

```
template<typename T>
typename T::size_type getLength(const T&c)
{
    if (c.empty())
        return 0;
    return c.size();
}
```

//一: typename的使用场合

//(1)在模板定义里,表明其后的模板参数是类型参数。

//typename

//template<typename T, int a, int b> //typename后边跟的是一个类型

//int funcaddv2(T c) { } I

//类模板

//template<class T> //名字为T的模板参数。这里虽然可以用class,但是这里的class不是类定义的意思,不能和类定义时的class混为一

//class myvector { ... };

//typename可以写为class,

//(2)使用类的类型成员,用typename来标识这是一个类型;

//::作用域运算符 访问类中的静态成员的时候, 类名::静态成员名,

//int Time::mystatic = 5;

//::还可以用来访问类型成员

//typename的第二个用法:通知编译器,一个名字代表的是一个类型。这里的typename不能换成class