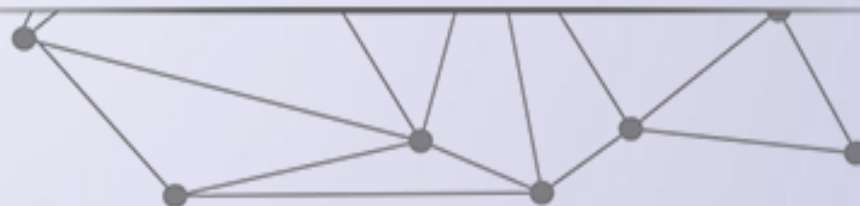




# 图形库的应用方法



黄天羽

北京理工大学



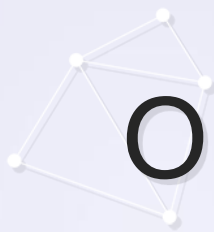


- Graphwin对象
- 图形对象
- 图形颜色系统



# GraphWin对象

- 一个程序可以定义任意数量的窗体
- `GraphWin()`
- 默认标题是 “Graphics Window”
- 默认大小为200\*200



# OS平台编程的需求

- 目录和文件的操作

对系统目录、文件的处理 and 操作方法

- 程序定时执行

- Py文件转化为exe文件：针对Windows平台，将Python的.py文件转化为可执行程序





# GraphWin对象常用方法

方法名称	方法含义
<b>plot(x, y, color)</b>	在窗口中(x,y)位置绘制像素。颜色参数可选，默认值为黑色。
<b>plotPixel(x, y, Color)</b>	在“原始”位置(x,y)处绘制像素，忽略setCoords()方法设置的坐标变换。
<b>setBackground(color)</b>	将窗口背景颜色设为指定颜色，默认值为灰色。
<b>close()</b>	关闭屏幕上的窗口。
<b>getMouse()</b>	程序等待用户在窗口内点击鼠标，返回值为点击处的位置，并以Point对象返回。
<b>setCoords(xll, yll, xur, yur)</b>	设置窗口的坐标系。左下角是(xll,yll)，右上角是(xur,yur)。所有后面的绘制都以这个坐标系做参照(plotPexil除外)




## ■ 图形对象

- 点、线段、圆、椭圆、矩形、多边形以及文本

## ■ 默认初始化

- 黑色边框
- 没有被填充



# 图形对象通用方法

方法名称	方法含义
<b>setFill(color)</b>	设置对象内部填充颜色。
<b>setOutline(color)</b>	设置对象边框颜色。
<b>setWidth(pixels)</b>	设置对象的宽度(对Point类不起作用)。
<b>draw(aGraphWin)</b>	在指定的窗口中绘制对象。
<b>undraw()</b>	从窗口中删除该对象。如该对象没有在窗口中画出将会报错。
<b>move(dx,dy)</b>	将对象沿x轴和y轴分别移动dx和dy单位长度。
<b>clone()</b>	返回该对象的副本。





# Point对象方法

方法名称	方法含义
<b>Point(x,y)</b>	以指定坐标的值(x, y)构造一点
<b>getX()</b>	返回该点的x坐标值
<b>getY()</b>	返回该点的y坐标值





# Line对象方法

方法名称	方法含义
<b>Line(point1, point2)</b>	构造一个从点point1到点point2的线段
<b>setArrow (string)</b>	设置线段的箭头样式。箭头可以绘制在左端，右端，或者两端都有。string参数值为'first', 'last', 'both',或 'none'，默认值为'none'。
<b>getCenter()</b>	返回线段中点的坐标值。
<b>getP1(), getP2()</b>	返回线段相应端点的坐标值。



# Circle对象方法：

方法名称	方法含义
<b>Circle(centerPoint, radius)</b>	根据给定圆心和半径构建圆
<b>getCenter()</b>	返回圆心的值
<b>getRadius()</b>	返回圆的半径长度
<b>getP1(), getP2()</b>	返回值为该圆边框对应点，对应点指的是该圆外接正方形的对角点。



# Rectangle对象方法

方法名称	方法含义
<b>Rectangle (point1, point2)</b>	以point1和point2为对角点创建一个矩形。
<b>getCenter()</b>	返回矩形的中心点的克隆值。
<b>getP1(), getP2()</b>	返回构造矩形的对角点的克隆值



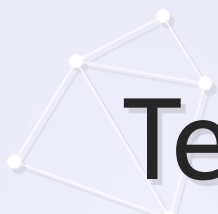
# Oval对象方法

方法名称	方法含义
<b>Oval(point1, point2)</b>	在点point1和point2指定的边界框中创建一个椭圆。
<b>getCenter()</b>	返回椭圆的中心点的坐标值
<b>getP1(), getP2()</b>	返回构造椭圆的对角点的坐标值



# Polygon 对象方法

方法名称	方法含义
<b>Polygon</b> <b>(point1, point2,</b> <b>point3, ...)</b>	根据给定的顶点构造一个多边形。也可以只用一个顶点列表作为参数
<b>getPoints()</b>	返回构造多边形的顶点值的列表



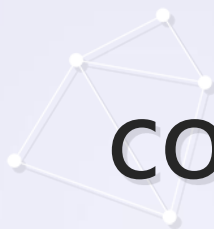
# Text 对象方法

方法名称	方法含义
<b>Text(anchorPoint, string)</b>	以anchorPoint点的位置为中心，构建了一个内容为string的文本对象。
<b>setText(string)</b>	设置文本对象的内容
<b>getText()</b>	返回当前文本内容。
<b>getAnchor()</b>	返回文本显示中间位置点anchor的坐标值。
<b>setFace(family)</b>	设置文本字体。family可选值为：'helvetica', 'courier', 'times roman', 以及 'arial'。
<b>setSize(point)</b>	设置字体大小为给定点point的大小。合法数值为5-36。
<b>setStyle(style)</b>	设置字体的风格。可选值为'normal', 'bold', 'italic', 以及 'bold italic'。
<b>setTextColor(color)</b>	设置文本颜色。与setFill效果相同。



# 图形颜色

- Python中颜色由字符串指定
- 很多颜色具有不同深浅
  - 红色逐渐加深
  - 'red1' 'red2' 'red3' 'red4'



# color\_rgb(red,green,blue)函数

- 设定颜色数值获得颜色
- 三个参数为0-255范围内的整数
- 返回一个字符串

`color_rgb(255,0,0)` 亮红色 ,

`color_rgb(130,0,130)` 中度洋红色。





# 温度转换程序示例

计算温度值设定窗口颜色：

- 温度越高，颜色越偏红
- 温度越低，颜色越偏蓝

setBackground(Newcolor)设置窗口背景颜色。



假定输入温度范围为0-100 ,

颜色权重 $\text{weight} = \text{输入温度} / 100$


newcolor的rgb计算：

红色分量 $= 255 * \text{weight}$

绿色分量 $= 66 + 150(1 - \text{weight})$

蓝色分量 $= 255 * (1 - \text{weight})$





# 完整代码如下


```
from graphics import *

def convert(input):
    celsius = eval(input.getText())    # 输入转换
    fahrenheit = 9.0/5.0 * celsius + 32
    return fahrenheit

def colorChange(win,input):
    cnum = eval(input.getText())
    weight = cnum / 100.0
    newcolor =color_rgb(255*weight,66+150*(1-weight),255*(1-weight))
    win.setBackground(newcolor)

def main():
    win = GraphWin("Celsius Converter", 400, 300)
    win.setCoords(0.0, 0.0, 3.0, 4.0)
    # 绘制输入接口
    Text(Point(1,3),
         " Celsius Temperature:").draw(win)
    Text(Point(2,2.7),
         " (Please input 0.0-100.0 )").draw(win)
    Text(Point(1,1),
         "Fahrenheit Temperature:").draw(win)
    input = Entry(Point(2,3), 5)
    input.setText("0.0")
    input.draw(win)
```





# 完整代码如下

```
output = Text(Point(2,1),"")
output.draw(win)
button = Text(Point(1.5,2.0),"Convert It")
button.draw(win)
rect = Rectangle(Point(1,1.5), Point(2,2.5))
rect.draw(win)
# 等待鼠标点击
win.getMouse()
result = convert(input)      # 转换输入
output.setText(result)      # 显示输出
# 改变颜色
colorChange(win,input)
# 改变按钮字体
button.setText("Quit")
# 等待点击事件,退出程序
win.getMouse()
win.close()
```

```
if __name__ == '__main__':
    main()
```



# 输入不同摄氏温度

0、20、40、60、80、100

Celsius Converter

Celsius Temperature:   
(Please input 0.0-100.0)

Quit

Fahrenheit Temperature: 32.0

Celsius Converter

Celsius Temperature:   
(Please input 0.0-100.0)

Quit

Fahrenheit Temperature: 68.0

Celsius Converter

Celsius Temperature:   
(Please input 0.0-100.0)

Quit

Fahrenheit Temperature: 104.0

Celsius Converter

Celsius Temperature:   
(Please input 0.0-100.0)

Quit

Fahrenheit Temperature: 140.0

Celsius Converter

Celsius Temperature:   
(Please input 0.0-100.0)

Quit

Fahrenheit Temperature: 176.0

Celsius Converter

Celsius Temperature:   
(Please input 0.0-100.0)

Quit

Fahrenheit Temperature: 212.0