



# 通用循环构造方法

礼 欣

北京理工大学



# 交互式循环

- 交互式循环是无限循环的一种
- 允许用户通过交互的方式重复程序的特定部分
- 让我们以交互循环的视角重新审视求平均数程序，伪码如下：

初始化sum为0

初始化count为0

初始化moredata为"yes"

当moredata值为"yes"时

    输入数字x

    将x加入sum

    count值加1

    询问用户是否还有moredata需要处理

输出sum/count





# 交互式循环代码

```
# average2.py
def main():
    sum = 0.0
    count = 0
    moredata = "yes"
    while moredata[0] == "y":
        x = eval(input("Enter a number >> "))
        sum = sum + x
        count = count + 1
        moredata = input("Do you have more numbers (yes or no)?")
    print("\nThe average of the numbers is", sum/count)
main()
```



# 交互式循环执行

- 程序的输出如下：

```
>>>
Enter a number >> 32
Do you have more numbers (yes or no)?yes
Enter a number >> 45
Do you have more numbers (yes or no)?y
Enter a number >> 34
Do you have more numbers (yes or no)?y
Enter a number >> 76
Do you have more numbers (yes or no)?y
Enter a number >> 45
Do you have more numbers (yes or no)?nope
```

The average of the numbers is 46.4

- 用户不再需要计数，但又总被提示信息打扰



# 哨兵循环

- 执行循环直到遇到特定的值，循环语句才终止执行的循环结构设计方法
- 哨兵循环是求平均数的更好方案，思路如下：
  - 设定一个哨兵值作为循环终止的标志
  - 任何值都可以做哨兵，但要与实际数据有所区别

伪码如下：

接收第一个数据

while 这个数据不是哨兵

    程序执行相关语句

    接收下一个数据项

在求考试分数平均数的程序中，可以设定负数为哨兵



# 哨兵循环版本1代码

```
# average3.py
def main():
    sum = 0.0
    count = 0
    x = eval(input("Enter a number (negative to quit) >> "))
    while x >= 0:
        sum = sum + x
        count = count + 1
        x = eval(input("Enter a number (negative to quit) >> "))
    print("\n The average of the numbers is", sum/count)
main()
```





# 哨兵循环版本1执行

- 下面是程序执行过程:

```
>>>
Enter a number (negative to quit) >> 32
Enter a number (negative to quit) >> 45
Enter a number (negative to quit) >> 34
Enter a number (negative to quit) >> 76
Enter a number (negative to quit) >> 45
Enter a number (negative to quit) >> -1

The average of the numbers is 46.4
>>>
```

- 没有那么多yes/no的干扰，执行结果更加清晰
- 但不能包含负数的平均数计算，为了更加通用化需要引入字符串



# 哨兵循环版本2

- 利用非数字字符串表示输入结束
- 所有其他字符串将被转换成数字作为数据处理
- 空字符串以" "（引号中间没有空格）代表，可以作为哨兵，用户输入回车Python就返回空字符串
- 伪码如下：

初始化sum为0

初始化count为0

接受输入的字符串数据，xStr

while xStr非空

    将xStr转换为数字x

    将x加入sum

    count值加1

    接受下个字符串数据，xStr

输出sum/count







# 哨兵循环版本2代码

```
# average4.py
def main():
    sum = 0.0
    count = 0
    xStr = input("Enter a number (<Enter> to quit) >> ")
    while xStr != "":
        x = eval(xStr)
        sum = sum + x
        count = count + 1
        xStr = input("Enter a number (<Enter> to quit) >> ")
    print("\nThe average of the numbers is", sum / count)
main()
```



# 哨兵循环版本2执行

## ■ 以下是程序执行过程

```
>>>
Enter a number (<Enter> to quit) >> 34
Enter a number (<Enter> to quit) >> 23
Enter a number (<Enter> to quit) >> 0
Enter a number (<Enter> to quit) >> -25
Enter a number (<Enter> to quit) >> -34.4
Enter a number (<Enter> to quit) >> 22.7
Enter a number (<Enter> to quit) >>

The average of the numbers is 3.3833333333333333
>>>
```





# 文件循环

- 面向文件的方法是数据处理的典型应用
- 之前求平均数的数字都是用户输入的，如果几百个数求平均，输入困难且容易出错
- 可以事先将数据录入到文件中，然后将这个文件作为程序的输入，避免人工输入的麻烦，便于编辑修改



# 文件循环代码

```
# average5.py
def main():
    fileName = input("What file are the numbers in? ")
    infile = open(fileName, 'r')
    sum = 0
    count = 0
    for line in infile:
        sum = sum + eval(line)
        count = count + 1
    print("\nThe average of the numbers is", sum / count)
main()
```



# 遍历文件

- 在这段代码中，循环变量line遍历文件的每一行，将每行都转成数字然后加到sum中。
- 通过Python的readline()来读取，readline()将文件的一行读取到字符串中。
- 在文件尾部，readline()返回的一个空字符串可以作为哨兵值。
- Python中采用readline()方法的end-of-file循环模式：

```
line = infile.readline()
while line != "":
    # 处理每一行
    line = infile.readline()
```





# 文件循环代码while

■ 将end-of-file哨兵循环应用到平均数问题的代码：

```
# average6.py
def main():
    fileName = input("What file are the numbers in? ")
    infile = open(fileName, 'r')
    sum = 0.0
    count = 0
    line = infile.readline()
    while line != "":
        sum = sum + eval(line)
        count = count + 1
        line = infile.readline()
    print("\nThe average of the numbers is", sum / count)
main()
```





# 嵌套循环

- 决策和循环互相嵌套可以实现复杂算法
- 之前实例中文件每行只存一个数字，这一次数字以逗号分隔出现在文件的同一行上
- 下面是处理一行的代码片段：

```
for xStr in line.split(","):
    sum = sum + eval(xStr)
    count = count + 1
```

# 嵌套循环代码

```
# average7.py
def main():
    fileName = input("What file are the numbers in? ")
    infile = open(fileName, 'r')
    sum = 0.0
    count = 0
    line = infile.readline()
    while line != "":
        # 为line中的值更新其count和sum
        for xStr in line.split(","):
            sum = sum + eval(xStr)
            count = count + 1
        line = infile.readline()
    print("\nThe average of the numbers is", sum / count)
main()
```

外循环：while语句对每行循环一次

内循环：for语句对一行中每个数字进行循环