摘自Python入门经典——William F.Punch & Richard Enbody（已阅读完）

编程规则：

1. 编写程序前要深思熟虑
2. 程序应该是具有可读性的短文。
3. 实践是提高编程能力和问题解决能力最好的办法。

如果你解决不了问题，那么就解决一个更简单的问题：去发现问题。--Polya

如果想插入一条多行的长注释，考虑使用三引号字符串。只需要在注释的开始和结束使用引号，这和#不同，他不必每行使用。

字符串：索引和分片

连接+ 和重复\* 如mystring\*3

strip方法用于除去字符串的开头和结尾与参数相同的符号。

如’,this.’.strip(‘,.’)得到this

result=’this is a test’.split()

result=[‘this’,’is’,’a’,’test’]

result=’field,field2’.split(‘,’)

result=[‘field’,’fields’]

‘:’.join([‘a’,’b’,’c’])得到’a:b:c’

sorted函数（只能用于列表上）

python维护命名空间，其中变量名与对象关联，称为“引用”，即变量名“引用”对象。（C++中也有）

元组：基本上是不可改变的列表，几乎具有列表所有的特征。

调用python函数时，参数和其对应的实参之间都复制了什么？答案是关联，复制的是引用关系，而不是对象本身。

虽然实参和形参最初引用了相同的对象，但在重新赋值后，形参的引用发生了变化，引用一个新的对象。实参则不会发生改变，保持原来的引用。

如果被引用的对象是不可变的（如数字，字符串，元组）,那么对象在函数运行前后不会做任何改变。

那如果传递的是可变的对象如列表，那么则会改变原来的实参值。

传递可变对象使得函数可以在主调程序中更改值。

事实上，Python不指定参数的类型，使得对可变或者不可变参数的处理比较复杂，影响可读性。

在Python中定义函数即创建一个对象。

与函数相关联的值包括：

\_\_name\_\_ 与函数相关联的名称

\_\_str\_\_ 用于构造函数字符串函数

\_\_dict\_\_函数的命名空间

\_\_doc\_\_文档字符串。

“传值”或者“传引用”

在从结构化语言派生出的编程语言（C++,JAVA,C#），将参数区分为“传值”或是“传引用”。

那么Python采用的是哪一种方式？

对于新手而言，有用的答案是“两者都不是”；但最好的答案：Python传递是对象引用。在Python中一切皆为对象，所以每个值的传递都是对象的引用。对象在这个过程中是不可复制的。如果对象是可变的，在函数中所做的更改将反应在对象的外部作用域中。如果对象是不可改变的，当更新引用时将引用新的对象。

深拷贝与浅拷贝：

只复制引用，而不复制对象本身的情况是浅拷贝。要复制内容，而不是简单复制引用的，则称为深拷贝。

字典和集合：实际上Python在很大程度上依赖于其基础组织结构中的字典。

字典是可变的。

字典方法：

items()

keys()

values()

copy(): 浅拷贝

集合：通过调用集合的函数来创建，这和其他数据类型结构不同。

set函数。

集合是可变的。

交集：aset.intersection(bset)

并集：aset.union(bset)

差集：aset.difference(bset)

对称差：aset.symmetric\_difference(bset)

子集：small.issubset(big)

超集：bigset.issuperset(small)

add(element)

clear()

remove(element)

copy():浅拷贝

命名空间是名称和对象之间的关系。将命名空间想象为字典，其中的键是名称，值是对象。命名空间不共享名称。

Python变量范围规则：

1. 本地L
2. 封闭E
3. 全局G
4. 内置B

文件

CSV文件是一种文件格式，用一种特殊方法来进行信息编码，并存储在计算机文件中。使用CSV格式编码的文件用于存储表格数据。

python 提供了CSV模块。(这一块很有用)

模块：OS（Operating System）(这一块很有用)

和math模块相同，os模块也是Python标准模块。

os.getcwd()

os.chdir(‘’)

os.listdir()

os.path.isfile()

os.path.isdir()

os.path.join()

os.path.walk()

type函数返回对象的类型。

if-elif-else

计算机科学观点：最小的通用计算

我们需要多少语言才能够写出任意的计算机程序？

为了计算出任何的内容，所需要的最小指令集合是多少？

结论是汇编语言（针对CPU）

最小指令集：

LODA A:

STORE A:

CLR

INC

BRZ X:如果累加器为0，跳转到RAM中X地址。

所有计算机程序都可以只用这5条指令完成。所有程序，任何程序。

Python是面向对象的

类和实例

类是建立新对象的模板，利用类模板所创建的对象称为该类的实例。

在Python中任何内置的数据结构都定义为类，列表、字符串、集合、元组、字典都是类。使用构造函数或快捷方式，能生产这些类的实例。

类也是对象；每个类、每个实例（实际上Python的一切）都是对象。

在类和实例之间存在着一种特殊的关系。这种关系称为“instance-of”关系。

类命名空间中定义的名称同样适用于该类的所有实例。类在某种程度上充当了全局存储库的角色，有所有的实例适用和共享。但是这种关系不是双向的。但是程序员可以在实例中定义来“覆盖”类的全局定义。

特殊参数：self

方法是类实例的接口

类对象的方法定义了对象能做什么，同时定义了对象的接口。也就是说，类实例和用户的交互是通过的方法来定义的。

Python类模型

Python标准方法：\_\_

例如：\_\_init\_\_; \_\_init\_\_没有显示的return语句，因为\_\_init\_\_是创建实例过程中的一部分，所以Python自动处理返回。

\_\_str\_\_生成字符串，同时返回字符串。

**在python中，在实例生产后，可以对它进行修改，程序员可以为实例添加新属性，甚至在实例的命名空间中添加方法。这不同其他的OOP语言，在JAVA,C++中，实例生成后，用户就不能修改。**

类设计者是特殊的程序员。程序员只能通过类设计者提供的接口认识类。程序员只知道接口，而不知道类的工作细节。这就是封装的目的。设计良好的类可以作为其他程序员使用的库，为这些程序员提供了抽象的功能单元已解决问题。

OOP三个概念：

1. 封装性
2. 继承性
3. 多态性：是指使用运算符（+），根据他们的对象类型，执行不同 的操作过程。

Python和其他OOP语言：

公有和私有：私有属性只提供给类设计者；而公共属性提供给每个人使用。

Python不强制进行类设计者和程序员之间的区分。所有方法和实例变量都是公共的。Python为类设计者提供了双下划线\_\_，来表示哪些属性程序员不能直接修改。每当类设计者在属性的名字前面加双下划线时，如果要在类外面使用该属性，Python会将属性名字进行变形，但是这种名称的变化并不能阻止程序员访问属性值，如用dir即可看到新的名字。它只是提供了模糊层。Python的宗旨“我们都是成年人”，如果说类设计者表示某个东西是私有的，那么程序员就不要冒险违反规则了。

使用from… import \*,则不必使用类名就能导入方法。如fracClassl.Rational(1,2)可以简化为Rational(1,2);

python自检函数：查询对象的类型

type(someVariable)

isinstance(someVariable, somtType)返回布尔值

自定义运算符重载：python提供了可重载的特殊运算符集合，只能重载该集合中的运算符。

重载+后，var1+var2即为var1.\_\_add\_\_(var2)

var2+var1即为var2.\_\_add\_\_(var1),两者有区别。

异常和异常处理（具体程序见D:Myphon\Try\_Except）.

所有的输入都是邪恶的，除非被证实不是。 —Michael Howard& David LeBlanc

1,有关异常的哲学：

1. Easier to Ask Forgiveness than Permission(EAFP)
2. Look Before you Leap(LBYL)

2,异常：else和finally

无论是否一场，在执行try-except程序快时，都会执行finally语句。

try程序块没有遇到一个错误，则执行else程序。

3，异常的用法

1. 检查输入
2. 检查文件打开

4，深入异常

1. raise
2. 自定义异常：Python中的异类是类。要创建异常，就必须创建一个异常类的子类，通过继承，将异常类的所有基本特点都保留下来。如：

class NewException ( Exception):

pass

测试

1.混合测试

（1）用异常捕捉用户错误

（2）用assert捕捉开发者犯的错误

Python中assert函数有两个参数。

2.自动测试

python提供了一些工具来帮助自动运行测试代码。

1. doctest 例：见D:\MyPython\doctest
2. 如果想看到所有测试运行的细节，可以调用命令行代码，并将‘-v’开关放在该行的末尾。

递归：另一种控制机制

有一类语言称为功能性的编程语言：Haskell, Scheme, Lisp,就是将递归作为语言的主要控制机制。

* 1. 递归如何实现

1. 栈的数据结构

计算机对函数调用的跟踪是以栈这种数据结构来实现的。

python有现成的栈类stackList.

Python特点（自己总结）：

1. 类：

1不同C++/Java等有成员变量和成员函数之说，Python类中定义的只有成员函数，Python的对象属性是可以随意添加并改变的。（和javascript一样）

2在类定义中的变量相当于C++/Java中的static变量，尽管没有static关键

3不同C++/Java/JS,Python中创建一个类实例不需要用new这个关键字，只需a = Person(“John”)即可

1. 函数：

1由于Python是弱数据类型的语言，所以函数中不需要加上变量的类型，只需形参变量名即可。(和javascript一样)

2Python函数可以返回多于1个的数据

如 def getCoords(self):

return self.x,self.y

xc,yc = cp.move(dist)

摘自《Python技术参考大全》

如果仅仅处理文本文件，用Perl;

要转化、翻译或封装数据到新的数据存储格式时用Python

需要在shell中过滤文本时用Awk

Perl的根本应用领域还是文本处理，包括Internet中处理CGI

Perl的缺陷是面向对象是作为完成语言开发工作后加入的。如果想在完全面向对象环境下工作并利用OO特征尤其是继承性和多态性，应选择Python.

JavaScript和Python在设计方面很相似，但目标不一样。

摘自《Python编程金典》

进程管理（一章已经看完，对应的代码见Ubuntu的my\_code/Python）

在多处理系统上，每个进程都可在单独的处理器中运行，从而实现真正的“并发性”