

文章目录

- [1 条件语句 \(if、elif、else\)](#)
- [2 循环语句 \(while、while...else\)](#)
- [3 循环语句 \(for、for...else\)](#)
- [4 break、continue语句](#)
- [5 pass 语句](#)

1 条件语句 (if、elif、else)

格式：

```
if 判断条件1:
    执行语句1...
elif 判断条件2:
    执行语句2...
elif 判断条件3:
    执行语句3...
else:
    执行语句4...
```

成绩等级判断（直接上代码，一看就懂）：

```
number = int(input("请输入你的成绩："))
if number >= 90:
    print("优秀")
elif number >= 80:
    print("良好")
elif number >= 70:
    print("不错")
elif number >= 60:
    print("及格")
else:
    print("不及格")
```

if 嵌套（把 if...elif...else 结构放在另外一个 if...elif...else 结构中）：

```
num=int(input("输入一个数字: "))
if num > 0:
    if num >= 18:
        print ("已成年")
    else:
        print ("未成年")
else:
    print("刚出生")
```

2 循环语句（while、while...else）

`while` 循环最常用的应用场景就是 **让执行的代码** 按照 **指定的次数 重复** 执行

格式：

```
while 判断条件(condition):
    执行语句...
```

1-100求和：

```
# 1.编写循环 确定 要计算的数字
# 2. 添加 结果 变量，在循环内部 处理计算结果
i = 0
while i <= 100:
    print(i)
    i += 1

# 最终代码如下
i = 0
sum = 0
while i <= 100:
    sum += i
    i += 1
print(sum)
```

死循环（表达式永远为true）：

```

while True:
    num = int(input("输入一个数字  :"))
    print ("你输入的数字是： ", num)

'''
输入一个数字  :10
你输入的数字是： 10
输入一个数字  :50
你输入的数字是： 50
输入一个数字  :
'''

```

while...else 语句：

```

# 如果 while 后面的条件语句为 false 时，则执行 else 的语句块
age = 0
while age < 3:
    print (age,"年龄小于3")
    age = age + 1
else:
    print (age,"年龄大于等于3")
'''
0 年龄小于3
1 年龄小于3
2 年龄小于3
3 年龄大于等于3
'''

```

while嵌套（九九乘法表）：

```

# 定义起始行
row = 1
# 最大打印 9 行
while row <= 9:
    # 定义起始列
    col = 1
    # 最大打印 row 列
    while col <= row:
        # end = "", 表示输出结束后，不换行
        # "\t" 可以在控制台输出一个制表符，协助在输出文本时对齐
        print("%d * %d = %d" % (col, row, row * col), end="\t")
        # 列数 + 1
        col += 1
    # 这行代码的目的是一行输出完后之后，添加换行！
    print("")
    # 行数 + 1
    row += 1
'''

```

```

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 循环语句（for、for...else）

for 循环可以遍历任何可迭代对象，如一个列表或者一个字符串

格式如下：

```

# <variable>: 存储元素的变量 ; <sequence>: 遍历的对象
for <variable> in <sequence>:
    执行语句

-----
for <variable> in <sequence>:
    执行语句1
else:
    执行语句2

```

案例：

```

# 案例1:
lists = ['小白',20,'四川']
for i in lists:
    print(i,end=" ")
# 小白 20 四川

# 案例2:
str = 'abcd'
for i in str:
    print(i,end=" ")
# a b c d

# 案例3:
lists = ['小白',20,'四川']
for i in lists:
    print(i)

```

```
else:
    print("遍历结束! ")
'''
小白
20
四川
遍历结束!
'''
```

range()函数:

```
# 案例1:
for i in range(5):
    print(i,end=" ")
# 0 1 2 3 4

# 案例2 (增加步长, 如数字3):
for i in range(0, 10, 3):
    print(i,end=" ")
# 0 3 6 9
```

4 break、continue语句

break: 可以跳出 for 和 while 的循环体。如果你从 for 或 while 循环中终止, 任何对应的循环 else 块将不执行。

continue: 跳过当前循环块中的剩余语句, 然后继续进行下一轮循环。

for循环案例:

```
# for循环中使用break
for i in range(1,10):
    if i%2 == 0:
        break
    print(i,end=" ")
# 输出结果: 1, 因为2对2取余等于0就结束循环了

# for循环中使用continue
for i in range(1,10):
    if i%2 == 0:
        continue
    print(i,end=" ")
# 输出结果: 1 3 5 7 9 , 跳过了偶数
```

while循环案例:

```
# while循环中使用break
n = 5
```

```
while n > 0:
    n -= 1
    if n == 2:
        break
    print(n,end=" ")
# 输出结果: 4 3, 当n=2时结束循环不输出

# while循环中使用continue
n = 5
while n > 0:
    n -= 1
    if n == 2:
        continue
    print(n,end=" ")
# 输出结果: 4 3 1 0 , 当n=2时就跳过继续下一次循环
```

5 pass 语句

pass 不做任何事情，一般用做占位语句，是为了保持程序结构的完整性。

```
while True:
    pass #没写语句时，加上pass就不会报错
```