**变量的定义（definition）**:用于为变量**分配存储空间**，还可以为变量指定初始值。在一个程序中，变量有且仅有一个定义；

**变量的声明（declaration）**：用于向程序**表明变量的类型和名字**。

定义也是声明：当定义变量时我们声明了它的类型和名字。可以通过使用extern关键字声明变量名而不定义它。

有分配空间的叫定义，没分配空间的叫声明。

另外，“定义”的同时一定耦合了一个“声明”

简单来说有下面四种情况：

1. 变量和对象不加extern永远是定义，类中的除外。

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/6568584)

1. **int** i; //定义
2. string str; //定义
3. extern **int** i; //声明

int i; //定义

string str; //定义

extern int i; //声明

2. 函数只有函数头是声明，有函数体是定义。

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/6568584)

1. void fun (**int** i, **char** c) ; // fun是声明, i, char, 也是声明
2. void fun (**int** i, **char** c) // fun是定义
3. {
4. ......
5. }

void fun (int i, char c) ; // fun是声明, i, char, 也是声明

void fun (int i, char c) // fun是定义

{

......

}

3. 如果声明有初始化式，就被当作定义，即使前面加了extern。

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/6568584)

1. extern **double** pi = 3.14159;

extern double pi = 3.14159;

4. 类的情况

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/6568584)

1. class A; //A的声明
2. class A{…};//A的定义
3. class A
4. {
5. static **int** x; //这里的x是声明，即x还没有分配空间
6. static const **int** a; //这里的a是声明，同上
7. void fun(); //这里的函数是声明
8. //非static变量在类定义对象时才分配内存.
9. **int** a; //a是定义，已分配空间
10. **char** c; //c是声明，同上
11. };
12. **int** A::x; //这个才是定义
13. const **int** A::a=11; //这个才是定义

class A; //A的声明

class A{…};//A的定义

class A

{

static int x; //这里的x是声明，即x还没有分配空间

static const int a; //这里的a是声明，同上

void fun(); //这里的函数是声明

//非static变量在类定义对象时才分配内存.

int a; //a是定义，已分配空间

char c; //c是声明，同上

};

int A::x; //这个才是定义

const int A::a=11; //这个才是定义

注：在VC里编译下面代码会出现以下错误

void fun(int a, int a); // a应该是个声明

error C2086: 'a' : redefinition

提示说是重复定义，其实a是声明，但总不能说重复声明吧，再说声明是可以多次的，所以这里提示的‘定义’非我们上面提到的‘定义’。