

# C

language is very simple

# 语言其实 很简单

张宁 编著



最接地气的C语言图书 拥有3个最显著的特点 那就是

**不用啃，不用啃，真的不用啃！**

LANGUAGE IS **VERY SIMPLE**

清华大学出版社

# C language is very simple

# 语言其实 很简单

○ 张宁 编著



清华大学出版社  
北京

## 内 容 简 介

本书是为零基础的 C 语言初学者量身定做的，特别适合非计算机专业的读者自学 C 语言。本书尽量避免使用专业术语，利用大量贴近生活的实例，用通俗易懂的方式讲解 C 语言的基本概念和基本编程方法，并提供许多独特的小窍门、小技巧、小口诀等，使读者在轻松的环境中花费很少的时间就能掌握 C 语言，并应用自如。

本书兼顾了全国计算机等级考试二级 C 语言程序设计考试大纲的相关要求，可以作为等级考试辅导教材和培训班教材使用。对于大、中专院校师生、各类 C 语言应试备考人员、广大 C 语言编程爱好者，都具有很好的学习参考价值。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

### 图书在版编目 (CIP) 数据

C 语言其实很简单 / 张宁编著. —北京：清华大学出版社，2015  
ISBN 978-7-302-39751-9

I. ①C… II. ①张… III. ①C 语言-程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2015) 第 071319 号

责任编辑：夏兆彦

封面设计：张 阳

责任校对：徐俊伟

责任印制：何 芊

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈：010-62772015，[zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者：北京鑫丰华彩印有限公司

装 订 者：三河市溧源装订厂

经 销：全国新华书店

开 本：185mm×260mm 印 张：25.25 字 数：631 千字

版 次：2015 年 7 月第 1 版 印 次：2015 年 7 月第 1 次印刷

印 数：1~3000

定 价：59.00 元

# 前 言

你是否以前尝试学习过 C 语言但又放弃了，或者是学习得一知半解，或者还是一位对 C 语言“一窍不通”的初学者？那么本书正适合你！

C 语言功能强大，内容繁多，最容易让初学者摸不到“门”。因此，本书并不像大多专业 C 语言教科书那样“板起面孔教人”，不罗列知识点，不使用专业术语来云山雾罩地分析问题，而是引用大量贴近生活的实例，用通俗易懂的方式与读者交流。同时，本书还提供了许多独特的小窍门、小技巧、小口诀、顺口溜等，用句流行话说，那是比较“接地气”的。目的只有一个：让不是科班出身的零基础初学者在轻松的环境中花费很少的时间来掌握 C 语言。

“大凡是讲编程的书，一定不好啃。”这恐怕是许多初学者和正在应对各种 C 语言考试的读者在学习道路上困惑的心声。笔者这里要告诉读者的是，本书拥有 3 个最显著的特点，那就是——不用啃，不用啃，真的不用啃！笔者已在 C 语言教学一线从教多年，深谙初学者的学习弱点。为此，从初学者角度出发，本书精炼了 C 语言教学的内容，在保证知识体系完整的基础上省去了许多无关紧要又晦涩难懂的专业知识，使本书内容既不过于复杂，又能满足一般编程的实际需要，更主要的是可以满足大多 C 语言考试（如全国计算机等级考试）的要求。尤其针对从一线教学中搜集到的许多初学者普遍认为的学习困难和容易误解的知识点，本书都用通俗易懂的方式做了大量的分析解读，尽最大努力帮助读者理清头绪、澄清概念，将对知识的误解消弭于无形。因此通过本书来学习 C 语言，读者会很快抓住知识的“根”，因而能达到事半功倍的效果！

学好 C 语言，方法是关键。本书会教给读者许多独特、有效的学习方法，但在这里笔者希望再强调一点：不少读者像学习英语一样来学习 C 语言，他们花大把的时间用来背诵程序，认为解决一个问题只对应着一个固定的程序：“熟读唐诗三百首，不会吟诗也会吟”，我“熟背程序三百篇，焉能还得不会编？”，这是大错特错的！因为实际问题千变万化，背诵的程序和实际问题不可能完全一样，因此纵使背下上千个程序，遇到实际问题还是难以下手。程序的运行是动态的，解决同一个问题不同的人编出的程序也不会完全相同。因此，学习程序设计实际是学习程序设计的思路和方法，完全没有必要背程序。而初学者的这种错误认识，究根溯源还是因为“照本宣科”的教科书。很多 C 语言的教科书，其中的程序例子确实很像“英文小短文”，一个问题对应着一个程序，并且在静态的纸上讲出程序运行的动态过程着实不易。为此，本书为典型程序例子都配有内存空间及变量值变化的插图，这些插图是比较“另类”的，以清晰反映程序的运行过程和变量值的变化为初衷，而不是只截个图给出程序的运行结果。这使读者可以更多地关注程序运行的过程，而不是只关注程序运行的结果。在“运动”中掌握程序，这也算是本书与大众化教科书所不同的另一个特点吧。

如何能掌握程序设计的方法，在遇到五花八门的实际问题时都能应对自如，编出对应

的程序？不要和我讲“融会贯通”“灵活运用”，那是颇有经验的编程高手们的事，而本书所关注的是初学者。为此，本书提供了许多编程“套路”，从典型的程序例子中理出“套路”，就能应对一大批的实际问题。学习武术有套路，没想到学习编程也有套路吧！用套路学习编程，让初学者迅速掌握编程方法，很快就能具有解决实际问题的编程能力，这也算是本书与大众化教科书所不同的又一个特点。

本书还抛砖引玉地介绍了数据结构、软件设计、数据库等最基本的知识，使读者在C语言学习之后向更高层次迈进。这些内容兼顾了全国计算机等级考试二级《公共基础》的相关考点。由于公共基础的考试内容对各类科目的二级考试都是相同的，不只局限于二级C语言，因此它们也可作为参加各类科目二级考试的读者备考《公共基础》的复习参考资料。

在本书最后还配有索引，“索引在手，遗忘不愁”。纵使有些学过的知识忘记了，也可以通过索引很快地找回来并复习巩固。本书索引既可以对C语言的基本概念进行速查，也可以对C语言的语句、关键字、运算符进行速查，还可以对基本的程序设计方法进行速查。

希望读者读过本书后，真正能把C语言用起来，让它成为我们身边的好朋友、好伙伴。倘能达到这个目的，笔者就感到心满意足了。

### 本书的独特栏目

在本书正文中，将穿插有以下栏目：

**【脚下留心】**针对初学者最容易犯的错误，或是在学习过程中，在编程实践时最应该引起注意的地方，都用“脚下留心”给出强调。零基础的初学者，紧紧抓住这些方面，就能在学习和编程实践中减少或避免很多不必要的弯路，为学习节省大量的时间。如果你正在应试，更要注意，这些内容往往都是高频出题但稍不留神就要丢分的。



脚下留心

忘记分号是初学者最易犯的错误之一。每条语句后的分号“;”千万别忘掉！

**【高手进阶】**是进一步提高水平的知识，一般比较深入或有些难度。“高手进阶”中的内容读者都可以根据兴趣选择阅读，跳过这些内容对后续章节知识体系的连贯性和整个C语言的学习都不会有影响。



高手进阶

数字字符与对应整数的二进制只有2位之差，例如字符'5'（即53）的二进制为0011 0101，整数5的二进制为0000 0101。前者第4、5位均为1（最右端为第0位），后者这两位均为0，而两者后4位是一致的0101，都表示十进制的5。还有，为什么'A'的ASCII码是65，'a'的ASCII码是97呢？作为字母表的第一个字母，为何大写从5开始，小写从7开始，似乎都不太“整”。把它们转换为二进制，答案立显！65的二进制是0100 0001，97的二进制是0110 0001，后5位0 0001都表示十进制的1，说明'A'、'a'是第一个字母。试

着把字母表的第二个字母'B'或'b'的 ASCII 码转换为二进制,你会发现后 5 位都表示十进制的 2。

【窍门秘笈】学习重在方法,方法得当,既可以节省学习时间,又能加深印象。“窍门秘笈”是学习方法的汇总,或是学习的小技巧、小窍门,或是概念的总结,还有轻松记忆知识点的顺口溜。我们的编程套路也将在窍门秘笈中给出。对于初学者,这些都是快速掌握 C 语言的捷径。



### 窍门秘笈

以变量类型为准的自动类型转换规则可总结为口诀如下:

变量定空间,塑身再搬迁。

若为空间窄,舍点也情愿。

【小游戏】编程不只是枯燥的工作,它也能充满乐趣。寓学于乐,寓编于乐,倘能达到这个境界,俨然不就是一位高手了吗?本书在正文中还穿插了一些小游戏,映衬相关的知识点,让读者在游戏中掌握编程!



### 小游戏

现有一架天平和 4 种重量的砝码,分别重 8 克、4 克、2 克、1 克,每种重量的砝码只有一个。现要用此天平称重 13 克的物体,物体放在左盘上,如图 1-20 所示。请问在右盘上应该怎样选放 4 种砝码,才能使天平左右两盘重量相同天平平衡呢?

## 程序示例和习题

【程序示例】学习编程,程序例子是必不可少的。对于程序示例,本书都精心做了安排。在突出知识点的基础上,本书程序示例所遵循的原则是:或者让趣味指数都在三星★★★以上,旨在提高读者的编程兴趣;或者让难度都在一星★以下,旨在简单明了、一针见血地说明问题。

【随讲随练】是本书的习题,这些都是针对大多 C 语言考试的高频考点精心设计的,其中一部分为全国计算机等级考试二级 C 语言程序设计的历年考试真题或无纸化考试改革后的题库真题,供读者巩固复习之用。本书习题的特色是“随讲随练”,每道题都安排在相应知识点讲解的正文之后,并在题后直接给出了答案。这避免了在章后统一安排习题所带来的向前查阅知识、向后查看答案的弊端,减少了读者反复翻书的无用功。读者可一气呵成,通读本书,就能有学有练。


【小试牛刀】这些不作为正式习题,而主要是思考题的性质,也将穿插在相应知识点的讲解中。读者利用刚刚所学知识可以马上试一试身手,或是巩固所学知识,或是举一反三,每试一次“牛刀”,都是一次能力的提高。

## 本书的卡通形象

在本书中,还有两个卡通形象,将陪伴读者整个的学习过程:

【小博士形象】是始终陪伴在我们身边的老师。或是学习的小贴士，或是需要注意的问题，“小博士”都会侃侃道来。这些都是比较关键的内容，请读者一定悉心体会。



用空格缩进无可厚非，但用 Tab 键  而不用一连串的空格，是更简便的做法。Tab（又称跳格、水平制表）和空格是两种不同的字符，但对于在程序中起的“空白间隔”作用是相同的。我们可以随意使用空格或 Tab 甚至空格和 Tab 的组合来作为程序中元素间的“空白间隔”。

【大零蛋形象】它是初学者的代言人，时不时地“冒出来”，或是提出初学者的常见疑问，或是以简单通俗的方式表达对知识的领悟。“零”就是零基础的意思，“我对 C 语言一窍不通，将伴随您一起从零开始学习！”。有大零蛋在身边，零基础的读者非但不必畏惧学习 C 语言，反而可以从中获得乐趣。随读者一起，随着学习的深入，“大零蛋”的水平也会逐步提高，由一窍不通的“大零蛋”最终摇身变为满腹经纶的“大灵蛋”！零基础的初学者们，请跟随它一起成长，相信读过本书之后，你也会由零基础的菜鸟摇身变成一位编程高手的。



我明白了：`printf` 函数的输出就是把“ ”中的内容原封不动地“抄”在屏幕上就可以了，但其中若遇到带%的“警察”则不要照原样抄，而要用后面的数据替换它。注意替换时要按照%所规定的“手势”，按相应格式替换。

有时候，大零蛋与小博士还会对起话来：

我英文没学好，不知道用哪个单词命名，怎么办？

用拼音吧，保存成绩的变量用 `chengji` 命名也不错哦！



## 致谢

---

感谢天津大学精密仪器与光电子工程学院的冯远明教授和课题组全体成员对本书写作的大力支持，没有他们的帮助和支持，不可能写出这样一部作品。特别感谢学院的何峰老师对 C 语言的知识介绍提供了许多有益的素材，这使很多枯燥乏味的概念讲解变得更为生动、有趣。

感谢赵佳为本书绘制或加工制作了全部精美的插图，这使本书更加妙趣横生，尤其是她设计的小博士和大零蛋形象，别有一番特色。

由于笔者水平有限，错谬之处在所难免，恳请专家和广大读者不吝赐教、批评指正。笔者的 E-mail 是：[zhni2011@163.com](mailto:zhni2011@163.com)，QQ 号码是：1307573198。

# 目 录

第 1 章 从这里爱上编程——程序设计和 C 语言概述	1
1.1 水面下的冰山——计算机程序和计算机语言	1
1.2 第一次亲密接触——纵览 C 语言	3
1.2.1 一窥程序之美——C 语言程序的结构	3
1.2.2 循序渐进, 快乐学习——语句的光和影	4
1.2.3 程序里的说明书——注释	6
1.2.4 免费翻译服务——编译和链接	8
1.3 先其利器——Visual C++ 6.0 上机指导	10
1.4 天平称物问题——进制转换	15
1.4.1 这些进制是哪来的——二进制、八进制、十六进制	15
1.4.2 你还应知道这几招——不同进制之间的转换	15
第 2 章 色彩斑斓的积木——数据类型、运算符和表达式	21
2.1 标识符、常量和变量	21
2.1.1 给我起个名字吧——标识符	21
2.1.2 有一说一——常量	22
2.1.3 程序里的储物盒——变量	23
2.2 追根“数”源——细说数据类型	25
2.2.1 整型数据用 C 语言怎样说	25
2.2.2 实型数据用 C 语言怎样说	28
2.2.3 字符型数据用 C 语言怎样说	32
2.2.4 字符串型常量用 C 语言怎样说	36
2.3 诸算达人——运算符和表达式	39
2.3.1 再谈加减乘除——算术运算	39
2.3.2 走, 给我进去! ——赋值	45
2.3.3 加加和减减——自增、自减	48
2.3.4 神秘的倒数第一——逗号运算	50
2.4 位在我心中——位运算	52
2.4.1 按位与 &	52
2.4.2 按位或	52
2.4.3 按位异或 ^	53
2.4.4 按位求反 ~	54



2.4.5	按位左移 << .....	54
2.4.6	按位右移 >> .....	54
2.4.7	位运算的复合赋值 .....	55
<b>第 3 章</b>	<b>一战到底——顺序结构 .....</b>	<b>56</b>
3.1	整装待发——C 语言中的语句 .....	56
3.2	别急，一个一个来——单个字符的输出与输入 .....	57
3.2.1	拿好了钥匙进仓库——输出与输入概述 .....	57
3.2.2	向屏幕开火——单个字符的输出 .....	58
3.2.3	饭要一口一口地吃——单个字符的输入 .....	59
3.3	更过瘾的输出与输入——格式输出与输入 .....	62
3.3.1	我有私人警察——格式输出函数 printf .....	62
3.3.2	我是快乐的快递员——格式输入函数 scanf .....	68
3.4	常用系统数学函数 .....	73
3.5	是不是有点专业级软件的意思了——顺序结构程序举例 .....	74
<b>第 4 章</b>	<b>程序也能跑捷径——选择结构 .....</b>	<b>76</b>
4.1	计算机的判断力——关系运算和逻辑运算 .....	76
4.1.1	判断真假的符号——关系运算符和逻辑运算符 .....	76
4.1.2	火眼金睛断真假——关系表达式和逻辑表达式 .....	78
4.1.3	挑剔的“吗+否则”——条件运算 .....	82
4.2	如果——if 语句 .....	83
4.2.1	教室停电就不上课了——if 语句的基本形式 .....	83
4.2.2	一朝天子一朝臣——if 语句的完整形式 .....	88
4.2.3	如果里的如果——if 语句的嵌套 .....	94
4.3	多路开关——switch 语句 .....	98
4.3.1	司令的锦囊——switch 语句的一般形式 .....	99
4.3.2	我爱读小说——switch 语句的执行过程 .....	100
4.4	goto 去哪？想去哪就去哪！——goto 语句 .....	105
<b>第 5 章</b>	<b>不必亲手愚公移山——循环结构 .....</b>	<b>107</b>
5.1	看好了情况再下手——while 语句 .....	107
5.2	先下手干了再说——do...while 语句 .....	110
5.3	我勤奋·我劳动·我光荣——for 语句 .....	112
5.3.1	按劳分配——for 语句的基本形式 .....	112
5.3.2	劳动者的工作模式——for 语句的常见应用 .....	114
5.3.3	有人接班我偷懒——表达式的变化 .....	119
5.4	循环里的循环——循环的嵌套 .....	122
5.5	埋头干活中的抬头看路——continue 语句和 break 语句 .....	125

5.5.1	来源于生活——continue 语句和 break 语句概述	125
5.5.2	下一个上——continue 语句	127
5.5.3	前方施工请绕行——break 语句	129
5.6	轻车熟路——程序控制结构小结和综合举例	130
<b>第 6 章</b>	<b>把平房升级为高楼大厦——数组</b>	<b>137</b>
6.1	直线升级——一维数组	137
6.1.1	一维数组的定义和引用	137
6.1.2	一维数组的初始化（定义时赋初值）	144
6.1.3	一维数组的应用	147
6.2	找东西和整理东西的艺术——查找和排序	152
6.2.1	这个经常有——查找技术	152
6.2.2	混乱之治——排序技术	156
6.3	立体升级——二维数组	163
6.3.1	二维数组的定义和引用	163
6.3.2	二维数组在内存中的存储形式	164
6.3.3	二维数组的初始化（定义时赋初值）	164
6.3.4	二维数组可被看做是由一维数组组成的	169
<b>第 7 章</b>	<b>蒙着面干活——函数</b>	<b>171</b>
7.1	从讲故事开始——函数概述	171
7.2	该是学写多段文章的时候了——函数定义和调用	173
7.2.1	自己创造函数——函数的定义	173
7.2.2	把自己写的“段落”用起来——函数的调用	175
7.2.3	你歇着，我劳动——函数调用的过程	177
7.2.4	做好的菜端出来——函数的返回值	179
7.3	喂！听到了吗——函数的声明	185
7.3.1	函数声明的形式	186
7.3.2	函数声明的位置	187
7.4	函数的嵌套调用和递归调用	189
7.4.1	函数里的函数——函数的嵌套调用	189
7.4.2	克隆函数——函数的递归调用	190
7.5	变量的时空范围——变量的作用域及存储类别	196
7.5.1	空间范围——局部变量和全局变量	196
7.5.2	时间范围——变量的存储类别	202
7.5.3	时空统一——局部变量和全局变量的存储类别	206
7.6	给编译做点儿手脚——预编译处理	207
7.6.1	潜伏代号——宏定义	207
7.6.2	程序的自动复制粘贴——文件包含	211

7.6.3 早知当初,何必如此——条件编译	212
<b>第8章 璀璨的星星——指针</b>	<b>213</b>
8.1 内存里的门牌号——地址和指针的基本概念	213
8.2 别拿地址不当值——指针变量	214
8.2.1 找张字条记地址——定义指针变量	215
8.2.2 这可不是说我——指针变量的基类型	217
8.2.3 把地址记下来——为指针变量赋值	218
8.2.4 指针运算俩兄弟——两个运算符	220
8.3 原来咱俩是一个朋友圈的——数组与指针	223
8.3.1 下一站到哪了——指针变量的运算	223
8.3.2 我原来是指针变量——一维数组的指针	227
8.3.3 我是你的上级——二维数组的指针和行指针	232
8.3.4 来自星星的数组——指针数组和指针的指针	238
8.3.5 指针三家人——指针小结	242
8.4 有了地址也可以找我帮忙啊——函数与指针	244
8.4.1 地址给我,我来帮忙——指针变量做函数参数	244
8.4.2 吃葡萄不抓葡萄粒,抓住葡萄的把柄——数组做函数参数	249
8.4.3 指针私房菜——返回地址值的函数	256
8.4.4 函数遥控器——函数的指针	256
8.5 一两拨千斤——字符串的指针	259
8.5.1 字符串的存储	259
8.5.2 字符串的输出和输入	263
8.5.3 字符串处理技术	267
8.6 另类运行程序——main函数的参数	286
<b>第9章 我的类型我做主——结构体与共用体</b>	<b>289</b>
9.1 多功能收纳盒——结构体	289
9.1.1 绘制收纳盒设计图——定义结构体类型	289
9.1.2 收纳盒制作——使用结构体变量	290
9.1.3 一本通讯录——结构体类型的数组	294
9.1.4 结构体的弓箭手——结构指针变量	296
9.1.5 重口味与轻口味——结构体类型数据做函数参数	297
9.1.6 大收纳盒里套小收纳盒——结构体类型的嵌套	301
9.2 公路桥洞——共用体	302
9.3 给类型起个“绰号”——类型定义符 typedef	304
9.4 内存空间的批发和零售——动态存储分配	306
9.5 电影院里的座次问题——链表	310
9.5.1 链表的建立和遍历	311

---

9.5.2	链表结点的插入和删除	317
9.5.3	链表的高级兄弟——高级链表简介	320
<b>第 10 章</b>	<b>得文件者得天下——文件</b>	<b>322</b>
10.1	一针 hold 住全文件——文件指针	322
10.1.1	与文件牵手——文件的打开	322
10.1.2	与文件分手——文件的关闭	326
10.1.3	文件操作流程	326
10.2	搬运流水线——文件的读写	327
10.2.1	手指和笔尖——文件位置指针	327
10.2.2	文本文件的读写	328
10.2.3	二进制文件的读写	332
10.3	这是手工活儿——文件的随机读写	337
<b>第 11 章</b>	<b>编程的经验财富——算法与数据结构基础</b>	<b>345</b>
11.1	“一招鲜”——算法	345
11.1.1	何谓算法	345
11.1.2	算法的控制结构	346
11.1.3	算法复杂度	346
11.2	数据结构概述	347
11.2.1	何谓数据结构	347
11.2.2	数据结构的类型	348
11.3	早出晚归的勤快人——栈（堆栈）	349
11.3.1	何谓栈（堆栈）	349
11.3.2	堆栈的基本运算	350
11.3.3	堆栈的逻辑结构和存储结构	350
11.3.4	堆栈的应用	351
11.4	先来后到——队列	352
11.4.1	何谓队列	352
11.4.2	队列的逻辑结构和存储结构	352
11.4.3	循环队列的基本运算	353
11.5	倒置的树——树与二叉树	354
11.5.1	树和树的基本概念	354
11.5.2	二叉树的基本概念	355
11.5.3	二叉树的存储结构	356
11.5.4	二叉树的遍历	356
<b>第 12 章</b>	<b>程林高手武功秘籍——软件开发基础</b>	<b>360</b>
12.1	编程之道——程序设计方法	360

12.1.1	编程 Style——程序设计风格	360
12.1.2	组装零件——结构化程序设计	361
12.1.3	这个 feel, 爽! ——面向对象程序设计	361
12.2	不懂门道看热闹, 看完咱也吊一吊——软件工程基础	364
12.2.1	何谓软件	364
12.2.2	软件生命周期	365
12.2.3	需求分析及其方法	366
12.2.4	软件设计及其方法	367
12.2.5	我是来找茬的——软件测试	369
12.2.6	谁来改正——程序的调试	371
12.3	信息时代是怎样炼成的——数据库和数据库设计初步	371
12.3.1	关系型数据库及相关概念	372
12.3.2	关系代数	374
12.3.3	数据库系统	378
12.3.4	数据库设计者眼里的世界——E-R 模型	380
12.3.5	数据库设计	381
附录一	常用字符 ASCII 码对照表	383
附录二	C 语言中的关键字	384
附录三	C 语言运算符的优先级和结合性	385
索引		386
参考文献		391

# 第 1 章 从这里爱上编程——程序设计和 C 语言概述

在生活中，许多工作都要遵照一定的程序完成。例如早上起床后的穿衣、刷牙、洗脸、梳头、吃早饭就是一个程序；做菜时的洗菜、切菜、炒菜、放盐、出锅也是一个程序。简而言之，程序就是一系列的操作步骤。计算机要实现某个功能也必须遵照一定的程序。然而遗憾的是，如果我们不会编写程序，计算机就只有遵照别人编写的程序来工作，而我们则只能在别人编写的程序的控制下使用计算机，只能成为计算机的奴隶。

你想摆脱别人程序的束缚，彻底地驾驭计算机吗？你想让计算机服从你的指挥，完全遵照你的意愿来工作吗？你想由计算机的奴隶变成它真正的主人吗？从今天开始，让我们走进计算机编程这个神秘的世界，一起掌握编程的本领、享受编程的乐趣、挖掘计算机更深层的魅力！

## 1.1 水面下的冰山——计算机程序和计算机语言

善于使用工具，是人类的特殊本领。人类的祖先在很早就会用石器作为工具，如图 1-1 所示。随着社会的进步，人们发明了各种各样的工具，计算机也是其中的一种，因此从工具的角度来说，一台计算机和一块原始人类的石器没有什么分别——因为它们都必须有“人”的操纵才能发挥功能，而工具本身是没有任何智能的。没有人的操纵，一块石头不可能自己飞出来猎杀一头野猪，一把螺丝刀也不可能自己转起来卸下一只螺钉。计算机也不例外，虽然它“能听会唱”、“能写会算”，一直被蒙上一层神秘的面纱，然而必须明确计算机的所有“智能”都是人类赋予的。人们针对特定的问题，详细告诉计算机“做什么”、“怎么做”，计算机只会傻乎乎地按照人们的指挥一步步地、老实地执行。计算机本身没有任何智能，在它“精明能干”的外表下隐藏着的却是“唯命是从”的憨厚本性，这就是计算机的世界。人类指挥计算机工作的过程，就是计算机程序。因此可以说，计算机的任何功能都是通过程序完成的，没有程序，计算机就不能做任何工作，如图 1-2 所示。



图 1-1 原始人类在使用石器



图 1-2 计算机必须有程序的指挥才能工作

一个程序是由一条条指挥计算机工作的命令（指令）组成的。人们为计算机解决某个问题事先设计操作步骤，“指挥”计算机如何工作，就是编写程序的过程。编写程序要通过某种计算机语言，也就是要通过某种计算机能听懂的语言来“告诉”它做法。

计算机“能听懂”的语言是怎样的呢？计算机的语言也有很多种，概括起来可以分为三大类：机器语言、汇编语言、高级语言。

机器语言仅由二进制的 0 和 1 组成，其中指挥计算机工作的每条指令都用二进制的数码表示，称为机器指令。用这种语言编写的程序，就是一串 010101.....的代码，甚至连标点符号都没有。显然，这种语言只有计算机能识别，而对我们人类来说则与天书无异，极少数人能看得懂。

由于学习和使用机器语言非常困难，人们又发明了其他类型的计算机语言——高级语言。在高级语言中，不必使用二进制代码，而可以使用英文单词、十进制数字、数学公式等编写程序，这使我们指挥计算机的过程大大简化。例如要指挥计算机计算一个算式，用高级语言只要编写下面的语句就可以了。

$$x = a + b - 24;$$

在计算机的世界中还有一种语言是汇编语言，它介于机器语言和高级语言之间。汇编语言使用的是英文助记符，也不是二进制的代码。然而它必须使用与机器语言相对应的指令，与人类思维习惯相距甚远。因此汇编语言难度也很高。

计算机只能直接读懂由 0 和 1 组成的二进制代码，也就是用机器语言编写的程序，如图 1-3 所示。高级语言程序和汇编语言程序由于都不是二进制代码，计算机都不能直接读懂。因此对用这两种语言编写的程序，还要将它们翻译成对应的二进制的机器语言程序，然后计算机才能够读懂执行，这个翻译过程被称为编译。将中文文章翻译成英文时，中文文章是要被翻译的来源，英文文章是要被翻译的目标。类似地，将高级语言和汇编语言程序翻译为机器语言程序时，高级语言程序和汇编语言程序被称为源程序，翻译后对应的机器语言程序被称为目标程序。

以上三种语言中，只有高级语言最接近人类的思维习惯，也是最简单、最容易掌握的。因此高级语言才是我们的学习目标。高级语言有很多种，如 Visual Basic 语言、Java 语言、Fortran 语言、C 语言等都属于高级语言。C 语言是当代最优秀的程序设计语言之一，也是 C++ 语言的基础，我们熟知的许多软件包括 Windows 系统本身都主要是用 C/C++ 语言编写的。在本书中，我们即将要学习的是 C 语言。

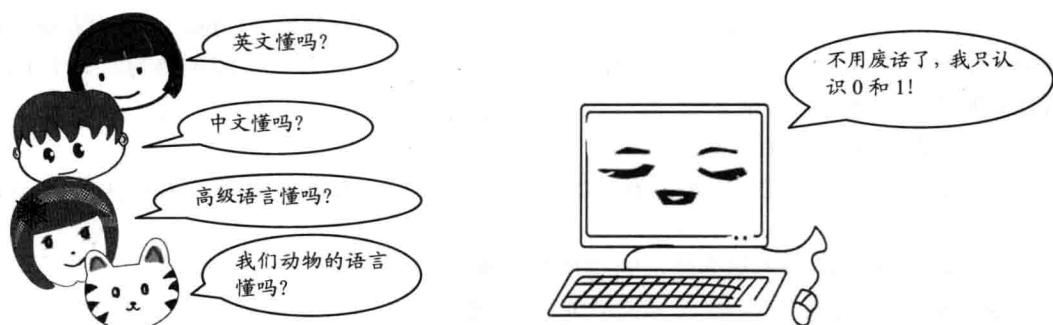


图 1-3 计算机只能直接识别由 0 和 1 组成的机器语言程序

## 1.2 第一次亲密接触——纵览 C 语言

### 1.2.1 一窥程序之美——C 语言程序的结构

用 C 语言编写一个程序类似于用中文写一篇文章。我们写文章时要分段落，段落中再分句子；一个 C 语言程序也要分函数编写，函数中又包含语句，如图 1-4 所示。函数相当于段落，函数中的语句相当于段落中的句子。可见 C 语言程序的基本组成单位是函数而不是语句。一个 C 语言源程序的基本结构如图 1-5 所示。

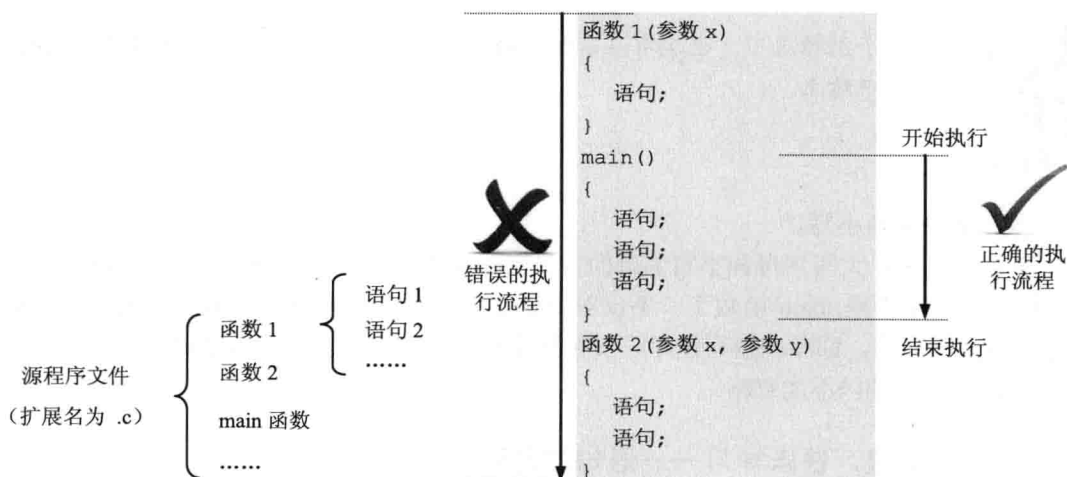


图 1-4 C 语言源程序的组成

图 1-5 C 语言源程序的基本结构及执行流程

从图 1-5 可以看出，每个函数由函数头部开始，函数头部包含“函数名+( )”，( ) 内为函数的参数（参数有些类似于数学中函数的自变量），一个函数可有多个参数，也可以没有参数。没有参数时，小括号本身( )不能省略，然后紧随一对大括号 { }，组成这个函数的语句都要被放在这对 { } 中。



一个C语言源程序可以包含多个函数，但其中一个函数必须名为 `main`，没有参数（或有两个参数，第8章将介绍），它称为主函数。顾名思义，主函数是最主要的。关于 `main` 函数需要理解以下几个方面。

(1) `main` 函数必须有，且只能有一个。

国不能一日无君，但一山不能容二虎。一个程序没有 `main` 函数或有两个以上 `main` 函数都是错误的。

(2) `main` 函数可位于程序函数间的任意位置。

如同皇帝出巡时，不一定走在最前面；当然走在最前面或最后面也无可厚非。同样 `main` 函数不一定是程序的第一个函数，也不一定是最后一个函数，它可位于函数之间的任意位置。

(3) `main` 函数是程序的入口和出口。

程序的执行过程，与我们阅读文章的过程是不同的。程序的执行是由 `main` 函数“这一段”起始，在 `main` 函数“这一段”中结束（而无论 `main` 函数位于什么位置）；而不是由第一个函数起始，在最后一个函数中结束，如图 1-5 所示。从 `main` 函数的第一条语句开始执行，这称为程序的入口；当 `main` 函数的最后一条语句执行完毕整个程序也就结束了，称为程序的出口。

其他函数似乎没有执行，不是形同虚设了吗？并没有形同虚设！其他函数是在 `main` 函数执行期间，由 `main` 函数调用执行的。其他函数与 `main` 函数的关系如同随从与皇帝的关系，当皇帝差遣随从去办事时随从才会活动，差遣就是调用。关于函数是如何调用的，我们将在以后讨论。现在读者只需理解程序的执行是从 `main` 函数开始，在 `main` 函数中结束就可以了。



个别情况下，也会有在其他函数中异常结束程序的可能，而没有在 `main` 函数中结束。

### 高手进阶

(4) `main` 必须小写。

在C语言中，大写字母和小写字母的含义是完全不同的。如将 `main` 写为 `Main`，就是两个概念，将不再是 `main` 函数了。不仅对 `main`，以后要学习程序中的其他元素也是严格区分字母大小写的，读者在编写程序时要特别注意，一定不要随便篡改字母的大小写。

(5) `main` 后的 `()` 不能省略。

## 1.2.2 循序渐进，快乐学习——语句的光和影

下面给出一个简单的C语言程序的例子，该程序只由一个 `main` 函数组成。

**【程序例 1.1】** C语言程序的简单实例。

```
main()
{
    int a,b,c;
    a=10;
```