**Day04笔记**

day04(上午)：

**1.流程控制之判断结构：**

**格式三：**

**①**

**if 条件表达式1:**

**语句块1**

**elif 条件表达式2:**

**语句块2**

**elif 条件表达式3:**

**语句块3**

**...**

**else:**

**语句块n**

**②**

**需求：**

age = 28  
**if** age < 0 **or** age > 130:  
 print(**'你可能不是人...'**)  
**elif** age <= 30:  
 print(**'伦家还很小的啦，么么哒...'**)  
**else**:  
 print(**'别追我了，我都有两个娃了...'**)  
print(**'判断结构学完了...'**)

**需求：  
从键盘输入小明的期末考试成绩  
当成绩为100分时，奖励一辆BMW；  
当成绩为(80,99]时，奖励一台8848；  
当成绩为[60,80]时，奖励一本从入门到精通只需33天；  
其它时，暴打一顿！  
【注意】：  
如果多个条件表达式是包含关系，那么范围小的一定要定义在范围大的上面  
如果多个条件表达式是互斥关系，那么谁上谁下无所谓**score = int(input(**"请输入小明的考试成绩：(分)"**))  
**if** score < 0 **or** score > 100:  
 print(**'输入的成绩有误！请重新输入...'**)  
**else**:  
 **if** score == 100:  
 print(**'奖励一辆BMW...'**)  
 **elif** score > 80:  
 print(**'奖励一台8848...'**)  
 **elif** score >= 60:  
 print(**'奖励一本参考书...'**)  
 **else**:  
 print(**'老子要揍你了，5555~~~~~~'**)

执行流程：

先执行①，遇到if关键字，执行条件表达式1，

如果条件表达式1的结果为True，那么立即进入if内部执行语句块1，

直到语句块1执行完毕后，整个if elif else结构就结束了，最终程序一定会去执行②

如果条件表达式1的结果为False，那么程序不会执行语句块1，会去执行第一个elif后面的条件表达式2，

如果条件表达式2的结果为True，那么立即进入到elif内部执行语句块2，

直到语句块2执行完毕后，整个if elif else结构就结束了，最终程序一定会去执行②

...

如果所有的条件表达式的结果都为False，那么程序就会进入到else的内部去执行语句块n，

执行完毕后，整个if elif else结构结束了，一定会去执行②

**2.循环结构：**

分类：while、for ... in ...

while循环：

理解：有四要素组成

①.初始化条件(执行一次) 就是一个起点(起始值)，代码的体现就是定义一个变量记录初始值

②.循环条件(可能执行一次或多次) 循环到底是继续还是结束，由它把控；如果执行结果为True，则循环继续；反之，循环结束

③.迭代条件(可能执行0次或多次) 不断的去改变起始值，以达到控制循环次数的目的

④.循环体(可能执行0次或多次) 需要重复执行的逻辑代码

**格式：**

**①**

**while ②:**

**④**

**③**

**⑤**

执行流程：① -> ② -> ④ -> ③ -> ② -> ④ -> ③ -> ② -> ⑤

False

**需求：打印10遍"hello python"**

i = 0  
**while** i <= 9:  
 print(**"hello python"**)  
 i += 1

**需求：打印1~100之间的偶数，得到个数和总和**

**引入计数和累加的思想**i = 1  
myCount = 0 *#定义mySum变量用于计数(记录个数)*mySum = 0 *#定义mySum变量用于求和(记录和值)***while** i <= 100:  
 **if** i % 2 == 0:  
 print(i)  
 myCount += 1  
 mySum += i  
 i += 1  
print(**"1~100之间偶数的个数为：%d"** %myCount)  
print(**"1~100之间偶数的总和为：%d"** %mySum)

**需求：打印1~150，如果是3的倍数，追加"foo"，如果是5的倍数，追加"biz"，  
如果是7的倍数，追加"baz"  
效果：  
1  
2  
3 foo  
4  
5 biz  
6 foo  
7 baz  
...  
15 foo biz  
...  
105 foo biz baz  
...  
150 foo biz**i = 1  
**while** i <= 150:  
 print(i,end=**""**)  
 **if** i % 3 == 0:  
 print(**" foo"**,end=**""**)  
 **if** i % 5 == 0:  
 print(**" biz"**,end=**""**)  
 **if** i % 7 == 0:  
 print(**" baz"**,end=**""**)  
 *#此处的print()函数的作用仅仅是用来换行* print()  
 i += 1

**需求：完成10次加法测试，并输出得分****import** random  
print(**'即将开始10次加法测试，请做好准备...'**)  
*#变量score用于记录最终的得分*score = 0  
i = 1  
**while** i <= 10:  
 *#随机生成+左右两侧整数数据，范围[1,100]* num1 = random.randint(1,100)  
 num2 = random.randint(1,100)  
 *#得到一个正确结果* result = num1 + num2  
 *#开始出题：* print(**"%d).%d + %d = ?"** %(i,num1,num2))  
 *#从键盘读入用户输入的结果* answer = int(input())  
 *#将用户输入的和正确答案做比较* **if** answer == result:  
 *#如果代码能够执行到此处，说明答对了，给与正向鼓励，以及+10分* print(**'答对了，你真棒'**)  
 score += 10  
 **else**:  
 print(**'答错了，你真笨'**)  
 i += 1  
*#输出用户的最终得分*print(**'10次加法测试已经结束，最终得分为：%d'** %score)

---------------------------------------------------------------------------------------------------------------------------------

day04(下午)：

**1.random模块的初体验：**

需求：获取一个随机整数，范围：[m,n]

步骤：

第一步：导入random模块 **import random**

第二步：通过模块名.函数完成需求 **random.randint(m,n)**

作用：randint(m,n)函数，返回一个[m,n]的随机整数值**演示random模块的导入，以及randint函数的调用  
需求：获取一个[1,5]之间的随机整数***1.导入random模块***import** random  
*2.模块名.函数完成需求*num = random.randint(1,5)  
print(**'随机整数为：%d'** %num)

2.无限循环(死循环)：

无限循环是非常占用系统资源的，我们是需要避免的；但是如果它一旦配合break等关键字使用，就会变得有意义；

**格式：**

**while True:**

**循环体**

**演示无限循环的定义：****import** time

**while True**:  
 time.sleep(1)  
 print(**'马上有钱'**)  
**while** 1: *#计算机执行代码时，底层隐式的存在bool(1)代码* time.sleep(1)  
 print(**'老郭好帅，萌萌哒...'**)

【注意】:

i = 1  
*# i = 0***if** i: *#计算机执行代码时，底层隐式的存在bool(i)代码* print(**'结果为真'**)  
**else**:  
 print(**'结果为假'**)

3.**time模块的初体验：**

需求：实现间隔打印效果

步骤：

第一步：导入time模块 **import time**

第二步：通过模块名.函数完成需求 **time.sleep(sec)**

作用：sleep(sec)函数，程序一旦执行到sleep(sec)函数位置，会立即休眠sec秒，等到时间到了，自动醒过来，往下继续执行

4.**break&continue关键字：**

对于break而言： 如果在执行循环的过程中，遇到了break关键字，那么程序立即结束当前循环

对于continue而言： 如果在执行村换的过程中，遇到了continue关键字，那么程序立即结束当次循环，开始下一次循环

**演示break&continue关键字配合循环使用：  
注意事项：和break和continue在同一作用域，它们后面不允许出现任何的代码，因为永远不可能被执行到**i = 1  
**while** i < 11:  
 **if** i == 5:  
 i += 1  
 *# break #只能的得到1-4* **continue** *#1-10之间没有5* print(i)  
 i += 1  
**#需求：打印1~100之间的奇数，得分个数和总和，使用关键字continue**myCount = 0  
mySum = 0  
j = 1  
**while** j <= 100:  
 **if** j % 2 == 0:  
 j += 1  
 **continue** print(j)  
 myCount += 1  
 mySum += j  
 j += 1  
print(**'1~100之间的奇数的个数为：%d'** %myCount)  
print(**'1~100之间的奇数的总和为：%d'** %mySum)

5.**else语句配合循环使用：**

**特点：如果循环是正常执行完毕，那么else中的代码一定会被执行；\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**如果循环是由于break关键字而强行终止，那么else中的代码就不会被执行；**

**演示else语句配合循环使用：**i = 1  
**while** i <= 10:  
 **if** i == 5:  
 **break** print(i)  
 i += 1  
**else**:  
 print(**'一旦遇到牛逼break我就不会被执行'**)

**需求：完成10次加法测试，并输出得分 附加需求：输入-1程序循环强行结束  
import** random  
print(**'即将开始10次加法测试，请做好准备...'**)  
*#变量score用于记录最终的得分*score = 0  
i = 1  
**while** i <= 10:  
 *#随机生成+左右两侧整数数据，范围[1,100]* num1 = random.randint(1,100)  
 num2 = random.randint(1,100)  
 *#得到一个正确结果* result = num1 + num2  
 *#开始出题：* print(**"%d).%d + %d = ?(退出：-1)"** %(i,num1,num2))  
 *#从键盘读入用户输入的结果* answer = int(input())  
 *#将用户输入的和正确答案做比较* **if** answer == -1:  
 print(**'太遗憾了，下次再挑战吧！'**)  
 **break  
 elif** answer == result:  
 *#如果代码能够执行到此处，说明答对了，给与正向鼓励，以及+10分* print(**'答对了，你真棒'**)  
 score += 10  
 **else**:  
 print(**'答错了，你真笨'**)  
 i += 1  
**else**:  
 *#输出用户的最终得分* print(**'10次加法测试已经结束，最终得分为：%d'** %score)

6.转换函数：

注意：看案例