

第一章

一些知识点

- 通过导入以下模块可以使“/”执行普通除法运算，“//”在任何情形下执行的都是整除运算。

其他运算包括：“*”乘方（pow(x,y)）、“%”取余

```
1 from __future__ import division
```

- python的整数运算没有位数限制
- 八进制数以0开头，十六进制以0x开头
- 获取用户输入input函数和raw_input函数，input函数要求输入的值是合乎python规范的，如输入一个str时应当有双引号。

```
1 x= input("please input x:")
```

- 小数取整

```
1 import math
2 math.floor(30.2) # 30
3 math.ceil(30.2) # 31
4 round(30.2) #30 四舍五入
```

- 复数模块cmath
- str、repr、反引号都个获得对象的字符串表示
- 长字符串 """ long string """
- 原始字符串 r" raw string ",注意结尾不能是“\”

```
1 # 需要反斜线结尾时
2 raw =r"D:\workspace\openstack_cinder_storage\cinder\ruijie_cinder\volume\drive
```

- Unicode字符串 `u"Unicode String"`，Unicode字符串以16位存储，而不是8位

第二章

知识点

通用序列操作

- 索引
- 分片
技巧1：当分片最左边的索引比右边的索引出现的晚时，获得的是一个空序列。
如：最后三个元素 `x[-3:]`，`x[-1:-3]`返回空
技巧2:使用负数步长时，分片最右边的索引比左边的索引出现的晚时，获得的是一个空序列
- 加法
- 乘法
- `in`运算（成员资格检查）
- 内建函数`len`、`min`、`max`

列表

- `list`函数
- 删除元素`del`函数
- 分片赋值，通过分片赋值可以修改、删除、插入`list`中的连续几个元素，通过指定不同的步长可以修改不连续的元素。
- 列表的方法

```
1 # 末尾追加元素
2 x.append(y)
3 # 统计出现次数
4 x.count(y)
5 # 追加另一个序列到末尾,加法操作会返回一个全新的列表。
6 x.extend(y)
7 # 索引元素的首次出现位置
8 x.index(y)
```

```
9 # 插入
10 x.insert(index_num,y)
11 # 删除最后一个源素,使用pop/append可以实现一个栈.
12 x.pop()
13 # 删除首个匹配项
14 x.remove(y)
15 # 反向
16 x.reverse()
17 # 排序
18 x.sort(cmp_func,key=key_func,reverse=False)
```

元组

- 单个元素的元组 (x,)
- tuple函数
- 可以作为字典的键

第六章

知识点

- 函数是否可调用

```
1 hasattr(func,__call__)
```

- 函数文档

```
1 def say_hello():
2     'print hello world'
3     return 'hellp world'
4
5 print say_hello.__doc__
```

- 不可变类型是按值传递的,可变类型传递引用
- 位置参数、关键字参数

- 参数收集

```
1 # 收集多余的参数
2
3 def func(*param,**name_param):
4     print param
5     print name_param
6
7 func(1,2,3,4,key1='a',key2='b')
8
9 """
10 返回值:
11 (1, 2, 3, 4)
12 {'key2': 'b', 'key1': 'a'}
13 """
14
15 # 逆过程
16
17 def func(a,b,key1='a',key2='b'):
18     print a,b
19     print key1,key2
20 param=(1, 2)
21 name_param={'key2': 'b', 'key1': 'a'}
22 func(*param,**name_param)
23
24
```

- 使用vars()函数可以返回全局作用域内包含所有变量的字典
- 使用locals()函数可以返回局部作用域内包含所有变量的字典
- 绑定全局变量

```
1 x=1
2 def change_global():
3     global x
4     x=x+1
5
```

- 嵌套作用域--闭包特性

里层函数可以访问外层函数的作用域.

- 函数工具map、filter、reduce、lambda

第七章

知识点

- 使用新式类，自定义类型继承object对象作为父类。
- 类的方法实际上是绑定self对象，称为绑定方法。
- 成员变量以双下滑线开头时，外部无法访问(被翻译成__classname__paramname)。
- 下滑线开头的方法不会被import语句导入。
- 类的静态变量
- 检查继承 issubclass(Child,Father),通过__bases__可以获取父类,通过__class__返回类类型
- 获取方法、属性:getattr (a,'param_name/func_name')
- 判断方法、属性是否存在:hasattr (a,'param_name/func_name')
- 添加方法、属性:setattr (a,'param_name/func_name','value')
- __dict__属性存储所有方法,属性

第八章

知识点

- 内建异常模块exceptions
- 常见的内建异常

表8-1 一些内建异常

类 名	描 述
Exception	所有异常的基类
AttributeError	特性引用或赋值失败时引发
IOError	试图打开不存在文件（包括其他情况）时引发
IndexError	在使用序列中不存在的索引时引发
KeyError	在使用映射中不存在的键时引发
NameError	在找不到名字（变量）时引发
SyntaxError	在代码为错误形式时引发
TypeError	在内建操作或者函数应用于错误类型的对象时引发
ValueError	在内建操作或者函数应用于正确类型的对象，但是该对象使用不合适的值时引发
ZeroDivisionError	在除法或者模除操作的第二个参数为0时引发

- 不带参数的raise能将捕获的异常向上传递
- 一次捕获多个异常

```
1 try:
2     pass
3 except (ExceptionType1,ExceptionType2) as e:
4     pass
```

- 不为except指定任意的异常类时，能够捕获到所有异常
- try/except /else/finally

```
1 try:
2     pass
3 except:
4     pass
5 else:
6     print "如果没有捕捉到异常，执行这个语句"
7 finally:
8     print "无论是否发生异常都会执行"
```

第九章

知识点

- 析构方法__del__，在垃圾回收的时候被调用。

- 使用super调用父类的方法，注意这里会调用一个超类的方法，这和继承顺序有关
- 基本的序列和映射接口：__len__、__getitem__、__setitem__、__delitem__
- 使用UserList、UserString、UserDict
- 使用property(getFunc,setFunc,delFunc,docfunc)指定函数的一个属性
- 静态方法、类方法，两者调用方式相同

```

1 class A(object):
2     @staticmethod
3     def say(self):
4         print "A"
5     @classmethod
6     def play(cls):
7         print "cls被绑定到类对象上"

```

- __getattr__(self,name)当特性name被访问时自动被调用
- __getattr__(self,name)当特性name被访问，且对象没有这个特性时被调用
- __setattr__(self,name,value)当试图给特性name赋值时会被自动调用
- __delattr__(self,name,value)当试图给特性name进行删除时会被自动调用
- __iter__迭代器接口，迭代器指具有next方法的对象，如果没有接下来的值则返回StopIteration异常
- iter函数可以返回可迭代对象的迭代器
- 生成器，任何包含yield语句的函数是生成器，调用生成器函数会返回一个迭代器

第十章

知识点

- 多次导入同一个模块，只在第一次导入时执行相应的代码，reload函数可以将一个模块重新导入。
- 在被导入的模块中，__name__的值为模块名
- python解释器的模块搜索路径 sys.path，存放自定义模块的最佳目录，site-packages

- 添加python环境变量

```
1 export PYTHONPATH=$PYTHONPATH:~/linqing
```

- 包是另一类模块，包含一个"__init__.py"文件，导入一个包时该文件内的代码块被执行。
- dir函数可以将模块的所有函数、类、变量的所有特性列出
- __all__定义了模块的所有公共接口
- __file__定义了模块源代码的位置
- 常用标准库sys

表 10-2 sys模块中一些重要的函数和变量

函数/变量	描 述
argv	命令行参数，包括脚本名称
exit([arg])	退出当前的程序，可选参数为给定的返回值或者错误信息
modules	映射模块名字到载入模块的字典
path	查找模块所在目录的目录名列表
platform	类似sunos5或者win32的平台标识符
stdin	标准输入流——一个类文件（file-like）对象
stdout	标准输出流——一个类文件对象
stderr	标准错误流——一个类文件对象

- 常用标准库os

表10-3 os模块中一些重要函数和变量

函数/变量	描 述
environ	对环境变量进行映射
system(command)	在子shell中执行操作系统命令
sep	路径中的分隔符
pathsep	分隔路径的分隔符
linesep	行分隔符('\n', '\r', or '\r\n')
urandom(n)	返回 <i>n</i> 字节的加密强随机数据

- sets模块包含Set类
- heapq模块包含堆heappop/heappush函数可以实现小端堆，heapify将任意一个列表转换成一个堆，nlargest/nsmallest用来寻找可迭代对象中第*n*大或第*n*小的元素
- collections模块能够实现双端队列
- 常用标准库time

表10-7 time模块中重要的函数

函 数	描 述
asctime([tuple])	将时间元组转换为字符串
localtime([secs])	将秒数转换为日期元组，以本地时间为准
mktime(tuple)	将时间元组转换为本地时间
sleep(secs)	休眠（不做任何事情）secs秒
strptime(string[, format])	将字符串解析为时间元组
time()	当前时间（新纪元开始后的秒数，以UTC为准）

表10-6 Python日期元组的字段含义

索 引	字 段	值
0	年	比如2000, 2001, 等等
1	月	范围1~12
2	日	范围1~31
3	时	范围0~23
4	分	范围0~59
5	秒	范围0~61
6	周	当周一为0时，范围0~6
7	儒历日	范围1~366
8	夏令时	0、1、或-1

。标准模块random

表10-8 random模块中的一些重要函数

函 数	描 述
random()	返回 $0 \leq n < 1$ 之间的随机实数 n ，其中 $0 < n \leq 1$
getrandbits(n)	以长整型形式返回 n 个随机位
uniform(a, b)	返回随机实数 n ，其中 $a \leq n < b$
randrange([start], stop, [step])	返回range(start, stop, step)中的随机数
choice(seq)	从序列seq中返回随意元素
shuffle(seq[, random])	原地指定序列seq
sample(seq, n)	从序列seq中选择 n 个随机且独立的元素

- 标准模块shelve，该模块能够实现简单的函数序列化
- 标准模块re

第十一章

知识点

- open函数的文件模式

表11-1 open函数中模式参数的常用值

值	描 述
'r'	读模式
'w'	写模式
'a'	追加模式
'b'	二进制模式（可添加到其他模式中使用）
'+'	读/写模式（可添加到其他模式中使用）

- 文件对象的方法，write/read，readline/readlines，writelines
- 使用sys.stdin、sys.stdout、sys.stderr进行读写
- 操作系统换行符os.linesep
- 文件在程序退出以后会自动关闭，但是一定要及时关闭。
- 通过with语句自动关闭文件
- 使用fileinput模块

```
1 import fileinput
2 for line in fileinput.input(filename):
3     process(line)
4 # fileinput只会读取需要加载的部分
```

第十五章

知识点

- 参考：Beautiful Soup的使用<http://cuiqingcai.com/1319.html>
- tidy模块：用来修复不规范的HTML文档的工具
- XHTML和HTML的最主要区别，前者对于显式关闭所有元素要求更加严格。
- HTMLParser库：用来解析标准XHTML的工具类。
- CGI 通用网关接口技术
- PSP：Python Server Page HTML和Python代码混合的

第十六章

知识点