

软件编程



科技入门一本通之 C 语言基础入门

一、语言语法基础

1、变量与数据类型

在 C 语言中，数据的存储是以变量的形式来实现的。不同类别的数据以不同类型的变量存储在内存中，以供程序使用。

最为常用的数据类型为——整型 (int)、字符型 (char) 和浮点型 (float)，这三种也称为基本数据类型，除此以外还有数组和指针等其他类型。

数据类型	关键字	存储的数据
整型	int	-128~127的整数
字符型	char	-2147483648~2147483647的整数
浮点型	float	-3.4*10 ³⁸ ~3.4*10 ³⁸ 的实数

(P 3-1-1) C 语言基本数据类型

(1) 整型变量 (int)

整型变量可以存储整数，也可以进行加减乘除等运算，在实际程序中使用频率最高。

//int 类型变量的声名及使用示例

```
int a;  
int b = 10;  
int c = a+b;
```

(2) 字符型变量 (char)

字符型变量可以存储单个字符，也可以直接用整数赋值，其值即为该数值在 ASCII 码表中与其对应的字符，常用字符的 ASCII 数值如下。

65	01000001	101	41	A	79	01001111	117	4F	O
66	01000010	102	42	B	80	01010000	120	50	P
67	01000011	103	43	C	81	01010001	121	51	Q
68	01000100	104	44	D	82	01010010	122	52	R
69	01000101	105	45	E	83	01010011	123	53	S
70	01000110	106	46	F	84	01010100	124	54	T
71	01000111	107	47	G	85	01010101	125	55	U
72	01001000	110	48	H	86	01010110	126	56	V
73	01001001	111	49	I	87	01010111	127	57	W
74	01001010	112	4A	J	88	01011000	130	58	X
75	01001011	113	4B	K	89	01011001	131	59	Y
76	01001100	114	4C	L	90	01011010	132	5A	Z
77	01001101	115	4D	M					
78	01001110	116	4E	N					
					97	01100001	141	61	a
					98	01100010	142	62	b
					99	01100011	143	63	c
					100	01100100	144	64	d
					101	01100101	145	65	e
					102	01100110	146	66	f
					103	01100111	147	67	g
					104	01101000	150	68	h
					105	01101001	151	69	i
					106	01101010	152	6A	j
					107	01101011	153	6B	k
					108	01101100	154	6C	l
					109	01101101	155	6D	m
					110	01101110	156	6E	n

111	01101111	157	6F	o
112	01110000	160	70	p
113	01110001	161	71	q
114	01110010	162	72	r
115	01110011	163	73	s
116	01110100	164	74	t
117	01110101	165	75	u
118	01110110	166	76	v
119	01110111	167	77	w
120	01111000	170	78	x
121	01111001	171	79	y
122	01111010	172	7A	z

//char 类型变量的声名及使用示例

```
char a, b, ch;
a = 65;
b = 'B';
ch = a+b;
```

(3) 浮点型变量 (float)

浮点型变量可以存储实数，通常用来进行除法运算并且保存除法运算的结果。

//float 类型变量的声名及使用示例

```
float a = 10;
float b = 3;
float c = a/b;
float d = 10/3;
float e = 10.0/3.0;
```

2、运算符

优先级数值越小，优先级越高；相同优先级的运算，按运算顺序依次进行；可在同级运算中加入圆括号改变运算次序。

3、数组

数组是一个可以存储一个固定大小的相同类型元素的顺序集合，通常用来存储一系列数据，也可认为数组是一系列相同类型的变量。

主要特点：数组在内存中的存储区域是连续的。

// 数组的声名及使用示例

```
int array[10] = {2, 3, 8, 13, 34, 85, 6, 10, 9, 15};
printf(“%d %d”, a[0], a[8]);
最终输出结果：2 9
```

4、程序流程控制

流程控制就是程序代码执行的顺序，总共分为顺序、分支、循环三种结构。

(1) 分支

常用的分支语句有 if 和 switch，根据不同的情况选择不同的程序操作。

if 语句的基本结构为 if-else

if(条件 1)

{

语句 1;

}

else if(条件 2)

{

语句 2;

}

else

{

语句 3;

}

// 只有条件 1 满足 -> 只运行语句 1,

只有条件 2 满足 -> 只运行语句 2

// 两条件都满足 -> 只运行语句 1,

两条件都不满足 -> 只运行语句 3

switch 语句的基本结构为 switch-case

switch(分支变量 n)

{

case x:

语句 1;

break;

case y:

语句 2;

break;

default:

语句 3;

break;

}

//n = x 时 -> 只运行语句 1

//n = y 时 -> 只运行语句 2

//n ≠ x 且 n ≠ y 时 -> 运行语句 3

(2) 循环

常用的循环语句有 `for` 和 `while`，可以进行迭代操作。

循环常用来处理某一项需重复执行的指令或操作，属于较常用的语句。

因为循环语句重复的特性，与数组结合的例子非常常见。

`for` 循环的主要参数有 3 个：

起始值：循环开始时循环变量的值。

循环条件：只有当循环条件满足时，循环才会持续运行，不满足，循环终止。

步长：循环变量每次增加或减少的量，可以控制循环的次数。

```
int i;
for(i = 0; i < 100; i++)
{
    printf(“%d”, i);
}
//i 从 0 增加到 99，每次增加 1
// 每增加一次，运行 printf(“%d”, i)
//i = 100 时，循环结束
```

与 `for` 循环类似，`while` 循环也有循环条件这一参数，但不同点在于 `while` 循环不需要循环变量也不需要设定步长。

常用 `while` 来实现死循环。

```
while(循环条件)
{
    语句 1;
}
// 运行语句 1 直到循环条件不满足为止
```

2、常用语句

```
scanf(“%d %c %f”, &a, &b, &c);
// 输入函数
printf(“%d %c %f”, &a, &b, &c);
// 输出函数
```

二、程序基本框架及常用语句

1、程序基本框架

如下所示为 C 程序的基本框架，实际操作中只需用其他代码替换其中的 [语句 ;] 即可。

```
#include <stdio.h>
int main(void) {
    语句 ;
    return 0;
}
```