# <string> <string.h> <cstring>差异总结

* **总结：**
* #include <cstring>   // 不可以定义 string s；可以用到 strcpy 等函数  
  using   namespace   std;  
    
  #include <string>   // 可以定义 string s；可以用到 strcpy 等函数  
  using   namesapce   std;
* #include <string.h>   // 不可以定义 string s；可以用到 strcpy 等函数
* 首先说 cstring 与 string.h:  
  cstring 和 string.h 其实里面都是 C 标准库提供的东西，某些实现中 cstring 的内容  
  就是:    
    namespace   std    
    {    
    #include   <string.h>    
    }    
  cstring 是 C++ 的组成部分，它可以说是把 C 的 string.h 的升级版，但它不是 C 的组成部分。  
  所以如果你用的是 C++，那么请用 cstring, 如果你用的是 C 请用 string.h。
* string 与 cstring:   
  一般一个 C++ 库老的版本带 “.h” 扩展名的库文件，比如 iostream.h，在新标准后的标准库中都有一个不带 “.h” 扩展名的相对应，区别除了后者的好多改进之外，还有一点就是后者的东东都塞进了 “std” 名字空间中。        
  string，它是 C++ 定义的 std::string 所使用的文件，是 string 类的头文件，属于 STL 范畴。它有很多对字符串操作的方法。
* string.h 是 C++ 标准化（1998 年）以前的 C++ 库文件，在标准化过程中，为了兼容以前，标准化组织将所有这些文件都进行了新的定义，加入到了标准库中，加入后的文件名就新增了一个 "c" 前缀并且去掉了. h 的后缀名，所以 string.h 头文件成了 cstring 头文件。但是其实现却是相同的或是兼容以前的。相当于标准库组织给它盖了个章，说 “你也是我的标准程序库的一份子了”
* cstring 代表的是 string.h，但是被封装到了 std 里面，譬如调用 strlen 函数，需要写成 std::strlen(yourstr) 才行，这个使用方法比较符合 C++ 的标准要求 string 就是 C++ 标准库里面的 string 模板（确切地说应该是一个特化的模板），但是他同样包含了 C 风格字符串操作函数的定义（应该是通过包含 string.h 实现的）string.h 就不需要使用名字空间了，这个是 C 风格字符串操作的一个函数库，strlen，strcpy，strcat，strcmp…… 都在这里面了，不过既然是 C 风格的库，当然不需要 namespace 支持了。

## 转自：*http://www.cnblogs.com/maowang1991/p/4181806.html*