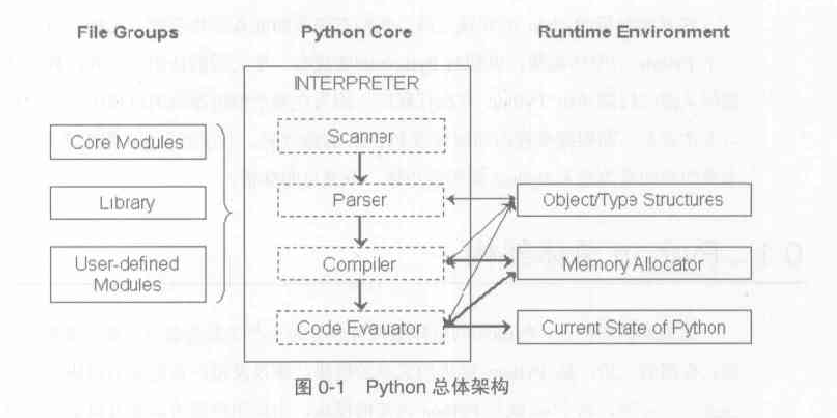
0.1 Python 总体架构

Python的整体架构可以分为三个主要部分。



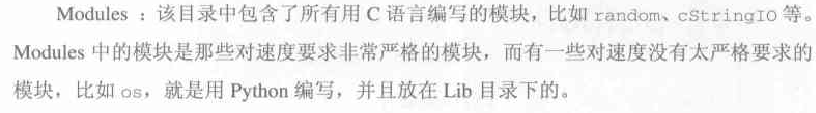
0.2 Python源代码的组织

Python源码可以从官网(http://www.python.org)下载

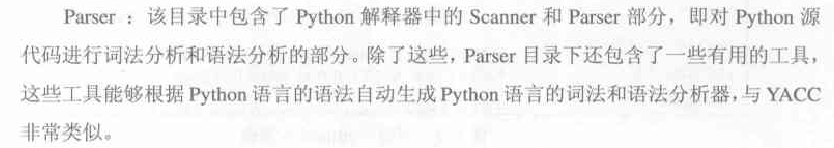
Include:包含了Python提供的所有头文件，如果用户需要自己用C或C++来编写自定义模块扩展Python，那么就需要用到这里提供的头文件。

Lib: 包含了Python自带的所有标准库，都是用Python语言编写的

Modules:



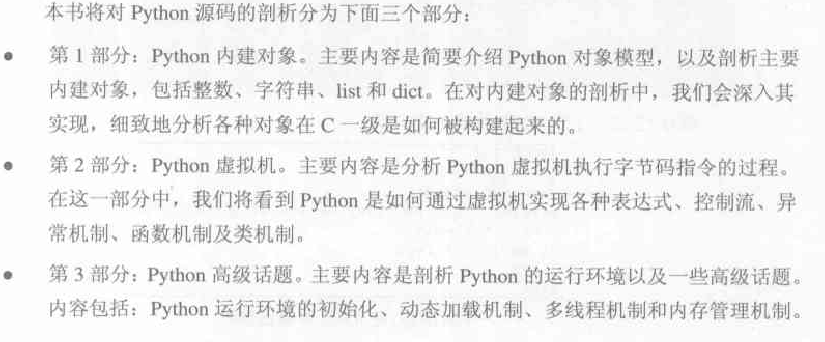
Parser:



Objects: 该目录中包含了所有Python的内建对象，包括整数，list,dict等。同时，该目录还包括了Python在运行时需要的所有的内部使用对象的实现。

Python: 该目录下包含了Python解释器中的Compiler和执行引擎部分，是Python运行的核心所在。

PCBuild:包含了VS的工程文件，研究Python源代码从这里开始



**第一部分**

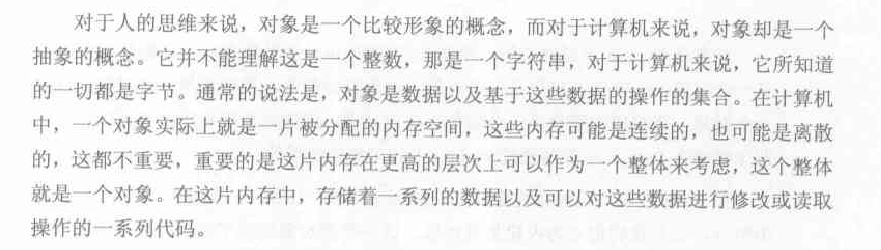
**Python内建对象**

**Python对象初探**

Python允许程序员通过class A(object)这样的表达式自己定义类型对象。

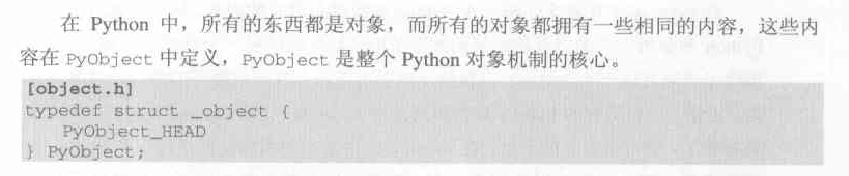
因为python是由C实现的，所以我们首先要弄清楚的一个问题就是：对象，这个神奇的东西，在C的层面，究竟长得是个什么模样

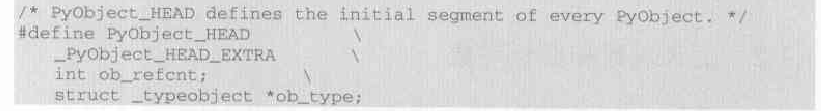
Python内的对象



在Python中，一个对象一旦被创建，它在内存中的大小就是不变的了。这意味着那些需要容纳可变长度数据的对象只能在对象内维护一个指向一块可变大小的内存区域的指针。

1.1 对象机制的基石——PyObject





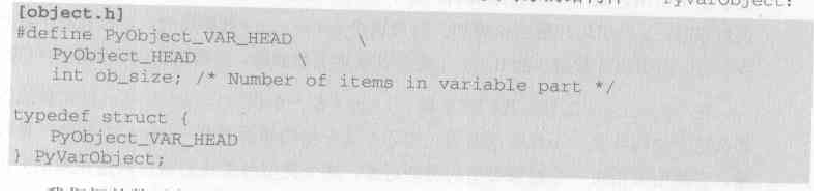
在PyObject的定义中，整型变量ob\_refcnt与python的内存管理机制有关，它实现了基于引用计数的垃圾收集机制。

ob\_type是一个指向\_typeobject结构体的指针，该结构体对应着Python内部的一种特殊对象，它是用来指定一个对象类型的类型对象。

在PyObject中定义了每一个Python对象都必须有的内容。

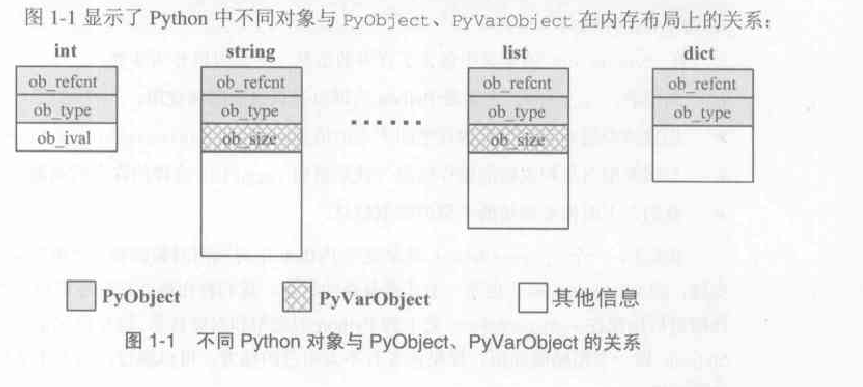
1.1.2 定长对象和变长对象

定长对象的不同对象占用的内存大小是一样的。



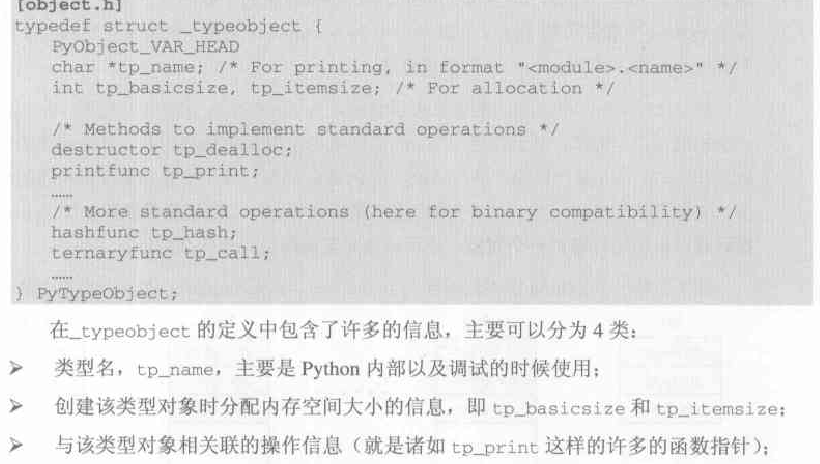
结构PyVarObejct 中的ob\_size指明了变长对象中一共容纳了多少个元素。比如，对于Python中最常用的list,它是一个PyVarObject对象。

PyVarObeject实际上只是对PyObject的一个扩展而已，我们只需要用一个PyObject指针就可以引用任意的一个对象



1.2 类型对象

占用内存空间的大小是对象的一种元信息，这样的元信息是与对象所属类型密切相关的，因此它一定会出现在与对象所对应的类型对象中



1.2.1 对象的创建

创建对象方法，第一种是通过Python C API来创建，第二种是通过类型对象PyInt\_Type。

C API，让用户可以从C环境中与Python交互。Python的C API分成两类，一类称为范型的API,这类API具有诸如PyObject\_\*\*\*的形式，可以应用在任何Python对象上。

另一类是与类型相关的API。这类API只能作用在某一种类型的对象上。

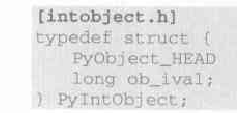
不论采用哪种C API，Python内部最终都是直接分配内存，因为Python对于内建对象是无所不知的，但对于用户自定义的类型，如class A(object),对于这种情况，Python会通过A所对应的类型对象创建实例对象。

1.2.2 对象的行为

第二章

Python中的整数对象

2.1 初识PyIntObject对象



Python中的整数对象PyIntObject 实际上就是对C中原生类型long的一个简单包装。