

目录

第 1 章 入门准备

01 开篇词：你为什么要学 Python ?

02 我会怎样带你学 Python ? 最近阅读

03 让 Python 在你的电脑上安家落户

04 如何运行 Python 代码 ?

第 2 章 通用语言特性

05 数据的名字和种类—变量和类型

06 一串数据怎么存—列表和字符串

07 不只有一条路—分支和循环

08 将代码放进盒子—函数

09 知错能改—错误处理、异常机制

10 定制一个模子—类

11 更大的代码盒子—模块和包

12 练习—密码生成器

第 3 章 Python 进阶语言特性

13 这么多的数据结构（一）：列表、元祖、字符串

14 这么多的数据结构（二）：字典、集合

15 Python大法初体验：内置函数

16 深入理解下迭代器和生成器

17 生成器表达式和列表生成式

18 把盒子升级为豪宅：函数进阶

19 让你的模子更好用：类进阶

20 从小独栋升级为别墅区：函数式编

02 我会怎样带你学 Python?

更新时间：2019-08-20 18:18:24



“受苦的人，没有悲观的权利。”
——尼采”

为什么需要程序语言

市面上程序语言很多，且各有不同，但无论你用什么语言来写程序，有一点你需要知道，那就是——计算机实际运行的是计算机指令序列。这些指令序列的直接表现形式是一连串的 0 和 1，也就是机器码。机器码不属于程序语言，所以程序的执行是与语言无关的，计算机在运行时感知不到你的语言。

既然计算机程序的执行和语言无关，那么为什么要搞出来语言这种东西来？

显然用机器码的 0 和 1 这两个字符来编程是困难和低效的。为了提高生产效率，我们需要一种容易书写和阅读的标记来表达机器码指令。使用方式是用这种标记进行程序编码，编码完成后再将这此标记翻译为机器码。最初达到这个目的是汇编语言，汇编语言一种是程序语言，不过是低级程序语言，它和机器码之间有着比较直接的一一对应关系。

扩展：这种「比较直接的一一对应关系」怎么理解呢？例如，你甚至可以通过查表的方式，得出汇编语言 `MOV AX,66H` 对应机器码的十六进制 `B866`，也就是二进制 `1011100001100110`。

所以汇编 `MOV AX,66H` 与机器码 `1011100001100110` 相对应。这个结果也可以反向推导。

十六进制怎么转二进制？有了 Python 就不要去手工算了，执行 Python 代码 `bin(int('B866', 16))` 轻松得到结果。

← 慕课专栏	≡ 你的第一本Python基础入门书 / 02 我会怎样带你学 Python ?
目录	如 Fortran，Pascal，以及当下广泛使用 C，Java，Python 等。之所以被称为高级程序语言，是因为它们相较于机器码和汇编语言有着更高级的抽象，同时不再与机器码有明显的对应关系。
第 1 章 入门准备	扩展：计算机的世界里「抽象」这次词表示屏蔽掉内部纷繁复杂的细节，直接触达其核心功能，为外部建立起更高效更易用的使用入口和操作方法。
01 开篇词：你为什么要学 Python ?	在之后的文章中「抽象」这个词会被频繁提及，因为它是计算机技术里非常重要的且具有指导意义的一个概念。
02 我会怎样带你学 Python ? 最近阅读	
03 让 Python 在你的电脑上安家落户	
04 如何运行 Python 代码？	
第 2 章 通用语言特性	为什么会有这么多高级语言？每种语言都因为有不同的使用理念和应用场景而被设计出来。例如：
05 数据的名字和种类—变量和类型	<ul style="list-style-type: none">• C 语言有非常高的执行效率，和马马虎虎的开发效率，在一些极度依赖执行效率和底层控制的领域独占鳌头，比如操作系统开发。• Java 借助于 JVM 得以跨越不同的操作系统和计算机架构而运行 Java 程序。Java 也拥有良好的执行效率和稳定性，它的执行效率与 C 语言相比，最高能保持在同一个数量级上。• Python 与 C 和 Java 相比有较大的不同，Python 是解释型语言，没有显式编译的过程，代码可直接由解释器解释执行，同时借助于解释器，其代码也可以跨平台运行。另一方面它是动态语言，在一定程度上更加灵活。Python 的理念是优雅和简单，虽然执行效率相较于 C 和 Java 大约慢一到两个数量级，不过 Python 的开发效率却数倍高于它们。
06 一串数据怎么存—列表和字符串	
07 不只有一条路—分支和循环	
08 将代码放进盒子—函数	
09 知错能改—错误处理、异常机制	
10 定制一个模子—类	
11 更大的代码盒子—模块和包	
12 练习—密码生成器	
第 3 章 Python 进阶语言特性	程序语言特性的层次
13 这么多的数据结构（一）：列表、元祖、字符串	各种语言有如此多的不同之处，那它们的本质是什么呢？从本质上来看，这些程序语言都是机器码或者汇编语言的高级抽象，抽象出来的各种语言特性，可以以更高效易用的方式去表达计算机的指令。
14 这么多的数据结构（二）：字典、集合	语言核心特性
15 Python大法初体验：内置函数	完整地表达计算机的指令只会用到语言的一部分特性，在这里我们把这部分特性叫作是核心特性。虽然每种语言都有着不同的表达方式（语法规则），但对于核心特性的表达却有着相近的模式，可以说它们是适用于大多数语言的通用特性。核心特性有：
16 深入理解下迭代器和生成器	<ul style="list-style-type: none">• 数据类型• 数值运算和比较• 变量定义、赋值• 数组、字符串• 分支跳转• 循环• 函数
17 生成器表达式和列表生成式	语言扩展特性
18 把盒子升级为豪宅：函数进阶	除上述必要的核心特性之外，为了让语言有更强的表达力和易用性，每种语言都有自己的一些扩展特性。如在 Python 中常用的有：
19 让你的模子更好用：类进阶	
20 从小独栋升级为别墅区：函数式编程	

<div>← 慕课专栏</div> <div>≡ 你的第一本Python基础入门书 / 02 我会怎样带你学 Python ?</div>	
目录	<ul style="list-style-type: none">• 模块与包• 内置数据结构• 内置函数• 迭代器• 生成器• 装饰器• 函数式特性• 动态类型• 以及其它常用语言特性
第 1 章 入门准备	
01 开篇词：你为什么要学 Python ?	
02 我会怎样带你学 Python ? 最近阅读	
03 让 Python 在你的电脑上安家落户	
04 如何运行 Python 代码？	
第 2 章 通用语言特性	<div>说明：虽然异常机制、类、模块与包是语言扩展特性，但它们是 Python 中的十分常用且无法回避的特性，所以本专栏在章节组织时将这三者放入语言核心特性章节中来作介绍。</div>
05 数据的名字和种类—变量和类型	
06 一串数据怎么存—列表和字符串	
07 不只有一条路—分支和循环	
08 将代码放进盒子—函数	
09 知错能改—错误处理、异常机制	
10 定制一个模子—类	
11 更大的代码盒子—模块和包	
12 练习—密码生成器	
第 3 章 Python 进阶语言特性	<ul style="list-style-type: none">• 标准库• IO 操作• 进程线程• 序列化• 代码规范• 以及运行环境相关，如 Python 解释器、虚拟环境，Java 的 JVM 等
13 这么多的数据结构（一）：列表、元祖、字符串	
14 这么多的数据结构（二）：字典、集合	
15 Python大法初体验：内置函数	
16 深入理解下迭代器和生成器	
17 生成器表达式和列表生成式	
18 把盒子升级为豪宅：函数进阶	
19 让你的模子更好用：类进阶	
20 从小独栋升级为别墅区：函数式编	

语言周边知识和功能

除了核心特性和扩展特性以外，语言一般还会整合一些语言特性之外的功能，或关联一些语言相关的知识，在这里我们将它们称为语言周边知识和功能。如常见的：

这些周边知识和功能严格来讲不属于语言特性的范畴，但却和语言紧密相关，其中有些十分有用甚至不可或缺。它们通常是操作系统、框架、运行环境相关的内容。

语言特性有层次之分，学习也有侧重点

上面的内容是为了让大家对程序语言有了一个全盘的认识，这样在学习时才能做到有的放矢。那具体该如何来学习 Python 呢？

首先，核心特性是学习重点对象，这部分内容也不多。只要掌握了语言的核心特性，就可以编写程序来完成所有的计算功能了。通用特性要是没掌握好，后续的学习如空中楼阁。

其次，语言扩展特性的学习多多益善。语言拓展特性很大程度上增强了语言的表达力，使得在开发中能大幅缩减代码量提升开发效率。但是扩展特性很繁杂，有很多细枝末节的东西，有些在开发时并不一定被用到，所以不要求把一门语言的拓展特性完全掌握，学习时只需掌握其中常用的部分，其它部分在以后的日常使用中逐渐积累即可。

最后，语言的周边知识和功能虽不属于语言特性的范畴，但是对其中一些知识的学习是有必要的。如标准库和 IO 操作中封装有大量的系统调用，为程序与操作系统（或系统外部）的交互提供了入口。另外标准库中也包含大量高效稳定的现成代码，省却了使用者造轮子的功夫。

其实无论是学习 Python 还是学习其它语言，认清语言特性的层次，然后有所侧重地学习，这才是高效掌握一门新语言的不二法门。我们也将以此为脉络来为大家开展后续的 Python 课程。

目录	
第 1 章 入门准备	精选留言 0
01 开篇词：你为什么要学 Python ?	
02 我会怎样带你学 Python ? 最近阅读	欢迎在这里发表留言，作者筛选后可公开显示
03 让 Python 在你的电脑上安家落户	
04 如何运行 Python 代码？	<div>!</div> <div>目前暂无任何讨论</div>
第 2 章 通用语言特性	
05 数据的名字和种类—变量和类型	
06 一串数据怎么存—列表和字符串	千学不如一看，千看不如一练
07 不只有一条路—分支和循环	
08 将代码放进盒子—函数	
09 知错能改—错误处理、异常机制	
10 定制一个模子—类	
11 更大的代码盒子—模块和包	
12 练习—密码生成器	
第 3 章 Python 进阶语言特性	
13 这么多的数据结构（一）：列表、元祖、字符串	
14 这么多的数据结构（二）：字典、集合	
15 Python大法初体验：内置函数	
16 深入理解下迭代器和生成器	
17 生成器表达式和列表生成式	
18 把盒子升级为豪宅：函数进阶	
19 让你的模子更好用：类进阶	
20 从小独栋升级为别墅区：函数式编	