

上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主编 李东方 支晓勇

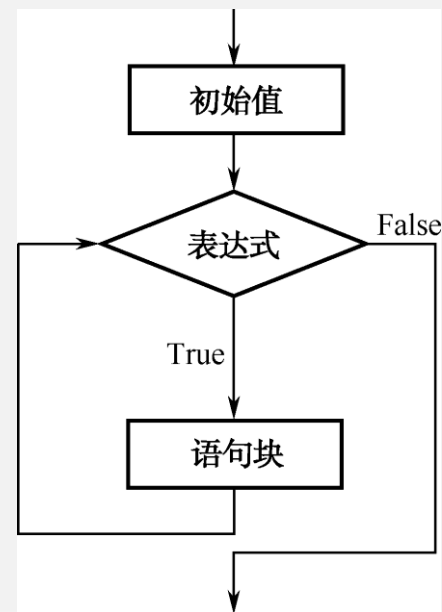
## 3.4 循环结构

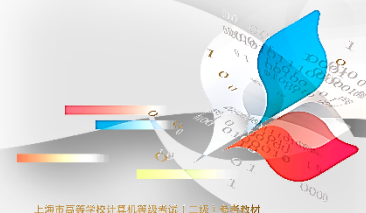
### • while语句

初始值

**while** 表达式:      #表达式的值为真  
          语句块A        #符合条件时执行的语句  
**else:**                #可选项  
          语句块B

```
time=8
while time<12:
    print("有效次数内")
    time=time+1
else:
    print("计次已满")
```





上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主 编 李东方 支晓勇

## 3.4 循环结构

### • while语句

- 与if语句类似，while语句的表达式可以是任意类型，如 $x \neq y$ ， $x > 3$  or  $x < 5$ ，-5等。
- 循环体中的语句块有可能一次也不执行，上例中若初始值 $time = 13$ ，则语句块不会执行。
- 语句块可以是一条或多条语句，上例中while子句中的语句块为两条语句，else子句中的语句块为一条语句。
- while循环中的else子句可以省略，上例中若没有else子句，则当计次等于12时，while语句结束，程序继续执行while循环后面的语句。
- 程序中需要包含使循环结束的语句，上例中若缺少语句 $time = time + 1$ ，则程序无法终止。

## 3.4 循环结构

### • 死循环

在while循环中，如果表达式的值恒真，循环将一直执行下去，无法靠自身终止，从而产生死循环。

```
while 1:
```

```
    print("Python是一门编程语言")
```

- 程序中需要包含使循环结束的语句，上例中若缺少语句`time=time+1`，则程序无法终止。





上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主 编 李东方 支晓勇

## 3.4 循环结构

- 【例3-9】 编写程序，统计并输出1 ~ 1000以内所有能够同时被3和7整除的数字个数

```
number=1
```

```
count=0
```

```
while number<=1000:
```

```
    if number%3==0 and number%7==0:
```

```
        count=count+1
```

```
    number=number+1
```

```
print("同时能够被数字3和7整除的数字个数为: ",count)
```

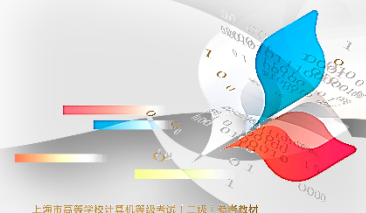


## 3.4 循环结构

- 【例3-10】 编写程序，用下列公式计算 $\pi$ 的近似值，直到最后一项的绝对值小于 $10^{-6}$ 为止。

```
import math
n=1 # 变量自增值
t=1 # 每项值
total=0 # 1/4π的值
flag=1 # 标记位
while math.fabs(t)>=1e-6: # 每项值的绝对值大于1e-6时进行
    total=total+t
    flag=-flag
    n=n+2
    t=flag*1.0/n
print("n=%f" % (total*4))
```

$$\frac{\pi}{4} \approx 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$$



上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主 编 李 芳 芳 支 晓 勇

## 3.4 循环结构

### • for循环和range()内建函数

**for** 循环变量 **in** 对象:

语句块A

**else:** #可选

语句块B

```
word = '山羊上山山碰山羊角'
```

```
sum = 0
```

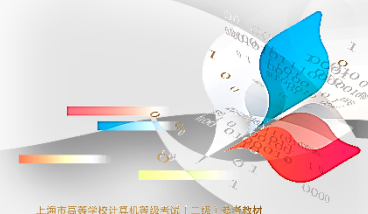
```
for letter in word:
```

```
    if letter == '山':
```

```
        sum += 1
```

```
print(sum)
```

程序运行结果为“4”，即以遍历方式计算出“山”在字符串中出现的次数



上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主编 李东方 支晓勇

## 3.4 循环结构

### • 内建函数range()

用于生成整数序列

**range(start, end, step)**

- start决定序列的起始值 (起始值可以省略, 省略时该值为0)
- end代表序列的终值 (半开区间, 不包括end的值)
- step代表序列的步长 (可以省略, 默认值是1)

```
for iNum in range(4,10,2):  
    print(iNum,end=" ")
```

## 3.4 循环结构

- 【例3-11】 编写程序，使用for语句计算1~10000的自然数之和。

```
total=0
for iNum in range(1,10001,1):
    total=total+iNum
print("1~10000的总和为：",total)
```



上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主 编 李 芳 芳 支 欣 芳



## 3.4 循环结构



- 【例3-12】 编写程序，解决以下问题。

4个人中有一人做了好事，已知有三个人说了真话，根据下面对话判断是谁做的好事。

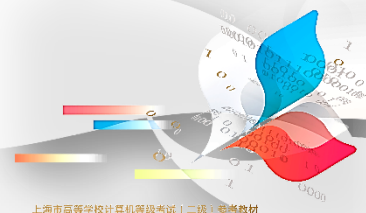
A说：不是我；

B说：是C；

C说：是D；

D说：C胡说。

```
for iNum in ['A','B','C','D']:
    if (iNum!='A') + (iNum=='C') + (iNum=='D') + (iNum!='D')==3:
        print(iNum,"做了好事！")
```



上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主編 李东方 支晓勇

## 3.4 循环结构

### • 循环语句嵌套

```
for i in range(1,3):  
    for j in range(1,4):  
        print (i*j,end=" ")
```

- 循环语句的嵌套层数不限
- 循环的内外层之间不能交叉
- 双层循环是一种常用的循环嵌套，循环的总次数等于内外层次数之积

## 3.4 循环结构



- 【例3-13】 编写程序，使用双重循环输出九九乘法表。

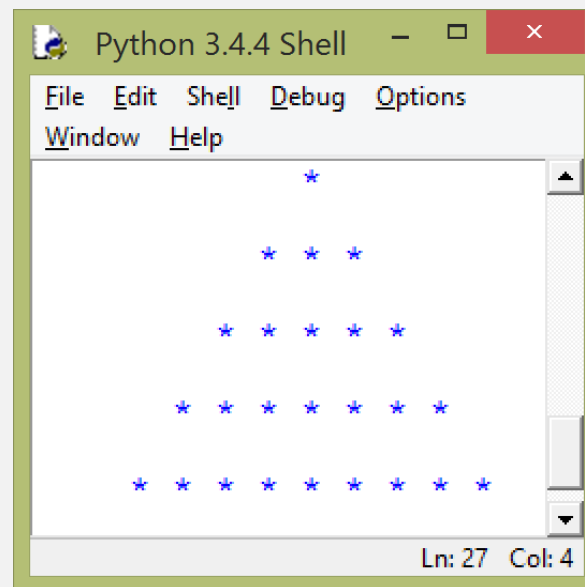
```
for i in range(1,10):  
    for j in range(1,10):  
        print("%s*%s=%2s" % (i,j,i*j), end="\t")  
    print("\n")
```



## 3.4 循环结构

- 【例3-14】 编写程序，使用双重循环输出所示三角形图案。

```
for i in range(1,6):
    for j in range(5-i):
        print(" ",end=" ")
    for j in range(1,2*i):
        print("*",end=" ")
    print("\n")
```



## 3.4 循环结构

Python程序设计基础

(第2版)

主編 李东方 支晓勇

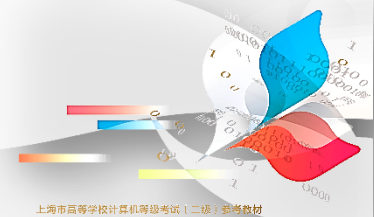
【例3-15】 以一道“奥数”题的解题，体会计算思维与逻辑思维的差别。

有一个以文字代替数字的算术表达式如图所示，已知4个替代数字的文字没有重复数字，编写程序求出文字所替代的数字。

$$\begin{array}{r} \text{山外山} \\ +) \text{青龙山} \\ \hline \text{青龙山外} \end{array}$$

**按逻辑思维：**由3位数和3位数相加等于4位数，则“青”只能是1；“山” + “青”  $\geq 10$ ，因此“山”只能是9，“龙”是0；个位的两个“山”相加，推得“外”等于8。

**按计算思维：**注重于程序的实现，用穷举法设计嵌套的4层循环，把所有的数字都试一遍，找出4个数字不相互重复的组合满足加法等式条件。



上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主 编 李东方 支晓勇

## 3.4 循环结构

```
for qing in range(10):
    for long in range(10):
        for shan in range(10):
            for wai in range(10):
                if (qing==long or qing==shan or qing==wai
                    or long==shan or long==wai or shan==wai):
                    continue
                elif (qing*1000+long*100+shan*10+wai==
                    shan*100+ wai*10 +shan+ qing*100+
long*10+shan):
                    print('qing=%d,long=%d,shan=%d,wai=%d'
                        %(qing,long,shan,wai))
                    break
```

$$\begin{array}{r} \text{山外山} \\ +) \text{ 青龙山} \\ \hline \text{青龙山外} \end{array}$$

## 3.4 循环结构

### • break语句

中断当前循环的执行，跳出循环结构。

- 对于包含else子句的while循环和for循环而言，在while或for子句中一旦执行break语句，else子句将没有机会执行。





## 3.4 循环结构

```
import random
point=random.randint(1,6)
count=1
while count<=3:
    guess=int(input("请输入您的猜测: "))
    if guess>point:
        print("您的猜测偏大")
    elif guess<point:
        print("您的猜测偏小")
    else:
        print("恭喜您猜对了")
        break
    count=count+1
else:
    print("很遗憾, 三次全猜错了! ")
```

- **【例3-16】** 编写程序，随机产生色子的一面（数字1~6），给用户三次猜测机会，程序给出猜测提示（偏大或偏小）。如果某次猜测正确，则提示正确并中断循环；如果三次均猜错，则提示机会用完。



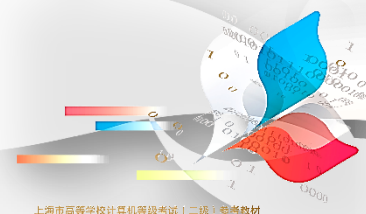
## 3.4 循环结构

### • 半路循环

在死循环程序中，通过添加break语句终止程序的执行

```
number=1
while 1:
    print("Python是一门编程语言")
    if number>=5:
        break
    number=number+1
```





## 3.4 循环结构

- **continue语句**

用于中断本次循环的执行，进入下一轮循环条件是否满足的判断

- 【例3-17】 编写程序，从键盘输入一段文字，如果其中包括“密”字（可能出现0次、1次或者多次），则输出时过滤掉该字，其他内容原样输出。

```
sentence=input("请输入一段文字：")  
for word in sentence:  
    if word=="密":  
        continue  
    print(word,end="")
```

## 3.4 循环结构

```
while 1:
    password=input("请输入密码：")
    if len(password)<6:
        print("长度为6位，请重试！")
        continue
    if password=="123456":
        print("恭喜您，密码正确！")
        break
    else:
        print("密码有误，请重试！")
```

- 【例3-18】编写程序，从键盘输入密码，如果密码长度小于6，则要求重新输入。如果长度等于6，则判断密码是否正确，如果正确则中断循环，否则提示错误并要求继续输入。

