

## 1、九九乘法表

```
for i in range(1, 10):
    for j in range(1, i + 1):
        print('{}x{}={} \t'.format(j, i, i * j), end='')
    print()
```

自然语言解释流程：

首先，九九乘法表需要1~9共9个数，于是首先用`range(1,10)`进行遍历，`i`便会依次取值1、2.....9，于是引入新的变量`j`进行遍历，`range(1,i+1)`，这样做的思想是一行一行的绘制，最后拼接！

于是，用`end=""`进行拼接，用`format`进行对应传参，`\t`是为了格式好看  
最后`print()`相当于换行的意思

效果图如下：

```
1x1=1
1x2=2   2x2=4
1x3=3   2x3=6   3x3=9
1x4=4   2x4=8   3x4=12  4x4=16
1x5=5   2x5=10  3x5=15  4x5=20  5x5=25
1x6=6   2x6=12  3x6=18  4x6=24  5x6=30  6x6=36
1x7=7   2x7=14  3x7=21  4x7=28  5x7=35  6x7=42  7x7=49
1x8=8   2x8=16  3x8=24  4x8=32  5x8=40  6x8=48  7x8=56  8x8=64
1x9=9   2x9=18  3x9=27  4x9=36  5x9=45  6x9=54  7x9=63  8x9=72  9x9=81
```

## 2、判断某年是否是闰年

```
year = int(input("请输入一个年份："))
if (year % 4) == 0 and (year % 100) != 0 or (year % 400) == 0:
    print("{}是闰年".format(year))
else:
    print("{}不是闰年".format(year))
```

自然语言流程解释：

首先，使用遍历`year`来接收用户传入的年份的整数形式

接着，使用表达式进行判断，如果`year`可以被400整除或者是被4整除且不能被100整除，那么`year`就是闰年，否则`year`不是闰年

`if-else`是条件判断语句，先判断`if`的条件是否成立，成立的话就执行`if`的执行语句，否则执行`else`里面的执行语句！

效果图如下：

```
请输入一个年份：200
200不是闰年

进程已结束，退出代码 0
```

### 3、turtle绘制五角星

```
import turtle as t

window = t.Screen()
t.fillcolor('red')
t.begin_fill()
for i in range(5):
    if (i < 5):
        t.forward(100)
        t.right(144)
        i += 1
t.end_fill()
window.exitonclick()
```

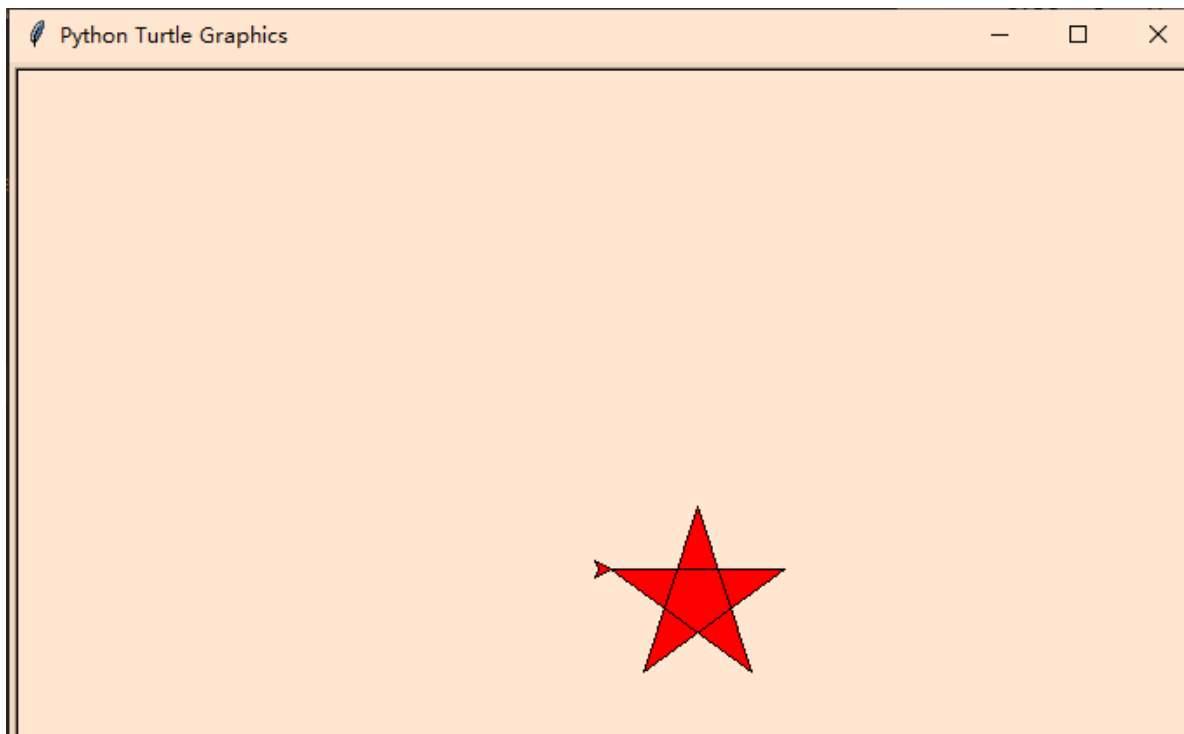
自然语言解释流程：

使用turtle库绘制五角星，首先创建一个窗口`window = turtle.Screen()`，接着开始填充颜色`turtle.fillcolor('red')`，让绘制的图形显示红色，`turtle.begin_fill()`开始填充，后面使用for循环来进行五角星的一笔勾勒，`forward(100)`代表的是当前箭头方向前进100像素，然后`right(144)`指的是顺时针旋转144度

这样循环5次后，能绘制出完整的五角星

最后，`turtle.end_fill()`结束填充，`window.exitonclick()`设置作用是使用当前窗口展示五角星

效果图如下：



### 4、turtle绘制五星红旗

```
import turtle as t
```

```
window = t.Screen()
t.bgcolor("red")
t.fillcolor("yellow")
t.color('yellow')
t.speed(10)
#主星
t.begin_fill()
t.up()
t.goto(-600,220)
t.down()
for i in range (5):
    t.forward(150)
    t.right(144)
t.end_fill()
#第1颗副星
t.begin_fill()
t.up()
t.goto(-400,295)
t.setheading(305)
t.down()
for i in range (5):
    t.forward(50)
    t.left(144)
t.end_fill()
#第2颗副星
t.begin_fill()
t.up()
t.goto(-350,212)
t.setheading(30)
t.down()
for i in range (5):
    t.forward(50)
    t.right(144)
t.end_fill()
#第3颗副星
t.begin_fill()
t.up()
t.goto(-350,145)
t.setheading(5)
t.down()
for i in range (5):
    t.forward(50)
    t.right(144)
t.end_fill()
#第4颗副星
t.begin_fill()
t.up()
t.goto(-400,90)
t.setheading(300)
t.down()
for i in range (5):
    t.forward(50)
    t.left(144)
t.end_fill()
window.exitonclick()
```

自然语言解释：

**turtle**绘制五星红旗难点在于，几个五角星的方位

首先，创建一个窗口`window = t.Screen()`，设置窗口背景色是红色，设置绘制图形填充色是黄色，设置绘制速度**10**，这样红旗的背景和长方体已经完成！

接着绘制主星，先找到适合绘制的方位点`t.goto(-600,220)`，然后就开始**for**循环，5次遍历，每次**forward(100)**以及**right(144)**，最后填充结束`end_fill()`就可以绘制完毕主星

4颗副星操作类似，也是找到方位，然后**for**循环遍历操作

最后使用当前窗口展示此五星红旗

效果图如下：

