

# 任务3-4 PYTHON其他语句

# break语句

这两个结果有什么不同？

break语句用于**结束整个循环**。

```
for i in range(5):  
    print("-----")  
    print i
```

整个程序运行了5次。

```
i=1  
for i in range(5):  
    i+=1  
    print("-----")  
    if i==3:  
        break  
    print(i)
```

当i=3时就会结束循环，整个程序只运行了3次。

# continue语句

continue的作用是用来**结束本次循环**，紧接着执行下一次的循环。

这两个结果有什么不同？

```
for i in range(5):  
    print("-----")  
    print i
```

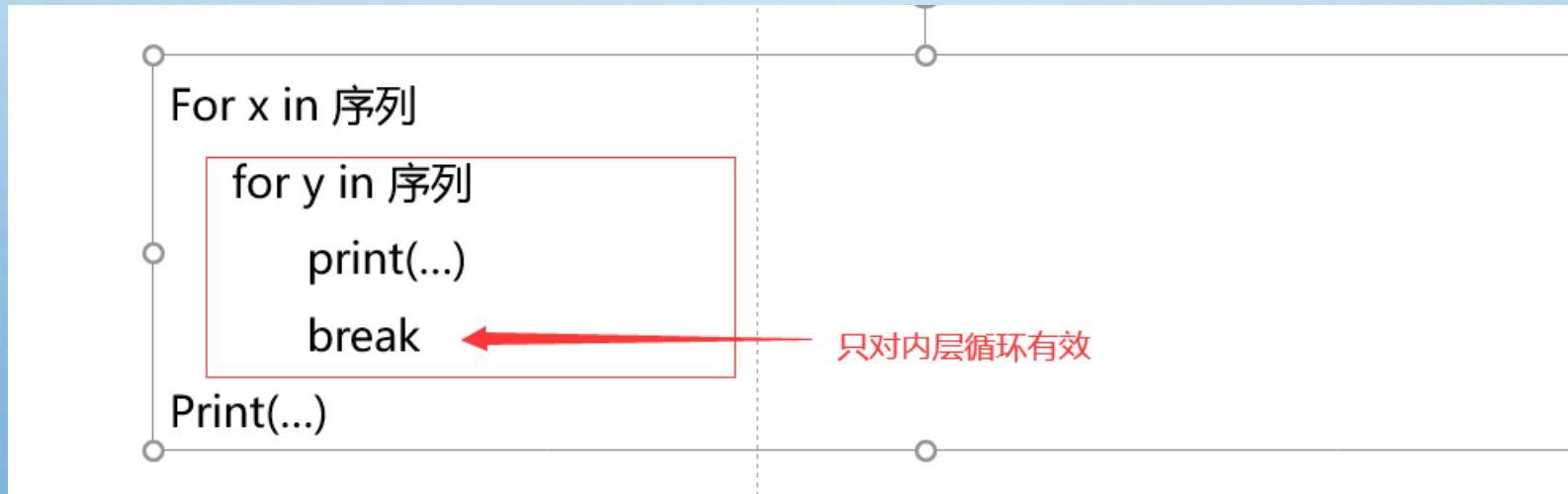
整个程序运行了5次。

```
i=1  
for i in range(5):  
    i+=1  
    print("-----")  
    if i==3:  
        continue  
    print(i)
```

当i=3时会结束本次循环，跳到循环开头重新开始，整个程序只运行了4次。i=3那次的横线不会打印出来。

## 注意：

- (1) break/continue 语句只能在循环中使用，不能单独使用。
- (2) break/continue 语句用在嵌套循环时，只会对其所处的最近一层循环起作用。



# pass语句

**pass是空语句**，它是为了保持程序结构完整性。

```
for letter in 'Runoob':  
    if letter == 'o':  
        pass  
        print ('执行 pass 块')  
        print ('当前字母:', letter)  
print ("Good bye!")
```

pass语句不做任何事情，  
用作占位。

# else语句

当 **while/for** 循环正常执行完的情况下，执行 **else** 输出；  
如果当 **while/for** 循环中执行了跳出循环的语句，比如 **break**，将不执行 **else** 代码块的内容；

```
count = 0
while count < 5:
    print("%d小于5"%count)
    count = count+1
else:
    print("%d等于5"%count)
print("循环结束")
```

```
count = 0
while count < 5:
    if count == 2:
        break
    print("%d小于5"%count)
    count = count+1
else:
    print("%d等于5"%count)
print("循环结束")
```

对比左右两段代码，运行结果有什么不同？