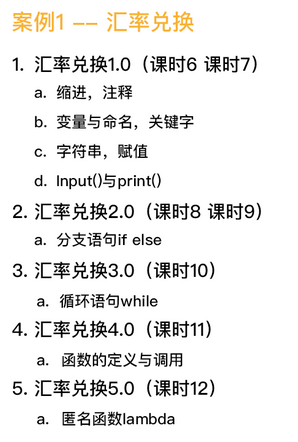
零基础Python入门



# 汇率兑换5.0

## 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）

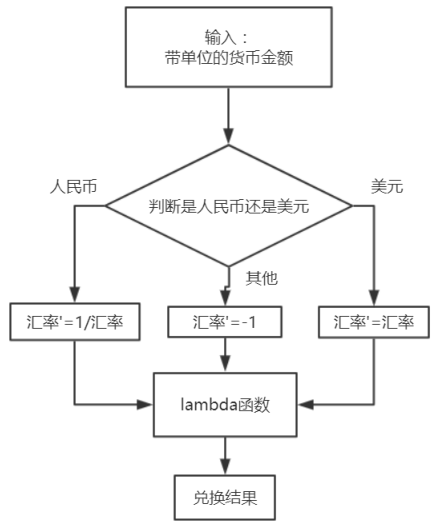
3、2.0增加功能：根据输入判断是人民币还是美元，进行相应的转换计算

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

6、5.0增加功能：(1) 使程序结构化 (2) 简单函数的定义 lambda

## 案例分析





## 上机实验

"""

功能：汇率兑换

版本：5.0

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

5.0 增加功能：(1) 使程序结构化 (2) 简单函数的定义 lambda

"""

# def convert\_currency(im, er):

# """

# 汇率兑换函数

# """

# out = im \* er

# return out

**def** main**():**

"""

主函数

"""

# 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**])**

# 使用lambda定义函数

convert\_currency2 **=** **lambda** x**:** x **\*** exchange\_rate

# # 调用函数

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

# 调用lambda函数

out\_money **=** convert\_currency2**(**in\_money**)**

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

**else:**

**print(**'不支持该种货币！'**)**

**if** \_\_name\_\_ **==** '\_\_main\_\_'**:**

main**()**

## Python语言的基本语法

### lambda函数

• 特殊函数—匿名函数

• 使用方法：

＜函数名> =lambda<参数列表＞：＜表达式＞

• 用于简单的能够在—行内表示的函数，计算结果为返回值

### 什么样的函数能够改成lambda函数

一般函数：

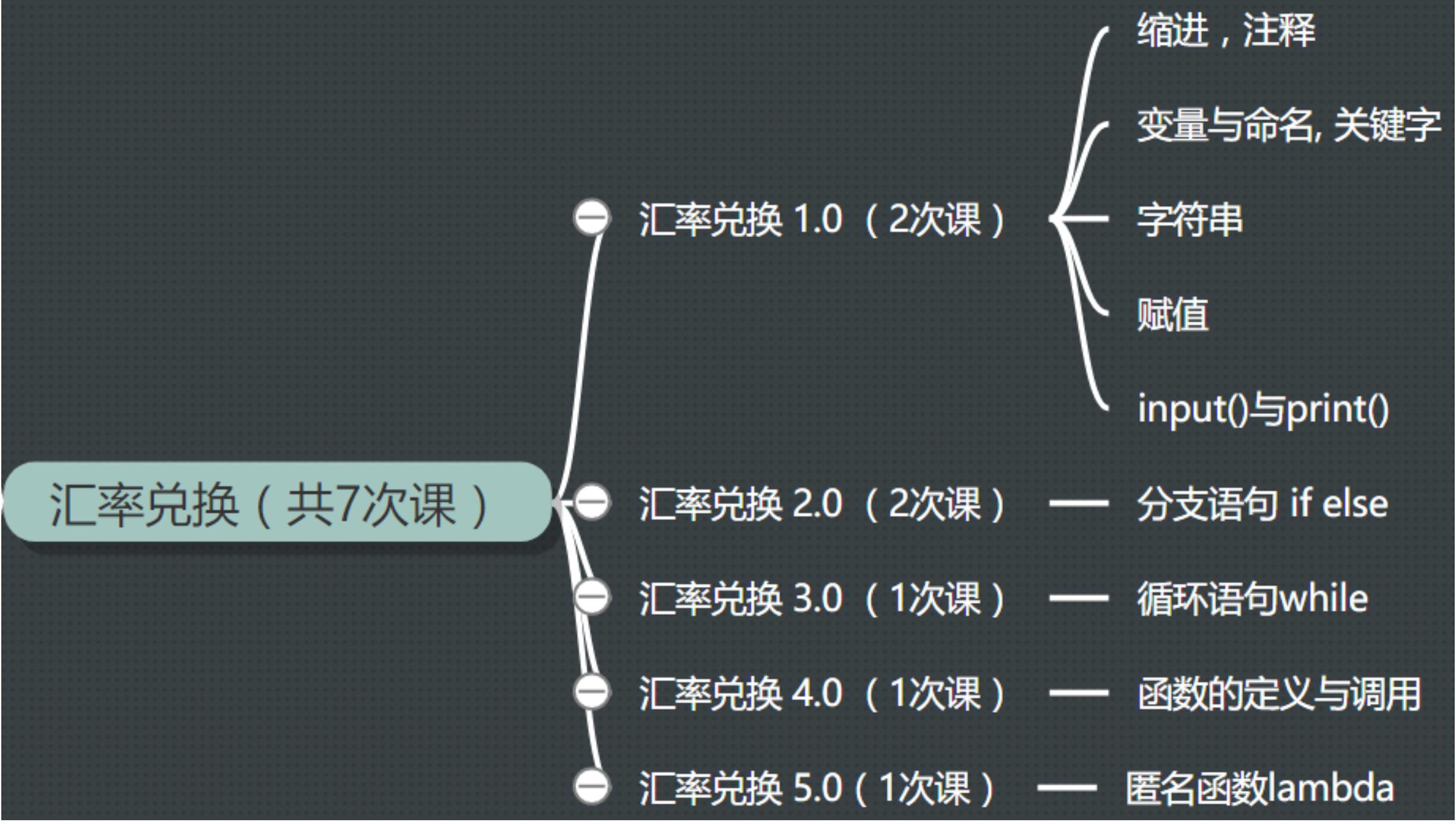
def<函数名＞（＜参数列表＞）：

＜函数体＞

return<返回值列表＞

一般函数的函数体只有一行语句（或者把多行语句改写成一行语句）的才能改写成lambda函数

## 小结



## 习题

1/3、【单选题】‘\_\_name\_\_’是python的内置属性。

A:

正确

B:

错误

解析:答案选A。

2/3、【单选题】函数定义时形参不可以为空。

A:

正确

B:

错误

解析:答案选B。函数定义可以不传入参数。

3/3、【单选题】以下调用主函数的代码正确的是（main函数都含有缩进，这里平台原因没有显示）：

A:

if \_\_name\_\_ == '\_\_main\_\_': main()

B:

if \_\_name\_\_ = '\_\_main\_\_': main()

C:

if \_\_name\_\_ == \_\_main\_\_: main()

D:

if \_name\_= \_main\_: main()

解析:答案选A。