4. 定义函数

1.简介

在前面的章节中，我已经解释了：

1.如何去安装MIT-Scheme

2.Scheme解释器是如何对S-表达式求值

3.基本的表操作

在本章中，我会介绍如何自定义函数。正如Scheme是函数式编程语言，你需要通过构建小的函数来构造你的程序。因此，明白如何构造并结合这些函数对掌握Scheme是非常关键的。

因为在前端定义函数非常不方便，因此你通常需要在编译器中编辑好代码，并在解释器中加载他们。

2.如何定义简易的函数并加载它们

你可以使用define来将一个符号与一个值绑定。通过这个操作符，你可以定义比如数，字符，表等任何类型的参数与函数。

让我们使用任意一款编辑器（记事本亦可）来编辑代码片段1中展示的代码，并将它们存储为‘hello.scm’，放置在类似于‘C:\doc\scheme\’文件夹下。如果可能的话，把这些文件放在你在第一章定义的MIT-Scheme默认文件夹下。

代码片段1（hello.scm）

; Hello world as a variable

(define vhello "Hello world") ;1

; Hello world as a function

(define fhello (lambda () ;2

"Hello world"))

接下来，给Scheme解释器下面的命令：

(cd "C:\\doc\\scheme")

;Value 14: #[pathname 14 "c:\\doc\\scheme\\"]

(load "hello.scm")

;Loading "hello.scm" -- done

;Value: fhello

hello.scm就通过这些命令。如果