**extern用法总结**

　　在C语言中，修饰符extern用在变量或者函数的声明前，用来说明“此变量/函数是在别处定义的，要在此处引用”。

**1. extern修饰变量的声明。**

　　如果文件a.c需要引用b.c中变量int v，就可以在a.c中声明extern int v，然后就可以引用变量v。这里需要注意的是，被引用的变量v的**链接属性**必须是**外链接（external）的**，也就是说a.c要引用到v，不只是取决于在a.c中声明extern int v，还取决于变量v**本身是能够被引用到**的。

　　这涉及到c语言的另外一个话题－－变量的作用域。能够被其他模块以extern修饰符引用到的变量通常是**全局变量**。**还有很重要的一点是，extern int v可以放在a.c中的任何地方**，比如你可以在a.c中的函数fun定义的开头处声明extern int v，然后就可以引用到变量v了，只不过这样只能在函数fun作用域中引用v罢了，这还是变量作用域的问题。对于这一点来说，很多人使用的时候都心存顾虑。好像extern声明只能用于文件作用域似的。

**2. extern修饰函数声明。**

　　从本质上来讲，变量和函数没有区别。函数名是指向函数二进制块开头处的指针。如果文件a.c需要引用b.c中的函数，比如在b.c中原型是int fun(int mu)，那么就可以在a.c中声明extern int fun（int mu），然后就能使用fun来做任何事情。

就像变量的声明一样，extern int fun（int mu）可以放在a.c中任何地方，而不一定非要放在a.c的文件作用域的范围中。对其他模块中函数的引用，最常用的方法是包含这些函数声明的头文件。**使用extern和包含头文件来引用函数有什么区别呢？**extern的引用方式比包含头文件要简洁得多！extern的使用方法是直接了当的，想引用哪个函数就用extern声明哪个函数。

这样做的一个明显的好处是，会加速程序的编译（确切的说是预处理）的过程，节省时间。在大型C程序编译过程中，这种差异是非常明显的。  
**3. 此外，extern修饰符可用于指示C或者C＋＋函数的调用规范。**

　　比如在C＋＋中调用C库函数，就需要在C＋＋程序中用extern “C”声明要引用的函数。

**这是给链接器用的，告诉链接器在链接的时候用C函数规范来链接。主要原因是C＋＋和C程序编译完成后在目标代码中命名规则不同。**