

Important part of C language

1. 关键字

C的关键字共有32个

- 数据类型关键字（12个）
char, short, int, long, float, double, unsigned, signed, struct, union, enum, void
- 控制语句关键字（12个）
if, else, switch, case, default
for, do, while, break, continue, goto, return
- 存储类关键字（5个）
auto, extern, register, static, const
- 其他关键字（3个）
sizeof, typedef, volatile

2. 占位符

打印格式	含义
%d	输出一个有符号的 10 进制 int 类型
%o(字母 o)	输出 8 进制的 int 类型
%x	输出 16 进制的 int 类型，字母以小写输出
%X	输出 16 进制的 int 类型，字母以大写输出
%u	输出一个 10 进制的无符号数
%hd	输出 short 类型
%d	输出 int 类型
%ld	输出 long 类型
%lld	输出 long long 类型
%hu	输出 unsigned short 类型
%u	输出 unsigned int 类型
%lu	输出 unsigned long 类型
%llu	输出 unsigned long long 类型

打印格式	含义
%f	输出 float 类型
%lf	输出 double 类型
%e, %E	以科学计数法输出
%c	输出 char 类型
%s	输出 char *类型（字符串）
%p	输出 void *类型（指针的值——地址）
%%	输出一个%

3. ASCII 码

ASCII 值	控制字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符
0	NUL	32	(space)	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	,	71	G	103	g
8	BS	40	(72	H	104	h
9	HT	41)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s

20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	TB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[123	{
28	FS	60	<	92	/	124	
29	GS	61	=	93]	125	}
30	RS	62	>	94	^	126	`
31	US	63	?	95	_	127	DEL

4. 转义字符

转义字符	含义	ASCII 码值（十进制）
\a	警报	007
\b	退格(BS)，将当前位置移到前一行	008
\f	换页(FP)，将当前位置移到下页开头	012
\n	换行(LF)，将当前位置移到下一行开头	010
\r	回车(CR)，将当前位置移到本行开头	013
\t	水平制表(HT)（跳到下一个TAB位置）	009
\v	垂直制表(VT)	011
\\	代表一个反斜线字符“\”	092
\'	代表一个单引号（撇号）字符	039
\"	代表一个双引号字符	034
\?	代表一个问号	063
\0	数字0	000
\ddd	8进制转义字符，d范围0~7	3位8进制
\xhh	16进制转义字符，h范围0~9，a~f，A~F	3位16进制

5. 进制转换

十进制	二进制	八进制	十六进制
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

6. 类型限定符

限定符	含义
extern	声明一个变量，extern 声明的变量没有建立存储空间。 extern int a;//变量在定义的时候创建存储空间
const	定义一个常量，常量的值不能修改。 const int a = 10;
Volatile	防止编译器优化代码
register	定义寄存器变量，提高效率。register 是建议型的指令，而不是命令

限定符	含义
	型的指令，如果 CPU 有空闲寄存器，那么 register 就生效，如果没有空闲寄存器，那么 register 无效。

类型	作用域	生命周期
auto 变量	一对 {} 内	当前函数
static 局部变量	一对 {} 内	整个程序运行期
extern 变量	整个程序	整个程序运行期
static 全局变量	当前文件	整个程序运行期
extern 函数	整个程序	整个程序运行期
static 函数	当前文件	整个程序运行期
register 变量	一对 {} 内	当前函数
全局变量	整个程序	整个程序运行期

7. 运算符优先级

优先级	运算符	名称或含义	使用形式	结合方向	说明
1	[]	数组下标	数组名[常量表达式]	左到右	--
	()	圆括号	(表达式) / 函数名(形参表)		--
	.	成员选择（对象）	对象.成员名		--
	->	成员选择（指针）	对象指针->成员名		--
2	-	负号运算符	-表达式	右到左	单目运算符
	~	按位取反运算符	~表达式		
	++	自增运算符	++变量名/变量名++		
	--	自减运算符	--变量名/变量名--		
	*	取值运算符	*指针变量		
	&	取地址运算符	&变量名		
	!	逻辑非运算符	!表达式		
	(类型)	强制类型转换	(数据类型)表达式		--
	sizeof	长度运算符	sizeof(表达式)		--
3	/	除	表达式/表达式	左到右	双目运算符
	*	乘	表达式*表达式		
	%	余数（取模）	整型表达式%整型表达式		
4	+	加	表达式+表达式	左到右	双目运算符

	-	减	表达式-表达式		
5	<<	左移	变量<<表达式	左到右	双目运算符
	>>	右移	变量>>表达式		
6	>	大于	表达式>表达式	左到右	双目运算符
	>=	大于等于	表达式>=表达式		
	<	小于	表达式<表达式		
	<=	小于等于	表达式<=表达式		
7	==	等于	表达式==表达式	左到右	双目运算符
	!=	不等于	表达式!= 表达式		
8	&	按位与	表达式&表达式	左到右	双目运算符
9	^	按位异或	表达式^表达式	左到右	双目运算符
10		按位或	表达式 表达式	左到右	双目运算符
11	&&	逻辑与	表达式&&表达式	左到右	双目运算符
12		逻辑或	表达式 表达式	左到右	双目运算符
13	?:	条件运算符	表达式 1? 表达式 2: 表达式 3	右到左	三目运算符
14	=	赋值运算符	变量=表达式	右到左	--
	/=	除后赋值	变量/=表达式		--
	=	乘后赋值	变量=表达式		--
	%=	取模后赋值	变量%=表达式		--
	+=	加后赋值	变量+=表达式		--
	-=	减后赋值	变量-=表达式		--
	<<=	左移后赋值	变量<<=表达式		--
	>>=	右移后赋值	变量>>=表达式		--
	&=	按位与后赋值	变量&=表达式		--
	^=	按位异或后赋值	变量^=表达式		--
	=	按位或后赋值	变量 =表达式		--
15	,	逗号运算符	表达式,表达式,...	左到右	--

8. 常用函数

函数	声明	作用	库
printf	int printf(const char *format, ...)	发送格式化输出到标准输出 stdout	stdio

函数	声明	作用	库
scanf	int scanf(const char *format, ...)	从标准输入 <code>stdin</code> 读取格式化输入。	stdio
getchar	int getchar(void)	从标准输入 <code>stdin</code> 获取一个字符（一个无符号字符）	stdio
gets	char *gets(char *str)	从标准输入 <code>stdin</code> 读取一行，并把它存储在 <code>str</code> 所指向的字符串中。当读取到换行符时，或者到达文件末尾时，它会停止	stdio
fprintf	int fprintf(FILE *stream, const char *format, ...)	发送格式化输出到流 <code>stream</code> 中	stdio
fscanf	int fscanf(FILE *stream, const char *format, ...)	从流 <code>stream</code> 读取格式化输入	stdio
fopen	FILE *fopen(const char *filename, const char *mode)	使用给定的模式 <code>mode</code> 打开 <code>filename</code> 所指向的文件	stdio
fread	size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream)	从给定流 <code>stream</code> 读取数据到 <code>ptr</code> 所指向的数组中（可以读取二进制数据）	stdio
fseek	int fseek(FILE *stream, long int offset, int whence)	设置流 <code>stream</code> 的文件位置为给定的偏移 <code>offset</code> ，参数 <code>offset</code> 意味着从给定的 <i>whence</i> 位置查找的字节数（三个文件位置参数：SEEK_SET 开头，SEEK_CUR 当前位置，SEEK_END 结尾）	stdio
fwrite	size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream)	把 <code>ptr</code> 所指向的数组中的数据写入到给定流 <code>stream</code> 中（可以写入二进制数据）	stdio
fgetc	int fgetc(FILE *stream)	从指定的流 <code>stream</code> 获取下一个字符（一个无符号字符），并把位置标识符往前移动	stdio
fgets	char *fgets(char *str, int n, FILE *stream)	从指定的流 <code>stream</code> 读取一行，并把它存储在 <code>str</code> 所指向的字符串内。当读取 (n-1) 个字符时，或者读取到换行符时，或者到达文件末尾时，它会停止	stdio
fputc	int fputc(int char, FILE *stream)	把参数 <code>char</code> 指定的字符（一个无符号字符）写入到指定的流 <code>stream</code>	stdio

函数	声明	作用	库
		中，并把位置标识符往前移动	
fputs	int fputs(const char *str, FILE *stream)	把字符串写入到指定的流 stream 中，但不包括空字符（注意只能操作字符串）	stdio
putchar	int putchar(int char)	把参数 char 指定的字符（一个无符号字符）写入到标准输出 stdout 中	stdio
puts	int puts(const char *str)	把一个字符串写入到标准输出 stdout，直到空字符，但不包括空字符。换行符会被追加到输出中（注意它只能操作字符串）	stdio
fclose	int fclose(FILE *stream)	关闭流 stream。刷新所有的缓冲区	stdio
sprintf	int sprintf(char *str, const char *format, ...)	发送格式化输出到字符串	stdio
sscanf	int sscanf(const char *str, const char *format, ...)	从字符串读取格式化输入	stdio
atof	double atof(const char *str)	把参数 str 所指向的字符串转换为一个浮点数（类型为 double 型）	stdlib
atoi	int atoi(const char *str)	把参数 str 所指向的字符串转换为一个整数（类型为 int 型）	stdlib
atol	long int atol(const char *str)	把参数 str 所指向的字符串转换为一个长整数（类型为 long int 型）	stdlib
calloc	void *calloc(size_t nitems, size_t size)	分配所需的内存空间，并返回一个指向它的指针（所有值被初始化为 0）	stdlib
malloc	void *malloc(size_t size)	分配所需的内存空间，并返回一个指向它的指针	stdlib
realloc	void *realloc(void *ptr, size_t size)	尝试重新调整之前调用 malloc 或 calloc 所分配的 ptr 所指向的内存块的大小（以上 3 个内存分配函数的 size 都是以字节为单位）	stdlib
free	void free(void *ptr)	释放之前调用 calloc、malloc 或 realloc 所分配的内存空间	stdlib

函数	声明	作用	库
exit	void exit(int status)	使程序正常终止（于非 main 函数中使用也会使整个程序终止）	stdlib
abs	int abs(int x)	返回 x 的绝对值（注意这个玩意是在 <code>stdlib.h</code> 里的）	stdlib
div	div_t div(int numer, int denom)	分子除以分母（div_t 类型是一个结构体，里面包含了 int quot——商以及 int rem——余数）	stdlib
labs	long int labs(long int x)	返回 x 的绝对值	stdlib
ldiv	ldiv_t ldiv(long int numer, long int denom)	分子除以分母	stdlib
rand	int rand(void)	返回一个范围在 0 到 RAND_MAX 之间的伪随机数（RAND_MAX 是一个常量，于环境不同会有所不同，但至少大于 32767）	stdlib
srand	void srand(unsigned int seed)	该函数播种由函数 rand 使用的随机数发生器，一般每次程序使用的时候在主程序里调用一次就行	stdlib
abort	void abort(void)	使一个异常程序终止	stdlib
memchr	void *memchr(const void *str, int c, size_t n)	在参数 str 所指向的字符串的前 n 个字节中搜索第一次出现字符 c（一个无符号字符）的位置，返回指向匹配字节的指针或 NULL	string
memcmp	int memcmp(const void *str1, const void *str2, size_t n)	把 str1 和 str2 的前 n 个字节进行比较，若返回值<0，则 str1 小于 str2，返回值=0，str1 等于 str2，返回值>0，str1 大于 str2	string
memcpy	void *memcpy(void *dest, const void *src, size_t n)	从 src 复制 n 个字符到 dest	string
memmove	void *memmove(void *dest, const void *src, size_t n)	另一个用于从 src 复制 n 个字符到 dest 的函数	string
memset	void *memset(void *str, int c, size_t n)	复制字符 c（一个无符号字符）到参数 str 所指向的字符串的前 n 个字符	string
strcat	char *strcat(char *dest,	把 src 所指向的字符串追加	string

函数	声明	作用	库
	const char *src)	到 dest 所指向的字符串的结尾	
strncat	char *strncat(char *dest, const char *src, size_t n)	把 src 所指向的字符串追加到 dest 所指向的字符串的结尾, 直到 n 字符长度为止	string
strchr	char *strchr(const char *str, int c)	在参数 str 所指向的字符串中搜索 第一次 出现字符 c (一个无符号字符) 的位置	string
strcmp	int strcmp(const char *str1, const char *str2)	把 str1 所指向的字符串和 str2 所指向的字符串进行比较	string
strncmp	int strncmp(const char *str1, const char *str2, size_t n)	把 str1 和 str2 进行比较, 最多比较前 n 个字节 (字符串里一个字节就是一个字符)	string
strcpy	char *strcpy(char *dest, const char *src)	把 src 所指向的字符串复制到 dest	string
strncpy	char *strncpy(char *dest, const char *src, size_t n)	把 src 所指向的字符串复制到 dest, 最多复制 n 个字符	string
strlen	size_t strlen(const char *str)	计算字符串 str 的长度, 直到空结束字符, 但不包括空结束字符 (即 '\0')	string
strrchr	char *strrchr(const char *str, int c)	在参数 str 所指向的字符串中搜索 最后一次 出现字符 c (一个无符号字符) 的位置	string
strstr	char *strstr(const char *haystack, const char *needle)	在字符串 haystack 中查找第一次出现字符串 needle (不包含空结束字符) 的位置	string
clock	clock_t clock(void)	返回程序执行起 (一般为程序的开头), 处理器时钟所使用的时间	time
time	time_t time(time_t *timer)	计算当前日历时间, 并把它编码成 time_t 格式	time
acos	double acos(double x)	返回以弧度表示的 x 的反余弦	math
asin	double asin(double x)	返回以弧度表示的 x 的正弦	math
atan	double atan(double x)	返回以弧度表示的 x 的正切	math
atan2	double atan2(double y, double x)	返回以弧度表示的 y/x 的正切。y 和 x 的值的符号决定了正确的象限	math

函数	声明	作用	库
cos	double cos(double x)	返回弧度角 x 的余弦	math
cosh	double cosh(double x)	返回 x 的双曲余弦	math
sin	double sin(double x)	返回弧度角 x 的正弦	math
sinh	double sinh(double x)	返回 x 的双曲正弦	math
tanh	double tanh(double x)	返回 x 的双曲正切	math
exp	double exp(double x)	返回 e 的 x 次幂的值	math
log	double log(double x)	返回 x 的自然对数（基数为 e 的对数）	math
log10	double log10(double x)	返回 x 的常用对数（基数为 10 的对数）	math
pow	double pow(double x, double y)	返回 x 的 y 次幂	math
sqrt	double sqrt(double x)	返回 x 的平方根	math
ceil	double ceil(double x)	返回大于或等于 x 的最小的整数值（向下取整函数）	math
fabs	double fabs(double x)	返回 x 的绝对值（注意只有浮点型的这个 abs 函数是在 math.h 里面）	math
floor	double floor(double x)	返回小于或等于 x 的最大的整数值（向下取整函数）	math
fmod	double fmod(double x, double y)	返回 x 除以 y 的余数	math
isalnum	int isalnum(int c)	该函数检查所传的字符是否是字母和数字	ctype
isalpha	int isalpha(int c)	该函数检查所传的字符是否是字母	ctype
isctrl	int isctrl(int c)	该函数检查所传的字符是否是控制字符	ctype
isdigit	int isdigit(int c)	该函数检查所传的字符是否是十进制数字	ctype
isgraph	int isgraph(int c)	该函数检查所传的字符是否有图形表示法（除了 ' ' 以外任何可以打印的字符）	ctype
islower	int islower(int c)	该函数检查所传的字符是否是小写字母	ctype

函数	声明	作用	库
isprint	int isprint(int c)	该函数检查所传的字符是否是可打印的	ctype
ispunct	int ispunct(int c)	该函数检查所传的字符是否是标点符号字符（非数字、字母、转义字符）	ctype
isspace	int isspace(int c)	该函数检查所传的字符是否是空白字符（' ', '\t', '\n', '\v', '\f', '\r'）	ctype
isupper	int isupper(int c)	该函数检查所传的字符是否是大写字母	ctype
isxdigit	int isxdigit(int c)	该函数检查所传的字符是否是十六进制数字（从 0 - f/F）（若不是则返回 0，以上的 ctype.h 函数同理）	ctype

9. 文件打开的格式

打开模式	含义
r 或 rb	以只读方式打开一个文本文件（不创建文件，若文件不存在则报错）
w 或 wb	以写方式打开文件(如果文件存在则清空文件，文件不存在则创建一个文件)
a 或 ab	以追加方式打开文件，在末尾添加内容，若文件不存在则创建文件
r+或 rb+	以可读、可写的方式打开文件(不创建新文件)
w+或 wb+	以可读、可写的方式打开文件(如果文件存在则清空文件，文件不存在则创建一个文件)
a+或 ab+	以添加方式打开文件，打开文件并在末尾更改文件, 若文件不存在则创建文件