高等教育质量工程信息技术系列示范教材

Python 大学教程

张基温 编著

清华大学出版社

高等教育质量工程信息技术系列示范教材

Python大学教程

张基温 编著

清华大学出版社
北京

内 容 简 介

本书是高等学校 Python 基础课程的教材。全书由 7 个单元组成。第 1 单元介绍 Python 的基本知识，内容包括 Python 的特点、数据对象、变量、输入输出等，使读者对 Python 有一个初步了解；第 2 单元为 Python 程序结构，内容包括选择结构、循环结构、函数、模块、异常处理等；第 3 单元为容器，内容包括序列容器、无序容器、迭代器、生成器与推导表达式；第 4 单元为面向类的程序设计，内容包括类与对象、类与对象的通用属性与操作、类的继承；第 5 单元为 Python 数据处理，内容包括文件操作、数据库操作、文件与目录管理；第 6 单元为 Python 网络编程，内容包括 Python Socket 编程、Python WWW 应用开发；第 7 单元为 Python GUI 开发，内容包括 GUI 三要素、GUI 程序结构、GUI 制作示例。

本书力求内容精练、概念准确、代码便于阅读、习题丰富全面、适合教也容易学。为了便于初学者很快能使用以丰富的模块支撑的 Python 环境，书后给出了 Python 运算符、Python 内置函数、Python 标准模块库目录和 Python 3.0 标准异常类结构。

本书适合作为高等学校零基础开设 Python 课程的教材，也适合作为程序设计爱好者和有关专业人员学习的参考书。

图书在版编目（CIP）数据

Python 大学教程/张基温编著. —北京：清华大学出版社，2018
（高等教育质量工程信息技术系列示范教材）
ISBN 978-7-302-50454-2

Ⅰ. ①P…　Ⅱ. ①张…　Ⅲ. ①软件工具– 程序设计– 高等学校– 教材　Ⅳ. ①TP311.561

中国版本图书馆 CIP 数据核字（2018）第 128338 号

责任编辑： 白立军
封面设计： 常雪影
责任校对： 焦丽丽
责任印制： 丛怀宇

出版发行： 清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn,010-62795954

印 装 者： 北京亿浓世纪彩色印刷有限公司
经　　销： 全国新华书店
开　　本： 185mm×260mm　　**印　张：** 20.75　　**字　数：** 493 千字
版　　次： 2018 年 9 月第 1 版　　**印　次：** 2018 年 9 月第 1 次印刷
定　　价： 69.00 元

产品编号：077037-01

前　言

（一）

在多年从事 C 语言、C++和 Java 教学工作中，少不了有学生要求解释如 fun(i ++，i ++)这样的问题。有时，到外校进行学术交流时，也不乏同行教师问到这个问题。我感觉，能问到这个问题的学生，无疑是好学生。因为，这个问题虽小，但要解释清楚它，需要涉及非定义行为、赋值表达式的副作用、序列点、程序设计风格等方面的概念，这些概念在相关教材中几乎不见提到，许多教师也不清楚。更让我吃惊的是，当我给一位从事了 30 多年 C 与 C++教学的大学副教授讲赋值表达式的副作用时，他竟然回了我一句:“我不这样认为。”这足以说明问题的严重性了。

实际上，与其说是赋值操作的副作用，不如说是“变量”的副作用。这似乎是一个不可逾越的鸿沟。因为“值的变化”是变量的基本性质。然而，这个问题在 Python 中被解决了，因为它的数据多数属于不可变类型。对于不可变类型的变量赋值，就成为引用指向另外一个对象了。这确实是 Python 的一大突破。Python 有许多让人耳目一新的特点，正是这些特点，使它得到了快速推广，并赢得广泛的支持。

2017 年 7 月 19 日，IEEE（美国电气电子工程师学会）出版的旗舰杂志 *IEEE Spectrum* 发布了第 4 届顶级编程语言交互排行榜。这个排行榜由读者需求、用户增速、开源、设计自由度、雇主需求 5 个子排行榜组成。其中，前 4 个子排行榜中都是 Python 力压群雄，只有雇主需求一榜位于 C 和 Java 之后，排名第三。图 1 为 *IEEE Spectrum* 2017 编程语言 Top 10 排名情况。

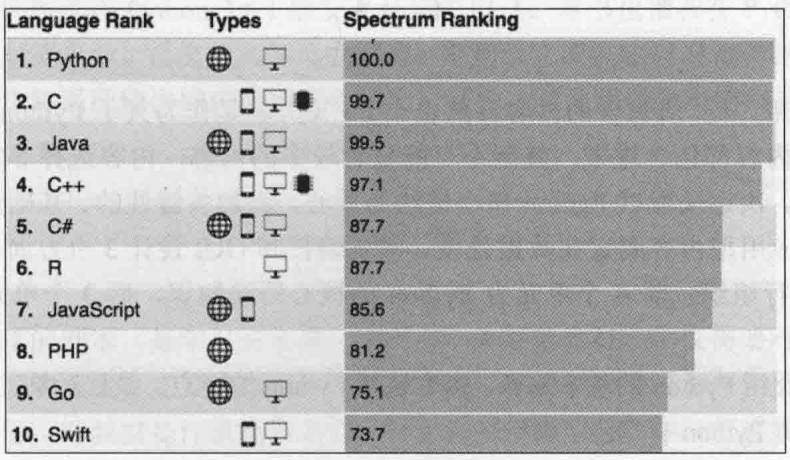

图 1　*IEEE Spectrum* 2017 编程语言 Top 10 排名情况

据 *IEEE Spectrum* 介绍，这个排行依据数据记者 Nick Diakopoulos 提供的数据，结合 10 个线上数据源的 12 个标准，对 48 种语言进行了排行。因为不可能顾及每一个程序员的想法，*IEEE Spectrum* 使用多样化、可交互的指标权重来评测每一种语言的现行使用情况。显然，这个排行的客观性、权威性是极高的。

另一个影响极大的程序设计语言排行榜是 TIOBE。TIOBE 排行榜是根据互联网上有经验的程序员、课程和第三方厂商的数量，并使用搜索引擎（如 Google、Bing、百度）以及 Wikipedia、Amazon、YouTube 统计出排名数据，但只是反映某个编程语言的热门程度，并不能说明一门编程语言好不好，也不反映就同一算法使用不同语言编写时代码数量多少。表 1 是其 2017 年 10 月发布的 Top 10 榜单。可以看出，Python 虽位居第 5，但它有上升趋势，而前 4 名均有下降趋势。

表 1　TIOBE 2017 年 10 月发布的程序设计语言 Top 10 榜单

2017 年 10 月	2016 年 10 月	变化	编程语言	评级/%	变化/%
1	1		Java	12.431	-6.37
2	2		C	8.374	-1.46
3	3		C++	5.007	-0.79
4	4		C#	3.858	-0.51
5	5		Python	3.803	+0.03
6	6		JavaScript	3.010	+0.26
7	7		PHP	2.790	+0.05
8	8		Visual Basic .NET	2.735	+0.08
9	11	↑	Assembly language	2.374	+0.14
10	13	↑	Ruby	2.324	+0.32

（二）

Python 应用广泛，所包含的内容自然也十分广泛。但是作为关于 Python 的基础教程，不可能把所有内容都包含进来，甚至不可能包含较多的内容，内容选择非常重要。作者经过反复斟酌，决定采取以 Python 核心语法为重心，添加关键性的、基础性的应用型内容。最后，将应用型内容圈定在数据处理、网络编程和 GUI 设计 3 个方面，并把全书按照 7 个单元进行组织。前 4 个单元为 Python 的核心语法知识，后 3 个单元为 3 个应用方面。

第 1 单元介绍 Python 的基本知识，内容包括 Python 的特点、数据对象、变量、输入输出等，使读者对 Python 有初步了解。

第 2 单元为 Python 程序结构，内容包括选择结构、循环结构、函数、模块、异常

处理。

第 3 单元为容器，内容包括序列容器、无序容器、迭代器、生成器与推导表达式。

第 4 单元为面向类的程序设计，内容包括类与对象、类与对象的通用属性与操作、类的继承。

第 5 单元为 Python 数据处理，内容包括文件操作、数据库操作、文件与目录管理。

第 6 单元为 Python 网络编程，内容包括 Python Socket 编程、Python WWW 开发。

第 7 单元为 Python GUI 开发，内容包括 GUI 三要素、GUI 程序结构、GUI 制作示例。

著名心理学家皮亚杰创建的结构主义，把教师的主要职责定义为为学习者创建学习环境。作为 Python 教材，本书把附录和习题作为正文之外的两个重要的学习环境。本书的附录包括 Python 运算符、Python 内置函数、Python 标准模块库目录和 Python 3.0 标准异常类结构。

除了语言的内核和内置函数，模块是 Python 的最大支撑。在 Python 中，每一项应用都要由相应的模块支持。每一个应用程序的开发都需要按照“熟悉领域知识—导入相关模块—设计相应算法—编写相应代码”的过程。由于 Python 开源代码的特点和社区广大热心者的支持，目前 Python 已经有上千的模块可以利用。读者知道哪些模块可用，不仅可以开阔思路，而且可以浏览这些模块目录得到通向该应用领域的线索。不过，要把这些模块全罗列出来不仅没有必要，也没有可能。这是将 Python 3.0 标准模块库目录作为附录的原因。虽然仅仅只有 29 项，但足可以对 Python 的应用范围画出一个轮廓。

习题也是重要的学习环境。为此本书收集并设计了多种类型的习题，并且在每节后面都给出相应的练习题。本书习题量虽多，却还是无法满足不同的练习需要。希望学习者和使用本书的老师们，不要囿于本书给出的习题，要开发出更多课后练习，开辟更好的 Python 学习环境。还需要说明的是，不是每一个题目都能直接在正文中找到答案。要找到正确的答案，需要深刻理解基本概念，或需要自己设计一些代码测试分析。这样才能培养出举一反三的能力、创新的能力。

本书所有例题都在 Python 3.6.1 的交互环境中调试。本书也推荐在 Python 3.0 的交互环境平台上教学或自学，在交互式环境中学习，有利于立即发现错误和理解错误原因。为便于阅读，文中将系统输出的内容用蓝色印出。其中，蓝色粗体为出错信息（在 IDLE 中是红色）。

（三）

教材是教学的剧本，是学习的向导。要编写一本好的教材，不仅需要对本课程涉及内容有深刻的了解和感悟，还要熟悉相关领域的知识，更要不断探讨和深化贯穿其中的教学理念和教育思想，写教材是件很难的事情。特别是在不断的写作中，常感到自己知识和能力的不足。由于是已经有了一些想法才开始写作的，又不忍将这些想法隐藏起来，还由于

已经得到一些亲朋的支持和鼓励，也不忍辜负他们的一片热情，只能硬着头皮写下去，也幸有他们的帮助，才最后得以完成本书。在本书的写作过程中，赵忠孝教授、姚威博士、张展为博士，以及魏士婧、刘砚秋、张秋菊、史林娟、张有明、戴璐、张展赫、吴灼伟（插图）等参加了有关部分的编写工作，在此谨表谢意。

本书就要出版了。它的出版，是我在这项教学改革工作中跨上的一个新台阶。本人衷心希望得到有关专家和读者的批评与建议，也希望能多结交一些志同道合者，把这本书改得更好。

张基温

丁酉菊月于穗小海之畔

目　录

第1单元　Python起步

1.1　程序设计语言与Python

1.1.1　计算机程序设计语言

程序（program）是关于问题求解、任务执行的可执行描述，通常由一组操作指令组成。用计算机进行问题求解的可执行描述，就称为计算机程序。计算机程序通常由一组计算机指令组成。为了便于设计与应用，人们用一套符号系统描述计算机的指令系统，这套用符号描述的指令系统称为计算机程序设计语言（computer programming language）。

计算机程序设计语言随着应用的拓展得到了快速发展，迄今已经形成4个层次：机器语言（machine language）、汇编语言（assembly language）、高级程序设计语言（high-level programming language）和脚本语言（scripting language）。

1. 机器语言

电子数字计算机用电子开关作为基本元件。这些元件一般只能有开/关、高电平/低电平两种状态特征。这两种状态可以形式化地表示为0和1，并进一步形成数字、文字、图像、图形、声音以及指令的编码。一种CPU的所有指令，就组成了其指令系统。在一台计算机上执行的程序，就是由这台计算机指令系统的指令组成的。所以，一台计算机的指令系统也称为这台计算机的机器语言。由于每种CPU的设计目标和技术手段不相同，所形成的指令组合规则、可以执行的指令种类和数量各不相同，所以，不同的CPU具有不同的机器语言。

用机器语言编写程序，不仅要思考如何用计算机的基本操作组成解题程序，还要熟悉机器的指令系统。此外，早期的计算机程序就是用图1.1所示的穿孔纸带通过光电设备输入到计算机的。程序员编写完程序，还要花费大量时间进行纸带穿孔，并像图1.2中两名科学家那样检查有没有把孔打错的地方。当时的程序规模不大，花费在编程上的时间不算太多，但在熟悉语言、穿孔纸带检查上却要花费大量时间。由于这些原因，机器语言仅适用于早期机器速度较慢、程序较为简单的情况。

图1.1　穿孔纸带

图1.2　科学家在检查穿孔纸带

2. 汇编语言

随着计算机速度的加快，应用范围扩大，人们开始考虑把纸带穿孔的简单而又烦琐的工作，交给已经在电报传送中使用的电传打字机完成。图 1.3 为由电传打字机控制的纸带穿孔机。后来，又用电传打字机把程序直接输入计算机的磁鼓和磁带存储器中，大大提高了程序输入的效率和可靠性。但是，电传打字机输入的是字符信息。为了与之匹配，人们开始把每条指令都用字符代替，建立起与机器指令相对应的一套助记符（mnemonics）。例如，把一条用 0、1 码表示的加指令操作用 add 表示，把一条用 0、1 码表示的减指令操作用 sub 表示，把一条用 0、1 码表示的跳转指令操作用 jump 表示等。这样，在编写程序时就比看一串 0、1 码方便多了，不仅容易理解，也容易记忆、检查错误。正如 Microsoft 公司的一名高级经理 Nathan Myhrvoid 所说："软件是一种可以膨胀到充满整个空间的气体。"软件生产的加速，无疑成为计算机进步和应用拓展的强劲动力。

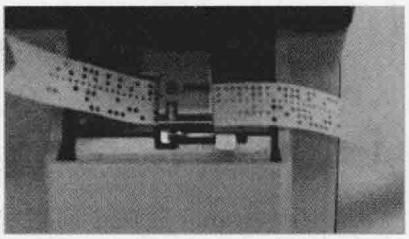

图 1.3　电传打字机控制的纸带穿孔机

这种用助记符代替机器语言所编出来的程序是机器不能直接辨认和理解的。为了让 CPU 理解和执行，还必须将这些助记符按照指令为单位进行一一代真。人们把这种代真过程称为汇编，把完成汇编过程的软件称为汇编程序（assembler），把这些由助记符组成的编程语言称为汇编语言（assembly language）或符号语言。

说汇编语言方便了程序设计和输入，仅仅是相对而言，它还存在两大遗憾：一是汇编语言还是因机器而异，不能通用，用一种 CPU 的汇编语言编写出来的程序，不能用于其他 CPU；二是若要换一种 CPU，就必须再去熟悉另一种 CPU 的指令系统。

3. 高级程序设计语言

为了屏蔽因机器不同而造成的程序设计障碍，让人能直接用人类积累起来的普适科学思维进行程序设计，也为了能使程序设计者方便地与领域的专家和用户交流，以提高程序的正确性，人们开始考虑如何用贴近自然语言（主要是英语）的符号系统编写程序，并相对于那些依赖于机器的程序设计语言，将它们称为计算机高级程序设计语言，简称高级语言。

最早的高级语言尝试，是德国人楚泽（Konrad Zuse，1910—1995）于 1945 年为他的 Z-4 计算机设计的 Plan Calcul。楚泽与美国贝尔实验室的研究员乔治·斯蒂比兹（George Robert Stibitz，1904—1995）几乎同时研发成功世界上最早的继电器式计算机，并奠定了数字计算机的雏形。图 1.4 为楚泽和他的继电器式计算机。

图 1.4　楚泽和他的继电器式计算机

在电子计算机上实现的第一个高级语言是美国尤尼法克公司于 1952 年研制成功的 ShortCode；而真正得到推广使用，至今仍在流行的第一个高级语言是

由美国的计算机科学家巴科斯设计，并于 1956 年首先在 IBM 公司的计算机上实现的 FORTRAN 语言。从此，高级语言犹如雨后春笋般涌现出来，并快速从科学计算扩散到商业、行政管理等多个行业。图 1.5 为 2001 年之前的高级语言发展情况。

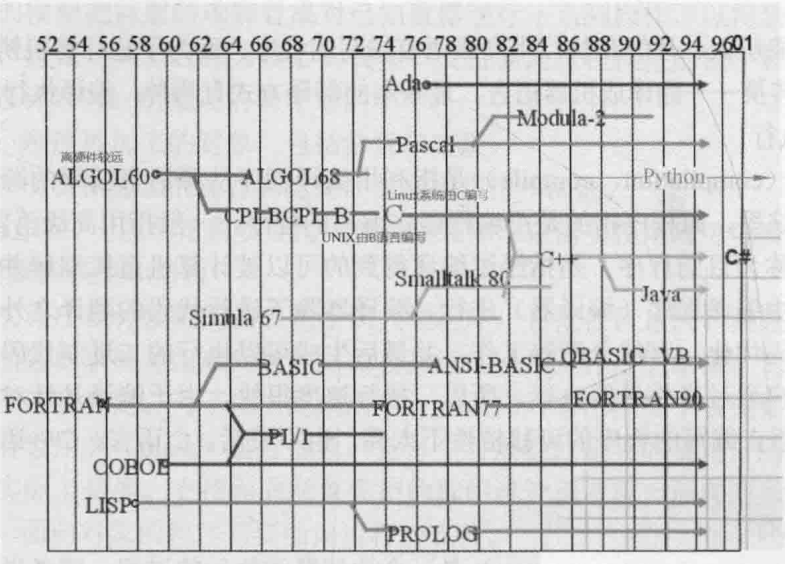

图 1.5　2001 年之前的高级语言发展状况

高级语言接近自然语言，但不能直接使用自然语言，因为自然语言会受情景、情绪、知识、修养等多种因素影响，而呈现表达的多样性，或理解的多样性。为了让机器理解，必须进行各种限制，使之规范化。所以，每一种高级语言都有自己的词法、语法和语义规则。

4. 脚本语言

脚本一般指戏剧表演、电影拍摄等所依据的底本或者书稿的底本。在程序中，脚本就是一条条的文字命令，它们可以由应用程序临时调用执行，而不需要传统的编写→编译→连接→运行（edit→compile→link→run）过程。早期的脚本语言经常被称为批处理语言或工作控制语言，例如 UNIX 的 shell 可以像胶水一样，将 UNIX 下的许多功能连接在一起。WWW 技术出现以后，人们又用一些脚本语言编写客户端脚本程序，用于说明网页的有关内容应当以什么格式表现。这种脚本不仅可以减小网页的规模和提高网页的浏览速度，而且可以丰富网页的表现，如动画、声音等。常见的脚本语言有 HTML、Scala、JavaScript、VBScript、ActionScript、MAXScript、ASP、JSP、PHP、SQL、Perl、Shell、Python、Ruby、JavaFX、Lua 和 AutoIt 等。

1.1.2　高级程序设计语言分类

不同的人，为了不同的应用目的，基于不同的开发背景，所开发出来的高级语言是不同的。这就造就了种类繁多的高级语言。据有人统计，在 2013 年时，计算机高级编程语言总数已经超过 600 种。

可以从不同角度对众多的高级语言进行分类。下面仅从与本书有关的角度介绍几种分类。

1. 编译与解释

计算机不能辨认、理解和执行用高级语言编写的程序，而为了让计算机辨认、理解和执行，必须将其转换——翻译成机器语言。最基本的翻译方式有两种：编译执行和解释执行。

1）编译执行

程序编译（compilation，compile）是指利用编译程序从源语言编写的源程序产生目标程序的过程。这里，源程序指的是用编程语言编写的程序，一般指用高级语言编写的程序；目标程序（也称为目的程序）是指经过编译得到的可以被计算机直接理解并执行的机器代码集合。编译由编译程序（编译器）进行。编译器除了进行代码的翻译之外，还进行语法错误检查、代码优化、存储分配等工作，并最后生成可以执行的二进制代码文件保存，在需要时才由用户通过操作系统执行。所以，执行速度很快。由于编译是针对具体的机器进行的，用这种语言编写的程序的可移植性不太高，也不灵活。C 语言、C++语言都是编译型程序设计语言。

2）解释执行

程序解释（program interpretation）是一个边解释边执行的过程，或者说是解释一句，执行一句，即翻译与执行不是分开的，而是流水式进行的。如果把编译比作书面翻译，那么解释就像是口头翻译。程序解释由解释器进行。为了能边解释边执行，解释器不仅要有一个解释模块，还需要一个执行模块。也就是说，执行的控制权不在用户程序而在解释器。由于要边解释边执行，所以执行速度比较慢。但是比编译式程序灵活性好，可移植性高，占用的空间也大。BASIC 语言、脚本语言都是典型的解释型程序设计语言。

3）半编译半解释

Java 语言采用的是半编译半解释的方式。半编译是指它不直接编译成目标代码，而是由编译器先编译成一种中间的二进制代码——字节码文件。在需要执行时，再由目标机器中的虚拟机对字节码文件进行解释并执行。这样一种方式，在前一段的编译中，可以解决代码优化、错误检查等工作，并且由于源代码与字节代码都与具体机器无关，所以获得了较好的可移植性和效率；在后一段，虚拟机只对字节代码进行解释执行，任务轻多了。这样，既保持了编译的优点，又兼具了解释的好处，实现了 Java 开发初期预计的“一次编译，到处执行”的承诺，从而适应了网络环境下的跨平台开发需求。此外，把可执行程序放在虚拟机环境中运行，虚拟机可以随时对程序的危险行为，比如缓冲区溢出、数组访问过界等进行控制，可以提高安全性。

2. 面向过程与面向对象

程序的实质可以描述为程序=模型+表现，即程序的灵魂是模型（model），程序设计语言就是用来描述解题模型的。模型是人的意识的表达，它可以借助实体表达，也可以借助虚拟表达。模型也有很多分类因素，如按表现形式分类、按产品属性分类、按技术分类、按材料分类、按用途分类、按学科领域分类。

在程序设计领域，从表现形态上，人们已经构造出两大类模型：面向过程（procedure oriented）的模型和面向对象（object oriented）的模型。

1）面向过程的模型和语言

面向过程的模型把问题的求解看成对已知数据进行一系列操作，以得到目标数据的过程。简单地说，它是一类以过程为核心的模型。支持面向过程模型的程序设计语言称为面向过程的程序设计语言。所描述的程序元素如下。

（1）数据：程序被加工的对象，包括常量和变量。

（2）操作：对数据施加的操作和运算，包括操作符和函数。

（3）函数/过程：一段经常需要使用的代码封装体，在需要使用时可以直接调用。

2）面向对象的模型和语言

面向对象的程序设计模型的初衷是组织大型程序，它把任何系统和问题都看成是一组对象的运动。问题的求解是组成问题的对象自身运动或相互作用的结果。每个对象都用属性、行为进行描述和界定。为了便于组织对象，在面向对象的模型中首先要关注对象的分类，并将一类对象用类（class）进行抽象，把每个对象看成是类的实例。所以，面向对象的模型的核心实际上是类。支持面向对象模型的程序设计语言称为面向对象的程序设计语言。一般说来，面向对象的程序需要面向过程的支持。

3. 数据类型

数据是程序处理的对象。为了规范化数据的存储和计算，现代程序设计语言都要求程序中的数据必须属于某种类型。但是不同的程序设计语言在对待类型的处理上遵循了不同的原则，采用了不同的形式。归纳起来，可以分为强类型（strong type）与弱类型（weak type）、静态类型（static type）与动态类型（dynamic type）。这些概念一般是针对变量而言的。

1）强类型与弱类型

鉴别一种程序设计语言是强类型还是弱类型的标准，是看它是否允许类型方面的 untrapped errors（出错后继续执行）。如果一旦发现类型出错就不可继续执行，就是强类型程序设计语言；反之，就是弱类型程序设计语言。

强类型程序设计语言的优点如下。

（1）编译时刻能检查出错误的类型匹配，以提高程序的安全性。

（2）可以根据对象类型优化相应运算，以提高目标代码的质量。

（3）减少运行时刻的开销。

强类型程序设计语言的缺点是灵活性差。弱类型与之相反。

2）静态类型与动态类型

鉴别一种程序设计语言是静态类型还是动态类型的标准，是看它对于类型错误的检查是在什么时间进行的：静态类型语言的数据类型检查仅在编译时进行，动态类型语言的数据类型检查在运行期间进行。

静态类型语言可以分为两种。

（1）如果类型是语言语法的一部分，则是显式类型（explicitly typed）。

（2）如果类型通过编译时推导，则是隐式类型（implicitly typed）。