

# “如果” 和 “复读机”

## 实验指导手册

版本:1.0



华为技术有限公司

**版权所有 © 华为技术有限公司 2020。保留一切权利。**

未经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 华为技术有限公司

地址：                  深圳市龙岗区坂田华为总部办公楼                  邮编：518129

网址：                  <https://edu.huaweicloud.com/>

# 目录

<b>背景介绍</b> .....	<b>1</b>
<b>实验目的</b> .....	<b>1</b>
<b>1 Python 中数据类型的使用</b> .....	<b>2</b>
1.1 实验介绍.....	2
1.2 实验目的.....	2
1.3 资源准备.....	2
1.4 判断语句.....	2
1.5 循环语句.....	5
1.6 小结任务.....	9

## 背景介绍

Python 是一个高层次的结合了解释性、编译性、互动性和面向对象的脚本语言，由于其简单的语法、强大的工具库等优势被很多人青睐，应用于各个行业之中。

## 实验目的

本实验指导主要实了解 Python 中控制流语句的使用。通过本实验，您将能够：

- 了解 Python 中判断语句的使用
- 了解 Python 中循环语句的使用

# 1 Python 控制流语句的使用

## 1.1 实验介绍

实验介绍了本对于Python中判断语句的使用和循环语句的使用。

## 1.2 实验目的

通过以下小的实验可以帮助我们掌握Python这门编程语言中循环判断语句的使用。

## 1.3 资源准备

安装 Python 解释器/anaconda。

安装好 jupyter lab。

(方便起见，实验所使用的环境为云上环境 modelarts)

## 1.4 判断语句

步骤 1 新建 notebook (python3) 文件。并修改名称为 Python 控制流语句。

步骤 2 if 语句的使用

使用代码实现课程中程序员买西瓜 (买一个西瓜，如果看见西红柿就买两个)：

```
watermelon_num = 1# 买一个西瓜
tomato = input("是否看到了卖西红柿的? 看到请输入 1; 没看到输入 0")
tomato = int(tomato)# 转化为 int 类型, 用户后续判断
if tomato: # 判断
    watermelon_num = 2
print("程序员带来了%d个西瓜"%watermelon_num)
```

是否看到了卖西红柿的？看到请输入1；没看到输入0 1  
程序员带来了2个西瓜

使用代码实现：晚上回来买一个西瓜，如果看到西红柿，就再买两个，如果还买两个瓜，你就别回来了

```
watermelon_num = 1
tomato = input("是否看到了卖西红柿的？看到请输入1；没看到输入0")
tomato = int(tomato)
if tomato:
    watermelon_num += 2
if watermelon_num == 2:
    print("程序员不能回家了")
print("程序员带来了%d个西瓜"%watermelon_num)
```

是否看到了卖西红柿的？看到请输入1；没看到输入0 1  
程序员带来了3个西瓜

### 步骤 3 隐式的判定条件

```
if 0:
    print("0")
if 1:
    print("1")
if "":
    print("空字符串")
if []:
    print("空列表")
if -1:
    print(-1)
if None:
    print('None')
```

输出：

```
1
-1
```

### 步骤 4 使用 else 实现如果...否则....

实现：如果看到西红柿就买两个，否则就吃西瓜炒蛋。

```
tomato = input("是否看到了卖西红柿的？看到请输入1；没看到输入0")
tomato = int(tomato)
if tomato:
```

```
print("老板, 来两个西红柿")
else:
    print("吃西瓜炒蛋")
```

是否看到了卖西红柿的? 看到请输入1; 没看到输入0 0  
吃西瓜炒蛋

#### 步骤 5 使用 elif 实现多分支的判断

实现: 如果看到西红柿就买两个, 没有西红柿, 有青椒就买两个青椒, 否则晚上就只能吃西瓜炒蛋了。

```
tomato = input("是否看到了卖西红柿的? 看到请输入1; 没看到输入0")
pepper = input("是否看到了卖青椒的? 看到请输入1; 没看到输入0")
tomato = int(tomato)
pepper = int(pepper)
if tomato:
    print("老板, 来两个西红柿")
elif pepper:
    print("老板, 来两个青椒")
else:
    print("吃西瓜炒蛋")
```

是否看到了卖西红柿的? 看到请输入1; 没看到输入0 0  
是否看到了卖青椒的? 看到请输入1; 没看到输入0 1  
老板, 来两个青椒

#### 步骤 6 使用逻辑运算符实现多条件的判断

实现: 如果有西红柿和鸡蛋就吃番茄炒蛋, 如果有米饭或者面条就配上主食一起吃。

```
egg = 3
tomato = 2
noodles = 1
rice = 0
if egg and tomato:
    print("番茄炒蛋做好了")
if noodles or rice:
    print("可以搭配主食一起吃了")
else:
    print("只能吃菜了")
```

输出:

番茄炒蛋做好了  
可以搭配主食一起吃了

## 1.5 循环语句

### 步骤 1 for 循环语句的使用

```
for i in "python":  
    print(i)
```

输出:

```
p  
y  
t  
h  
o  
n
```

### 步骤 2 加入 else 语句

```
for i in [1,2,3]:  
    print("循环第%d次"%(i))  
else:  
    print("循环完毕")
```

输出:

```
循环第 1 次  
循环第 2 次  
循环第 3 次  
循环完毕
```

### 步骤 3 while 循环

```
i = 5  
while i>0:  
    print(i)  
    i -= 1
```

输出:

```
5  
4  
3  
2  
1
```

### 步骤 4 嵌套循环

使用循环语句实现美观的 12 以内的乘法口诀表:

```
# 使用 for 循环完成乘法口诀  
for i in range(1,13):  
    for j in range(1,i+1):
```

```
print("%2dX%2d=%-3d"%(j,i,j*i), end=" ")
print() # 外循环结束后用作换行
```

```
1X 1=1
1X 2=2   2X 2=4
1X 3=3   2X 3=6   3X 3=9
1X 4=4   2X 4=8   3X 4=12   4X 4=16
1X 5=5   2X 5=10  3X 5=15  4X 5=20  5X 5=25
1X 6=6   2X 6=12  3X 6=18  4X 6=24  5X 6=30  6X 6=36
1X 7=7   2X 7=14  3X 7=21  4X 7=28  5X 7=35  6X 7=42  7X 7=49
1X 8=8   2X 8=16  3X 8=24  4X 8=32  5X 8=40  6X 8=48  7X 8=56  8X 8=64
1X 9=9   2X 9=18  3X 9=27  4X 9=36  5X 9=45  6X 9=54  7X 9=63  8X 9=72  9X 9=81
1X10=10  2X10=20  3X10=30  4X10=40  5X10=50  6X10=60  7X10=70  8X10=80  9X10=90  10X10=100
1X11=11  2X11=22  3X11=33  4X11=44  5X11=55  6X11=66  7X11=77  8X11=88  9X11=99  10X11=110  11X11=121
1X12=12  2X12=24  3X12=36  4X12=48  5X12=60  6X12=72  7X12=84  8X12=96  9X12=108  10X12=120  11X12=132  12X12=144
```

while 循环和 for 循环一起使用实现乘法口诀：

```
for i in range(1,13):
    j = 1
    while j<= i:
        print("%2dX%2d=%-3d"%(j,i,j*i), end=" ")
        j+=1
    print()
```

```
1X 1=1
1X 2=2   2X 2=4
1X 3=3   2X 3=6   3X 3=9
1X 4=4   2X 4=8   3X 4=12   4X 4=16
1X 5=5   2X 5=10  3X 5=15  4X 5=20  5X 5=25
1X 6=6   2X 6=12  3X 6=18  4X 6=24  5X 6=30  6X 6=36
1X 7=7   2X 7=14  3X 7=21  4X 7=28  5X 7=35  6X 7=42  7X 7=49
1X 8=8   2X 8=16  3X 8=24  4X 8=32  5X 8=40  6X 8=48  7X 8=56  8X 8=64
1X 9=9   2X 9=18  3X 9=27  4X 9=36  5X 9=45  6X 9=54  7X 9=63  8X 9=72  9X 9=81
1X10=10  2X10=20  3X10=30  4X10=40  5X10=50  6X10=60  7X10=70  8X10=80  9X10=90  10X10=100
1X11=11  2X11=22  3X11=33  4X11=44  5X11=55  6X11=66  7X11=77  8X11=88  9X11=99  10X11=110  11X11=121
1X12=12  2X12=24  3X12=36  4X12=48  5X12=60  6X12=72  7X12=84  8X12=96  9X12=108  10X12=120  11X12=132  12X12=144
```

使用嵌套循环时需要注意，嵌套层数最好不要超过三层。

两层循环时：

```
n = 0
for i in range(10):
    for j in range(10):
        n+=1
    print(n) # 输出 100
```

三层循环时：

```
n = 0
for i in range(10):
    for j in range(10):
        for k in range(10):
            n+=1
    print(n) # 1000
```

每嵌套一层计算量会大量增加。

### 步骤 5 使用 continue 语句跳过本次循环

```
for letter in 'Python': # 第一个实例
    if letter == 'h':
```

```
        continue
    print ('当前字母 :', letter)
var = 10                # 第二个实例
while var > 0:
    var = var -1
    if var == 5:
        continue
    print('当前变量值 :', var)
print("Good bye!")
```

输出:

```
当前字母 : P
当前字母 : y
当前字母 : t
当前字母 : o
当前字母 : n
当前变量值 : 9
当前变量值 : 8
当前变量值 : 7
当前变量值 : 6
当前变量值 : 4
当前变量值 : 3
当前变量值 : 2
当前变量值 : 1
当前变量值 : 0
Good bye!
```

#### 步骤 6 使用 break 语句终止本次循环

```
for letter in 'Python':    # 第一个实例
    if letter == 'h':
        break
    print('当前字母 :', letter)

var = 10                # 第二个实例
while var > 0:
    print('当前变量值 :', var)
    var = var -1
    if var == 5:    # 当变量 var 等于 5 时退出循环
        break
print("Good bye!")
```

输出:

```
当前字母 : P
当前字母 : y
当前字母 : t
当前变量值 : 10
当前变量值 : 9
当前变量值 : 8
当前变量值 : 7
当前变量值 : 6
Good bye!
```

break 的作用位置:

```
for i in range(3):
    print("当前在外循环: ",i)
    for j in range(3):
        print("当前在内循环: ",j)
        if j==2:
            break
```

输出:

```
当前在外循环: 0
当前在内循环: 0
当前在内循环: 1
当前在内循环: 2
当前在外循环: 1
当前在内循环: 0
当前在内循环: 1
当前在内循环: 2
当前在外循环: 2
当前在内循环: 0
当前在内循环: 1
当前在内循环: 2
```

### 步骤 7 死循环的应用

实现: 输入一个数, 小于 100 就继续输入, 只知道输入数的和超过 100 循环停止。

```
temp = 0
while True:
    num = input("请输入一个数字: ")
    num = float(num)
    temp += num
    if temp >= 100:
        break
```

## 1.6 小结任务

1.当前有一组成绩单:

```
score = [100,79,65,87,97,65,87,97,67]
```

请去除成绩单中成绩大于 80 的成绩,并将取出后的列表打印出来

2. 计算猴子吃桃问题。猴子第 1 天摘了若干个桃子,当即吃了一半零一个;第 2 剩下的吃了一半零一个,一次循环。到第十天时想吃就剩下一个桃子。求第一天摘了几个桃子?