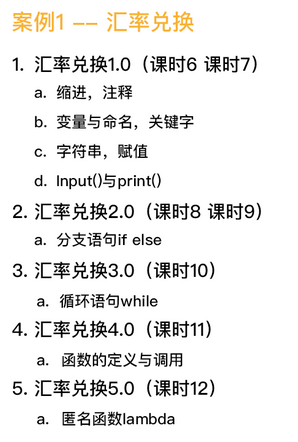
零基础Python入门



# 汇率兑换4.0

## 案例描述

1.设计一个汇率换算器程序，其功能是将外币换算成人民币，或者相反

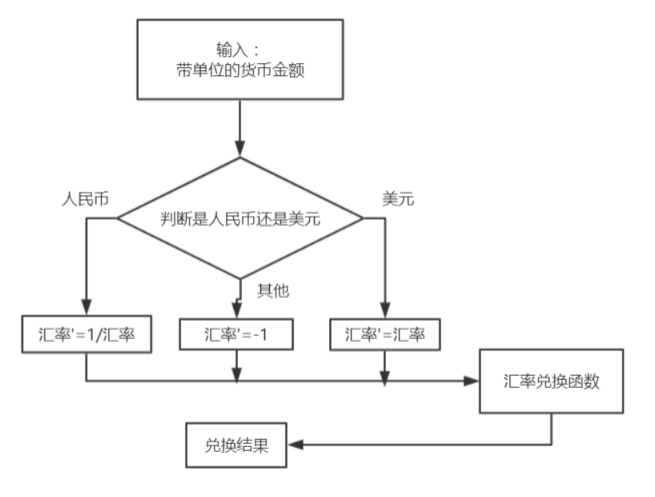
2.为了使程序简单，目前只考虑一种外币（如：美元）

3、2.0增加功能：根据输入判断是人民币还是美元，进行相应的转换计算

4、3.0增加功能：程序可以—直运行，直到用户选择退出

5、4.0增加功能：将汇率兑换功能封装到函数中

## 案例分析





## 上机实验

"""

功能：汇率兑换

版本：4.0

2.0 新增功能：根据输入判断是人民币还是美元，进行相应的转换计算

3.0 增加功能：程序可以一直运行，直到用户选择退出

4.0 增加功能：将汇率兑换功能封装到函数中

"""

**def** convert\_currency**(**im**,** er**):**

"""

汇率兑换函数

"""

out **=** im **\*** er

**return** out

# 汇率

USD\_VS\_RMB **=** 6.77

# 带单位的货币输入

currency\_str\_value **=** input**(**'请输入带单位的货币金额：'**)**

unit **=** currency\_str\_value**[-**3**:]**

**if** unit **==** 'CNY'**:**

exchange\_rate **=** 1 **/** USD\_VS\_RMB

**elif** unit **==** 'USD'**:**

exchange\_rate **=** USD\_VS\_RMB

**else:**

exchange\_rate **=** **-**1

**if** exchange\_rate **!=** **-**1**:**

in\_money **=** eval**(**currency\_str\_value**[:-**3**])**

# 调用函数

out\_money **=** convert\_currency**(**in\_money**,** exchange\_rate**)**

**print(**'转换后的金额：'**,** out\_money**)**

**else:**

**print(**'不支持该种货币！'**)**

## Python语言的基本语法

### 函数

1. 函数

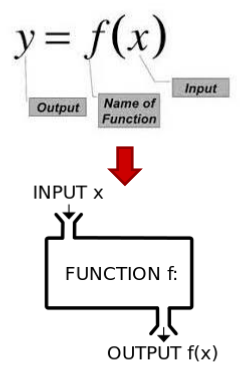
• 将特定功能代码编写在—个函数里

• 便于阅读和复用

• 对—组表达特定功能表达式的封装

• 使程序模块化

• Python内置函数 input(), print(), eval() ...



1. 函数的定义

def<函数名＞（＜参数列表＞）：

＜函数体＞

return<返回值列表＞

1. 函数的调用过程

• 调用程序在调用函数处暂停执行

• 调用时将参数（实参）赋值给函数的参数（形参）

• 执行函数体

• 返回函数结果，回到调用处继续执行

## 习题

1/3、【单选题】 下列不是使用函数的优点的是：

A:

减少代码重复

B:

使程序更加模块化

C:

使程序便于阅读

D:

代码量较大

解析:答案选D。函数可以减小代码量

2/3、【单选题】调用函数时输入的参数叫做形参。

A:

正确

B:

错误

解析:答案选B。调用函数时输入的参数叫做实参，编写函数时输入的参数叫做形参。

3/3、【单选题】函数编写完成之后可以直接运行。

A:

正确

B:

错误

解析:答案选B。函数需要调用才可执行。