

18. 什么情况下必须使用C++的初始化列表

理论而言：

1. 初始化 != 赋值. a. 初始化代表为变量分配内存. 变量在其定义处被编译器初始化(编译时). 在函数中, 函数参数初始化发生在函数调用时(运行时).
b. 赋值代表"擦除对象当前值, 赋予新值". 它不承担为对象分配内存的义务.
2. C++中, 类成员的初始化于初始化列表中完成, 先于构造函数体执行. 即成员真正的初始化发生在初始化列表中, 而不是构造函数体中.

再给说明。

1. 如果类中有一个成员是一个引用, 由于引用必须给予初始值, 因此, 引用必须使用初始化列表.
2. 同理, const属性必须给予初始值, 必须使用初始化列表.
3. 继承类中调用基类初始化构造函数, 实际上就是先构造基类对象, 必须使用初始化列表.

别的不再说明, 什么时候必须使用初始化列表是很明显的; 另外, 简单的说, 任何时候都鼓励使用初始化列表, 一些别的事情(比如在构造函数中分配资源之类的, 请参考RAII, Resource Acquisition Is Initialization)可以放在构造函数体内完成.

参考资料: [什么情况下必须使用C++的初始化列表_WingC的博客-CSDN博客_c++什么时候用初始化列表](#)