Python中语句：

判断语句使用if

语法格式：

if 条件判断1: #必须以:结尾，否则会报错

执行命令1 #必须以制表位分隔，不能顶行写

elif 条件判断2:

执行命令2

elif 条件判断3:

执行命令3

else:

执行的其他命令

Python中条件等于需要写“==” "="代表变量赋值，"=="代表条件判断

if语句中同一个判断可以判断多个条件，条件之间可以使用and、or、not进行连接

and 逻辑与，前后都成立

or 逻辑或，前后成立任何一个

not 逻辑否，不需要成立

vim 03-if判断.py

#encoding=utf-8

sex=input("请输入您的性别:")

age=input("请输入您的年龄:")#默认input读取的都是字符类型(str)，不能和数字(int)类型做比较

if int(age) >=18 and sex=="男":#将年龄变为字符类型，否则无法和后面数字比较大小

print("成年男性，可以工作")

if多分支

vim 03-if判断.py

#encoding=utf-8

sex=input("请输入您的性别:")

age=int(input("请输入您的年龄:"))#默认input读取的都是字符类型(str)，不能和数字(int)类型做比较,通过int可以直接将输入的内容>转为int类型，下面使用的时候不需要转换

if age >=18 and sex=="男":

print("成年男性，可以工作")

elif age < 18 :

print("未成年")

elif sex=="女":

print("女性")

not代表逻辑否

not 条件x 代表x不成立的时候才返回True

not 条件x and 条件y x条件不成立，y条件成立才返回True

not 条件x or 条件y x条件不成立或y条件成立都返回True

not (条件x and 条件y)

判断变量是否存在

vim 04-判断是否存在.py

#age=int(input("请输入您的年龄:"))

age=""

if age:

print("age非0")

else:

print("age为0")

计算BMI指数：

vim 05-bmi.py

#encoding=utf-8

#由于身高和体重是由用户输入的，需要转换类型才能做数字运算，由于身高和体重可能会有小数点以后的数字不能使用int类型，int只支持整数，可以使用float浮点类型

weight=float(input("请输入您的体重:(kg)"))

height=float(input("请输入您的身高:(m)"))

bmi=weight/(height\*\*2)#体重除以身高的平方

if bmi <= 18.5:

print("BMI指数为:%.2f,体重过轻，需要增重"%bmi)#%.2f代表浮点数只要小数点后两位

elif bmi >18.5 and bmi <=25:

print("BMI指数为:%.2f,体重正常，请注意保持"%bmi)

elif bmi >25 and bmi <= 28:

print("BMI指数为:%.2f,体重过重，请注意节食"%bmi)

elif bmi >28 and bmi <=32:

print("BMI指数为:%.2f,体重肥胖，请注意减肥"%bmi)

else:

print("BMI指数为:%.2f,体重过于肥胖，一定要减肥"%bmi)

循环语句：while和for

while循环是条件成立的时候不停的执行某个动作，直到条件不成立为止

语法格式：

while 条件: #结尾必须是:

执行命令1

执行命令2

执行命令3

...

直到条件不成立结束循环

计算从1加到100的总和

vim 06-从1加到100-while.py

#encoding=utf-8

num=1

sum=0

while num <=100: #定义while循环的条件

sum+=num

num+=1

print("从1加到100的数值是:%s"%sum)#这个需要顶行写，代表是while循环外的语句，如果tab制表位也是属于循环内

从1加到100只加偶数

vim 07-从1加到100只加偶数.py

#encoding=utf-8

num=1

sum=0

while num <= 100:

if num%2==0:#取模(能整除2)

sum+=num

num+=1 #这个需要和if对齐代表属于if语句之外，

print("从1加到100只要偶数的总和是:%s"%sum)

可以使用while嵌套以\*来输出一个矩形，类似于之前的九九乘法表

vim 08-矩形输出.py

x=int(input("请输入矩形的长:"))

y=int(input("请输入矩形的宽:"))

Y=1

print("---矩形输出开始---")

while Y<=y:

X=1

while X<=x:

print("\*",end="")#默认每一次循环的print都会换行，end就是定义换行符

X+=1

print("")#当一行输出完成后，立马换行，必须在第二个while循环外

Y+=1

print("---矩形输出完毕---")

循环语句中也可以使用else，只会执行一次，当循环不成立的时候直接执行else内容

使用for循环从1加到100

vim 09-从1加到100-for.py

sum=0

for num in range(1,101): #range是含头不含尾

sum+=num

num+=1

print("从1加到100的总和是:%s"%sum)

猜数字小游戏脚本

vim 10-猜数字小游戏.py

#encoding=utf-8

from random import randint#载入随机数模块，否则下面无法调用

NUM=randint(1,100)#随机取值1--99

count=0

while True:#永远循环不会停止

num=int(input("请输入一个数字(1--99):"))

count+=1

if num < NUM:

print("太小了，请重新输入")

elif num > NUM:

print("太大了，请重新输入")

else:

print("恭喜你，猜对了，尝试次数为:%s"%count)

break #一旦猜对结束循环

字符串

a=100 定义了一个int类型

b="100" 定义了一个str类型

int可以累积

str可以累积

int和str不能累加

下标位：用来标识字符串中每一个字符

含头不含尾

name="abcdefghijk"

name[2] 显示第三个字符，0是第一个

name[0:2] 显示第一个到第二个字符

name[0:-1] 显示除了最后一个字符

name[2:-1] 显示第三个到最后字符（最后一个不显示）

name[2::] 显示第三个到最后一个字符

name[0::2] 显示第一个、第三个、第五个一直到最后隔一个显示一个

name[-1::-1] 从最后一个字符显示到最前面的字符，颠倒字符串