|  |  |
| --- | --- |
| **Learn Python .4th Edition** |  |
| **Chapter 2 Python 如何运行程序** |  |
| **Python是一个解释器软件包，解释器将读取程序，并按照其中的命令执行得出运算结果。实际上解释器是代码与机器的计算机硬件之间的软件逻辑层。**  **Python安装在系统上后，包含了最小化的组件：一个解释器和支持的库（Library）。** | **Python解释器本身由C程序实现** |
| **【字节码编译】**  **在把语句输入到文本后，Python 从头到尾按顺序一个一个运行文件中的语句。**  **当程序执行时，Python Interpreter 将源代码编译成字节码，字节码与平台无关，运行速度快。程序被编译成字节码，然后被发送到执行引擎Python　Virtual　Machine（PVM）。Python　Code　没有编译阶段，均发生在运行阶段。** | **.pyc 能够加速启动，如果Python无权限在操作系统写入字节码文件，将在内存中生成，退出程序时丢弃。** |
| **Chapter３如何运行程序** |  |
| **模块导入是一种运行代码文件的方法，模块是Python程序最大的结构。模块是变量的封装，每个文件都是一个完备的命名空间。** |  |
| **Chapter４Introduction Python Object Types** |  |
| **Python Programs are composed of modules.**  **Modules contain statements.**  **Statements contain expressions.**  **Expressions create and process objects.** |  |
| **Built-in Objects preview[内置对象]**  **Python 数据是一个个对象。**  **[ Numbers、String、Lists、Dictionaries、Tuples、Files、Sets ]**  **Other Core type: Boolean、types、None**  **Program unit types: Functions、modules、classes、pattern**  **Implementation-related types: Complied code、stack tracebacks**  **运行一个表达式，生成一个特定的对象。对象分为可变和不可变对象。**  **一个操作符的意义取决于被操作的对象。**  **调用dir函数，返回一个列表，其中包含了对象所有的属性。Help函数查询对象的使用方法。** | **用于多种类型的通用操作都是以内置函数或表达式的形式出现。如len(X),X[1]。**  **特定类型的的操作是以方法调用的形式出现的。**  **String.upper()** |
| **Set 是不可变的对象的集合，元素唯一。集合可迭代，集合只能包含不可变的对象类型，因此字典和列表不能嵌入到集合里。** | **初始化空集合S = set(),而不是{}** |
| **当变量出现在表达式中，它会被当前引用的对象所替代。变量在赋值的时候才创建，它可以引用任何类型的对象，并且在引用之前赋值。在内存中，变量和对象保存在不同的部分，从变量到对象叫引用。对象有两个标准头信息，包含从技术上说，一个类型标志符去标识这个对象的类型，以及一个引用计数器，以决定此对象是否可以回收。** |  |
| **Chapter7 字符串** |  |
| **字符串是一个有序的字符集和，用来表现文本信息，如名字、内存中的文本文件内容、Internet 地址、Python程序。Python字符串比C字符数组更高层。**  **常见的字符串常量和表达式：**  **S= ’ ’**  **S=”Spaam”**  **S = r‘’’ Raw string Here ‘’’**  **支持操作：分片、合并和索引**  **一个字符叫做一个字节，len()返回字符串的字节数。** | **Python字符串不可变，即各个单字符从左往右的位置顺序和自身都不能被改变。** |
|  |  |