

# 山东建筑大学 计算机学院 实验报告

班级： 软件 182 姓名： 吴官峰 学 号： 201811104062 实验成绩： \_\_\_\_\_

课程： python 编程基础 \_\_\_\_\_ 同组者： \_\_\_\_\_ 实验日期： \_\_\_\_\_

---

## 实验二 Python 程序控制结构

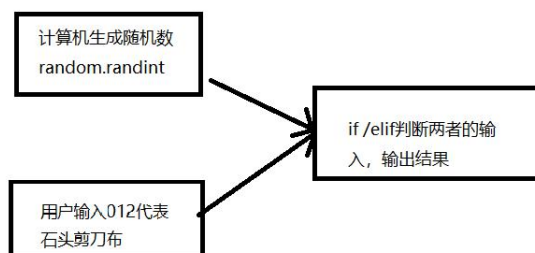
### 一、实验目的

学习 python 分支结构和循环结构编程。

### 二、实验内容

#### 1、编程实现剪刀石头布游戏。

设计思路（流程图，算法等）



代码截图

```
import sys
import random
handle='C'
while handle=='C' :
    computer = random.randint(0, 2)
    user = int(input('请根据对应编号出拳：0.剪刀，1.石头，2.布'))
    if computer == 0:
        computer = '剪刀'
    elif computer == 1:
        computer = '石头'
    else:
        computer = '布'
    if user == 0:
        user = '剪刀'
    elif user == 1:
        user = '石头'
    else:
        user = '布'
    print('电脑出的是%s, 用户出的是%s' % (computer, user))
    if ((computer == '剪刀' and user == '石头')
        or (computer == '石头' and user == '布')
        or (computer == '布' and user == '剪刀')):
        print('用户获胜')
    elif user == computer:
        print('双方平局')
    else:
        print('电脑获胜')
    handle = input('用户是否选择继续，C.继续，E.结束')
    if handle == 'E':
        print('游戏结束')
        sys.exit()
    elif handle != 'C' and handle != 'E':
        print('输入有误，请重新输入')
        handle = input()
```

# 山东建筑大学 计算机学院 实验报告

班级： 软件 182 姓名： 吴官峰 学 号： 201811104062 实验成绩： \_\_\_\_\_

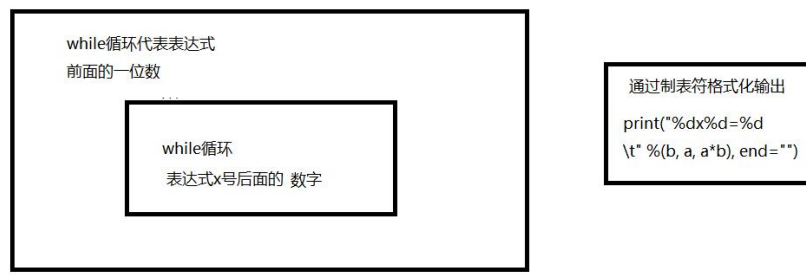
课程： python 编程基础 同组者： \_\_\_\_\_ 实验日期： \_\_\_\_\_

## 运行结果截图

```
=====
请根据对应编号出拳：0. 剪刀， 1. 石头， 2. 布0
电脑出的是布，用户出的是剪刀
用户获胜
用户是否选择继续：C. 继续， E. 结束C
请根据对应编号出拳：0. 剪刀， 1. 石头， 2. 布1
电脑出的是剪刀，用户出的是石头
用户获胜
用户是否选择继续：C. 继续， E. 结束C
请根据对应编号出拳：0. 剪刀， 1. 石头， 2. 布2
电脑出的是剪刀，用户出的是布
电脑获胜
用户是否选择继续：C. 继续， E. 结束E
游戏结束
>>>
```

2、编程显示一个三角形状的乘法口诀表。

## 设计思路（流程图，算法等）



## 代码

```
a = 1
while a < 10:
    b = 1
    while b <= a:
        print("%dx%d=%d\t" % (b, a, a*b), end=" ")
        b = b + 1
    print("\n")
    a = a + 1
```

## 运行结果

```
=====  
===== RESTART: E:\大三上\Python\实验二\乘法口诀.py =====  
=====
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
>>>
```

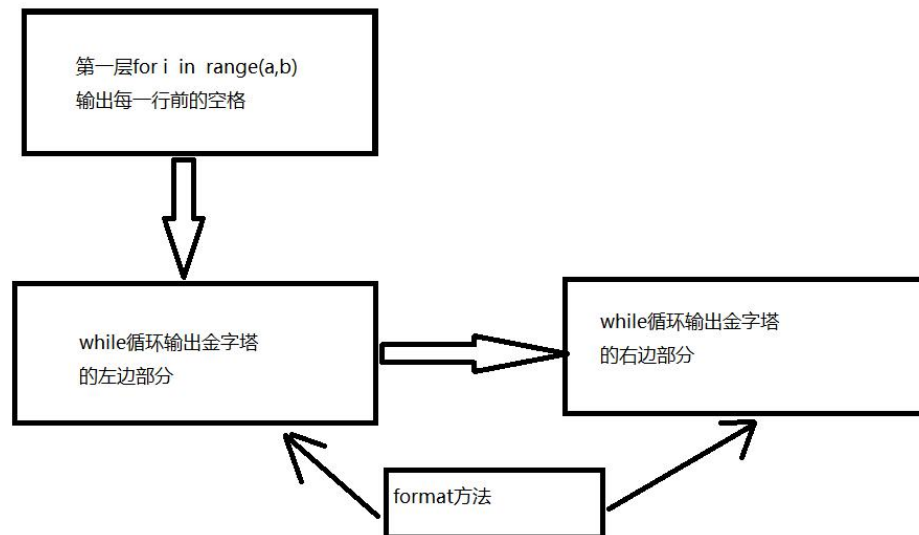
# 山东建筑大学 计算机学院 实验报告

班级： 软件 182 姓名： 吴官峰 学 号： 201811104062 实验成绩： \_\_\_\_\_

课程： python 编程基础 \_\_\_\_\_ 同组者： \_\_\_\_\_ 实验日期： \_\_\_\_\_

## 3、编程显示一个金字塔。

设计思路（流程图，算法等）



代码

```
a = int(input('请输入层数：'))
for i in range(a):
    print(' ' * (a-i), end='')
    j = i+1
    while j != 0:
        print(format(j, "3d"), end='')
        j = j - 1
    m=2
    while m != i+2:
        print(format(m, "3d"), end='')
        m = m + 1
    print()
```

运行结果

```
===== RESTART: E:\大三上\Python\实验二\金字塔.py =====
请输入层数： 11
      1
     2 1 2
    3 2 1 2 3
   4 3 2 1 2 3 4
  5 4 3 2 1 2 3 4 5
 6 5 4 3 2 1 2 3 4 5 6
 7 6 5 4 3 2 1 2 3 4 5 6 7
 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8
 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
10 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9 10
11 10 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9 10 11
>>> |
```

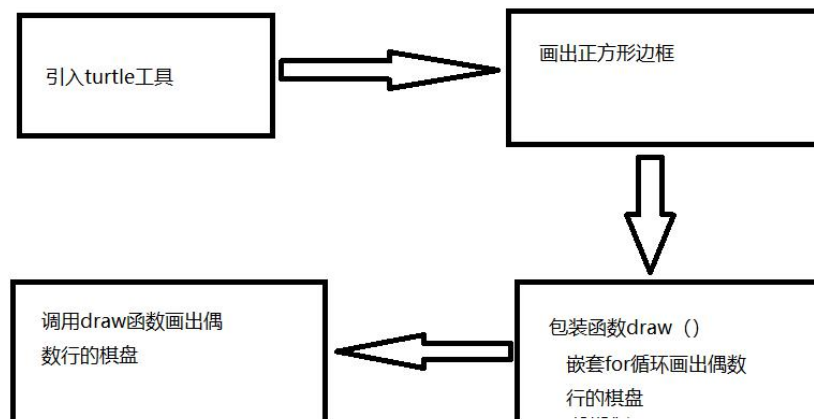
# 山东建筑大学 计算机学院 实验报告

班级： 软件 182 姓名： 吴官峰 学 号： 201811104062 实验成绩： \_\_\_\_\_

课程： python 编程基础 \_\_\_\_\_ 同组者： \_\_\_\_\_ 实验日期： \_\_\_\_\_

## 4、编写程序绘制一个棋盘

设计思路（流程图，算法等）



代码

```
import turtle

L = 320

def draw():
    for a in range(4):
        for i in range(4):
            turtle.begin_fill()
            for j in range(4):
                turtle.forward(L/8)
                turtle.left(90)
            turtle.end_fill()
            turtle.penup()
            turtle.forward(L/4)
            turtle.pendown()
        turtle.penup()
        turtle.backward(L)
        turtle.left(90)
        turtle.forward(L/4)
        turtle.right(90)
        turtle.pendown()

turtle.pensize(3)
turtle.color("red")
turtle.penup()
turtle.goto(-L/2, -L/2)
turtle.setheading(0)
turtle.pendown()
for i in range(4):
    turtle.forward(L)
    turtle.left(90)
turtle.speed(11)
turtle.color("black")
draw()
turtle.penup()
turtle.goto(-120, -120)
turtle.pendown()
draw()
```

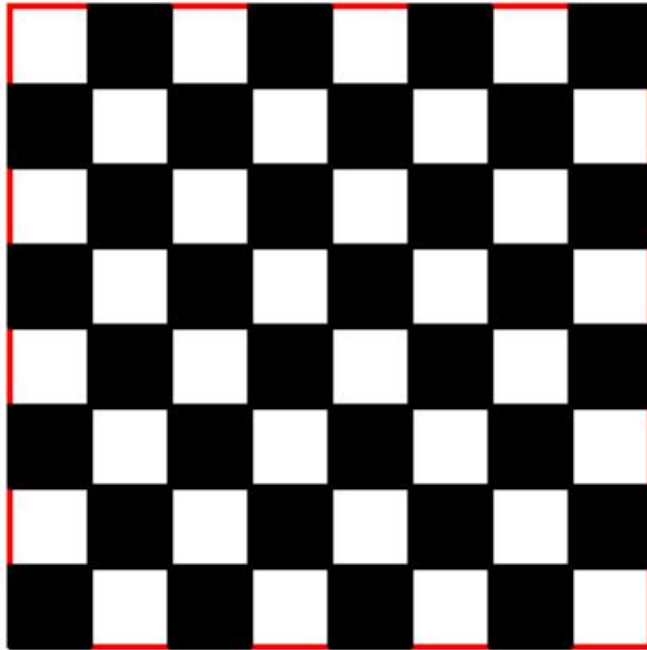
# 山东建筑大学 计算机学院 实验报告

班级： 软件 182 姓名： 吴官峰 学 号： 201811104062 实验成绩： \_\_\_\_\_

课程： python 编程基础 同组者： \_\_\_\_\_ 实验日期： \_\_\_\_\_

---

## 运行结果



## 三、实验总结

1.和上次实验的比较，出现代码缩进的问题频率减少了很多，即使出现后也都能够及时的发现错误。

2.在绘制乘法口诀表时，由于输出格式问题出现了问题

解决方法：格式化输出，利用制表符\t 进行空格的处理

3. 绘制金字塔时也是在处理左半部分的空格时出现了问题

解决办法：由于本题题目在输出时最多只输出两位数，因此可以利用 `format` 方法进行格式输出，每个数字无论长短都占三个字符的长度