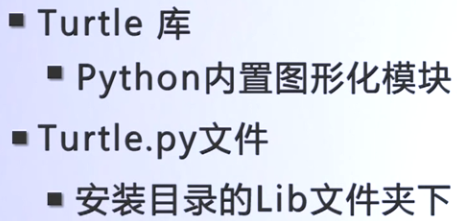
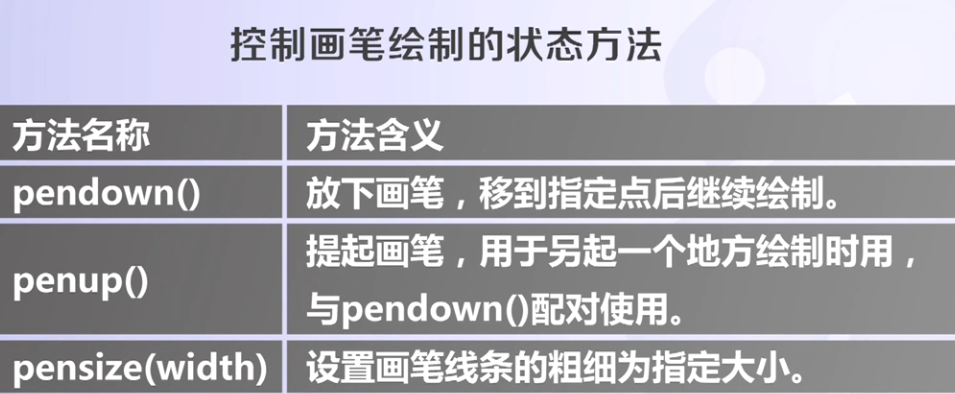
Python的turtle库

# turtle库



# turtle库的基本方法

## 控制画笔绘制的状态方法

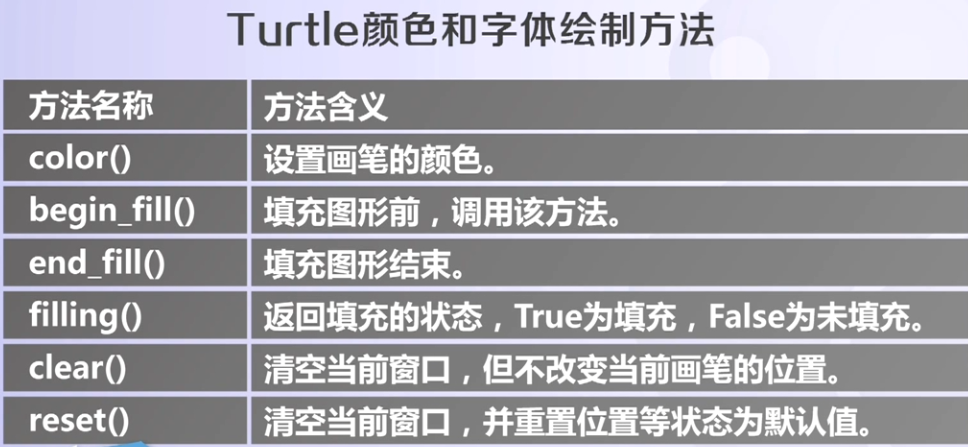


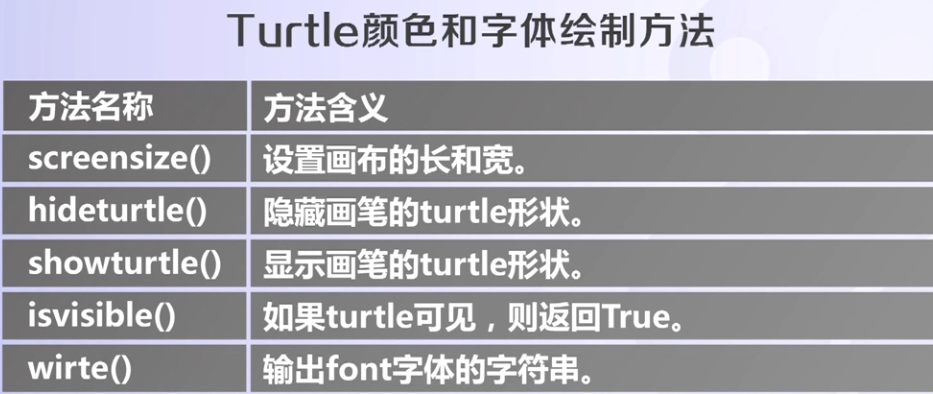
## turtle运动方法



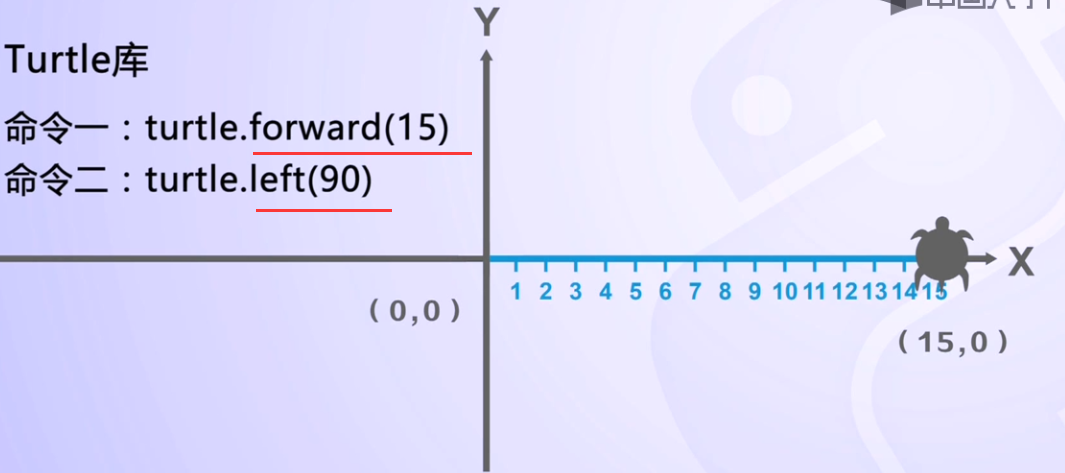


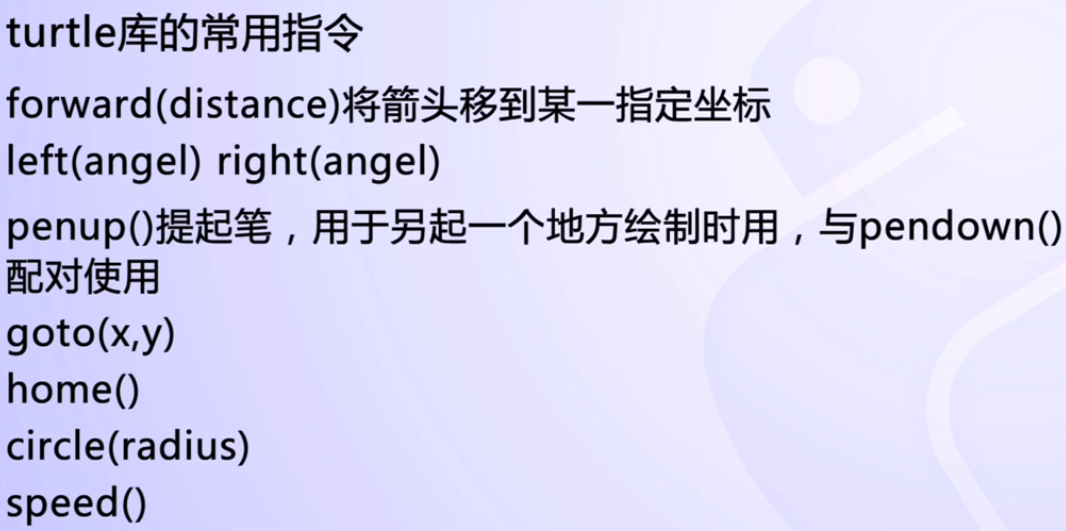
## Turtle颜色和字体绘制方法





# turtle库：





**import turtle**

**turtle模块**：它可以让你使用**海龟图形（turtle graphics）**绘制图像

其中的函数：

### turtle.pensize()：设置线条的粗细；

### turtle.speed()：设置绘制的速度，1-10，1最慢，10最快；

### turtle.begin\_fill()：准备开始填充图形；

### turtle.circle(50,steps=3)：circle函数在之前用到过，是画一个半径为radius的圆，这里是扩展，steps表示在半径为50的圆内的内置steps多边形；

### turtle.end\_fill()：填充完成；

### turtle.write(s,font=(“font-name”,font\_size,”font\_type”))：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；

### turtle.hideturtle()：隐藏箭头显示；

另外，还有其他一些turtle函数，如：

### turtle.backward(d)：与forward()函数对应，这里是从尾部绘制线条和箭头到头部；

### turtle.left(angle)：逆时针转动箭头方向；

### turtle.undo()：撤销上一个turtle动作；

### turtle.screensize(w,h)：设置turtle窗口的长和宽；

### turtle.clear()：清空turtle窗口，但是turtle的位置和状态不会改变；

### turtle.reset()：清空窗口，重置turtle状态为起始状态；

### turtle.showturtle()：与hideturtle()函数对应；

### turtle.filling()：返回当前是否在填充状态；true为filling，false为not filling；

### turtle.isvisible()：返回当前turtle是否可见。

# 示例:

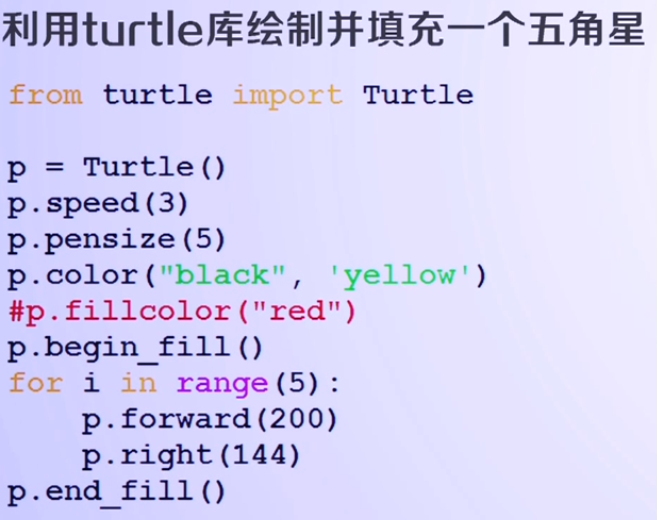
## 绘制Python小蟒蛇

import turtle  
  
def drawSnake(rad,angle,len,neckrad):  
 for i in range(len):  
 turtle.circle(rad,angle)  
 turtle.circle(-rad,angle)  
 turtle.circle(rad,angle/2)  
 turtle.fd(rad)  
 turtle.circle(neckrad+1,180)  
 turtle.fd(rad\*2/3)  
  
def main():  
 turtle.setup(1300,800,0,0)  
 pythonsize = 30  
 turtle.pensize(pythonsize)  
 turtle.pencolor('blue')  
 turtle.seth(-40)  
 drawSnake(40,80,5,pythonsize/2)  
main()

结果：



## 绘制五角星

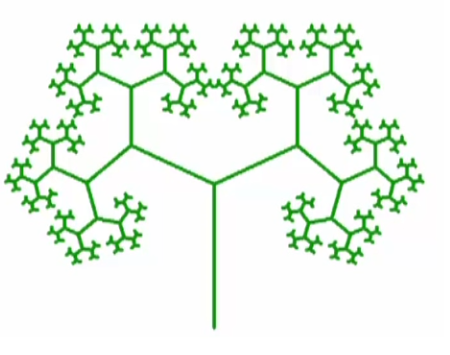




## 利用turtle库绘制Tree

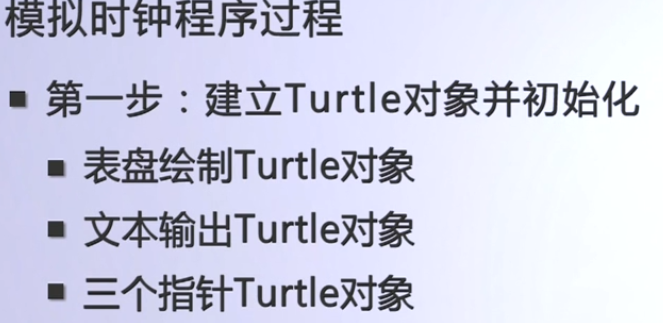
# drawtree.py  
  
from turtle import Turtle, mainloop  
  
  
def tree(plist, l, a, f):  
 *""" plist is list of pens  
 l is length of branch  
 a is half of the angle between 2 branches  
 f is factor by which branch is shortened  
 from level to level."""* if l > 5: #  
 lst = []  
 for p in plist:  
 p.forward(  
 l) # 沿着当前的方向画画Move the turtle forward by the specified distance, in the direction the turtle is headed.  
 q = p.clone() # Create and return a clone of the turtle with same position, heading and turtle properties.  
 p.left(a) # Turn turtle left by angle units  
 q.right(  
 a) # turn turtle right by angle units, nits are by default degrees, but can be set via the degrees() and radians() functions.  
 lst.append(p) # 将元素增加到列表的最后  
 lst.append(q)  
 tree(lst, l \* f, a, f)  
  
  
def main():  
 p = Turtle()  
 p.color("green")  
 p.pensize(5)  
 # p.setundobuffer(None)  
 p.hideturtle() # Make the turtle invisible. It’s a good idea to do this while you’re in the middle of doing some complex drawing,  
 # because hiding the turtle speeds up the drawing observably.  
 # p.speed(10)  
 # p.getscreen().tracer(1,0)#Return the TurtleScreen object the turtle is drawing on.  
 p.speed(10)  
 # TurtleScreen methods can then be called for that object.  
 p.left(90) # Turn turtle left by angle units. direction 调整画笔  
  
 p.penup() # Pull the pen up – no drawing when moving.  
 p.goto(0,  
 -200) # Move turtle to an absolute position. If the pen is down, draw line. Do not change the turtle’s orientation.  
 p.pendown() # Pull the pen down – drawing when moving. 这三条语句是一个组合相当于先把笔收起来再移动到指定位置，再把笔放下开始画  
 # 否则turtle一移动就会自动的把线画出来  
  
 # t = tree([p], 200, 65, 0.6375)  
 t = tree([p], 200, 65, 0.6375)  
  
  
main()

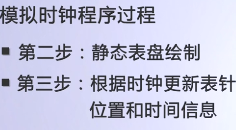
结果：

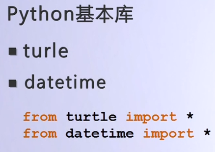


## 模拟时钟

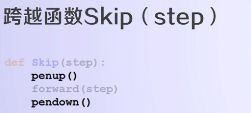




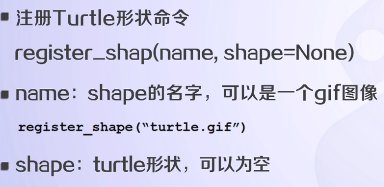








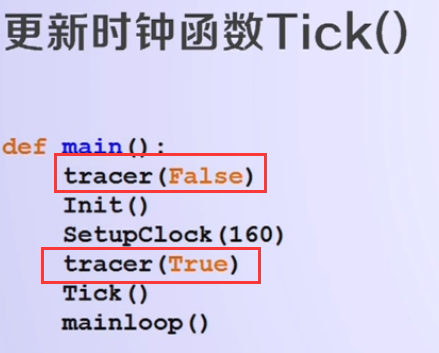






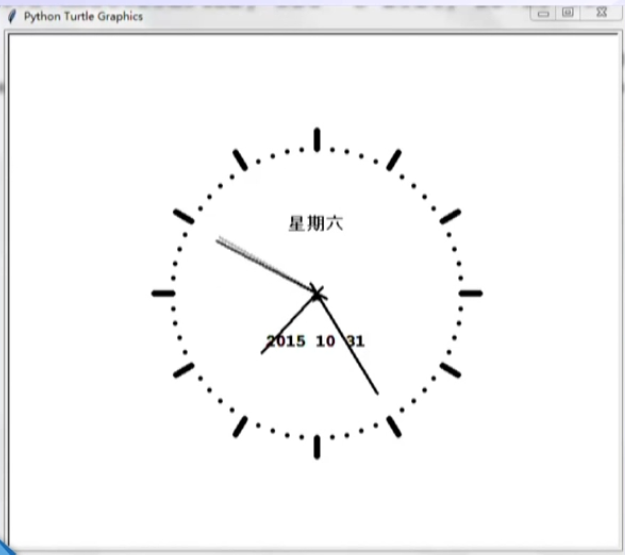






clock.py

from turtle import \*  
from datetime import \*  
  
  
def Skip(step):  
 penup()  
 forward(step)  
 pendown()  
  
  
def mkHand(name, length):  
 # 注册Turtle形状，建立表针Turtle  
 reset()  
 Skip(-length \* 0.1)  
 begin\_poly()  
 forward(length \* 1.1)  
 end\_poly()  
 handForm = get\_poly()  
 register\_shape(name, handForm)  
  
  
def Init():  
 global secHand, minHand, hurHand, printer  
 mode("logo") # 重置Turtle指向北  
 # 建立三个表针Turtle并初始化  
 mkHand("secHand", 125)  
 mkHand("minHand", 130)  
 mkHand("hurHand", 90)  
 secHand = Turtle()  
 secHand.shape("secHand")  
 minHand = Turtle()  
 minHand.shape("minHand")  
 hurHand = Turtle()  
 hurHand.shape("hurHand")  
 for hand in secHand, minHand, hurHand:  
 hand.shapesize(1, 1, 3)  
 hand.speed(0)  
 # 建立输出文字Turtle  
 printer = Turtle()  
 printer.hideturtle()  
 printer.penup()  
  
  
def SetupClock(radius):  
 # 建立表的外框  
 reset()  
 pensize(7)  
 for i in range(60):  
 Skip(radius)  
 if i % 5 == 0:  
 forward(20)  
 Skip(-radius - 20)  
 else:  
 dot(5)  
 Skip(-radius)  
 right(6)  
  
  
def Week(t):  
 week = ["星期一", "星期二", "星期三",  
 "星期四", "星期五", "星期六", "星期日"]  
 return week[t.weekday()]  
  
  
def Date(t):  
 y = t.year  
 m = t.month  
 d = t.day  
 return "%s %d %d" % (y, m, d)  
  
  
def Tick():  
 # 绘制表针的动态显示  
 t = datetime.today()  
 second = t.second + t.microsecond \* 0.000001  
 minute = t.minute + second / 60.0  
 hour = t.hour + minute / 60.0  
 secHand.setheading(6 \* second)  
 minHand.setheading(6 \* minute)  
 hurHand.setheading(30 \* hour)  
  
 tracer(False)  
 printer.forward(65)  
 printer.write(Week(t), align="center",  
 font=("Courier", 14, "bold"))  
 printer.back(130)  
 printer.write(Date(t), align="center",  
 font=("Courier", 14, "bold"))  
 printer.home()  
 tracer(True)  
  
 ontimer(Tick, 100) # 100ms后继续调用tick  
  
  
def main():  
 tracer(False)  
 Init()  
 SetupClock(160)  
 tracer(True)  
 Tick()  
 mainloop()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()



## 绘制彩虹

from turtle import \*  
  
  
def HSB2RGB(hues):  
 hues = hues \* 3.59 # 100转成395范围  
 rgb = [0.0, 0.0, 0.0]  
 i = int(hues / 60) % 6  
 f = hues / 60 - i  
 if i == 0:  
 rgb[0] = 1;  
 rgb[1] = f;  
 rgb[2] = 0  
 elif i == 1:  
 rgb[0] = 1 - f;  
 rgb[1] = 1;  
 rgb[2] = 0  
 elif i == 2:  
 rgb[0] = 0;  
 rgb[1] = 1;  
 rgb[2] = f  
 elif i == 3:  
 rgb[0] = 0;  
 rgb[1] = 1 - f;  
 rgb[2] = 1  
 elif i == 4:  
 rgb[0] = f;  
 rgb[1] = 0;  
 rgb[2] = 1  
 elif i == 5:  
 rgb[0] = 1;  
 rgb[1] = 0;  
 rgb[2] = 1 - f  
 return rgb  
  
  
def rainbow():  
 hues = 0.0  
 color(1, 0, 0)  
 # 绘制彩虹  
 hideturtle()  
 speed(100)  
 pensize(3)  
 penup()  
 goto(-400, -300)  
 pendown()  
 right(110)  
 for i in range(100):  
 circle(1000)  
 right(0.13)  
 hues = hues + 1  
 rgb = HSB2RGB(hues)  
 color(rgb[0], rgb[1], rgb[2])  
 penup()  
  
  
def main():  
 setup(800, 600, 0, 0)  
 bgcolor((0.8, 0.8, 1.0))  
 tracer(False)  
 rainbow()  
 # 输出文字  
 tracer(False)  
 goto(100, -100)  
 pendown()  
 color("red")  
 write("Rainbow", align="center",  
 font=("Script MT Bold", 80, "bold"))  
 tracer(True)  
  
 mainloop()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

结果：

