# 第二章 Python起步

## 2.1 程序输出，print 语句及“Hello World!”

在Python 中未指定返回值的函数会自动返回None，等价于NULL

在交互式解释器中,可使用变量名查看该变量的原始值

交互式解释器是调用repr()函数来显示对象，print 语句调用str()函数显示对象

下划线(\_)在解释器中，表示最后一个表达式的值

print 语句，与字符串格式运算符( % )结合使用，可实现字符串替换功能

如 print "%s is number %d!" % ("Python", 1) #Python is number 1!

符号 >> 用来重定向输出，将输出重定向到标准错误输出：

import sys

print >> sys.stderr, 'Fatal error: invalid input!'

logfile = open('/tmp/mylog.txt', 'a')

print >> logfile, 'Fatal error: invalid input!'

logfile.close()

带逗号的print 语句输出的元素之间会自动添加一个空格

在print 语句中可以使用逗号来抑制自动生成的换行符号

## 2.2 程序输入和raw\_input()内建函数

help(), 如help(raw\_input)

## 2.3 注释

# 单行注释

“”， “”“ ”“” 文档字符串，在模块、类或者函数的起始添加一个字符串

## 2.4 运算符

标准算术运算符

+ - \* / // % \*\*

标准比较运算符

< <= > >= == != <>

逻辑运算符

and or not

3 < 4 < 5 合法且等同于 3 < 4 and 4 < 5, 合理使用括号可增强代码的可读性

## 2.5 变量和赋值

变量名仅仅是一些字母开头的标识符，所谓字母开头，意指大写或小写字母，另外还包括下划线( \_ )， 其它的字符可以是数字，字母， 或下划线

变量名是大小写敏感

Python 是动态类型语言， 也就是说不需要预先声明变量的类型，变量的类型和值在赋值那一刻被初始化。

赋值： = \*= +=

## 2.6 数字

Python 支持五种基本数字类型

􀁺 int (有符号整数)

􀁺 long (长整数)

􀁺 bool (布尔值)

􀁺 float (浮点值)

􀁺 complex (复数)

decimal， 用于十进制浮点数, 非内建函数， import decimal

## 2.7 字符串

使用索引运算符( [ ] ), 第一个字符的索引是 0，最后一个字符的索引是 -1

切片运算符( [ : ] )可以得到子字符串

加号( + )用于字符串连接运算

星号( \* )则用于字符串重复

## 2.8 列表和元组

元组可以看成是只读的列表。通过切片运算( [ ] 和 [ : ] )可以得到子集

## 2.9 列表

aDict = {'host': 'earth', 'port': 80}

aDict.keys()

for key in aDict：

print(key, aDict[key])

## 2.10 代码块及缩进对齐

## 2.11 if 语句

if expression1:

if\_suite

elif expression2:

elif\_suite

else:

else\_suite

## 2.12 while 循环

while expression:

while\_suite

## 2.13 for 循环和range()内建函数

for item in string/list/dic/:

do something

range()内建函数来生成列表

range(3) 等于range(0,3)

for i in range(3):

print(i)

输出为：

0

1

2

range()函数经常和len()函数和enumerate()一起用于索引

foo = 'abc'

for i, ch in enumerate(foo):

print(i, '(d%)' % i)

a (0)

b (1)

c (2)

## 2.14 列表解析(表达式生成列表)

i = [x for x in 'abc'] # i = ['a', 'b', 'c']

squared = [x \*\* 2 for x in range(4)]

sqdEvens = [x \*\* 2 for x in range(8) if not x % 2]

## 2.15 文件和内建函数open() 和 file()

打开文件：

handle = open(file\_name, access\_mode = 'r')#如果 open() 成功， 一个文件对象句柄会被返回。

handle = file(file\_name, access\_mode = 'r')#返回文件对象

handle.close()

'r' 表示读取，默认值

'w' 表示写入

'a' 表示添加

+' 表示读写

'b' 表示二进制访问

对于很大的文件来说， 一次性读取会占用太多的内存

filename = raw\_input('Enter file name: ')

fobj = open(filename, 'r')

for eachLine in fobj:

print eachLine,（每行文本已经自带了换行字符， 如果我们不抑制print 语句产生的换行符号， 文本在显示时就会有额外的空行产生）

fobj.close()

## 2.16 错误和异常

try:

filename = raw\_input('Enter file name: ')

fobj = open(filename, 'r')

for eachLine in fobj:

print eachLine, fobj.close()

except IOError, e:

print 'file open error:', e

## 2.17 函数

函数的参数可以有一个默认值

def addMe2Me(x,debug=True):

'apply + operation to argument'

return (x + x)

## 2.18 类

class ClassName(base\_class[es]):

"optional documentation string"

static\_member\_declarations

method\_declarations

当一个类实例被创建时， \_\_init\_\_() 方法会自动执行， 在类实例创建完毕后执行， 类

似构建函数。它的目的是执行一些该对象的必要的初始化工作

每个方法都有的一个参数， self.它是类实例自身的引用

self.\_\_class\_\_.\_\_name\_\_

## 2.19 模块

import module\_name

module.function()

module.variable

## 2.20 实用的函数

dir([obj]) 显示对象的属性，如果没有提供参数， 则显示全局变量的名字

help([obj]) 以一种整齐美观的形式 显示对象的文档字符串， 如果没有提供任何参

数， 则会进入交互式帮助。

int(obj) 将一个对象转换为整数

len(obj) 返回对象的长度

open(fn, mode) 以 mode('r' = 读， 'w'= 写)方式打开一个文件名为 fn 的文件

range([[start,]stop[,step]) 返回一个整数列表。起始值为 start, 结束值为 stop - 1; start

默认值为 0， step默认值为1。

raw\_input(str) 等待用户输入一个字符串，可以提供一个可选的参数 str 用作提示信息。

str(obj) 将一个对象转换为字符串

type(obj) 返回对象的类型（返回值本身是一个type 对象！）

# 第三章