

ANSI C

彭光泽 编译
刘乃琦 主审

C 语言入门

C LANGUAGE



电子科技大学出版社

C 语言入门

彭光泽 编译
刘乃琦 主审

电子科技大学出版社

•1992•

内 容 简 介

本书是基于新的 ANSI C 进行讲解的入门读物。内容有: C 的基本常识; 控制台与输入输出; 控制结构; 数组; 指针; 函数; C 特有的运算符; 结构与共享件; 数据类型与存储类; 预处理程序; 标准库函数; 文件处理。

通俗易懂, 行文流畅; 层次分明, 图解有趣; 恰当的举例, 适量的练习等是本书的特色。本书将使初学者及在 C 的学习中有过挫折的人倍感亲切。

ANSI C

C 语言入门

彭光泽 编译

刘乃琦 主审

*

电子科技大学出版社出版

(中国成都建设北路二段四号)

电子科技大学出版社激光照排中心照排

四川省平武县印刷厂胶印

四川省新华书店经销

*

开本 787×1092 1/16 印张 12.875 版面字数 310 千字

版次 1992 年 6 月第一版 印次 1992 年 6 月第一次印刷

印数 1—7000 册

中国标准书号 ISBN 7-81016-401-5/TP·29

[川]016 (15452·176) 定价(压膜)5.40 元

译 序

有人说 C 语言是职业程序员的语言,其理由是 C 语言学习起来比 BASIC 语言难。的确,C 语言中引入了不少诸如宏、流、指针、结构等令人生畏的概念。再加上许多有关 C 语言的书籍都出自设计 C 语言本身的高水平专家之手,其内容比较深奥,因此就难免使一些初学 C 语言的人望而却步了。

其实,万事只在开头难。C 语言也不例外,只要入了门,就未必见得比 BASIC 更难驾驭。或许,使用起来,C 语言还有比 BASIC 更为方便之处。

《C 语言入门》,将同大家一道来叩开 C 语言的大门。这次之所以选择这本书并把它翻译出来介绍给国内读者,是因为该书具有以下几个对初学者来说特别理想的特点:

- 一、书中每一章节都采取了步步为营的教学方式,即:基本概念的学习——案例学习——训练巩固。
- 二、抽象问题全部形象化。
- 三、凡遇专业术语均作了个别说明。
- 四、图文并茂,举例生动。
- 五、也是最重要的一点,能让人懂。

另外,值得说明的是,随着这些年来 C 语言的普及,从各种 C 处理体系中派生出了不少方言。为此,ANSI(美国标准规格协会)于 1988 年制定了 C 语言规范。其后,遵从该规范的 C 语言便称为 ANSI C。也许读者早已从本书的封面上注意到,这正是一本基于崭新的 ANSI C 进行讲解的 C 语言入门读物。所以,此书不仅初学者可以放心地学习下去,即使已具备某种程度 C 语言基础的人,也不妨拾来一阅,兴许对自己先前所掌握的较老的 C 语言知识的规范化会有一定帮助。

目前,BASIC 热方兴未艾,C 语言又异军突起,计算机世界总给人以日新月异之感。如果本书能够赶上科学进步的潮流而对为祖国现代化努力奋斗的读者多少有点帮助的话,那么,作为译者,将会从内心感到莫大宽慰。

在这里,要感谢远道为我们提供原版资料的日本朋友安岐康治先生。同时,要感谢对本书给予充分肯定并对其出版给以大力支持的电子科技大学出版社社长杨旭明教授和承担本书主审的电子科技大学计算机系副系主任刘乃琦副教授。

此书虽颇具特色,但译者水平有限,疏漏之处海望读者批评指正。

彭光泽
1992 年元月

目 录

第 1 章 C 的基本常识	1
§ 1.1 什么是 C 语言	1
●——C 的起源	1
●——从 BASIC 时代迈向 C 的时代	2
§ 1.2 最简单的程序	2
●——显示“Hello”的程序	2
●——名叫 main 的函数	3
●——构成程序的单词	3
●——排列单词造句	3
§ 1.3 构成程序的单词	4
●——保留字(关键字)	4
●——函数名	4
●——运算符	5
●——变量与常数	6
●——标识符的命名规则	6
§ 1.4 数据类型与变量的类型说明	7
●——什么是类型	7
●——变量的类型说明	7
●——C 的基本数据类型	8
§ 1.5 C 的程序设计格式	8
●——依据 ANSI C 的书写格式	8
●——#include	9
●——自由格式与空格	10
●——注释	11
第 2 章 控制台与输入输出	12
§ 2.1 与计算机的沟通	12
●——scanf/printf 函数的概况	12
●——格式控制字符串	13
§ 2.2 向显示器输出数据:printf	13
●——整数数据的输出	13
●——实数数据的输出	14
●——多个数据的输出	14
●——\n(换行)	14

§ 2.3 输出位数的指定	15
●——整数数据的位数指定	16
●——实数数据的位数指定	16
§ 2.4 键盘数据输入,scanf	17
●——整数数据的输入	17
●——实数数据的输入	17
§ 2.5 字符与信息交换标准代码	19
●——ASCII 码	19
●——2 进制数与 16 进制数	20
§ 2.6 字符串的输入与输出	20
●——字符串与数组	20
§ 2.7 单个字符的输入与输出:getchar/putchar	22
●——getchar/putchar	22
●——换码符	23
▲——char 型与 int 型的混用	24
第 3 章 控制结构(1)	26
§ 3.1 控制结构与结构化程序设计	26
●——什么是控制结构	26
●——结构化程序设计	27
§ 3.2 if else 语句(条件判定)	27
●——if else 语句与程序块	27
●——关系运算符	28
§ 3.3 复杂的条件表达式	29
●——复杂的条件表达式	29
●——逻辑运算符	29
§ 3.4 for 语句	31
§ 3.5 while 语句	33
§ 3.6 数据的结尾与 EOF	35
●——判定数据结尾的方法	35
▲——表达式与表达式的值	37
§ 3.7 多重循环	37
[算法训练 1] 直角三角形的组合	39
第 4 章 数组	41
§ 4.1 对数据进行管理的数组	41
●——数组的作用	41
●——数组的说明	42
§ 4.2 数组数据的初始化	42
§ 4.3 向数组输入数据	44
§ 4.4 二维数组	46
[算法训练 2] 求最大值	48

第 5 章 指针	50
§ 5.1 什么是指针	50
●——变量的逻辑映象与物理映象	50
●——指针变量的说明与访问	51
▲——指针运算符	51
●——指针的大小	52
§ 5.2 指针与字符串	52
§ 5.3 指针与一维数组	54
§ 5.4 由指针决定的数组表达	56
§ 5.5 指针与二维数组	57
§ 5.6 指针数组	59
§ 5.7 指针的地址计算	61
[算法训练 3] 排序	62
第 6 章 函数	64
§ 6.1 什么是函数	64
●——函数的作用	64
●——函数的定义与自变量	64
§ 6.2 局部变量	67
§ 6.3 传值调用 (call by value)	69
§ 6.4 传址调用 (call by reference)	71
§ 6.5 传递一维数组	73
§ 6.6 传递二维数组	74
§ 6.7 传递指针数组	76
§ 6.8 函数原型	78
●——void	79
▲——以往的函数定义格式	81
[算法训练 4] 哈诺依塔(递归算法)	81
第 7 章 控制结构(I)	83
§ 7.1 do while 语句	83
§ 7.2 switch case 语句	85
§ 7.3 else if 语句	87
§ 7.4 break 语句	89
§ 7.5 其它控制结构	91
●——continue 语句	91
●——goto 语句	91
●——无限循环	92
[算法训练 5] 分类	93
第 8 章 C 特有的运算符	95
§ 8.1 运算符的种类与优先级	95
●——运算符的种类	95

●——优先级与结合规则	95
§ 8.2 位运算符	97
●——位(bit)逻辑运算	97
●——移位运算	98
§ 8.3 前缀运算与后缀运算	100
§ 8.4 条件运算符	102
§ 8.5 逗号运算符	103
§ 8.6 其它运算符	104
●——赋值运算符	104
●——sizeof 运算符	105
●——指针运算符	105
●——强制型运算符	106
●——结构元素访问运算符	106
●——函数调用与下标表达式	106
[算法训练 6] 对分检索	106
第 9 章 结构与共享件	108
§ 9.1 什么是结构	108
●——结构的说明	108
●——结构元素的访问	109
§ 9.2 结构数组	110
§ 9.3 结构的成批赋值	111
§ 9.4 针对结构的指针	113
§ 9.5 共享件	116
[算法训练 7] 爱拉托逊斯筛法	118
第 10 章 数据类型与存储类	121
§ 10.1 数据类型	121
●——数据类型的种类	121
●——基本数据类型的尺寸与范围	122
●——有符号数与无符号数	122
§ 10.2 enum 型	123
§ 10.3 类型转换与强制型	125
●——混合运算与类型转换	125
●——零扩充与符号扩充	126
●——强制型	126
§ 10.4 存储类	128
§ 10.5 作用域(scope)	129
●——作用域与寿命	129
●——自动变量/静态变量/外部变量	130
§ 10.6 静态变量	132
§ 10.7 外部变量	134

§ 10.8	初始化	135
●	——编译时的初始化与执行时的初始化	135
●	——数组的初始化	136
第 11 章	预处理程序	137
§ 11.1	什么是预处理程序	137
●	——#include	138
§ 11.2	简单宏功能	139
§ 11.3	带自变量的宏	141
§ 11.4	宏与函数	142
	[算法训练 8] 万年历	143
第 12 章	标准库函数	147
§ 12.1	ANSI 程序库	147
●	——什么是标准库函数	147
●	——ANSI 程序库	147
§ 12.2	数值运算函数	149
§ 12.3	随机数	151
§ 12.4	字符串处理函数	153
●	——strcpy(字符串的复制)	154
●	——strcat(字符串的连接)	154
●	——strcmp(字符串的比较)	155
●	——strlen(取得字符串的长度)	155
§ 12.5	字符校验/转换宏	156
●	——is~(字符校验宏)	156
●	——to~(字符转换宏)	157
	[算法训练 9] 蒙特卡罗法的 π 计算	159
第 13 章	文件处理	160
§ 13.1	文件处理概要	160
●	——文件指针与文件打开	160
●	——文件输入/输出的概念	161
§ 13.2	文件处理函数	161
●	——fopen(文件的打开)	162
●	——fclose(文件的关闭)	162
§ 13.3	单字符文件输入/输出	163
●	——getc(来自文件的单字符输入)	163
●	——putc(针对文件的单字符输出)	163
§ 13.4	系统指令的使用与命令行自变量的获取	165
§ 13.5	单行文件输入/输出	166
●	——fgets(来自文件的单行输入)	166
●	——fputs(针对文件的单行输出)	167
§ 13.6	格式文件输入/输出	169

●——fprintf(格式文件输出)	169
●——fscanf(格式文件输入)	169
§ 13.7 顺序文件	170
●——文件结构	170
●——顺序文件的读出/写入	170
▲——I/O 重定向(redirect)	172
附录 ANSI C 主要函数参考手册	174
问题解答	192
习题解答	194

第 1 章 C 的基本常识

【本章目的】

学习一门新的计算机语言,最明智的作法是先编程序,千万不要一开头就去啃过细的语法。

本章将让读者了解用 C 语言编制程序所需具备的起码知识。

§ 1.1 什么是 C 语言

学习要点

- ☆ C 语言的起源
- ☆ C 在计算机语言中的地位

●——C 的起源

【名词解释】

▶C 1967 年,英国剑桥大学的 Martin Richards 发明了一种计算机语言,取名 BCPL。1970 年,美国贝尔研究所的 Ken Thompson 又发明了一种语言取名 B。后来的 C 语言在开发过程中曾受到以上两种语言的重大影响。出于这种单纯的理由,C 语言便继 B 之后取名为“C”。

▶操作系统 操作系统是计算机的运用系统。建立操作系统的目的是为了更有效地利用硬件资源、软件资源以及人力资源,从而提高系统整体的效率。

* * * *

1972 年,美国贝尔研究所的 Dennis Ritchie 等人,专门为配备在 DEC 公司生产的微机 PDP-11 上的 UNIX 开发出一种描述语言。他们给这个语言取了一个相当直率的、仅由 1 个孤单的字母所构成的名字——C。

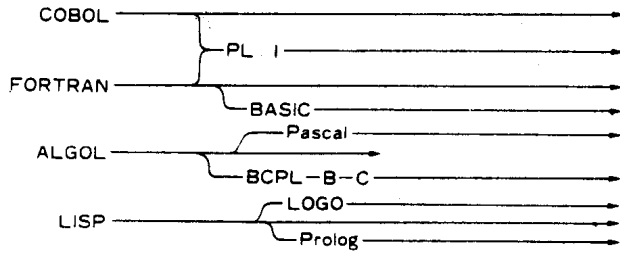
所谓 UNIX,是指由贝尔研究所建立起来的一种计算机操作系统,是当今最为普及的标准操作系统之一。

起初,C 语言只是在以贝尔研究所为中心的一部分有专业志向的爱好者小组内部使用。但后来随着 UNIX 系统的成功,C 语言便迅速地向各大学和研究机关普及。同时,由于 CP/M 和 MS-DOS 操作系统的采用,C 语言得以在个人计算机上崭露头角。如今,C 的用户范围已扩展到普通的个人计算机用户,而不再局限于大学和企业中的专门研究人员了。

C 语言以 ALGOL 语言为母体,是在结构化程序设计这一强烈的概念意识下作成的。同样以 ALGOL 为母体的,还有 Pascal 语言,它与 C 语言非常相似。

简单地说来,C 与 Pascal 的不同之处在于,Pascal 是重视语言规范美的教育性语言,而 C 是重视易于使用的实用性语言。

下图显示了几种主要计算机语言的发展历程。



● —— 从 BASIC 时代迈向 C 的时代

【名词解释】

▶ **分时系统** 对中央处理机的处理时间进行时间分片后再分配给用户的一种操作系统的方式。利用分时系统,各终端的多个用户可同时使用一台主计算机。

▶ **独立 BASIC** 指不通过操作系统,而以 BASIC 单独的系统去完成从启动到执行之间全部动作的方式。

▶ **结构化程序设计** 结构化程序设计这一观点,是荷兰人 E. W. Dijkstra 所提倡的程序设计的方法论。这个方法论建立在明确的算法与数据结构之上,主张应用现代流程控制结构来进行设计,从而产生出效率高、可靠性强的程序。

* * * *

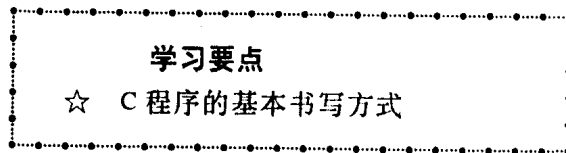
BASIC 是 1964 年在美国达特茅斯大学专门为初学者开发出来的一种会话型语言。那时候,BASIC 作为通用计算机的分时系统用语,被应用于学生的学习。通用机水平时代的 BASIC,并非是一帆风顺的。不过到了 1975 年,Bill Gates 将 BASIC 移植到个人计算机上,这才使 BASIC 作为个人计算机的主要语言而大显风采。随着个人计算机的迅速普及,BASIC 终于在众多计算机语言的角逐中获得稳固的地位。

BASIC 能有今天的地位,应当归功于以下两个方面的背景因素。其一,是个人计算机厂商在当时只向用户提供独立 BASIC。其二,是 BASIC 本身所具有的使用方便、平易近人等特色,形成了令用户倍感亲切的气氛环境,因而受到广大初学者的欢迎。

然而,从计算机语言的角度来衡量,传统 BASIC 在语言规范方面还不能说是很出色的。比如,它没有块结构,现代流程控制结构不完备,数据类型也不丰富。同时,它还不能向子程序传递自变量。从结构化程序设计的概念来看,这些都是传统 BASIC 所难以突破的极限。

“C 语言的崛起,使结构化程序设计的理想得以实现。同时,C 也具备了在实际应用中使用的方便的特色。因此,作为一种超过传统 BASIC 的语言,C 语言近来特别引人注目。”

§ 1.2 最简单的程序



● —— 显示“Hello”的程序

如果用 C 语言来书写在屏幕上显示“Hello”的程序,则有如下表达方式。

```
main( )
{
    printf("Hello\n");
}
```

这就是 C 最简单的程序。我们再进一步应用这个程序,让“Hello”显示 10 次。

```
main( )
{
    int i;
    for (i=1;i<=10;i++)
        printf("Hello\n");
}
```

现在,从上面的程序出发,归纳一下已经了解到的有关 C 的程序。

●——名叫 main 的函数

C 的程序是以 main() 这种形式开头的。程序的主体,安置在 { } 之间。这样,C 的程序便有如下格式:

```
main( )
{
    程序主体
}
```

C 的基本单位称为函数,几个函数汇集起来,便构成了程序。简单的程序,只由 main 一个函数构成。

关于函数,第 6 章中将作详细的讲解,眼下尽可听之任之。

●——构成程序的单词

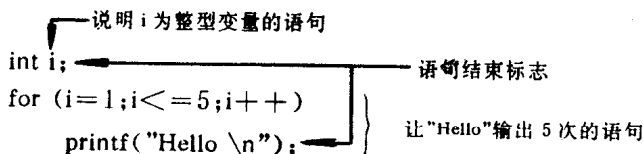
我们日常所使用的语言,是将名词、动词等单词 按规则排列起来形成文章的。程序的情况也一样,是将如下表分类的单词排列起来写成文章的。

保留字	运算符	函数	变量	常数
int for	= <= ++	printf main	i	10 "Hello"

●——排列单词造句

应用 for(保留字)、printf(标准函数)、i(变量)等单词来描述一件工作内容的语言单位叫做语句。

在汉语中,句子的结尾要加句号(。),而在 C 中,是加分号(;)来作为语句的结束记号的。例如:

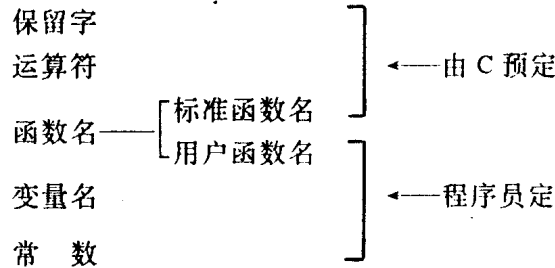


§ 1.3 构成程序的单词

学习要点

- ☆ 构成 C 语言程序的是些什么样的单词
- ☆ 这些单词有何作用

构成程序的单词可分为两大类。一类是 C 预先定好的单词比如保留字；一类是由程序员来决定的单词。



●——保留字(关键字)

【名词解释】

▶ ANSI C ANSI 是美国国家标准学会的简称。为了使多种 C 语言规范达到标准化,ANSI 的 X3 委员会从 1983 年开始工作,到 1985 年,制定出了 ANSI C 语言标准规范方案并公诸于世。

▶ K&R 曾经在贝尔研究所参加过 C 语言研究的 B. W. Kernighan 和 D. M. Ritchie 合写了一本叫《程序语言 C》的书来对 UNIX C 进行解说。书中提及到的 C 的规范,便叫做 K&R 规范。

* * * *

被称为保留字的词汇,是 C 编程时的最基本的词汇。它们在 ANSI C 中规定如下:

· 保留字

auto	double	int	struct
break	else	long	switch
case	<u>enum</u>	register	typedef
char	extern	return	union
<u>const</u>	float	short	unsigned
continue	for	<u>signed</u>	<u>void</u>
default	goto	sizeof	<u>volatile</u>
do	if	static	while

(注)带 的是对 K&R 规范的补充。

●——函数名

【名词解释】

▶ 标准函数 厂商作为标准库提供的基本函数。有 printf、scanf 等。

▶ 用户函数 用户自我定义的函数。

* * * *

编制程序时,如果没有基本处理是很不方便的。比如,数据的输入处理和输出处理就是如

此。提供 C 的厂商，已经预先把这些基本的处理放在叫做标准函数的程序里了。给这些标准函数所取的名字，便是标准函数名。

在 ANSI C 中，规定有如下所示标准函数。但在实际的 C 处理系统中，还有其它一些标准函数。

• ANSI 库

abort	ferror	isctrll	putc	strlen
abs	fflush	isdigit	putchar	strncat
acos	fgetc	isgraph	puts	strncmp
asctime	fgetpos	islower	qsort	strncpy
asin	fgets	isprint	raise	strpbrk
assert	floor	ispunct	rand	strchr
atan	fmod	isspace	realloc	strspn
atan2	fopen	isupper	remove	strstr
atexit	fprintf	isxdigit	rename	strftime
atof	fputc	labs	rewind	strtod
atoi	fputs	ldexp	scanf	strtok
atol	fread	ldiv	setbuf	strtol
bsearch	free	localtime	setjmp	strtoul
calloc	freopen	log	setvbuf	system
ceil	frexp	log10	signal	tan
clearerr	fscanf	longjmp	sin	tanh
clock	fseek	malloc	sinh	time
cos	fsetpos	memchr	sprintf	tmpnam
cosh	ftell	memcmp	sqrt	tmpfile
ctime	fwrite	memcpy	srand	tolower
difftime	getc	memmove	sscanf	toupper
div	getchar	memset	strcat	ungetc
exit	getenv	mktime	strchr	va_arg
exp	gets	modf	strcmp	va_end
fabs	gmtime	perror	strcpy	va_start
fclose	isalnum	pow	strncpy	vprintf
feof	isalpha	printf	strerror	vsprintf

除了厂商提供的标准函数外，程序员还可自己编制函数。这类自制函数被称为用户函数。用户函数的编制方法在第 6 章中讨论。

●——运算符

【名词解释】

- ▶ 增量运算符 对变量的内容进行+1的运算符。以++表示。
- ▶ 算术运算符 进行加减乘除的运算符。

运算这个词显得有点生硬，读者只要简单地联想到加、减、乘、除等的计算就行了。进行这类运算的，便是运算符。

* * * *

在 C 中也算做运算符

`i = 5 + 6;`

↑
运算符

上面这个语句的意思是,5 与 6 相加,将其结果赋入变量 i。+, 是进行加算的运算符。

在 C 中,除了加减乘除的运算符外,还有各种各样的奇特的运算符。比如上面的=,是将右边的结果代入左边的赋值运算符。另外,前面的程序中曾经出现过的 i++ 的 ++,被称为增量运算符,是对 i 的内容进行 +1 的运算符。

以下是 C 所使用的运算符中专门进行加减乘除的运算符(算术运算符)。

• 算术运算符

运算符	-	*	/	%	+	-
功能	负号	乘法	除法	求余(模运算)	加法	减法

●——变量与常数

只靠 for 或 printf 等由 C 预先规定的单词,是写不了程序的。

比如,把重复 5 次的指令写成

```
for(5)
```

这虽然很简单,但却是不管用的。实际上,应该这样写才对:

```
for(i=1;i<=5;i++)
```

也就是说,计算机还没有那么强的理解力,重复 5 次这项工作,还需要借用 i 这个工具来进行管理。

在程序的运行中,改变其值的称为变量,其值不变的称为常数。一般可把变量看作是存放值的盒子,为了区别这类盒子而命名的标识符,称为变量名。比如,

```
a=10;
```

这个赋值语句中的 a 是变量,10 是常数。此时,变量 a 的值是 10。现在,我们这样来写

```
a=a+1;
```



这个语句的意思是,给 a 的内容加 1,并将其作为 a 的值代入。于是,a 的值就变成了 11。

这就是说,变量 a 好比一个盛值的容器,根据赋入的值(数)的不同,其内容也将发生变化。因此,从“变化的量”这个意思出发,将其称之为变量。

●——标识符的命名规则

变量名和函数名都是由程序命名的标识符。因此,从原则上讲,命名什么样的标识符都可以。不过,在命名标识符时,必须遵从以下各项规则。

- 可以使用的字符是 A~Z、a~z、0~9 以及 _ (下划线)。
- 标识符必须由字母或下划线开头。因而,像 2a 这样的标识符是错误的。
- 标识符没有长短限制,但从前往后数,只到第 32 位字符有效。
- 保留字 (if, for 等) 不能使用,但可作为标识符的一部分。比如,for 不能得到认可,而 force 却能得到认可。
- 字母的大写和小写是被区别对待的。比如,eof 与 EOF 是不相同的。

问 1

① 设变量 a 的值为 10。语句 $a = a * a + 1$ 执行后,a 的值是多少?

②从下面找出错误的变量名

auto _abc a\$ xl Eof

§ 1.4 数据类型与变量的类型说明

学习要点

- ☆ 什么是数据的类型
- ☆ C 中有怎样的数据类型

●——什么是类型

【名词解释】

▶ 整型 不带小数的数据类型。有 short、int、long 和它们的无符号类型 unsigned short、unsigned int、unsigned long 等共 6 个。

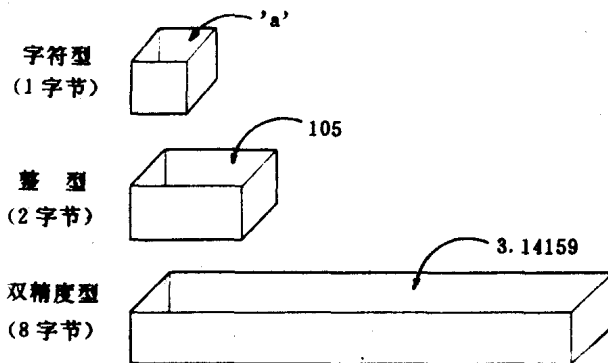
▶ 实型 表达带小数点的数的数据类型。有 float、double、long double 等。

* * * *

我们通常把 5 和 5.0 当作一回事。可在计算机世界里，却非常重视数据的类型，5 和 5.0 会被认作是不同的类型而区别对待。5 这种不带小数点的类型叫做整型，而 5.0 这种带小数点的类型叫做实型。

为什么要这样来区别数据的类型呢？这是因为，将数据赋入变量的时候，还需要解决数据的大小问题。变量这个容器的容量，并不是变幻自如的，它也有定量。因此，它只能接受规定大小的数据。这样，就必须决定好数据的类型，原则上杜绝不同类型数据的混淆。

比如，如果将实型数据强制赋入整型变量，那么，小数点以后就会被舍去。



●——变量的类型说明

程序中所使用的变量，要在程序的前面（所有执行语句之前）进行变量类型的说明。比如，

```
main( )
{
    int a, sum;    ← 说明变量 a 与 sum 为 int 型
    char code;    ← 说明变量 code 为 char 型
```