

# 一，关键字：作为C语言的保留字 32

- (1) 存储类型（表示数据的存储位置）：**auto static extern register**
- (2) 基本数据类型（表示数据的格式及数据在内存上所占空间大小）：**unsigned signed char short int long float double enum**
- (3) 控制语句类型：**if else switch case default for while do**
- (4) 辅助控制语句类型：**break continue return goto**
- (5) 构造数据类型：**struct、union、typedef**
- (6) 其他类型：**volatile、void、const、sizeof**

注释：

a, **goto**无条件转移语句

b, 存储类型 (1) **auto**

注：① **auto**定义的变量被称为自动型变量

② **auto**定义的变量可以省略**auto**

③ **auto**定义的变量若未初始化，则为随机值，由系统赋值

④ **auto**定义的变量保存在内存的栈区，由系统开辟空间，释放空间

(2) **static**

注：① **static**修饰的变量被称为静态变量

② **static**修饰的变量保存在静态区/全局区

(3) **extern**

注：① **extern**修饰的变量被称为外部变量

② **extern**修饰的变量可以调用工程中其他文件的同名全局变量

③ **extern**调用外部变量时，需要将**extern**所在文件与被调用的全局变量所在文件同时编译

④ **extern**不能调用由**static**修饰的全局变量

(4) **register**

注：① **register**修饰的变量被称为寄存器变量

② **register**修饰的变量需要向操作系统进行寄存器空间的申请，若没有足够的寄存器空间，将会退化成为**auto**类型变量

c、**void**不是数据类型，但可作为函数的返回值，参数以及修饰指针等，不能定义变量

作业：**volatile**的用法？并举例说明

答：1，用法：告诉编译器，该变量随时会发生变化，每次使用该变量，直接到内存中去取而不是采用暂存在寄存器中的值

2，作用：防止被编译器优化。如果一个变量是易变的，不希望编译器做优化，就在这个变量定义是加volatile

3，举例：（1）对硬件寄存器进行操作 （2）一个中断服务子程序中会访问到的非自动变量 （3）多线程应用中被几个任务共享的变量

## 2、标识符命名规则

一般使用于变量、数组、指针、函数以及结构体等命名

- (1) 不能以数字开头
- (2) 只能由字母、数字、下划线构成
- (3) 不能与关键字重名

```
int_123 _234 _char short 5_t
```

## 3、语句与注释

语句属于程序的基本构成部分，以“;” 结尾

注释不属于程序，一般用于语句的解释说明，有两种格式：**//**注释内容（单行注释，在**//**之后并且在当前行的内容属于注释内容），**/\***注释内容**\*/**（多行注释，在**\*\***之间，不论跨越多少行均属于注释内容）

## 4、程序结构

**#include <stdio.h> //include standard input and output headerfile** 包含标准输入输出头文件，可以使用标准输入输出函数

**int main()**//表示程序起始位置，必须包含，无论是工程中还是单一文件都只能包含一个**main**函数

**{**//表示**main**函数范围

```
printf("helloworld!\n"); //表示函数体，包含函数功能的具体实现过程
```

```
return 0; //表示程序结束
```

```
}
```

# 二，数据类型、命名

**5、数据类型（在32位操作系统下）： long 4/8(64位操作系统)**

32位操作系统->2^32, 1个地址就是一个字节

表示数据的格式和数据在内存上所占空间大小

前提:

- ① 32位操作系统和64位操作系统 一字节对应一个地址
- ② 原码、反码、补码

原码: 表示数据的二进制形式, 一般用于数据显示

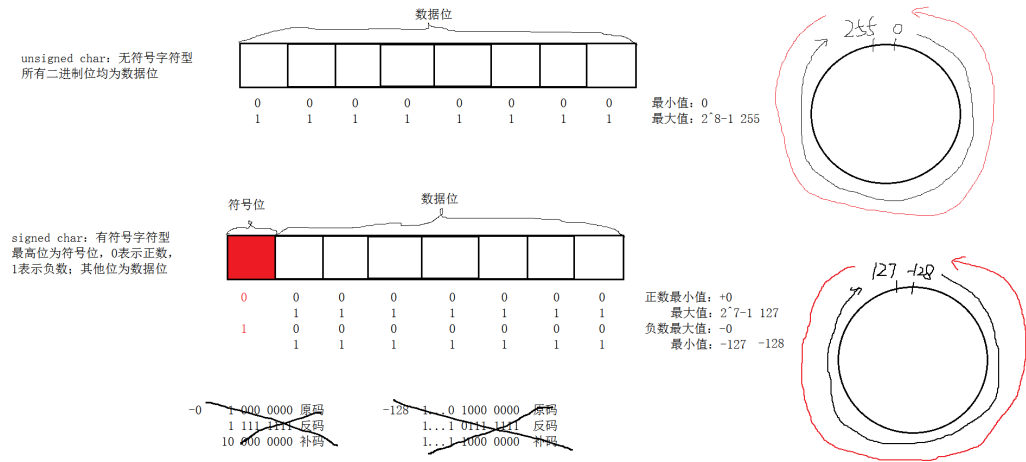
反码: 表示原码除符号位外按位取反, 一般用于过渡

补码: 表示反码除符号位外+1, 一般用于数据存储

- ③ 1字节 = 8bit(二进制位)

(1) 整数类型

数据类型	名称	在内存上所占空间大小(单位: 字节)	值域范围
unsigned char	无符号字符型	1	0 - 255(2^8-1)
signed char	有符号字符型		-128(-2^7) - 127(2^7-1)
unsigned short int	无符号短整型	2	0 - 65535(2^16-1)
signed short int	有符号短整型		-32768(-2^15) - 32767(2^15-1)
unsigned int	无符号整型	4	0 - 2^32-1
signed int	有符号整型		-2^31 - 2^31-1
unsigned long int	无符号长整型	4	0 - 2^32-1
signed long int	有符号长整型		-2^31 - 2^31-1



# 6、命名法

- (1) 驼峰命名法: 指混合使用大小写字母来构成变量和函数的名字

第一个单词以小写字母开始；从第二个单词开始以后的每个单词的首字母都采用大写字母，例如：**myFirstName**、**myLastName**，这样的变量名看上去就像骆驼峰一样此起彼伏，故得名。

（2）匈牙利命名法：变量名=属性+类型+对象描述，其中每一对象<sup>对象</sup>的名称都要求有明确含义<sup>含义</sup>，可以取对象名字全称或名字的一部分。要基于容易记忆容易理解的原则。

**hwnd**：**h**是类型描述，表示句柄，**wnd**是变量对象描述，表示窗口，所以**hwnd**表示窗口句柄；

**g\_cch**：**g\_**是属性描述，表示全局变量<sup>全局变量</sup>，**c**和**ch**分别是计数类型和字符类型<sup>字符</sup>，一起表示变量类型，这里忽略了对对象描述，所以它表示一个对字符进行计数的全局变量。

前缀类型还有：

a	数组 (Array)
b	布尔值 <sup>Q</sup> (Boolean)
by	字节 (Byte)
c	有符号字符 (Char)
cb	无符号字符 (Char Byte, 并没有神马人用的)
cr	颜色参考值 (Color Ref)
cx,cy	坐标差 (长度 Short Int)
dw	双字 (Double Word)
fn	函数 (Function)
h	Handle (句柄)
i	整形 (Int)
l	长整型 (Long Int)
lp	长指针 (Long Pointer)
m_	类成员 (Class Member)
n	短整型 (Short Int)
np	近程指针 <sup>Q</sup> (Near Pointer)
p	指针 (Pointer)
s	字符串 (String)
sz	以 Null 做结尾的字符串型 (String with
w	字 (Word)

### 3、预习浮点型、泊尔类型、常量、变量

泊尔类型: `_Bool` 0/1 头文件是 `#include<stdbool.h>`