1. **If语句**
2. 每个条件后面要使用冒号
3. 在同等级的判断语句中，一旦有一个符合条件那么其余的便不会执行
4. 使用缩进来划分语句块，相同缩进数的语句在一起组成一个语句块
5. if-elif-else的可进行层层嵌套

形式如：

if 表达式1:

语句

if 表达式2:

语句

elif 表达式3:

语句

else:

语句

elif 表达式4:

语句

else:

语句

例题：输入一个数字，判断能不能被2或3整除，并将结果打印

num=int(input("请输入数字："))

if num%2==0:

if num%3==0:

print("可以被2和3整除")

else:

print("可以被2整除,不能被3整除")

else:

if num%3==0:

print("能被3整除，不能被2整除")

else:

print("不能被2和3整除")

1. **while循环**

格式形如：

While 判断条件：

语句

while还可以配合else使用。

end可在print中常用作不换行间隔符，形如：

print(11,end=",")

print(22)

输出结果：11,22

但end也支持转义字符，即如果end="\n"，那么

print(11,end="\n")

print(22)

输出结果：11

22

例题：使用while实现1到100的累加

num=0

n=1

while n<=100:

num += n

n+=1

print(num,end=",")

print(n)

1. **for循环**

　for循环的一般格式如下：

　　for var in iterable:

　　　　语句块

每次循环, var 迭代变量被设置为可迭代对象iterable的当前元素, 提供给语句块使用.

for循环可以遍历任何序列的项目，如一个列表或者一个字符串,也就是说必须符合这一要求才能被便利，那么如果想遍历数字，如1到10，那么手动创建一个这样的数列是非常麻烦的，此时我们可以使用range（）函数。

range()函数会自动生成数列。

range()语法及格式：

range(start,end,step=1):

1. range(10):默认step＝1，start＝0,生成可迭代对象，

包含[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

1. range(1,10):指定start＝1，end＝10，默认step＝1，生成可迭代对象，

包含[1, 2, 3, 4, 5, 6, 7, 8, 9]

1. range(1,10,2):指定start＝1，end＝10，step＝2，生成可迭代对象，

包含[1, 3, 5, 7, 9]

**注：数列中不包含end**

例题：使用for和range实现1到100的累加

n=0

for i in range(1,101):

n+=i

print(n)

1. **break、continue和else**
2. break 语句可以跳出 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。

1. continue语句被用来跳过当前循环块中的剩余语句，然后继续进行下一轮循环。
2. 循环语句可以有 else 子句，它在穷尽列表(for循环)或条件变为 false (while循环)导致循环终止时被执行,但循环被break终止时不执行。

例题：将1到100的所有偶数相加

n=0

for i in range(1,101):

if i%2==0:

continue

n+=i

else:

print(n)

例题：将1到100的所有偶数逐数相加，当累加的结果大于1000时结束

n=0

for i in range(1,101):

if n>1000:

break

n+=i

else:

print(n)

**5. 冒泡排序**

str=[10,48,15,-45,9]

i=0

j=0

lenth=len(str) # length值为5

while i<lenth:

j=i+1

while j<lenth:

if str[i]<str[j]:

#替换语句

a=str[i]

str[i]=str[j]

str[j]=a

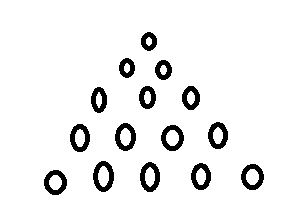
j=j+1# j+1指向下一个索引，同时开始下一while循环判断是否越界

i=i+1 # i+1指向下一个索引，同时开始下一while循环判断是否越界

else:

print(str)

1. **打印图形**
2. 分析行
3. 分析有没有空格
4. 分析打印的内容

****

line=5

i=j=n=0

for i in range(0,line): #控制行数

for j in range(0,line-i-1): #控制空格数量

print(" ",end="")

for n in range(0,i+1): #控制空格数量

print(0,end=" ")

print("\n")