Turtle 实例

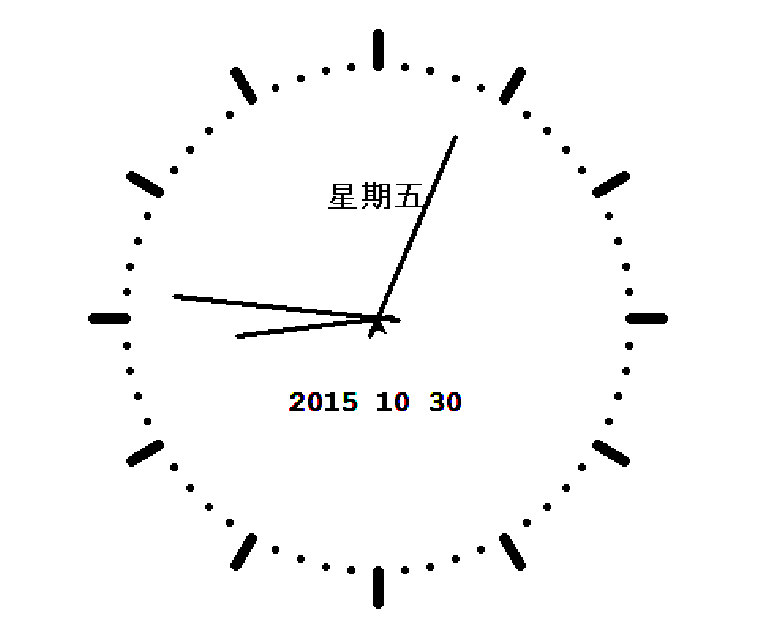
黄天羽

北京理工大学

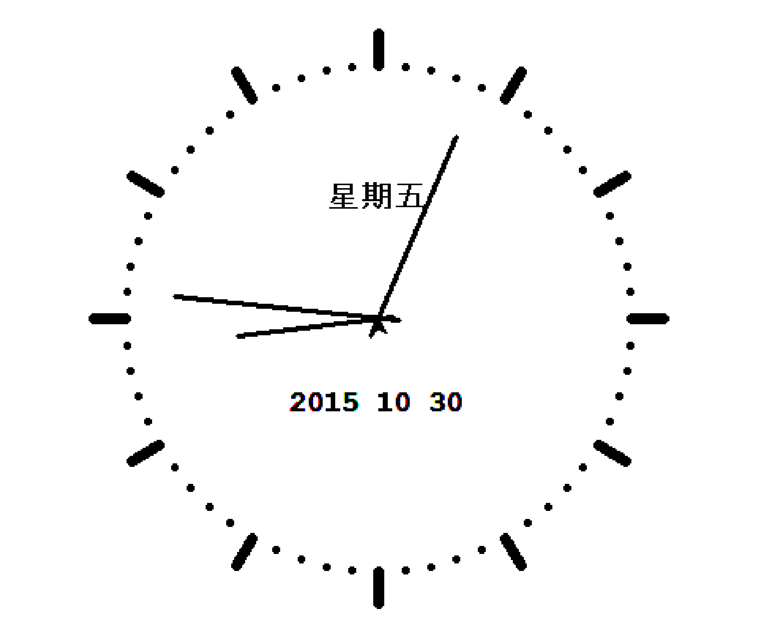
python 语言程序设计

时钟模拟

- 编写Python程序模拟时钟，要求时钟根据计算机系统时间实时动态更新。



- 5个turtle对象
  - 1个turtle: 绘制外表盘
  - 3个turle: 模拟表针行为
  - 1个turtle: 输出表盘上文字

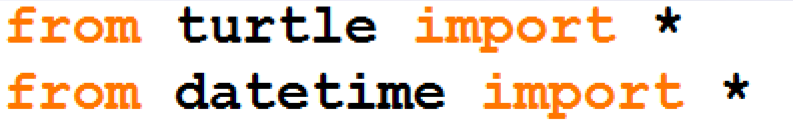


模拟时钟程序过程

- 第一步：建立Turtle对象并初始化。
  - 表盘绘制Turtle对象
  - 文本输出Turtle对象
  - 3个指针Turtle对象
- 第二步：静态表盘绘制
- 第三步：根据时钟更新表针位置和时间信息

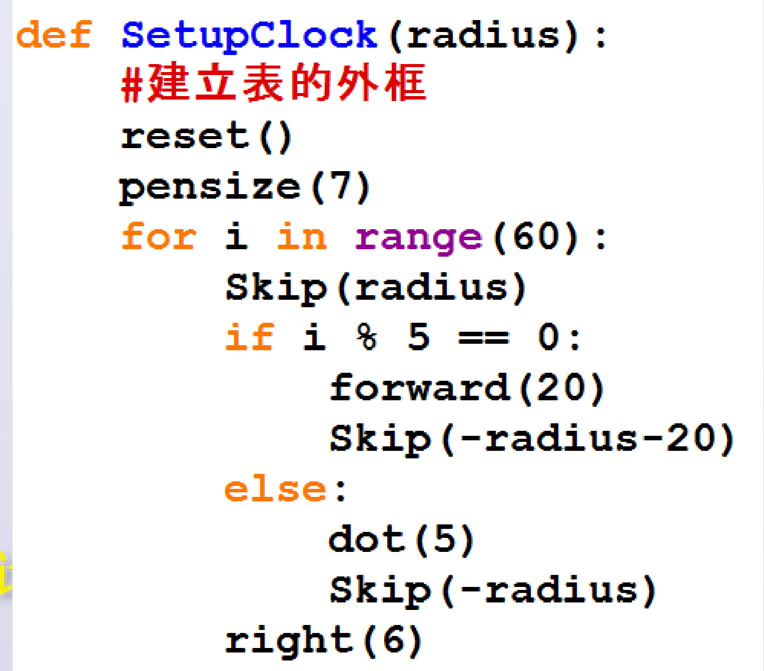
OS平台编程的需求

- Python基本库
  - Turtle
  - datetime



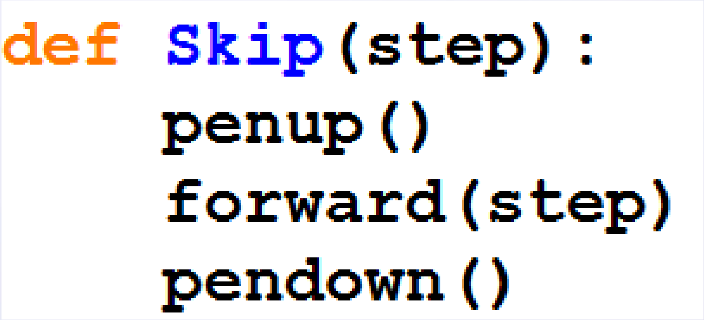
```
from turtle import *
from datetime import *
```

- 表盘绘制函数SetupClock(radius)



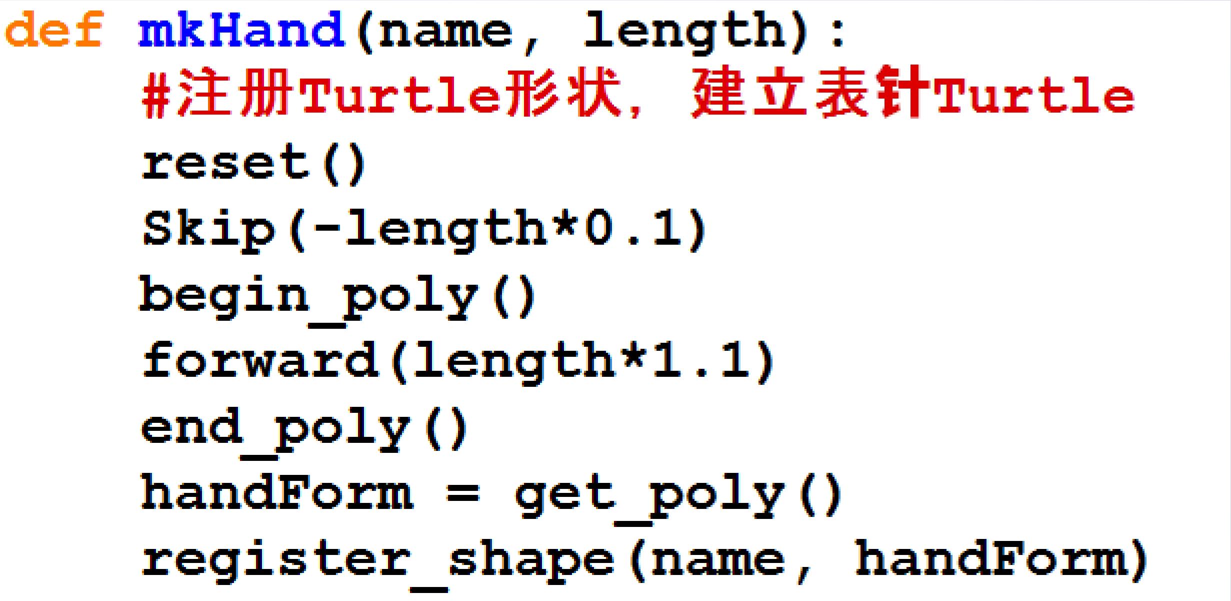
```
def SetupClock(radius):
    #建立表的外框
    reset()
    pensize(7)
    for i in range(60):
        Skip(radius)
        if i % 5 == 0:
            forward(20)
            Skip(-radius-20)
        else:
            dot(5)
            Skip(-radius)
        right(6)
```

- 跨越函数Skip（step）



```
def Skip(step):
    penup()
    forward(step)
    pendown()
```

- 定义表针函数mkHand（ ）



```
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)
```

OS平台编程的需求

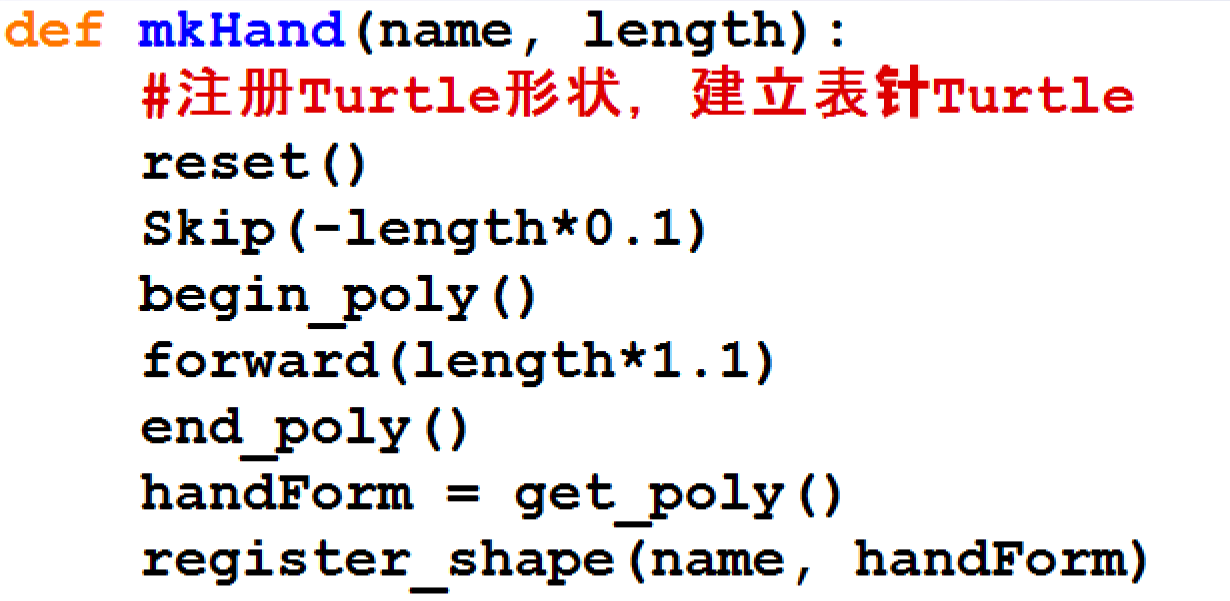
- 注册Turtle形状命令

  register_shap(name, shape=None)

- name：shape的名字，可以是一个gif图像

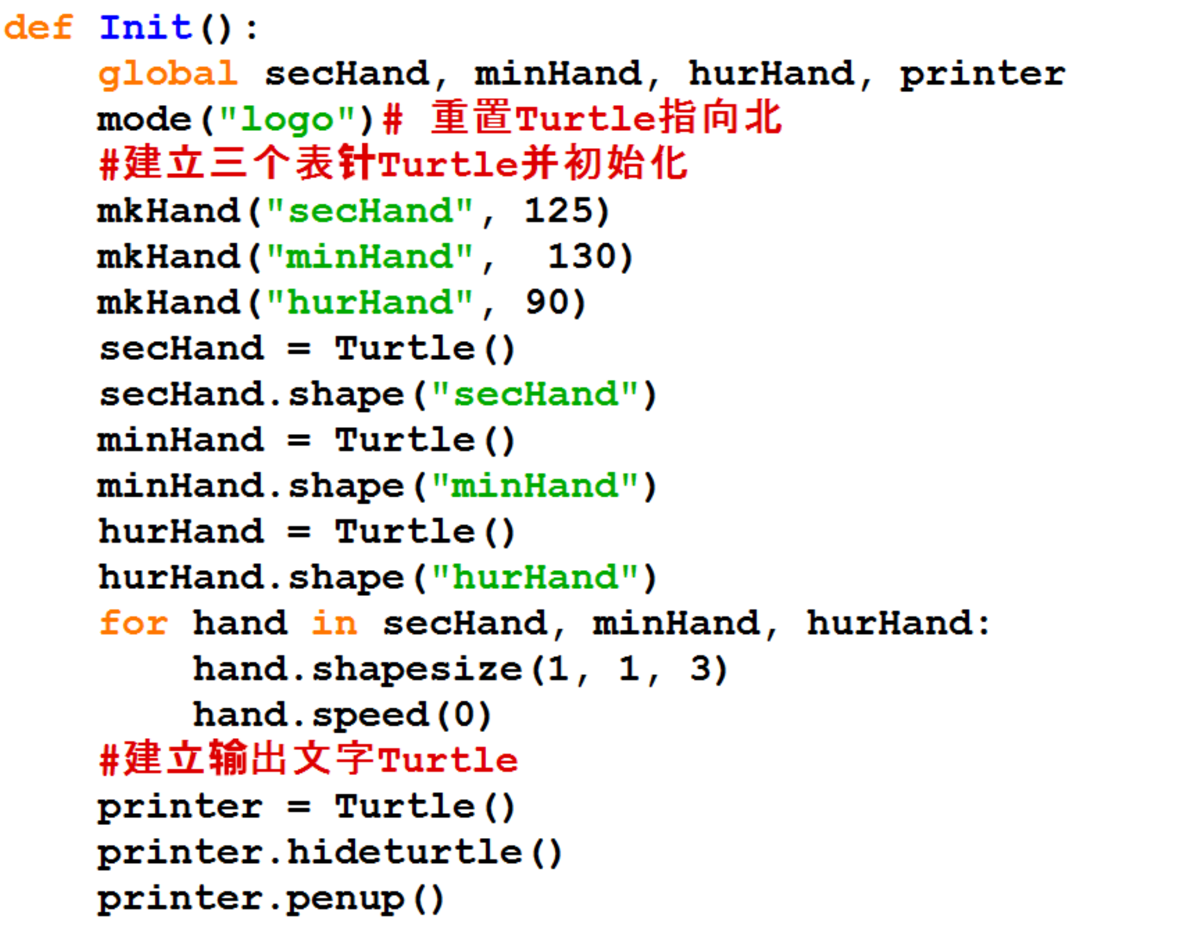
```
register_shape("turtle.gif")
```

- shape：turtle形状，可以为空



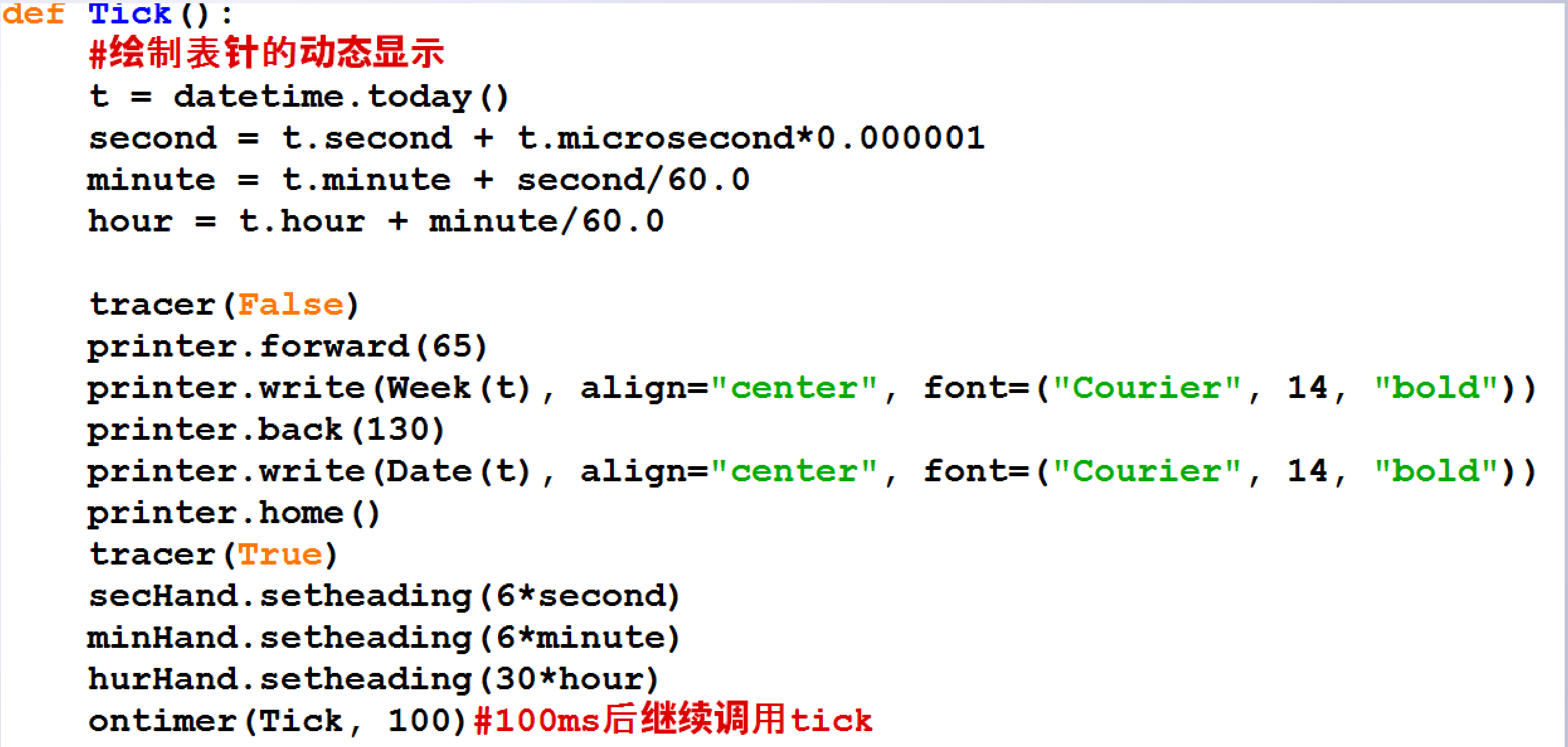
```
def mkHand(name, length):
    #注册Turtle形状，建立表针Turtle
    reset()
    Skip(-length*0.1)
    begin_poly()
    forward(length*1.1)
    end_poly()
    handForm = get_poly()
    register_shape(name, handForm)
```

初始化函数Init（ ）



```
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()
```

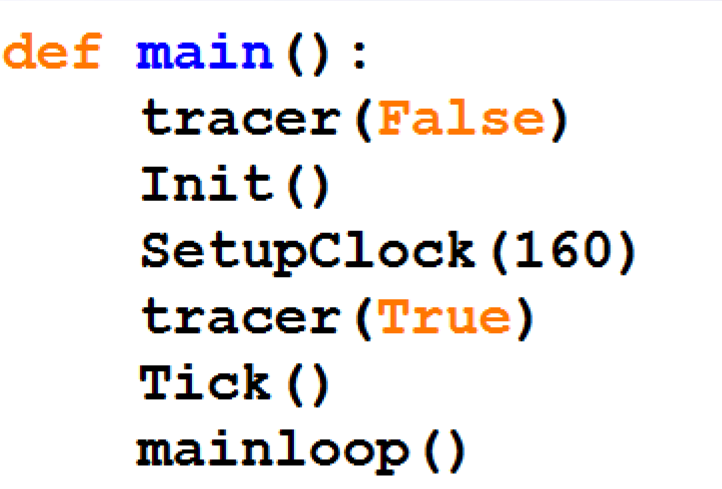
更新时钟函数Tick()



```
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

    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)
    secHand.setheading(6*second)
    minHand.setheading(6*minute)
    hurHand.setheading(30*hour)
    ontimer(Tick, 100)#100ms后继续调用tick
```

主函数main()



```python
def main():
    tracer(False)
    Init()
    SetupClock(160)
    tracer(True)
    Tick()
    mainloop()
```

程序Python代码见clock.py