081 纯Python代码玩转小迷宫

* 核心技术

在实现玩转小迷宫这个游戏时，分别使用了input()输入函数、print()输出函数、if…elif…else语句、二维列表、while循环、for循环，其中技术要点的内容如下：

1.input()输入函数

在 Python 中，使用内置函数 input() 可以接收用户的键盘输入。input() 函数的基本用法如下：

variable = input("提示文字")

其中，variable 为保存输入结果的变量，双引号内的文字用于提示要输入的内容。例如，想要接收 用户输入的内容，并保存到变量tip中，可以使用下面的代码：

1. tip = input("请输入文字：")

在ython3.x中，无论输入的是数字还是字符都将被作为字符串读取。如果想要接收数值，需要把接收到的字符串进行类型转换。例如，想要接收整型的数字并保存到变量age中，可以使用下面的代码：

1. age = int(input("请输入数字："))

说明：在Python2.x中，input()函数接收内容时，数值直接输入即可，并且接收后的内容作为数字类型；而如果要输入字符串类型的内容，需要将对应的字符串使用引号括起来，否则会报错。

实现根据身高、体重计算BMI指数的示例代码如下：

1. height = float(input("请输入您的身高（单位为米）：")) # 输入身高，单位：米
2. weight = float(input("请输入您的体重（单位为千克)：")) # 输入体重，单位：千克
3. bmi=weight/(height\*height) # 用于计算BMI指数，公式为“体重/身高的平方”
4. print("您的BMI指数为："+str(bmi)) # 输出BMI指数
5. # 判断身材是否合理
6. **if** bmi<18.5:
7. print("您的体重过轻 ~@\_@~")
8. **if** bmi>=18.5 **and** bmi<24.9:
9. print("正常范围，注意保持 (-\_-)")
10. **if** bmi>=24.9 **and** bmi<29.9:
11. print("您的体重过重 ~@\_@~")
12. **if** bmi>=29.9:
13. print("肥胖 ^@\_@^")

2.print()输出函数

默认的情况下，在Python中，使用内置的print()函数可以将结果输出到IDLE或者标准控制台上。其基本语法格式如下：

print(输出内容)

中，输出内容可以是数字和字符串（字符串需要使用引号括起来），此类内容将直接输出，也可以是包含运算符的表达式，此类内容将计算结果输出。例如：

1. a = 10 # 变量a，值为10
2. b = 6 # 变量b，值为6
3. print(6) # 输出数字6
4. print(a\*b) # 输出变量a\*b的结果60
5. print(a **if** a>b **else** b) # 输出条件表达式的结果10
6. print("成功的唯一秘诀——坚持最后一分钟") # 输出字符串“成功的唯一秘诀——坚持最后一分钟”

3.if…elif…else语句

在开发程序时，如果遇到多选一的情况，则可以使用if…elif…else语句，该语句是一个多分支选择语句，通常表现为“如果满足某种条件，就会进行某种处理，否则，如果满足另一种条件，则执行另一种处理……”。if…elif…else语句的语法格式如下：

if 表达式1:

语句块1

elif 表达式2:

语句块2

elif 表达式3:

语句块3

…

else:

语句块n

使用if…elif…else语句时，表达式可以是一个单纯的布尔值或变量，也可以是比较表达式或逻辑表达式，如果表达式为真，执行语句；而如果表达式为假，则跳过该语句，进行下一个elif的判断，只有在所有表达式都为假的情况下，才会执行else中的语句。if…elif…else语句的流程如下图1所示。

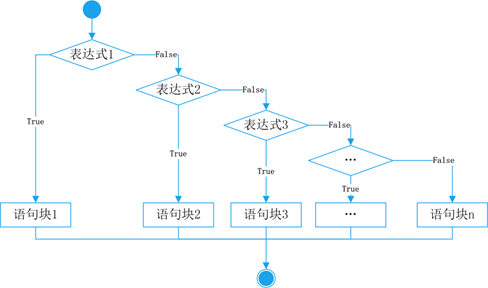


图1 if…elif…else语句的流程图

注意：if和elif都需要判断表达式的真假，而else则不需要判断；另外，elif和else都必须跟if一起使用，不能单独使用。

使用if…elif…else多分支语句实现根据用户输入的年龄输出相应的提示信息的功能，示例代码如下：

1. your\_age = int(input("请输入您的年龄：")) # 获取用户输入的年龄，并转换为整型
2. **if** your\_age <= 18: # 调用if语句判断输入的数据是否小于等于18
3. # 如果小于等于18则输出提示信息
4. print("您的年龄还小，要努力学习哦！")
5. **elif** 18 < your\_age <= 30: # 判断是否大于18岁，并且小于30岁
6. # 如果输入的年龄大于18岁并且小于30岁则输出提示信息
7. print("您现在的阶段正是努力奋斗的黄金阶段！")
8. **elif** 30 < your\_age <= 50: # 判断输入的年龄是否大于30岁小于等于50岁
9. # 如果输入的年龄大于30岁而小于等于50岁则输出提示信息
10. print("您现在的阶段正是人生的黄金阶段！")
11. **else**:
12. print("最美不过夕阳红！")

说明：第1行代码中的int()函数用于将用户的输入强制转换成整型。

4.二维列表

在Python中，由于列表元素还可以是列表，所以它也支持二维列表的概念。那么什么是二维列表？前文提到酒店有很多房间，这些房间都可以构成一个列表，如果这个酒店有500个房间，那么拿到499号房钥匙的旅客可能就不高兴了，从1号房走到499号房要花好长时间，因此酒店设置了很多楼层，每一个楼层都会有很多房间，形成一个立体的结构，把大量的房间均摊到每个楼层，这种结构就是二维列表结构。使用二维列表结构表示酒店每个楼层的房间号的效果如图2所示。

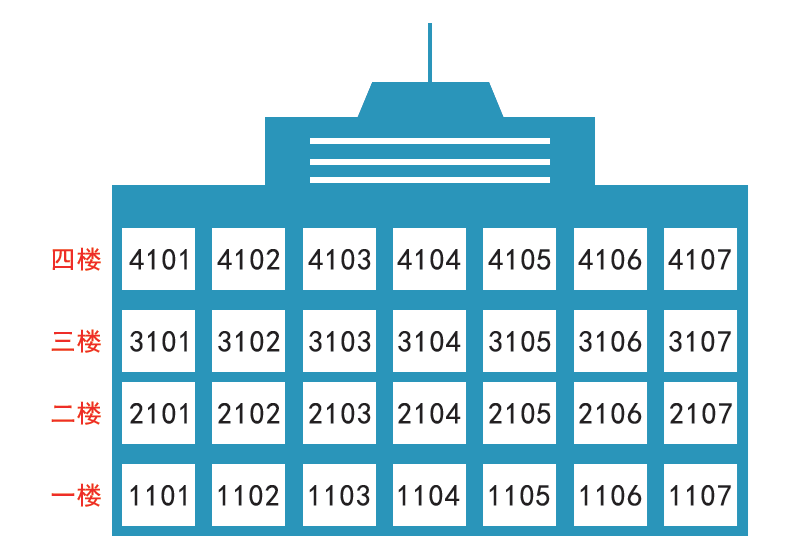


图2 二维列表结构的楼层房间号

在Python中，二维列表是包含列表的列表，即一个列表的每一个元素又都是一个列表。创建二维列表可以参考以下代码：

1. listname = [['千', '山', '鸟', '飞', '绝'],
2. ['万', '径', '人', '踪', '灭'],
3. ['孤', '舟', '蓑', '笠', '翁'],
4. ['独', '钓', '寒', '江', '雪']]

创建二维数组后，可以通过以下语法格式访问列表中的元素：

listname[下标1][下标2]

参数说明如下：

* listname ：列表名称。
* 下标 1 ：表示列表中第几行，下标值从 0 开始，即第一行的下标为 0。
* 下标 2 ：表示列表中第几列，下标值从 0 开始，即第一列的下标为 0。 例如，要访问二维列表中的第 2 行，第 4 列，可以使用下面的代码：

1. print(listname[1][3])

5.while循环

while循环是通过一个条件来控制是否要继续反复执行循环体中的语句。语法如下：

while 条件表达式:

循环体

说明：循环体是指一组被重复执行的语句。

当条件表达式的返回值为真时，则执行循环体中的语句，执行完毕后，重新判断条件表达式的返 回值，直到表达式返回的结果为假时，退出循环。while 循环语句的执行流程如图3所示。

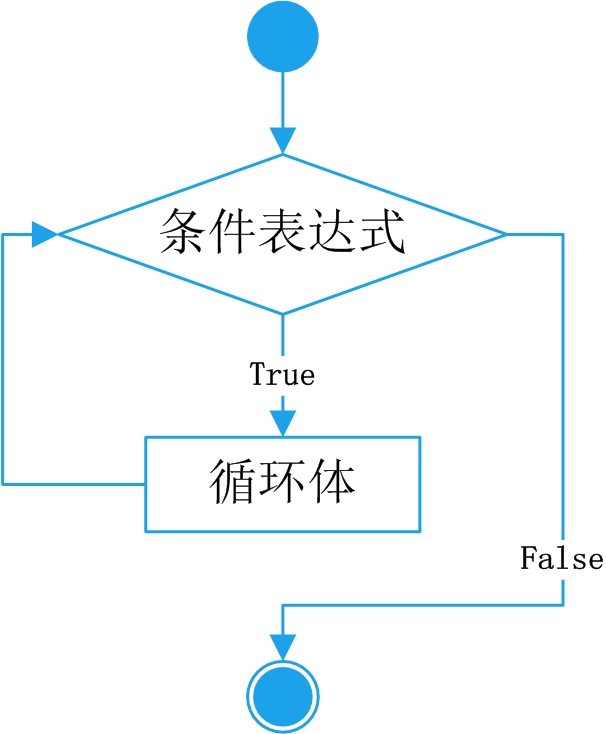


图3 while循环语句的执行流程图

使用while循环语句实现从1开始依次尝试符合条件的数，直到找到符合条件的数时，才退出循环。具体的实现方法是：首先定义一个用于计数的变量number和一个作为循环条件的变量none（默认值为真），然后编写while循环语句，在循环体中，将变量number的值加1，并且判断number的值是否符合条件，当符合条件时，将变量none设置为假，从而退出循环。具体代码如下：

1. print("今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问几何？**\n**")
2. none = **True** # 作为循环条件的变量
3. number = 0 # 计数的变量
4. **while** none:
5. number += 1 # 计数加1
6. **if** number%3 ==2 **and** number%5 ==3 **and** number%7 ==2: # 判断是否符合条件
7. print("答曰：这个数是",number) # 输出符合条件的数
8. none = **False** # 将循环条件的变量赋值为否

6.for循环

for循环是一个依次重复执行的循环。通常适用于枚举或遍历序列，以及迭代对象中的元素。语法如下：

for 迭代变量 in 对象:

循环体

其中，迭代变量用于保存读取出的值；对象为要遍历或迭代的对象，该对象可以是任何有序的序列对象，如字符串、列表和元组等；循环体为一组被重复执行的语句。

for循环语句的执行流程如图4所示。

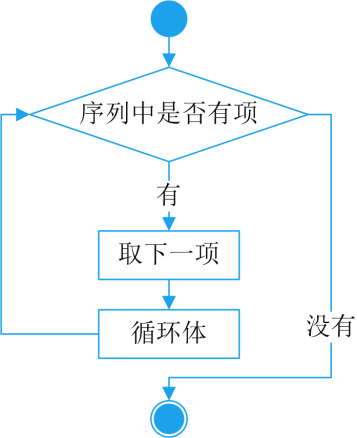


图4 for循环语句的执行流程图

在使用 for 循环时，最基本的应用就是进行数值循环。例如，想要实现从 1 到 100 的累加，可以 通过下面的代码实现：

1. print("计算1+2+3+……+100的结果为：")
2. result = 0 # 保存累加结果的变量
3. **for** i **in** range(101):
4. result += i # 实现累加功能
5. print(result) # 在循环结束时输出结果

上面的代码中，使用了range()函数，该函数是Python内置的函数，用于生成一系列连续的整数，多用于for循环语句中。其语法格式如下：

range(start,end,step)

参数说明如下：

* start ：用于指定计数的起始值，可以省略，如果省略则从 0 开始。
* end ：用于指定计数的结束值（但不包括该值，如 range(7)，则得到的值为 0~6，不包括 7）， 不能省略。当 range() 函数中只有一个参数时，即表示指定计数的结束值。
* step ：用于指定步长，即两个数之间的间隔，可以省略，如果省略则表示步长为 1。例如， range(1,7) 将得到 1、2、3、4、5、6。

注意：在使用range()函数时，如果只有一个参数，那么表示指定的是end；如果有两个参数，则表3

示指定的是start和end；如果3个参数都存在时，最后一个参数才表示步长。