窗体

运行结果如图4所示。



图4 注册多边图形

（6）**turtle模块中的shape()方法，用于将画笔设置为指定图形。其语法格式如下：**

**turtle.shape(name=None)**

**参数说明如下：**

* **name：表示需要设置图形的字符串名称。**

实现**将画笔设置为乌龟**的示例代码如下：

1. **import** turtle # 导入绘图海龟模块
2. turtle.shape('turtle') # 将画笔设置为乌龟
3. turtle.mainloop() # 不关闭窗体

运行结果如图5所示。



图5 **将画笔设置为乌龟**