# 1. if判断语句

## 1.1. 生活中的判断场景

火车站安检



上网吧



## 1.2. 开发中的判断场景

密码判断



重要日期判断

if 今天是周六或者周日:

约妹子

if 今天是情人节:

买玫瑰

if 今天发工资:

先还信用卡的钱

if 有剩余:

又可以happy了，O(∩\_∩)O哈哈~

else:

噢，no。。。还的等30天

小总结：

如果某些条件满足，才能做某件事情，而不满足时不允许做，这就是所谓的判断

不仅生活中有，在软件开发中“判断”功能也经常会用到

## 1.3. if判断语句介绍

### 1.3.1. if判断语句语法

if语句是用来进行判断的，其使用格式如下：

if 要判断的条件:

条件成立时，要做的事情

### 1.3.2. 案例

demo1:

age = 30

print("------if判断开始------")

if age>=18:

print("我已经成年了")

print("------if判断结束------")

运行结果:

------if判断开始------

我已经成年了

------if判断结束------

demo2:

age = 16

print("------if判断开始------")

if age>=18:

print("我已经成年了")

print("------if判断结束------")

运行结果:

------if判断开始------

------if判断结束------

小总结：

以上2个demo仅仅是age变量的值不一样，结果却不同；能够看得出if判断语句的作用：就是当满足一定条件时才会执行那块代码，否则就不执行那块代码

注意：

**代码的缩进为一个tab键，或者4个空格**

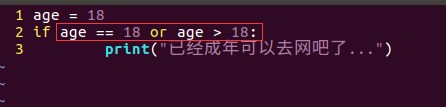
### 1.3.3. 练一练

要求：从键盘获取自己的年龄，判断是否大于或者等于18岁，如果满足就输出“哥，已成年，网吧可以去了”

1. 使用input从键盘中获取数据，并且存入到一个变量中
2. 使用if语句，来判断 age>=18是否成立
3. 支持Python2和Python3都能正常运行

## 1.5. 想一想

判断age大于或者等于18岁，使用的是 >=，还有哪些呢？



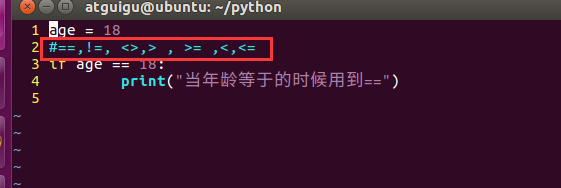
接下来学习比较运算

# **2. 运算符**

## 2.1. 比较(关系)运算符

python中的比较运算符如下表

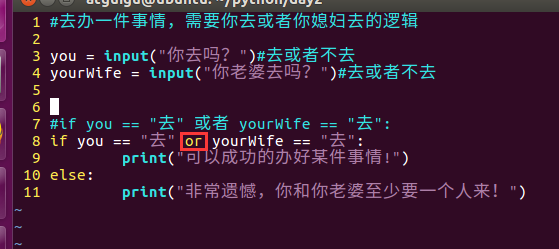
| 运算符 | 描述 | 示例 |
| --- | --- | --- |
| == | 检查两个操作数的值是否相等，如果是则条件变为真。 | 如a=3,b=3则（a == b) 为 true. |
| != | 检查两个操作数的值是否相等，如果值不相等，则条件变为真。 | 如a=1,b=3则(a != b) 为 true. |
| <> | 检查两个操作数的值是否相等，如果值不相等，则条件变为真。Python和在[Pascal](https://baike.baidu.com/item/Pascal" \t "https://baike.baidu.com/item/%3C%3E/_blank)等特有方式，java和c没有,**在Python3中废弃了** | 如a=1,b=3则(a <> b) 为 true。这个类似于 != 运算符 |
| > | 检查左操作数的值是否大于右操作数的值，如果是，则条件成立。 | 如a=7,b=3则(a > b) 为 true. |
| < | 检查左操作数的值是否小于右操作数的值，如果是，则条件成立。 | 如a=7,b=3则(a < b) 为 false. |
| >= | 检查左操作数的值是否大于或等于右操作数的值，如果是，则条件成立。 | 如a=3,b=3则(a >= b) 为 true. |
| <= | 检查左操作数的值是否小于或等于右操作数的值，如果是，则条件成立。 | 如a=3,b=3则(a <= b) 为 true. |



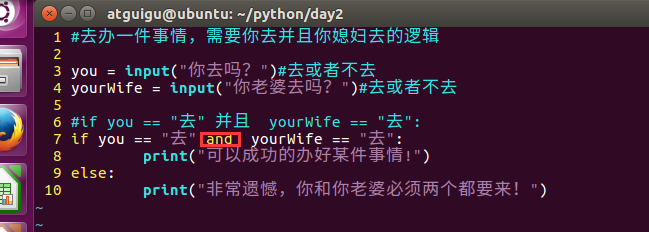
## 2.2. 逻辑运算符

| 运算符 | 逻辑表达式 | 描述 | 实例 |
| --- | --- | --- | --- |
| and | x and y | 布尔"与" - 如果 x 为 False，x and y 返回 False，否则它返回 y 的计算值。 |  |
| or | x or y | 布尔"或" - 如果 x 是 True，它返回 True，否则它返回 y 的计算值。 |  |
| not | not x | 布尔"非" - 如果 x 为 True，返回 False 。如果 x 为 False，它返回 True。 |  |

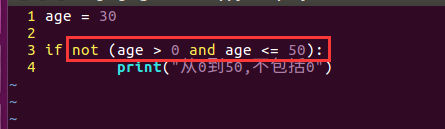
### 2.2.1. or



### 2.2.2. and



### 2.2.3. not



取的范围就变成了，<= 0并且>50的范围

# 3. if-else语句

想一想：在使用if的时候，它只能做到满足条件时要做的事情。那万一需要在不满足条件的时候，做某些事，该怎么办呢？

答：else

## 3.1. if-else的使用格式

if 条件:

满足条件时要做的事情1

满足条件时要做的事情2

满足条件时要做的事情3

...(省略)...

else:

不满足条件时要做的事情1

不满足条件时要做的事情2

不满足条件时要做的事情3

...(省略)...

### 车票案例

ticket = 1 # 用1代表有车票，0代表没有车票

if ticket == 1:

print("有车票，可以上火车")

print("终于可以见到Ta了，美滋滋~~~")

else:

print("没有车票，不能上车")

print("亲爱的，那就下次见了，一票难求啊~~~~(>\_<)~~~~")

结果1：有车票的情况

有车票，可以上火车

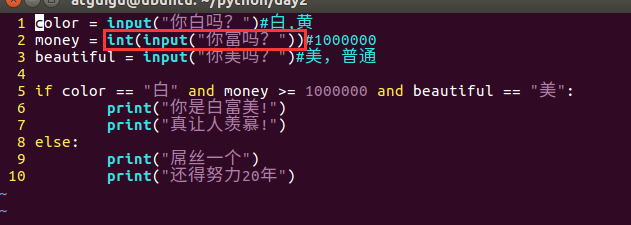
终于可以见到Ta了，美滋滋~~~

结果2：没有车票的情况

没有车票，不能上课

亲爱的，那就下次见了，一票难求啊~~~~(>\_<)~~~~

## 3.2. 使用if-else语句完成白富美案例





## 3.3. 练一练

要求：从键盘输入刀子的长度，如果刀子长度没有超过10cm，则允许上火车，否则不允许上火车；单词提醒：knife刀

# 4. elif

## 4.1. elif的功能

elif的使用格式如下:

if xxx1:

事情1

elif xxx2:

事情2

elif xxx3:

事情3

说明:

当xxx1满足时，执行事情1，然后整个if结束

当xxx1不满足时，那么判断xxx2，如果xxx2满足，则执行事情2，然后整个if结束

当xxx1不满足时，xxx2也不满足，如果xxx3满足，则执行事情3，然后整个if结束

### 4.1.1 判断年龄级别案例

### 4.1.2 判断等价考试案例

score = 77

if score>=90 and score<=100:

print('本次考试，等级为A')

elif score>=80 and score<90:

print('本次考试，等级为B')

elif score>=70 and score<80:

print('本次考试，等级为C')

elif score>=60 and score<70:

print('本次考试，等级为D')

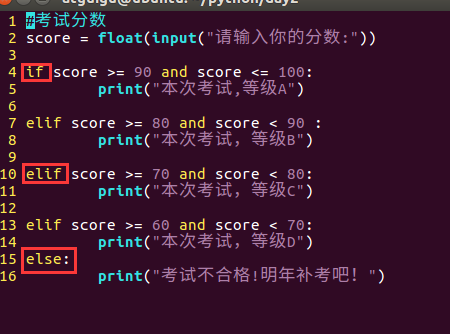
elif score>=0 and score<60:

print('本次考试，等级为E')



## 4.2. 注意点

### 4.2.1.可以和else一起使用



if 性别为男性:

输出男性的特征

...

elif 性别为女性:

输出女性的特征

...

else:

第三种性别的特征

...

说明:

当 “性别为男性” 满足时，执行 “输出男性的特征”的相关代码

当 “性别为男性” 不满足时，如果 “性别为女性”满足，则执行 “输出女性的特征”的相关代码

当 “性别为男性” 不满足，“性别为女性”也不满足，那么久默认执行else后面的代码，即 “第三种性别的特征”相关代码

### 4.2.2. elif必须和if一起使用，否则出错

# 5. if嵌套

通过学习if的基本用法，已经知道了

当需要满足条件去做事情的这种情况需要使用if

当满足条件时做事情A，不满足条件做事情B的这种情况使用if-else

想一想：

坐火车或者地铁的实际情况是：先进行安检如果安检通过才会判断是否有车票，或者是先检查是否有车票之后才会进行安检，即实际的情况某个判断是再另外一个判断成立的基础上进行的，这样的情况该怎样解决呢？

答：

if嵌套

## 5.1. if嵌套的格式

if 条件1:

满足条件1 做的事情1

满足条件1 做的事情2

...(省略)...

if 条件2:

满足条件2 做的事情1

满足条件2 做的事情2

...(省略)...

说明

外层的if判断，也可以是if-else

内层的if判断，也可以是if-else

根据实际开发的情况，进行选择

## 5.2. if嵌套的应用

### 进车站demo：

ticket= 1 # 用1代表有车票，0代表没有车票

knife\_lenght = 9 # 刀子的长度，单位为cm

if ticket== 1:

print("有车票，可以进站")

if knife\_lenght< 10:

print("通过安检")

print("终于可以见到Ta了，美滋滋~~~")

else:

print("没有通过安检")

print("刀子的长度超过规定，等待警察处理...")

else:

print("没有车票，不能进站")

print("亲爱的，那就下次见了，一票难求啊~~~~(>\_<)~~~~")

结果1：ticket= 1;knife\_lenght= 9

有车票，可以进站

通过安检

终于可以见到Ta了，美滋滋~~~

结果2：ticket= 1;knife\_lenght= 20

有车票，可以进站

没有通过安检

刀子的长度超过规定，等待警察处理...

结果3：ticket= 0;knife\_lenght= 9

没有车票，不能进站

亲爱的，那就下次见了，一票难求啊~~~~(>\_<)~~~~

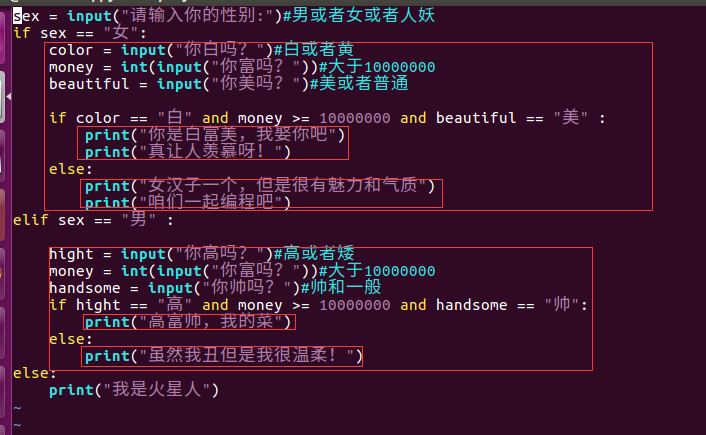
结果4：ticket= 0;knife\_lenght= 20

没有车票，不能进站

亲爱的，那就下次见了，一票难求啊~~~~(>\_<)~~~~

**想一想:为什么结果3和结果4相同？？？**

### 改造白富美案例



## 5.3. 练一练

情节描述：上公交车，并且可以有座位坐下

要求：输入公交卡当前的余额，只要超过2元，就可以上公交车；如果空座位的数量大于0，就可以坐下

## 5.4. 小结

if 语句可以相互嵌套

if 嵌套，可以嵌套多层，但是一般嵌套两层层就行了，如果嵌套多层的话便维护代码，如果需要嵌套多层，肯定可用用其他方式代替多层嵌套这种方式。

# 6. if语句应用:石头剪子布游戏

## 2.4.1 运行效果:

## 2.4.2 参考代码:

import random

player = input('请输入：剪刀(0) 石头(1) 布(2):')

player = int(player)

computer = random.randint(0,2)

# 用来进行测试

#print('player=%d,computer=%d',(player,computer))

if ((player == 0) and (computer == 2)) or ((player ==1) and (computer == 0)) or ((player == 2) and (computer == 1)):

print('获胜，哈哈，你太厉害了')

elif player == computer:

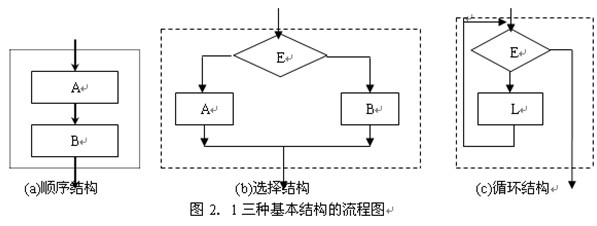
print('平局，要不再来一局')

else:

print('输了，不要走，洗洗手接着来，决战到天亮')

# 7. 循环语句介绍

## 7.1. 引入



程序执行有三种方式：顺序执行，选择执行，循环执行。

## 7.2. 生活中的循环场景

跑道



风扇

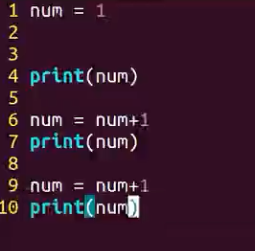


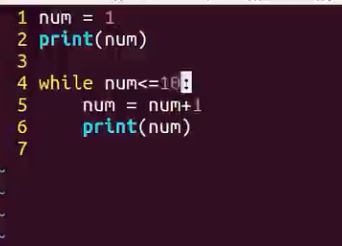
王者荣耀



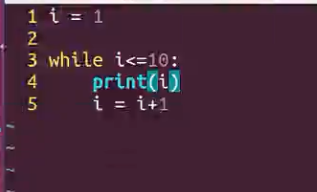
### 7.2.1. 理解循环案例代码

在屏幕上输出1~10





### 7.2.2. While循环正确的写法



### 7.2.3. 画图分析执行过程

## 7.3. 软件开发中循环的使用场景

跟媳妇承认错误，说一万遍"媳妇儿，我错了"

print("媳妇儿，我错了")

print("媳妇儿，我错了")

print("媳妇儿，我错了")

...(还有99997遍)...

使用循环语句一句话搞定

i = 0

while i < 10000:

print("媳妇儿，我错了")

i+=1

## 7.4 小总结

一般情况下，需要多次重复执行的代码，都可以用循环的方式来完成

**循环不是必须要使用的，但是为了提高代码的重复使用率，所以有经验的开发者都会采用循环**

# 8. while循环

## 8.1. while循环的格式

while 条件:

条件满足时，做的事情1

条件满足时，做的事情2

条件满足时，做的事情3

...(省略)...

Demo打印下标从1开始

i = 0

while i < 5:

print("当前是第%d次执行循环"%(i+1))

print("i=%d"%i)

i += 1

结果

当前是第1次执行循环

i=0

当前是第2次执行循环

i=1

当前是第3次执行循环

i=2

当前是第4次执行循环

i=3

当前是第5次执行循环

i=4

## 8.2. while循环的注意事项

i=i+1别忘记写,否则条件永远满足，一直执行

# 9. while循环应用

## 9.1. 计算1~100的累积和（包含1和100）

参考代码如下:

#encoding=utf-8

i = 1

sum = 0

while i<=100:

sum = sum + i

i += 1

print("1~100的累积和为:%d"%sum)

5050

## 9.2. 计算1~100之间偶数的累积和（包含1和100）

参考代码如下:

#encoding=utf-8

i = 1

sum = 0

while i <= 100 :

**if i%2 == 0**:

sum = sum + i

i+=1

print("1~100的累积和为:%d"%sum)

2550

# 10. while循环嵌套

前面学习过if的嵌套了，想一想if嵌套是什么样子的？

类似if的嵌套，while嵌套就是：while里面还有while

## 10.1. while嵌套的格式

while 条件1:

条件1满足时，做的事情1

条件1满足时，做的事情2

条件1满足时，做的事情3

...(省略)...

while 条件2:

条件2满足时，做的事情1

条件2满足时，做的事情2

条件2满足时，做的事情3

...(省略)...

## 10.2. while嵌套应用一打印三角形

### 打印矩形

\* \* \* \* \*

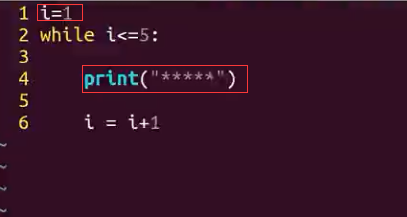
\* \* \* \* \*

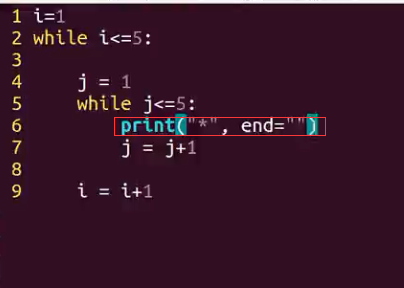
\* \* \* \* \*

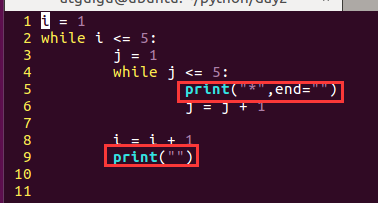
\* \* \* \* \*

\* \* \* \* \*

参考代码：







### 打印三角形

要求：打印如下图形：

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

参考代码：

i = 1

while i<=5:

j = 1

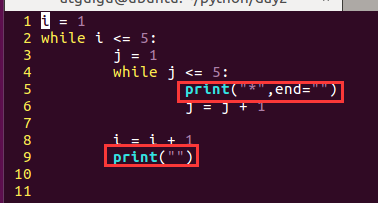
while j<=i:

print("\* ",end='')

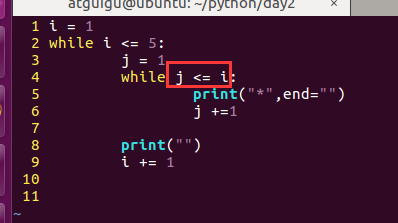
j+=1

print("\n")

i+=1

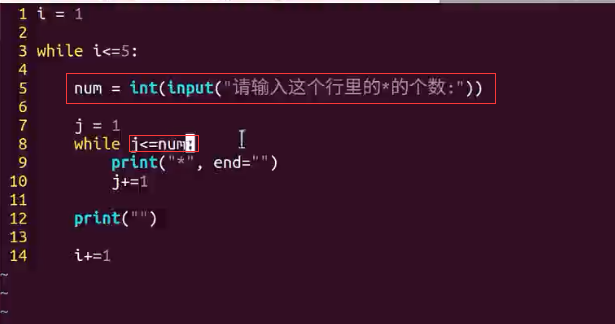


只要把打印矩形的基础上把内部的while循环的条件由j<=5修改成 j<=i即可

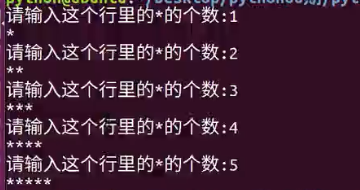


### 分析打印三角形的过程

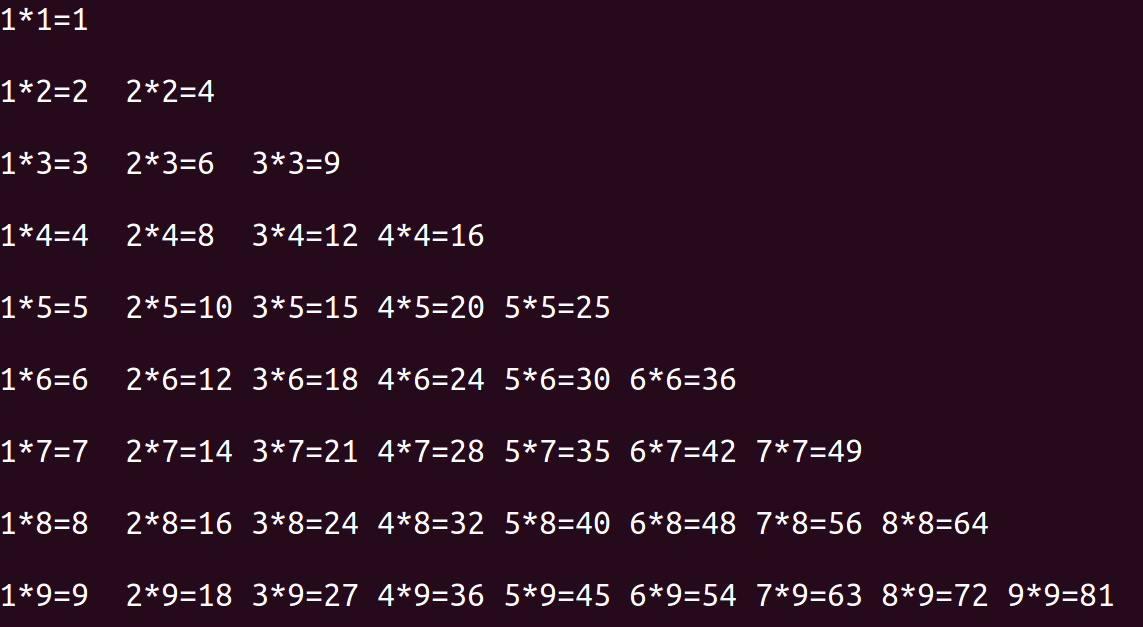
### 打印三角形扩展



执行效果



## 10.3. while嵌套应用二：九九乘法表



参考代码：

i = 1

while i <= 9 :

j = 1

while j <= i :

print("%d\*%d=%d\t" % (j,i,i\*j),end="")

j += 1

i += 1

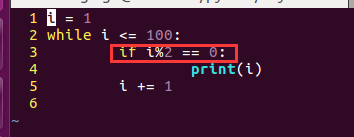
#print("\n",end="")

print("")

\t添加制表符

## 10.4. 打印1~100之间的偶数

思路：while循环嵌套if判断语句，代码如图：



# 11. for循环语句

像while循环一样，for可以完成循环的功能。

在Python中 for循环可以遍历任何序列的项目，如一个**列表**、**字符串、元组**等等。

## 11.1. for循环的格式

for 临时变量 in 列表或者字符串等:

循环满足条件时执行的代码

else:

循环不满足条件时执行的代码

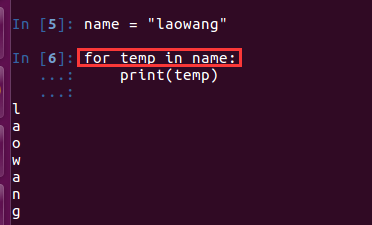
## 11.2. for的案例

name = 'atguigu'

for x in name:

print(x)

运行结果如下:



## 11.3. for-else

name = ''

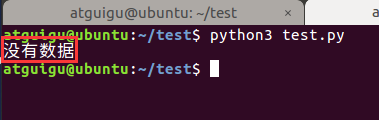
for x in name:

print(x)

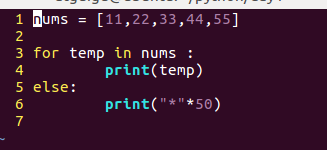
else:

print("没有数据")

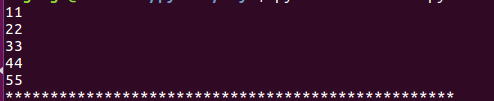
运行结果如下:



### 11.3.1其他语言没有这样的写法



执行执行结果



# 12. break和continue

## 12.1. break

### 12.1.1 for循环中的break

普通的循环示例如下：

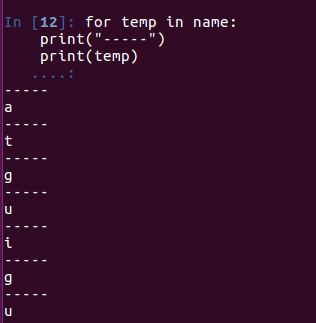
name = 'atguigu'

for x in name:

print('----')

print(x)

运行结果:



带有break的循环示例如下:

name = 'atguigu'

for x in name:

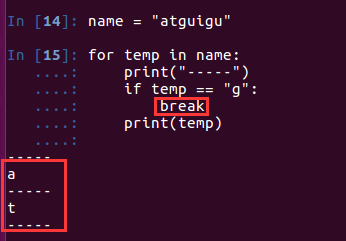
print('----')

if x == 'g':

break

print(x)

运行结果:



### 12.1.2. while循环中的break

普通的循环示例如下：

i = 0

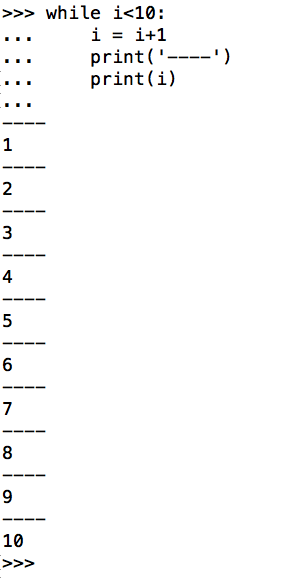
while i<10:

i = i+1

print('----')

print(i)

运行结果:



带有break的循环示例如下:

i = 0

while i<10:

i = i+1

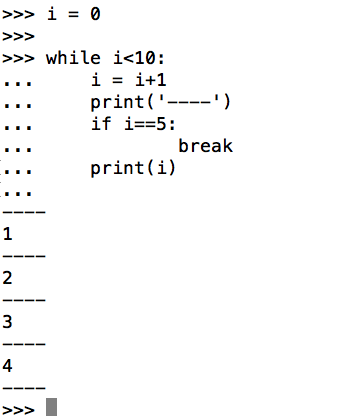
print('----')

if i==5:

break

print(i)

运行结果:



### 12.1.3 while循环中的break作用范围

如果是循环嵌套循环，break在内循环，退出的是内循环

### 12.1.4 break小总结

break的作用：用来结束整个循环

## 12.2. continue

### 12.2.1 for循环中的continue

带有continue的循环示例如下:

name = 'atguigu'

for x in name:

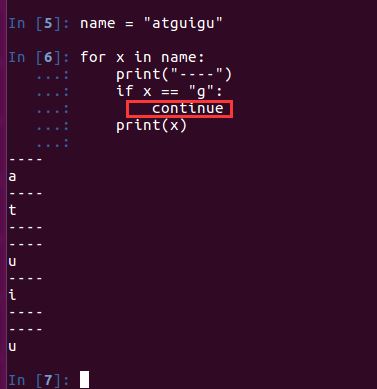
print('----')

if x == 'g':

continue

**print(x)**

运行结果:



### 12.2.1 while循环中的continue

带有continue的循环示例如下:

i = 0

while i<10:

i = i+1

print('----')

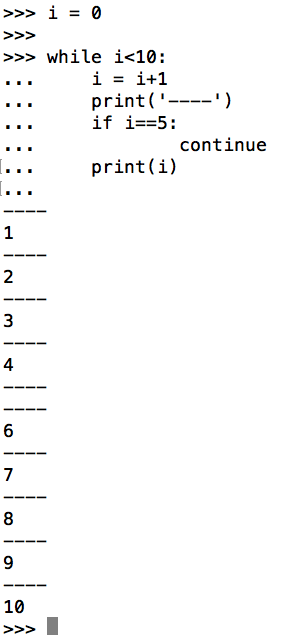
if i==5:

continue

print(i)

**注意：不要把i =+ 1写到最后，否则程序会停留.**

运行结果:



小总结:

**continue的作用：用来结束本次循环，紧接着执行下一次的循环**

## 12.3. 注意点

break/continue在嵌套循环中，只对最近的一层循环起作用



break/continue只能用在循环中，除此以外不能单独使用，只能在循环使用。

break/continue在嵌套循环中，只对最近的一层循环起作用

# 13. if语句和while循环和for循环总结

## 13.1 if语句总结

if往往用来对条件是否满足进行判断

if有4中基本的使用方法：

### 13.1.1. 基本方法

if 条件:

满足时做的事情

### 13.1.2. 满足与否执行不同的事情

if 条件:

满足时做的事情

else:

不满足时做的事情

### 13.1.3. 多个条件的判断

if 条件:

满足时做的事情

elif 条件2:

满足条件2时做的事情

elif 条件3:

满足条件3时做的事情

else:

条件都不满足时做的事情

### 13.1.4. if嵌套

if 条件:

满足时做的事情

这里还可以放入其他任何形式的

if判断语句:

做对应的事情

## 13.2. While语句总结

### 13.2.1. while循环一般通过数值是否满足来确定循环的条件

i = 0

while i< 10:

print("hello")

i+=1

## 13.3. for循环语句总结

### 13.3.1. for循环一般是对能保存多个数据的变量，进行遍历

name = 'atguigu'

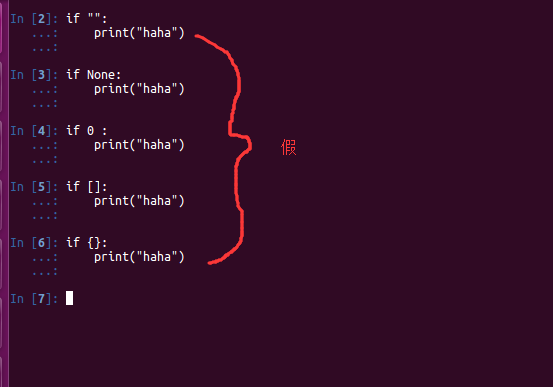
for x in name:

print(x)

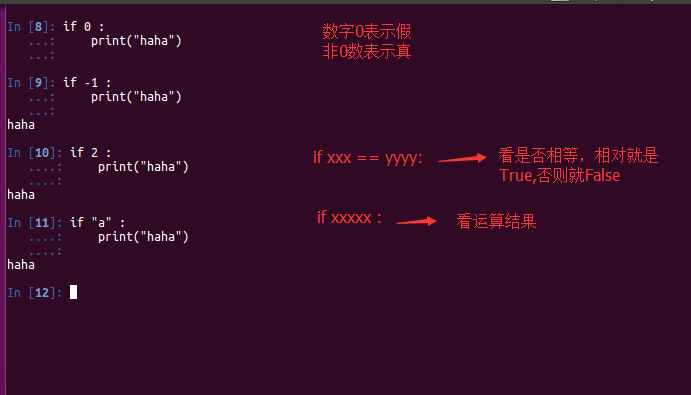
if、while、for等其他语句可以随意组合，这样往往就完成了复杂的功能

# 14.if的各种真假判断

## 14.1. 这些情况都是假



## 14.2. 其他情况



# 15. 作业

## 1. 使用if，编写程序，实现以下功能：

从键盘获取用户名、密码

如果用户名和密码都正确（预先设定一个用户名和密码），那么就显示“欢迎进入xxx的世界”，否则提示密码或者用户名错误

## 2. 使用while，完成以下图形的输出

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

## 打印99乘法表

## 计算1~100之间偶数的累积和（包含1和100）

## name = 'atguigu'，第一问：遍历字符串，并且当是u的时候跳出循环；第二问：变量字符串，并且但是u的时候，跳过本次循环，不打印u.