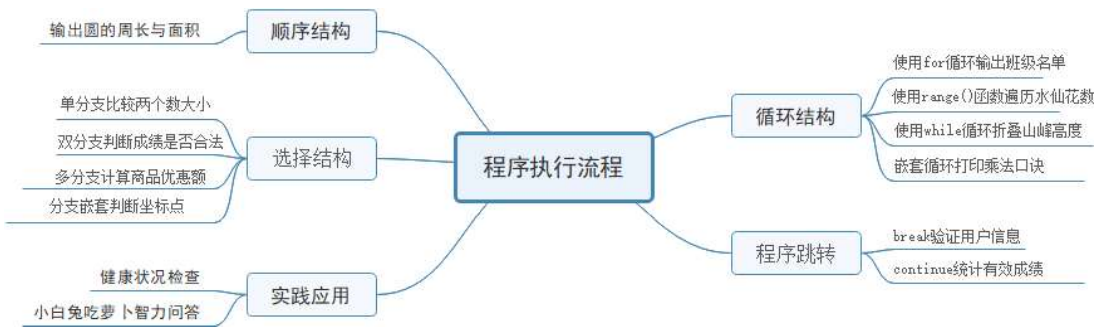


第 3 章 Python 程序流程控制

1. 知识图谱



2. 学习目标

- (1) 掌握单分支、双分支和多分支判断语句的使用。
- (2) 理解与掌握 for 循环语句。
- (3) 理解与掌握 range()函数。
- (4) 理解与掌握 while 循环语句。
- (5) 熟练掌握 break 和 continue 语句。

3.1 顺 序 结 构

程序中语句执行的基本顺序为各个语句出现的先后次序，称为顺序结构，其执行流程如图 3-1 所示。在该结构中先执行语句块 1，再执行语句块 2，最后执行语句块 3，三者是顺序关系。

【任务 3-1】编写程序，从控制台输入圆的半径，计算圆的周长和面积，并分别打印输出。

```
1. import math
2. radius=int(input("请输入的圆的半径:"))
3. area=math.pi*radius*radius
4. perimeter = 2*math.pi*radius
```

```
5. print ( "圆的周长为: %.2f" % perimeter)
6. print ( "圆的面积为: %.2f" % area)
```

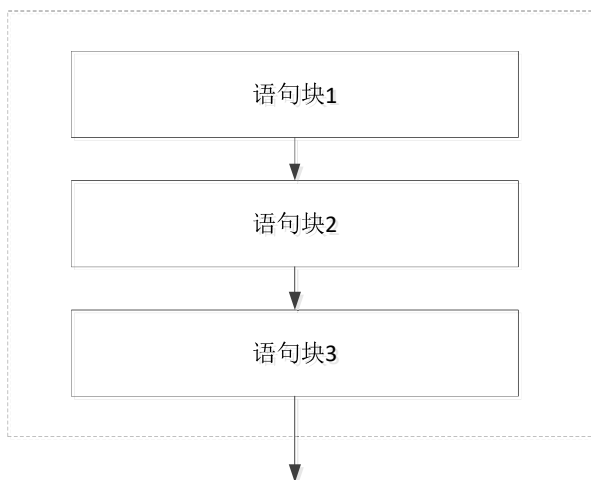


图 3-1 顺序结构的执行流程

代码说明：

第 1 行代码——计算圆的周长和面积需要使用 π 的值，Python 的 `math` 模块中包含常量 `pi`，通过导入 `math` 模块可以直接使用。

第 2 行代码——`input()`函数从控制台接收一个输入值，`int()`函数将输入的值转换为整型。

第 3 行代码——计算圆的面积。

第 4 行代码——计算圆的周长。

第 5~6 行代码——分别使用 `print()`函数输出圆的周长与面积，并保留两位小数。

运行程序，其输出结果如下：

```
请输入的圆的半径:5
圆的周长为:31.42
圆的面积为:78.54
```

3.2 选择结构

选择结构也称为分支结构，是程序根据条件判断结果而选择不同的执行路径的一种运行方式。Python 使用 `if` 语句来实现分支结构，常见的分支结构有单分支结构、双分支结构、多分支结构以及分支嵌套。

3.2.1 单分支比较两个数大小

if 单分支结构是最基本的分支结构，它允许基于某些条件执行相应的代码块。单分支结构语法格式如下：

```
if (条件表达式):  
    缩进代码块  
非缩进代码块
```

其中：

- (1) 判定条件可以是关系表达式、逻辑表达式或算术表达式。
- (2) 每个 if 条件后面要使用冒号，表示接下来是满足条件后要执行的缩进代码块。
- (3) 缩进代码块可以是单个语句，也可以是多个语句，多个语句的缩进必须对齐一致。

当条件表达式的值为真（True）时，执行 if 后面的缩进代码块，否则跳过缩进代码块。无论缩进代码块是否执行，都将继续执行与 if 语句相同缩进的非缩进代码块。单分支结构的程序执行流程如图 3-2 所示。

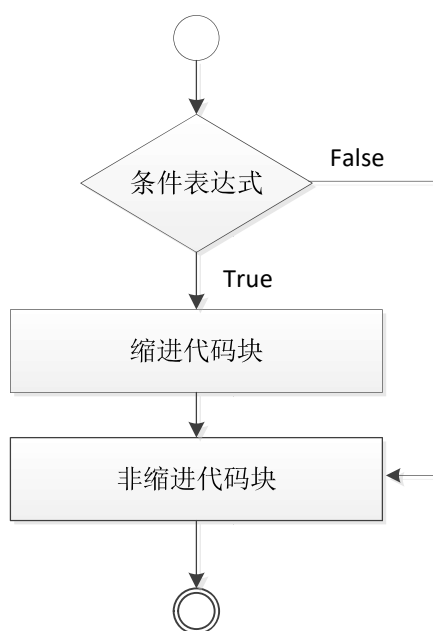


图 3-2 单分支结构执行流程

【任务 3-2】编写程序，从控制台输入两个整数 **a** 和 **b**，比较这两个数使得 **a>b**，并输出较大的 **a** 的值。

```
1. a=int(input("请输入 a 的值:"))
2. b=int(input("请输入 b 的值:"))
3. if(a<b):
4.     t = a
5.     a = b
6.     b = t
7. print("a 与 b 两个数, 较大的值为:",a)
```

代码说明：

第 1 行代码——使用 `input()` 函数从控制台接收一个输入，`int()` 函数将输入的值转换为整型，赋给变量 **a**。

第 2 行代码——使用 `input()` 函数从控制台接收一个输入，`int()` 函数将输入的值转换为整型，赋给变量 **b**。

第 3 行代码——比较 **a** 与 **b** 的大小，如果 **a** 小于 **b**，交换 **a** 与 **b** 的值。

第 7 行代码——输出较大的值 **a**。

运行程序，其输出结果如下：

```
请输入 a 的值:9
请输入 b 的值:6
a 与 b 两个数, 较大的值为:9
```

3.2.2 双分支判断成绩是否合法

使用单分支结构，只能在满足某些条件时做一些事情，那么当条件不满足，但依然需要完成某些任务时，可以使用双分支结构。双分支结构语法格式如下：

```
if (条件表达式):
    缩进代码块 1
else:
    缩进代码块 2
非缩进代码块
```

双分支结构的执行流程为：当条件表达式的值为真时（**True**），执行 `if` 后面的缩进代码块 1，否则执行 `else` 后面的缩进代码块 2。需要注意的是，双分支结构的两个代码块是互斥的，只能执行一个缩进代码块，无论哪种情况，程序都会执行其后的非缩进代码块。双分支结构的程序执行流程如图 3-3 所示。

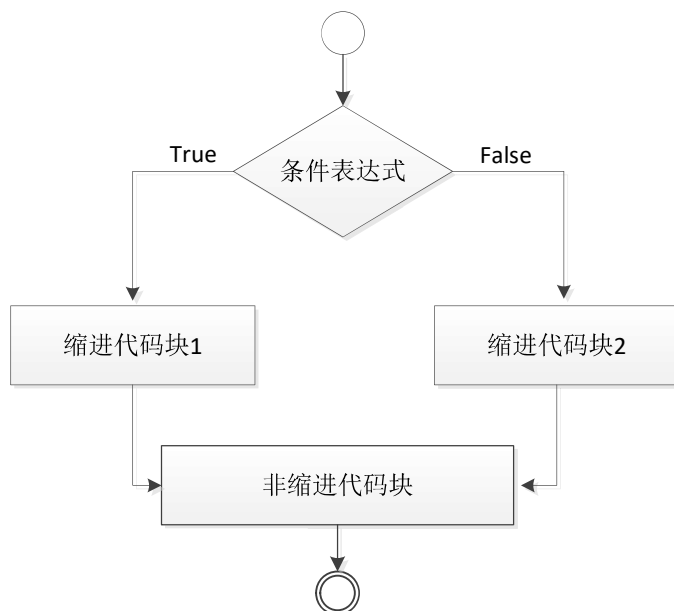


图 3-3 双分支结构执行流程

【任务 3-3】编写程序，输入学生成绩，判断成绩范围是否在 0~100，如果在此范围，输出“成绩有效！”，否则输出“成绩非法！”。

```
1. score = float(input("请输入学生成绩:"))
2. if(score>=0 and score<=100):
3.     print("成绩有效!")
4. else:
5.     print("成绩非法!")
```

代码说明：

第 1 行代码——获取从控制台输入的值，并将该值转变为 float 类型，赋给 score。

第 2~5 行代码——判断 score 值的范围是否在 0~100，如果在该范围，打印输出“成绩有效！”，否则输出“成绩非法！”

运行程序，输入成绩，其输出结果如下：

```
请输入学生成绩:88
成绩有效!
```

3.2.3 多分支计算商品优惠额

多分支结构通常用于判断同一个条件或一类条件的多个执行路径，Python 会按照多个

分支结构的代码顺序判断条件，寻找并执行第一个结果为 **True** 条件对应的语句。多分支结构语法格式如下：

```
if (条件表达式 1) :  
    缩进代码块 1  
elif (条件表达式 2) :  
    缩进代码块 2  
elif (条件表达式 3) :  
    缩进代码块 3  
...  
else:  
    缩进代码块  $n+1$   
非缩进代码块
```

多分支结构的执行流程为：如果条件表达式 1 的条件为 **True**，则执行缩进代码块 1；如果条件表达式 1 的条件为 **False**，但条件表达式 2 的值为 **True**，则执行缩进代码块 2；如果条件表达式 1 和 2 均为 **False**，但条件表达式 3 的值为 **True**，则执行缩进代码块 3，以此类推。如果所有的条件均不满足，则执行 **else** 中的缩进代码块。需要注意的是，无论哪种情况，程序都将执行随后的非缩进代码块。其具体执行流程如图 3-4 所示。

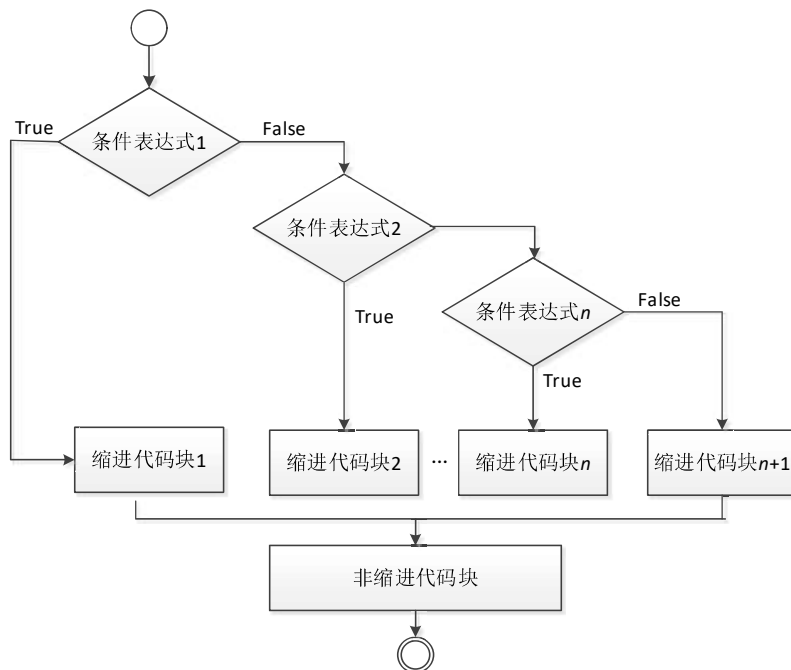


图 3-4 多分支结构执行流程

【任务 3-4】输入购买商品的总额，根据优惠规则，输出优惠后的应付金额。商品优惠规则如下：

- (1) 当购物的总额低于 1000 元时，打 9 折。
- (2) 当购物的总额在 1000~3000 元时（不含 3000）时，打 8 折。
- (3) 当购物的总额为 3000~8000 元时（不含 8000）时，打 7 折。
- (4) 当购物的总额为 8000~10000 元时（不含 10000）时，打 6 折。
- (5) 当购物的总额大于 10000 元时，打 5 折。

```
1. money = int(input("请输入您购买商品的总额:"))
2. if(money<1000):
3.     money= money*0.9
4. elif(money<3000):
5.     money = money*0.8
6. elif(money<8000):
7.     money=money*0.7
8. elif(money<10000):
9.     money=money*0.6
10. else:
11.     money = money*0.5
12. print( "优惠后的商品总额为: %.2f" % money)
```

代码说明：

第 1 行代码——获取从控制台输入的值，并将该值转换为 float 类型，赋给 money。

第 2~3 行代码——如果购买金额低于 1000 元，则应付金额为 money*0.9。

第 4~5 行代码——如果购买金额在 1000~3000 元（不含 3000），则应付金额为 money*0.8。

第 6~7 行代码——如果购买金额在 3000~8000 元（不含 8000），则应付金额为 money*0.7。

第 8~9 行代码——如果购买金额在 8000~10000 元（不含 10000），则应付金额为 money*0.6。

第 10~11 行代码——如果购买金额在 10000 元以上，则应付金额为 money*0.5。

第 12 行代码——输出优惠后的商品总额，并保留两位小数。

运行程序，输入商品总额，其优惠价格如下：

```
请输入您购买商品的总额:9800
优惠后的商品总额为: 5880.00
```

3.2.4 分支嵌套判断坐标点

在实际应用过程中，有时候分支执行场景比较复杂，比如我们乘坐飞机时，要首先买票办理登机手续，才能进入安检，只有安检通过，才能拿着登机牌登机。从上述场景可以看到，后面的条件是在前面条件成立的基础上才进行判断。针对这种情况，Python 提供了分支嵌套来实现，其语法格式如下：

```
if (条件表达式 1) :  
    if (满足条件表达式 2)  
        缩进代码块 1  
    else:  
        缩进代码块 2  
else:  
    if (条件表达式 3)  
        缩进代码块 3  
    else:  
        缩进代码块 4  
非缩进代码块
```

上述分支嵌套的执行流程为：如果条件表达式 1 的值为 True，则继续判断条件表达式 2 的值，如果也为 True，则执行缩进代码块 1；如果条件表达式 1 的值为 True，条件表达式 2 的值为 False，则执行缩进代码块 2；如果条件表达式 1 的值为 False，条件表达式 3 的值为 True，则执行缩进代码块 3；如果条件表达式 1 的值为 False，条件表达式 3 的值为 False，则执行缩进代码块 4。其具体执行流程如图 3-5 所示。

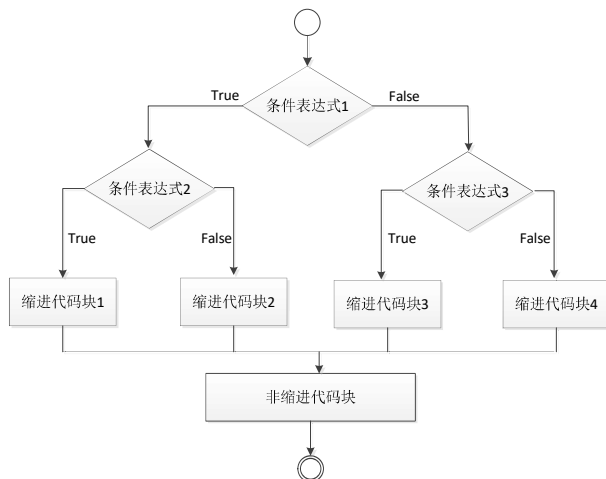


图 3-5 分支嵌套执行流程

【任务 3-5】编程实现从控制台输入 x 和 y 两个坐标点，判断该点是属于平面坐标系的第 1 象限、第 2 象限、第 3 象限还是第 4 象限。

```
1. x=int(input("请输入平面内的 x 坐标点:"))
2. y=int(input("请输入平面内的 y 坐标点:"))
3. if(x>0):
4.     if(y>0):
5.         print("(%d,%d) 在第 1 象限"%(x,y))
6.     else:
7.         print("(%d,%d) 在第 4 象限"%(x,y))
8. else:
9.     if(y>0):
10.        print("(%d,%d) 在第 2 象限"%(x,y))
11.    else:
12.        print("(%d,%d) 在第 3 象限"%(x,y))
```

代码说明：

第 1~2 行代码——分别从控制台输入两个数，并将它们转换为整型，赋给 x 和 y 。

第 3~5 行代码——判断 x 和 y 是否同时大于 0，如果满足该条件，则输出该坐标点在第 1 象限。

第 6~7 行代码——如果 x 大于 0，且 y 小于 0，则在第 4 象限。

第 8~10 行代码——如果 x 小于 0，且 y 大于 0，则在第 2 象限。

第 11~12 行代码——如果 x 小于 0，且 y 小于 0，则在第 3 象限。

运行程序，输入 x 和 y 坐标的值，输出如下：

```
请输入平面内的 x 坐标点:4
请输入平面内的 y 坐标点:-6
(4,-6) 在第 4 象限
```

3.3 循环结构

3.3.1 使用 for 循环输出班级名单

for 循环广泛应用于 Python 程序设计中，可以遍历任何序列，如列表、字符串和元组。在遍历过程中，列表或元组有几个元素，for 循环的循环体就执行几次，针对每个元素执行一次。for 循环的基本语法格式如下：

```
for 变量 in 序列:
    缩进代码块
非缩进代码块
```

在上述格式中，序列必须是一个可迭代的对象。执行 for 循环时，将序列中的每一项赋给变量，并针对每一个变量执行缩进代码块，执行完序列中的每一项后，退出循环，执行非缩进代码块。for 循环的程序执行流程如图 3-6 所示。

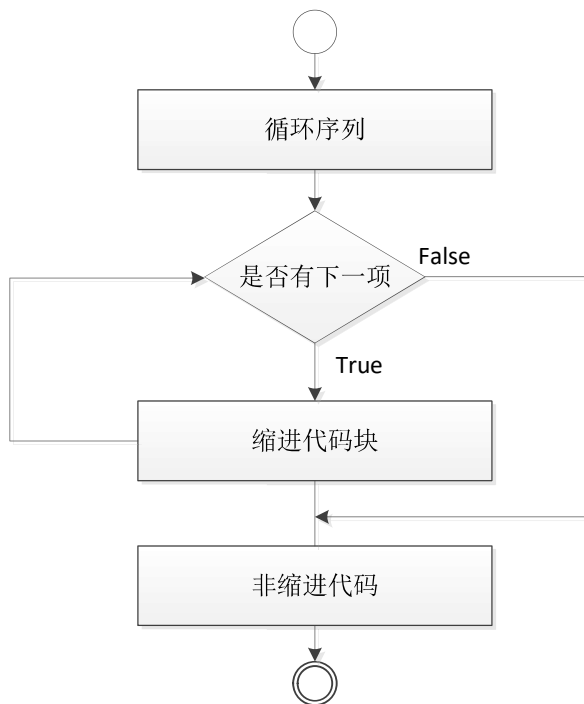


图 3-6 for 循环执行流程

【任务 3-6】 有一个班级姓名的元组集合，使用 for 循环遍历输出其中每一位同学的信息。

```
1. names = ("张三", "李四", "王五", "赵六", "田七")
2. for username in names:
3.     print(username, ', ', end='')
```

代码说明：

第 1 行代码——声明一个姓名元组，该元组中有 5 个元素。

第 2 行代码——使用 for...in 循环遍历其中的每一个元素。

第 3 行代码——使用 print() 函数输出遍历出的每一个元素。

运行程序，其输出结果如下：

```
张三 , 李四 , 王五 , 赵六 , 田七 ,
```

3.3.2 使用 range()函数遍历水仙花数

range()函数是 Python 内置的一个可迭代对象,可以产生指定范围的数字序列,配合 for 使用可以对该数字序列进行循环输出。其格式为:

```
range(start, stop[, step])
```

上述函数产生[start,stop)范围内的序列数字,其中计数从 start 开始,默认从 0 开始,例如 range(10)等价于 range(0,10);计数到 stop 结束,但不包括 stop,例如 range(3)表示计数[0,1,2],step 表示步长,默认步长为 1。

【任务 3-7】编程打印输出 100~999 的所有水仙花数。水仙花数是指一个 n 位数 ($n \geq 3$),其各位上的数字的 n 次幂之和等于它本身。例如 153 是一个水仙花数,因为 $153=1^3+5^3+3^3$ 。

```
1. for i in range(100,1000,1):
2.     baiwei = int(i/100)
3.     shiwei = int((i/ 10) % 10)
4.     gewei =int(i%10)
5.     result = baiwei*baiwei*baiwei+shiwei*shiwei*shiwei+gewei*gewei*
gewei
6.     if(result==i):
7.         print(i,"是一个水仙花数")
```

代码说明:

第 1 行代码——从 100 循环到 1000,步长为 1,但不包括 1000。

第 2 行代码——获得百位上的数字。

第 3 行代码——获得十位上的数字。

第 4 行代码——获得个位上的数字。

第 5 行代码——百位上数字的立方、十位上数字的立方及个位上数字的立方和。

第 6 行代码——如果求和结果等于循环的数字,则该数字是一个水仙花数,输出。

运行程序,其输出结果如下所示:

```
153 是一个水仙花数
370 是一个水仙花数
371 是一个水仙花数
407 是一个水仙花数
```

3.3.3 使用 while 循环折叠山峰高度

`while` 循环是“当”型循环控制语句，在每次执行循环体之前，都要先计算条件表达值的值，根据循环表达值的值，决定是否进入循环体。`while` 循环的语句格式如下：

```
while (条件表达式):  
    缩进循环体  
非缩进代码块
```

其中，条件表达式是循环条件，可以是任意类型的表达式，如关系表达式或逻辑表达式，缩进循环体由一条或多条语句组成。在循环开始时，首先判断 `while` 后面条件表达值的值，如果条件为 `True`，则执行缩进循环体；执行完循环体后，再次判断条件表达值的值，如此循环，直到条件表达式的值为 `False`，直接弹出循环体，进入非缩进代码块。`while` 循环的程序执行流程如图 3-7 所示。

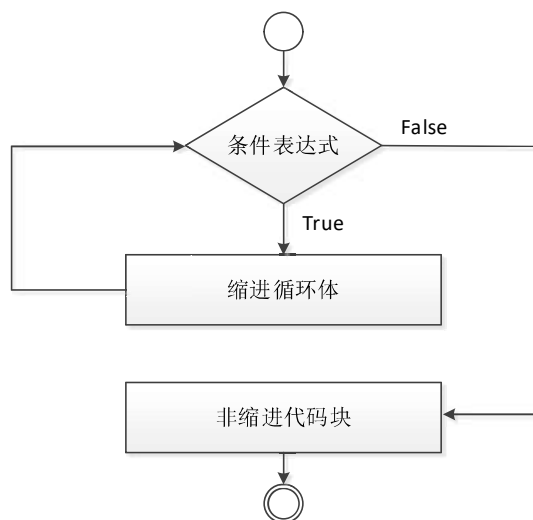


图 3-7 while 循环执行流程

【任务 3-8】世界最高山峰珠穆朗玛峰的高度为 8848m。现在有一张足够大的纸张，厚度为 0.001m，请问折叠多少次，可以保证厚度不低于珠穆朗玛峰的高度。

```
1. start = 0.001  
2. end = 8848  
3. count = 0  
4. while start < end:  
5.     count = count + 1
```

```
6.     start= start*2
7.  print("经过%d 次折叠，可以达到珠穆朗玛峰的高度! "%count)
```

代码说明:

第 1 行代码——声明开始时一张纸的厚度为 0.001。

第 2 行代码——声明最终纸张折叠后的高度。

第 3 行代码——声明经过 count 次折叠才能达到山峰高度。

第 4 行代码——循环条件，即折叠的厚度大于 8848 时循环终止。

第 5~6 行代码——每次执行循环体折叠次数加 1，而高度是上一次高度的 2 倍。

第 7 行代码——输出最终折叠的次数。

运行程序，其输出结果如下：

```
经过 24 次折叠，可以达到珠穆朗玛峰的高度！
```

3.3.4 嵌套循环打印乘法口诀

Python 语言允许在一个循环体内嵌入其他的循环体，如可以在 while 循环中嵌入 for 循环，也可以在 for 循环中嵌入 while 循环。

【任务 3-9】编程实现使用嵌套循环，输出乘法口诀表。

```
1.  row_num= 1
2.  while row_num <= 9:
3.      col_num= 1
4.      while col_num <= row_num:
5.          print("%d * %d = %d\t" % (col_num, row_num, col_num* row_num),
6.                end="")
7.          col_num += 1
8.          print()
9.      row_num += 1
```

代码说明:

第 1 行代码——row_num 声明行号，乘法口诀表有 9 行，因此 row_num 从 1 开始到 9 结束。

第 2 行代码——while 循环的条件是 row_num 小于等于 9。

第 3 行代码——声明列号，每次进入循环时，列号从 1 开始，循环到行号。

第 5 行代码——print() 函数默认每次输出都会换行，使用 end="" 使每次输出不换行。

第 6 行代码——输出一个元素之后，使列号加 1。

第 7 行代码——每次输出一行后换行。

第 8 行代码——行号加 1。

运行程序，其输出结果如下：

```
1*1=1
1*2=2    2*2=4
1*3=3    2*3=6    3*3=9
1*4=4    2*4=8    3*4=12    4*4=16
1*5=5    2*5=10   3*5=15    4*5=20    5*5=25
1*6=6    2*6=12   3*6=18    4*6=24    5*6=30    6*6=36
1*7=7    2*7=14   3*7=21    4*7=28    5*7=35    6*7=42    7*7=49
1*8=8    2*8=16   3*8=24    4*8=32    5*8=40    6*8=48    7*8=56    8*8=64
1*9=9    2*9=18   3*9=27    4*9=36    5*9=45    6*9=54    7*9=63    8*9=72    9*9=81
```

3.4 程 序 跳 转

3.4.1 break 验证用户信息

`break` 语句主要用于弹出当前循环体，如果有两层或两层以上的循环，在最内层使用 `break` 语句，则弹出的是最内层的循环，外层的循环不受任何影响。`break` 语句的执行流程如图 3-8 所示。

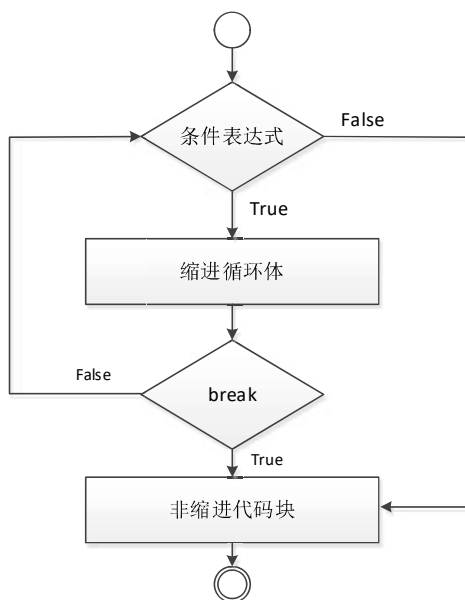


图 3-8 `break` 语句执行流程

【任务 3-10】编写程序,建立一个死循环,模拟用户登录操作,当输入的账号是 zhangsan 且密码为 123456 时,提示验证成功,否则继续验证;当连续输入 3 次错误时,弹出循环体。

```
1. count =1
2. while True:
3.     username = input("请输入您的用户名: ")
4.     password = input("请输入您的密码: ")
5.     if username=="zhangsan" and password=="123456":
6.         print("恭喜您, 成功登录!")
7.         break;
8.     if count==3:
9.         break
10.    count= count+1
```

代码说明:

第 1 行代码——count 主要记录循环的次数,初始值为 1。

第 2 行代码——建立一个 while 循环,其循环条件始终为 True。

第 3 行代码——从控制台获取一个字符串,赋给 username。

第 4 行代码——从控制台获取一个字符串,赋给 password。

第 6~7 行代码——判断用户名和密码是否是 zhangsan 和 123456,如果是输出“恭喜您,成功登录!”,并使用 break 弹出循环体。

第 8~9 行代码——如果 count 的值是 3,即循环了 3 次,则使用 break 弹出循环体。

第 10 行代码——每次循环完成后, count 的值加 1。

运行程序,输入正确的用户信息,其输出结果如下:

```
请输入您的用户名: zhangsan
请输入您的密码: 123456
恭喜您, 成功登录!
```

3.4.2 continue 统计有效成绩

continue 用于终止当前循环,忽略剩余的语句,然后回到循环的顶端,进入下一次循环。在开始下一次循环之前,如果是条件循环,先验证条件表达式;如果是迭代循环,则验证是否还有元素可以迭代。只有在成功的前提下,才开始新一轮循环。

【任务 3-11】输入一批考试分数,若输入大于 100,则为无效成绩,提示重新输入;若成绩小于 0,则结束成绩输入,弹出循环体,并计算输出所有成绩的最高分和平均分。

```
1. count = 0
2. avg_score=0
3. max_score=0
4. total_score=0
5. while True:
6.     score = float(input("请输入考生成绩:"))
7.     if score<0:
8.         break
9.     if score>100:
10.        print("输入成绩大于最高分数, 请重新输入!")
11.        continue
12.    if max_score<score:
13.        max_score=score
14.    count=count+1
15.    total_score = score+total_score
16. avg_score = total_score/count
17. print("一共输入%d 位同学有效成绩, 平均成绩为: %.2f, 最高成绩
    为: %.2f"% (count, avg_score, max_score))
```

代码说明:

第 1~3 行代码——分别声明学生数 `count`、平均成绩 `avg_score` 和最高成绩 `max_score`, 其初始值都为 0。

第 6 行代码——从控制台输入数据, 转换为 `float` 类型数据, 赋给 `score`。

第 7~8 行代码——如果输入的成绩小于 0, 则直接退出 `while` 循环。

第 9~11 行代码——如果输入的成绩大于 100, 提示输入成绩无效, 继续下一轮循环。

第 12~13 行代码——计算最高成绩。

第 14 行代码——统计有效成绩人数。

第 16 行代码——计算平均成绩。

第 17 行代码——输出成绩有效的人数、平均成绩和最高成绩。

运行程序, 其输出结果如下:

```
请输入考生成绩:30
请输入考生成绩:400
输入成绩大于最高分数, 请重新输入!
请输入考生成绩:-10
一共输入 1 位同学有效成绩, 平均成绩为:30.00, 最高成绩为:30.00
```

3.5 实践应用

3.5.1 健康状况检查

1. 项目介绍

输入一个人的身高(单位: cm)和体重(单位: kg), 根据公式“标准体重=(身高-150)*0.6+48”计算标准体重, 然后再根据公式“超重率=(实际体重-标准体重)/标准体重”计算超重率。若超重率<10%为正常体重, 10%≤超重率<20%属于体重超重, 20%≤超重率<30%属于轻度肥胖, 30%≤超重率<50%属于中度肥胖, 超重率≥50%属于重度肥胖。

2. 学习目标

- (1) 掌握 if 语句的格式和使用方法。
- (2) 掌握 if...elif 嵌套语句的格式和使用方法。
- (3) 培养利用分支嵌套解决实际问题的能力, 提高逻辑思维能力。

3. 项目解析

首先使用 input() 函数输入人的身高和体重, 然后根据提供的公式分别计算标准体重和超重率, 最后利用 if...elif 嵌套语句判断超重率, 并输出相应的结果。

4. 代码清单

本项目的代码清单如下:

```
1. height = float(input("请输入您的身高(cm):"))
2. weight = float(input("请输入您的体重(kg):"))
3. standard_weight = (height-150)*0.6+48
4. overweight_rate=(weight-standard_weight)/standard_weight
5. if overweight_rate<0.1:
6.     print("您的身高为%d, 体重为:%.2f, 属于体重正常"%(height,weight))
7. elif overweight_rate<0.2:
8.     print("您的身高为%d, 体重为:%.2f, 属于体重超重"%(height,weight))
9. elif overweight_rate<0.3:
10.    print("您的身高为%d, 体重为:%.2f, 属于轻度肥胖"%(height,weight))
11. elif overweight_rate<0.5:
12.    print("您的身高为%d, 体重为:%.2f, 属于中度肥胖"%(height,weight))
13. else:
14.    print("您的身高为%d, 体重为:%.2f, 你严重肥胖"%(height,weight))
```

代码说明：

第 1~2 行代码——分别从控制台输入两个数，转换成 float 类型后赋给 height 与 weight。

第 3 行代码——根据身高按照公式计算标准体重。

第 4 行代码——根据体重计算超重率。

第 5~14 行代码——根据超重率进行条件判断，输出相应的字符串。

运行程序，其输出结果如下：

```
请输入您的身高 (cm) :170
请输入您的体重 (kg) :75
您的身高为 170, 体重为:75.00, 属于轻度肥胖
```

3.5.2 小白兔吃萝卜智力问答

1. 项目介绍

小白兔特别喜欢吃萝卜，从第 1 天开始它每天吃掉原有萝卜的一半，再多吃 1 个，吃到第 10 天只剩下 1 个萝卜。编程输出原来一共有多少个萝卜。

2. 学习目标

- (1) 掌握 for 循环语句的使用。
- (2) 掌握 range() 序列函数的使用。
- (3) 掌握利用 for 和 range 进行循环结构设计的能力。

3. 项目解析

假设某一天有 m 个萝卜，下一天有 n 个萝卜，可以推算出 $n=m/2-1$ ，因此可以计算出 $m=2\times(n+1)$ ，也就是说已知某一天有 n 个萝卜，则前一天萝卜的数量为 $2\times(n+1)$ 。根据题意，第 10 天有 1 个萝卜，就 9 天应该有 $2\times(1+1)=4$ 个萝卜，依此类推可以计算出第 1 天的萝卜数量。

4. 代码清单

本项目的代码清单如下：

```
1. result = 1
2. for i in range(9,0,-1):
3.     result = (result+1)*2
4.     print("第%d天的萝卜数量为:%d"%(i,result))
```

代码说明：

第1行代码——声明 `result=1`，表示第10天有1个萝卜。

第2行代码——使用 `for...in range` 从序列9循环到1。

第3~4行代码——计算并输出每天的萝卜数目。

运行程序，其输出结果如下：

```
第9天的萝卜数量为:4
第8天的萝卜数量为:10
第7天的萝卜数量为:22
第6天的萝卜数量为:46
第5天的萝卜数量为:94
第4天的萝卜数量为:190
第3天的萝卜数量为:382
第2天的萝卜数量为:766
第1天的萝卜数量为:1534
```

3.6 本章小结

本章主要介绍了 Python 程序的控制流程，包含顺序结构、分支结构和循环结构。其中分支结构分为单分支结构、双分支结构、多分支结构以及分支嵌套结构；循环结构用来多次重复执行某个代码块，主要有 `for` 循环和 `while` 循环；在程序执行的过程中，根据需要还可以使用 `break` 弹出当前循环体，以及使用 `continue` 结束当前循环，并继续进行下一轮循环。

本章习题

一、选择题

1. 在 `if` 语句中进行判断，产生（ ）时会输出相应的结果。
A. 0 B. 1 C. 布尔值 D. 以上均不正确
2. 在 Python 中实现多个条件判断需要用到（ ）语句与 `if` 语句的组合。
A. `else` B. `elif` C. `pass` D. 以上均不正确
3. 循环中可以用（ ）语句来弹出深度循环。
A. `pass` B. `continue` C. `break` D. 以上均可以
4. 使用（ ）语句弹出当前循环的剩余语句，可以继续进行下一轮循环。
A. `pass` B. `continue` C. `break` D. 以上均可以

5. 在 `for i in range(5)` 语句中, `i` 的取值是 ()。
- A. [1,2,3,4,5] B. [1,2,3,4] C. [0,1,2,3,4] D. [0,1,2,3,4]
6. 以下 `for` 循环中, () 不能实现 1~10 的累加功能。
- A. `for i in range(10,0): sum+=i`
B. `for i in range(1,11): sum+=i`
C. `for i in range(10,0, -1): sum+=i`
D. `for i in (10,9,8,7,6,5,4,3,2,1): sum+=i`
7. 下面的 `if` 语句判断满足“性别(`gender`)为男, 职称(`employ`)为讲师, 年龄(`age`)小于 35 周岁”条件的为 ()。
- A. `if(gender=="男" or age<35 and employ=="教师"):n+=1`
B. `if(gender=="男" or age<35 and employ=="教师"):n+=1`
C. `if(gender=="男" or age<35 and employ=="教师"):n+=1`
D. `if(gender=="男" and age<35 and employ=="教师"):n+=1`
8. 阅读下面的程序代码:

```
sum=0
for i in range(100):
    if(i%10):continue
    sum=sum+1
print(sum)
```

上述程序的输出结果为 ()。

- A. 5050 B. 4950 C. 450 D. 45
9. 已知 `x=10`, `y=20`, `z=30`, 则执行以下程序后, `x`, `y`, `z` 的值分别为 ()。

```
if x<y:
    z=x;
    x=y;
    y=z;
```

- A. 10, 20, 30 B. 10, 20, 20 C. 20, 10, 10 D. 20, 10, 10
10. 有一个列表 `L=[4,6,8,10,12,5,7,9]`, 列表解析式 `[x for x in L if x%2==0]` 返回的结果是 ()。
- A. [4,8,12,7] B. [6,10,5,9] C. [4,6,8,10,12] D. [5,7,9]

二、填空题

1. 在 Python 中使用 `while True:` 循环的过程中, 使用_____可以退出循环。

2. 循环语句使用 `for i in range(1,10,2)`，则循环了_____次。
3. 执行下列 Python 语句后的输出结果是_____，循环执行了_____次。

```
i=-1
while(i<0):
    i*=i
print(i)
```

4. 在循环过程中，弹出循环使用_____语句。
5. 在循环过程中，终止本次循环，开始下一轮循环使用_____语句。

三、判断题

1. if 分支结构有单分支、双分支、多分支以及分支嵌套。()
2. 在循环过程中，`break` 用于结束本次循环，开始下一轮循环。()
3. `for` 循环和 `while` 循环可以实现相同的功能。()
4. 当循环过程中，遇到 `continue` 语句，则下面的代码在本次循环中不执行。()
5. `range()` 函数的作用是产生一个指定的序列。()

四、程序分析题

1. 分析以下程序，输出结果为_____。

```
x=False
y=True
z=False
if(x or y and z):
    print("True")
else:
    print("false")
```

2. 分析以下程序，输出结果为_____。

```
for I in range(2,5):
    print(i,end='')
```

3. 分析以下程序，输出结果为_____。

```
n=int(input("请输入图形的行数:"))
for I in range(0,10-i):
    print(" ",end=' ')
    for j in range(0,2*i+1)
        print("*",end=' ')
    print()
```

五、简答题

1. 简述 Python 中 `continue` 与 `break` 的区别。
2. 简述 `range()` 函数的使用规则。
3. 简述 `while` 循环和 `for` 循环语句的区别。

六、程序设计题

1. 编写一个程序，判断一个输入数是正数还是负数。
2. 编写一个程序，计算 $1+2+3+\cdots+100$ 的和。
3. 编写一个程序，输入两个数，计算这两个数的最大公约数和最小公倍数。
4. 编写一个程序，计算 $S = 1 + 1/2 + 1/3 + \cdots + 1/100$ 的和。
5. 编写程序，从键盘输入三角形的三条边 `a`、`b`、`c`，求三角形的周长和面积。提示：三角形的面积 $= \sqrt{p(p-a)(p-b)(p-c)}$ ， p 的值为三角形周长的 $1/2$ 。
6. 输入一个年份，判断该年份是否是闰年。判断标准：该年份除以 4 余数为 0 或者除以 100 余数为 0 且除以 400 余数为 0。
7. 编写程序，输入 x 的值，输出 y 的值。要求：当 $x \geq 0$ 时， $y = 3x - 1$ ；当 $x < 0$ 时， $y = x^2 + 5$ 。
8. 输入某门课的成绩，将其转换为五级制（优秀、良好、中等、及格和不及格）输出。评定条件如下：

$$\text{成绩等级} = \begin{cases} \text{优} & \text{score} \geq 90 \\ \text{良} & 80 \leq \text{score} < 90 \\ \text{中} & 70 \leq \text{score} < 80 \\ \text{及格} & 60 \leq \text{score} < 70 \\ \text{不及格} & \text{score} < 60 \end{cases}$$

9. 有一个球从 100m 高度自由落下，每次落地后反弹回原来高度的一半，再落下。求第 10 次落地时，共经过多少米以及第 10 次反弹的高度。
10. 小明的妈妈每天给他 2.5 元钱，他会存起来，但是每当这一天是存钱的第 5 天或者是 5 的倍数时，他都会花去 6 元钱。请问经过多少天，小明才可以存到 100 元钱。
11. 小明单位发了 100 元的购物卡，他到超市买 3 类洗化用品：洗发水（15 元）、香皂（2 元）和牙刷（5 元）。要把 100 元正好花掉，可有哪些购买组合。
12. 编写程序，要求不断接收控制台输入的字符串，并输出字符串的长度，如果接收的字符串是 `quit`，则弹出循环体。