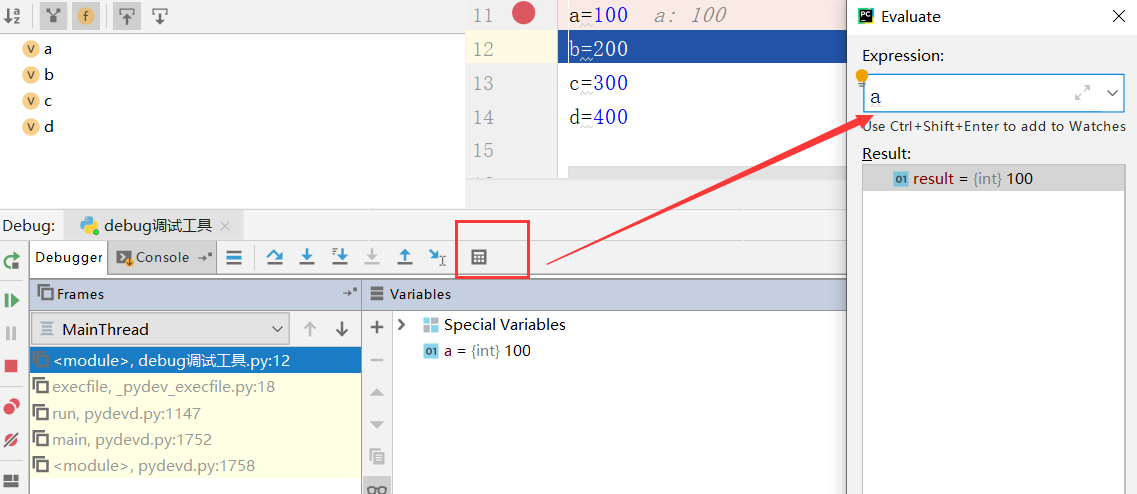
一、debug调试工具的使用

**F7**： 遇到函数调用，会进入到函数内部一步一步执行

**F8**： 往下走一步，不会进入到函数内部

**打断点**：行号后面点一下，会出现一个红点，使用dubug调试工具去运行代码，运行到断点处会暂停，暂停后可查看当前代码的变量值和类型等，调试的过程中不要更改还没运行的代码值。

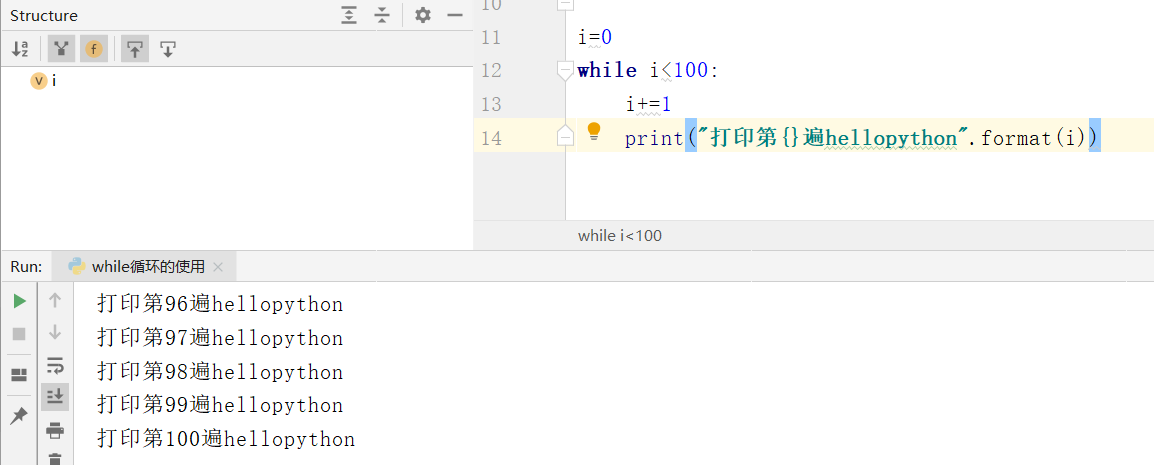
**小计算器**：在运行的过程中，可通过计算器来查找已运行过的代码数值。



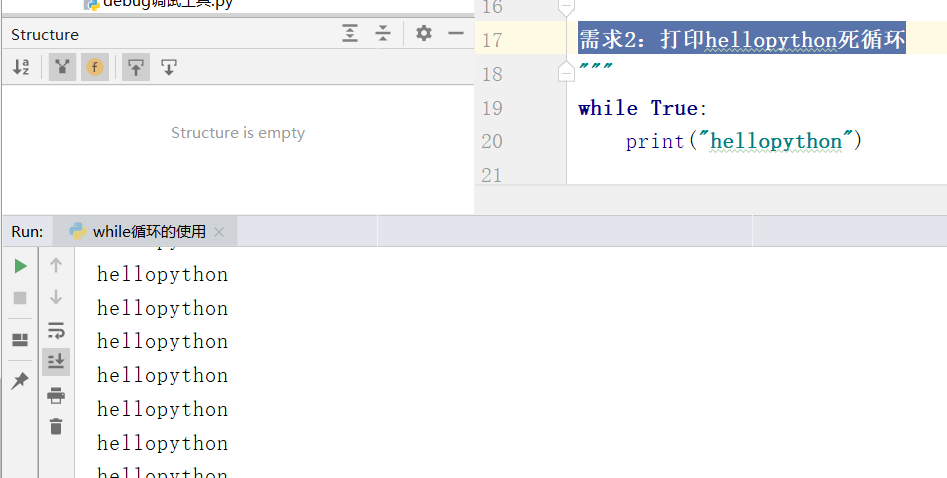
二、while循环

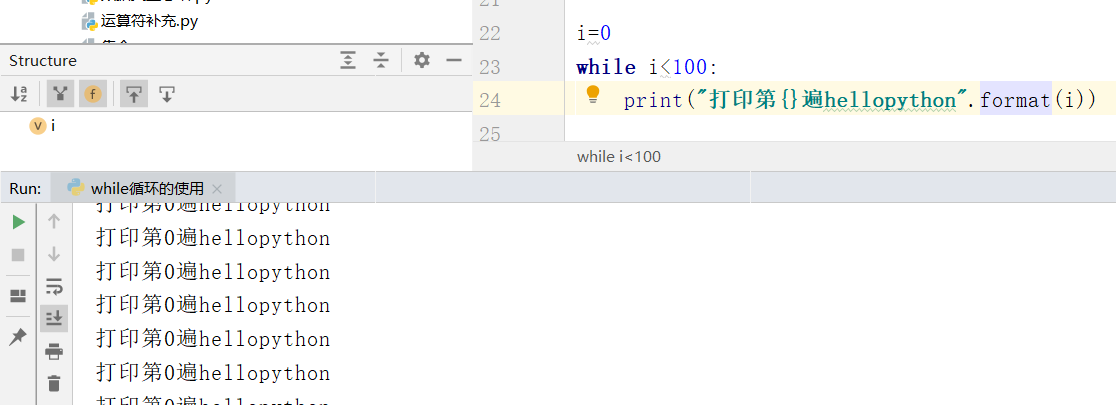
需求1：打印100遍hellopython

首先定义一个变量（i）来计数，记录打印了多少遍hellopython

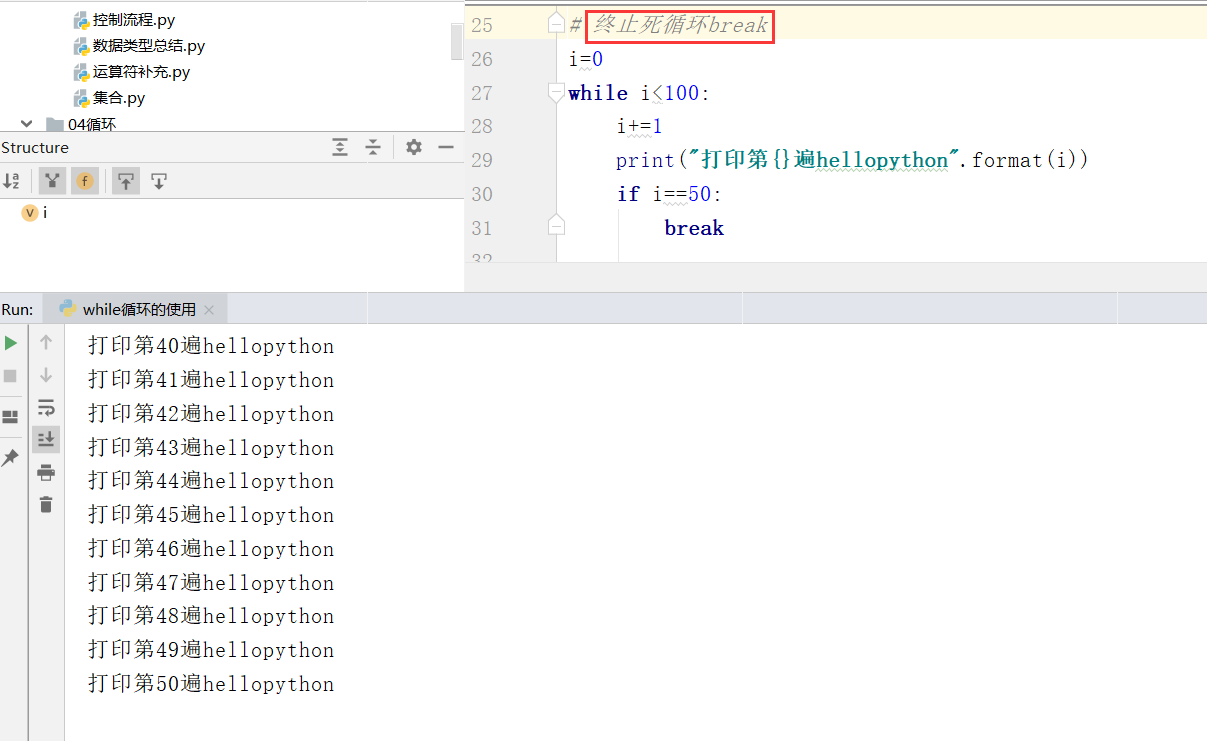


需求2：打印hellopython**死循环**

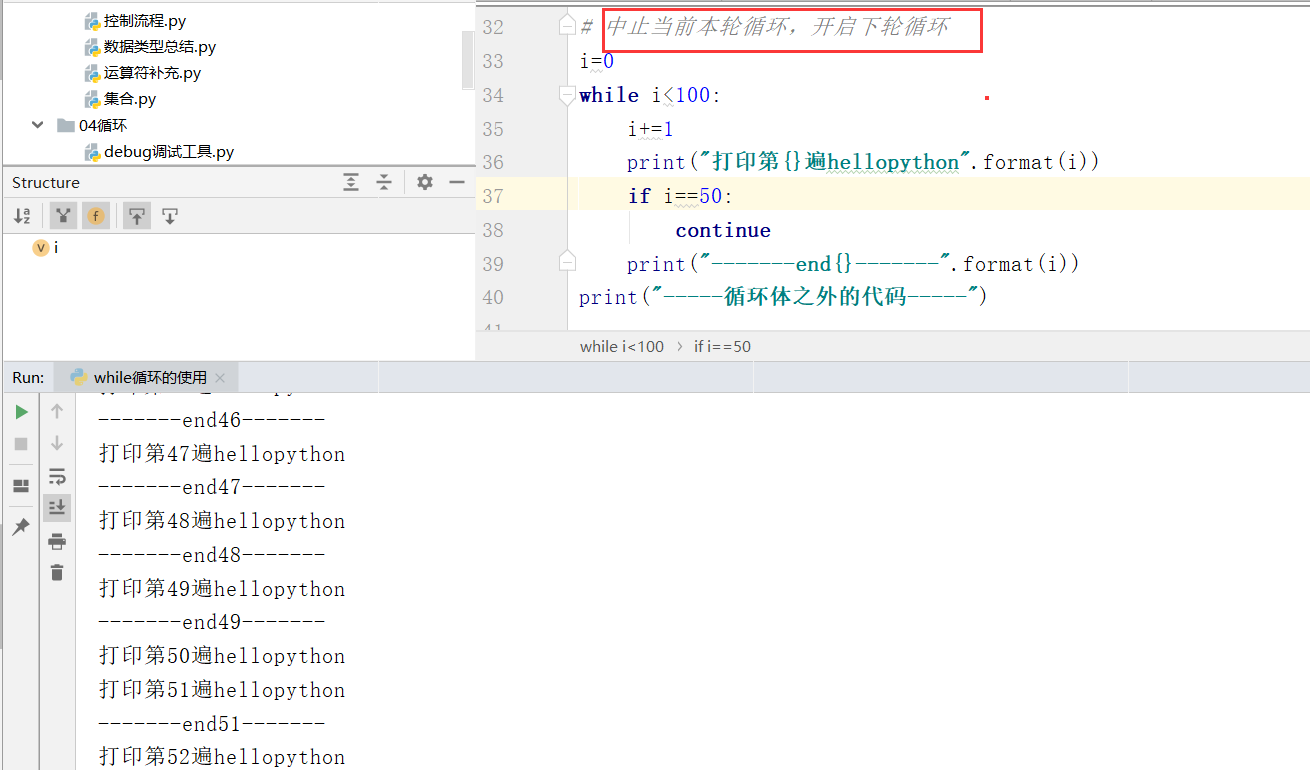




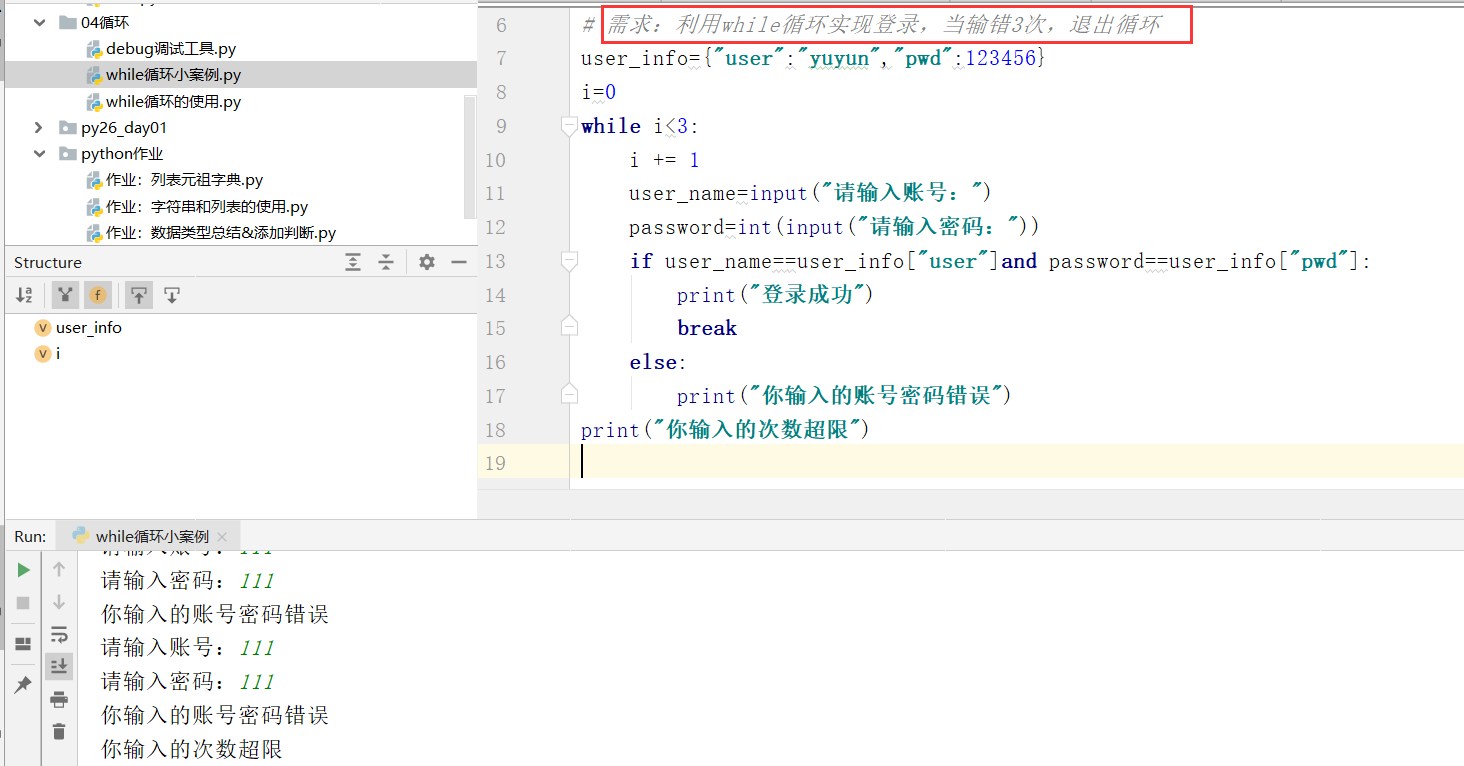
终止循环（包括死循环）**break，**break只能用在循环里面



继续循环**continue**：中止当前本轮循环，开启下一轮循环的条件判断（执行到continnue之后，会直接回到while后面进行条件判断）



需求3：利用while循环实现登录，当输错3次，提示输入次数超限

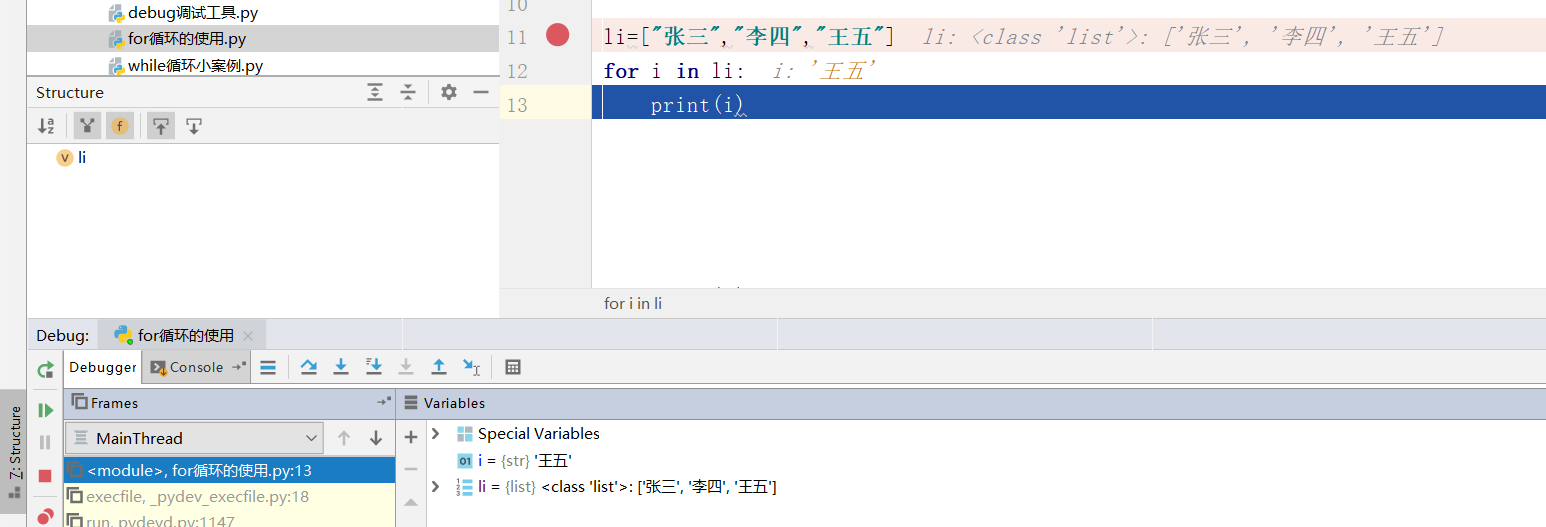


三、for循环（循环遍历）

* 语法：

for i in XXX:

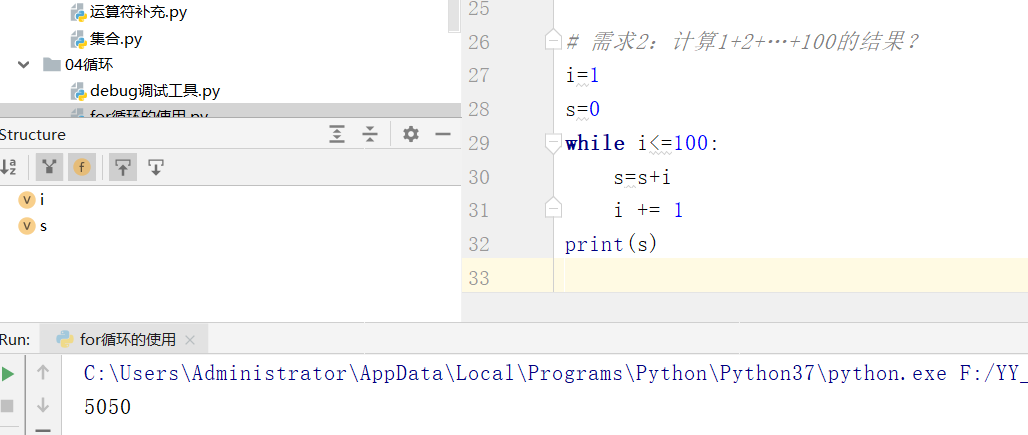
循环体

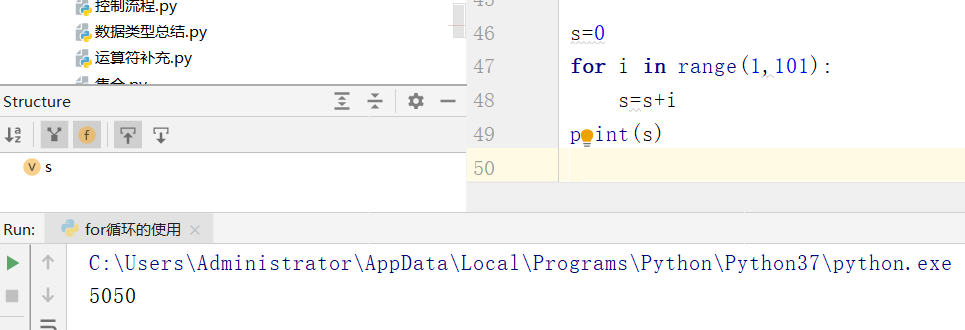


* 需求1：当前有10位同学的成绩，放在一个列表中，请区分成绩的等级



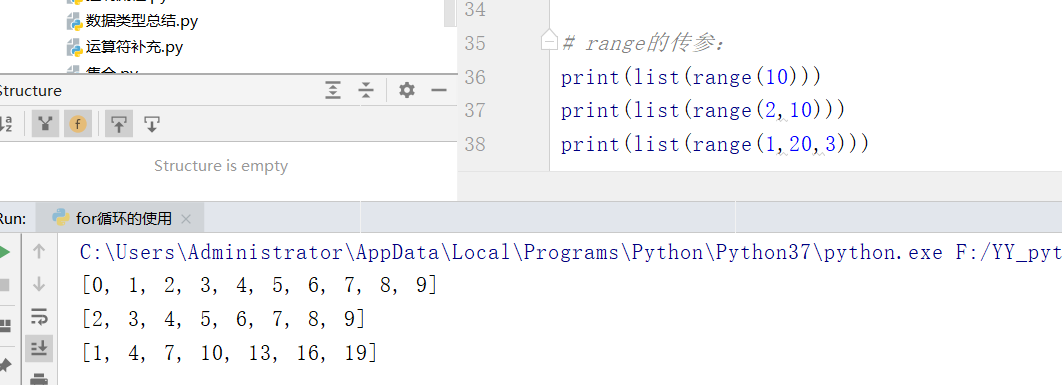
需求2：计算1+2+…+100的结果？



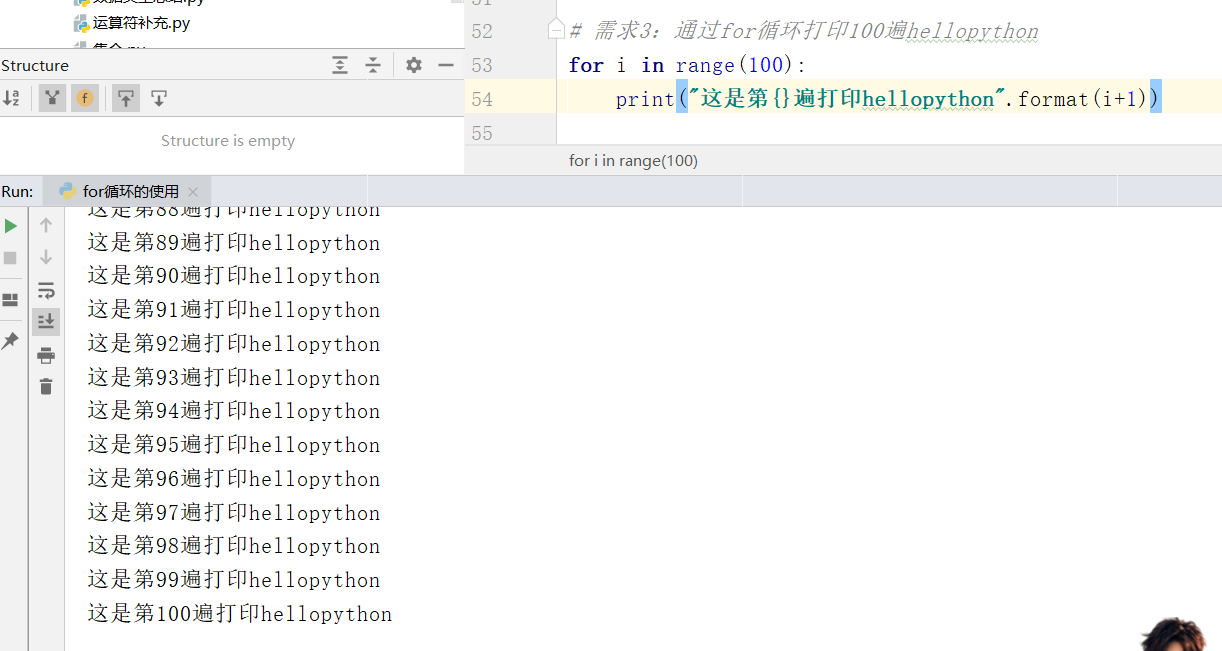


**内置函数range：**

1. range(n)：默认生成一个 0到n-1的整数序列，对于这个整数序列，我们可以通过list()函数转化为列表类型的数据
2. range(n,m)：默认生成一个n到m-1的整数序列，对于这个整数序列，我们可以通过list()函数转化为列表类型的数据
3. range(n,m,k)：相当于其他函数里面的for循环，n为初始值，m为结束值，k为步长，会生成初始值为n，结束值为m-1，递减或者是递增的整数序列
4. range返回的数据是支持使用for进行遍历的，也能够进行下标取值和切片（切片返回的还是range类型的数据）



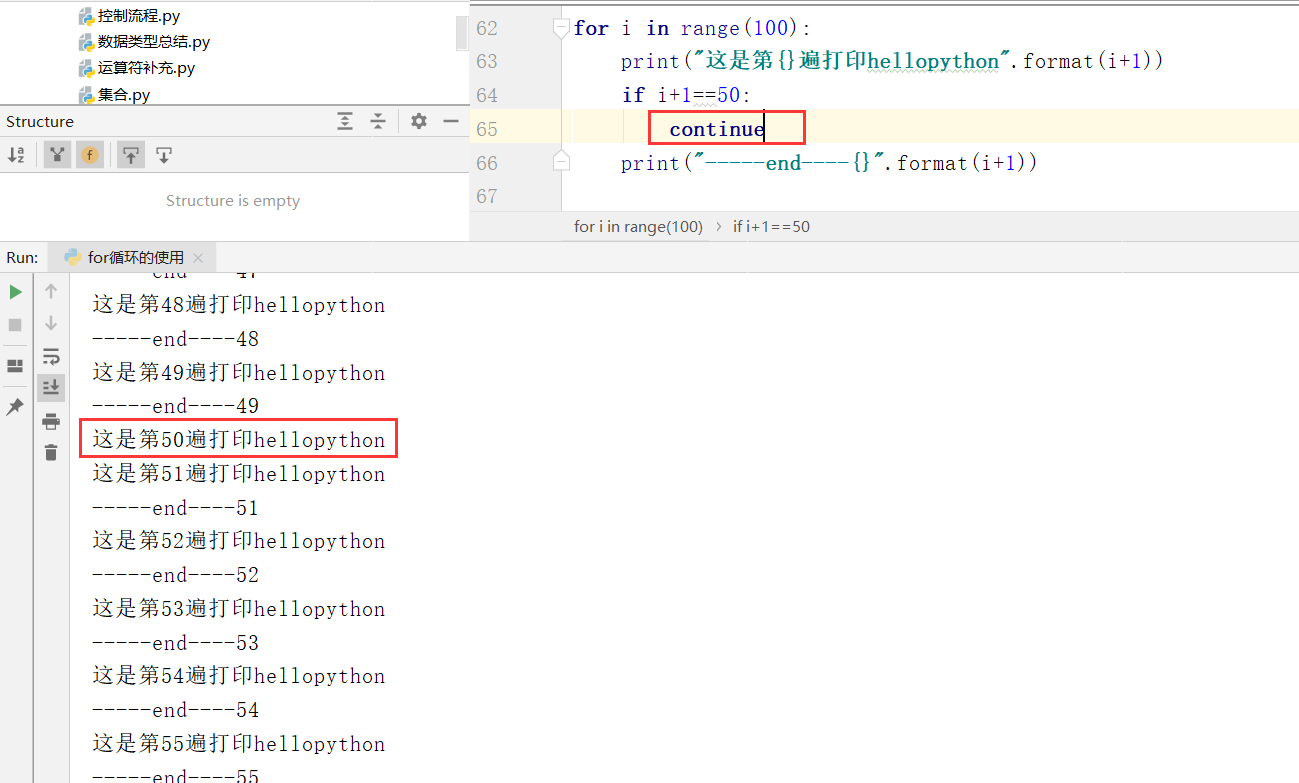
需求3：通过for循环打印100遍hellopython



需求4：通过for循环打印100遍hellopython，打印到第50遍终止。



需求5：通过for循环打印100遍hellopython，第50遍中止下一轮继续。

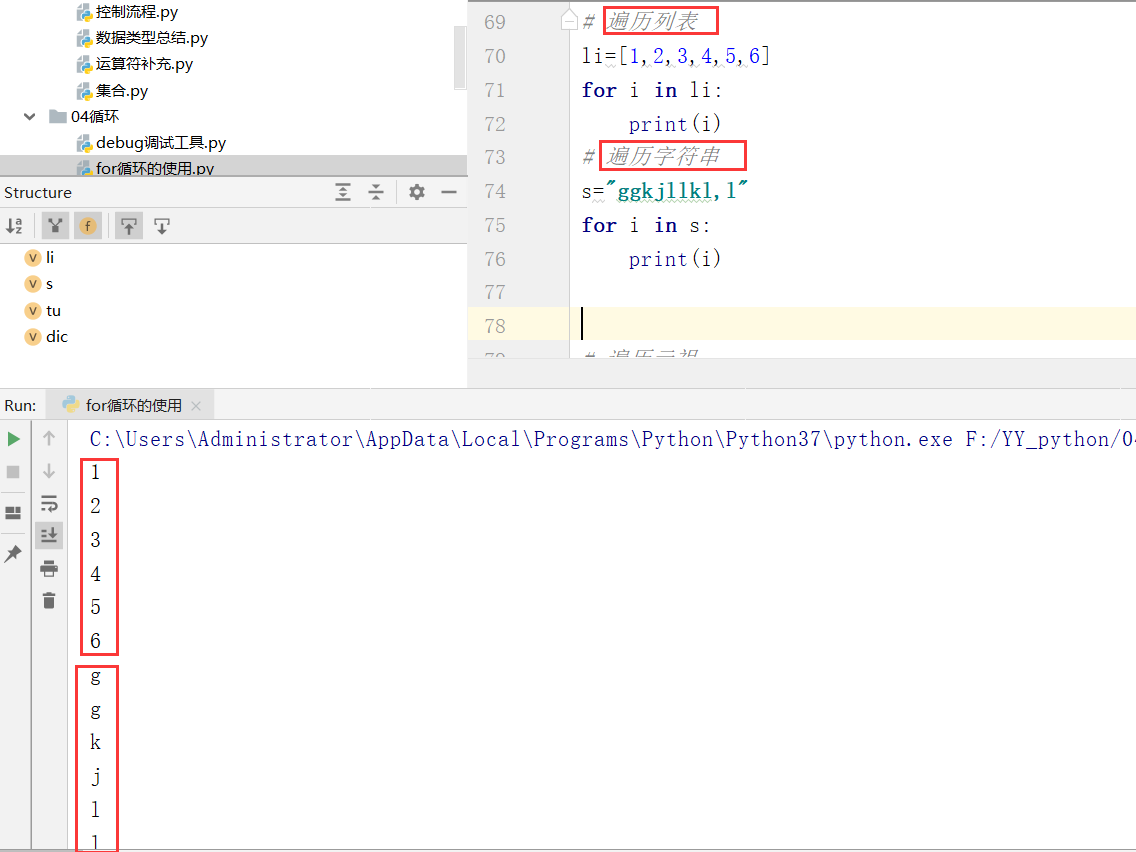


四、for循环的使用场景

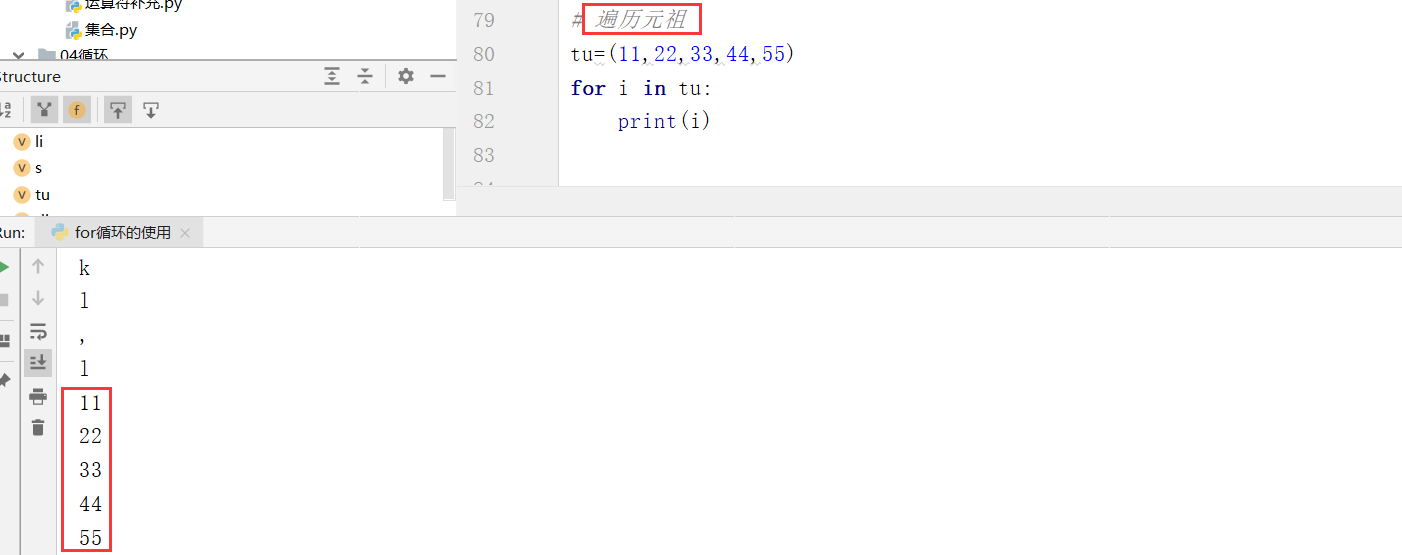
场景：可遍历列表、字符串、字典、元祖

# 遍历列表

# 遍历字符串



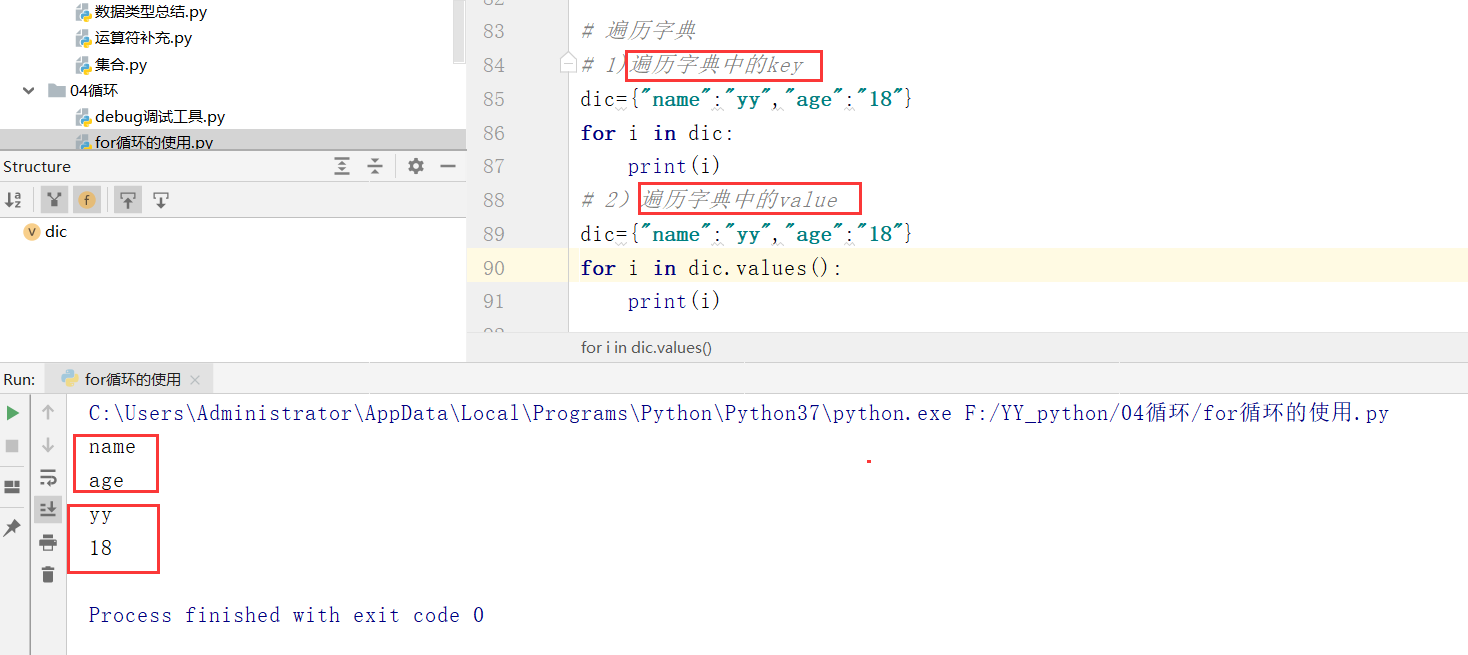
# 遍历元祖



**# 遍历字典**

# 1)遍历字典中的key

# 2）遍历字典中的value



# 3)遍历字典中的keys和values（返回的是元祖类型）

# 4)遍历字典中的keys和values（可以利用元祖拆包将返回的元祖拆开打印）

