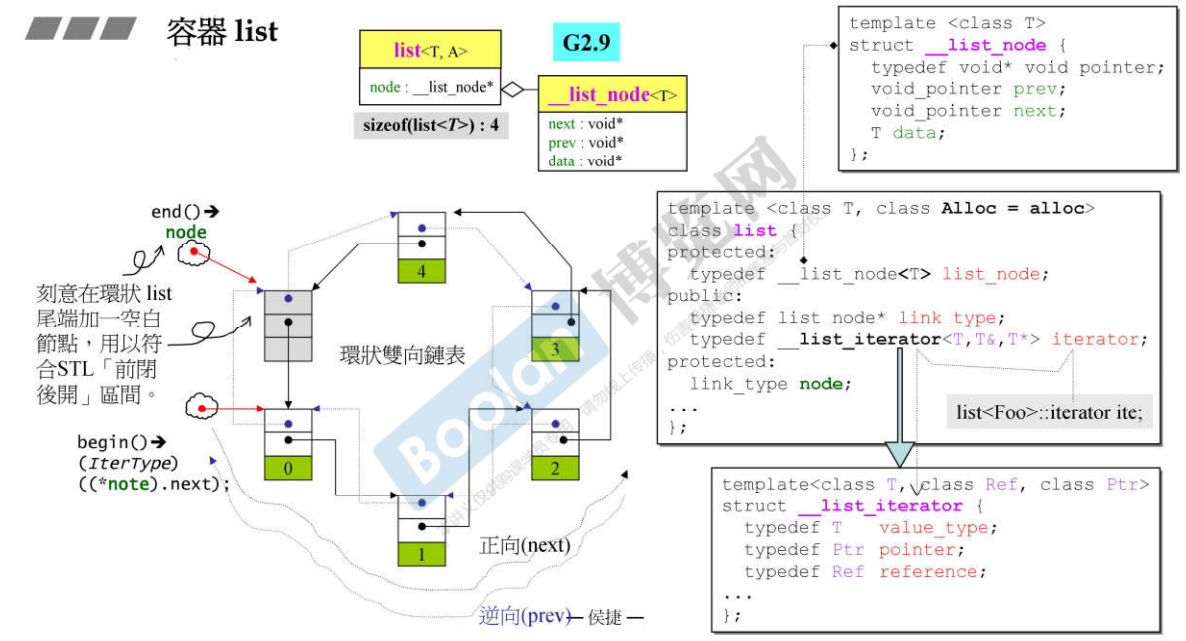
1、



只有一个数据node，类型是link\_type，是一个指向某个东西的指针，不管指向什么，它的大小是4个字节（在32位机器下），因此用sizeof去看一个list的对象，就是4个字节。

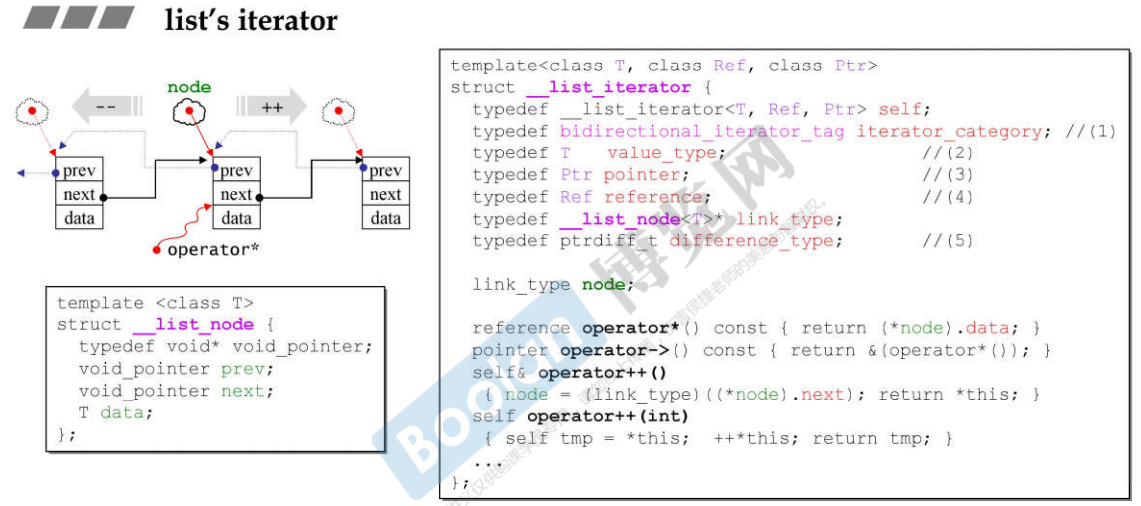
一个list跟它的分配器要内存时，要的是一个结点大小的内存，既两个指针和一个数据的大小内存。

图中红色线被圈起来的点是iterator，这是一个smart pointer，++之会指向下一个结点而不是连续内存的一下个内存。

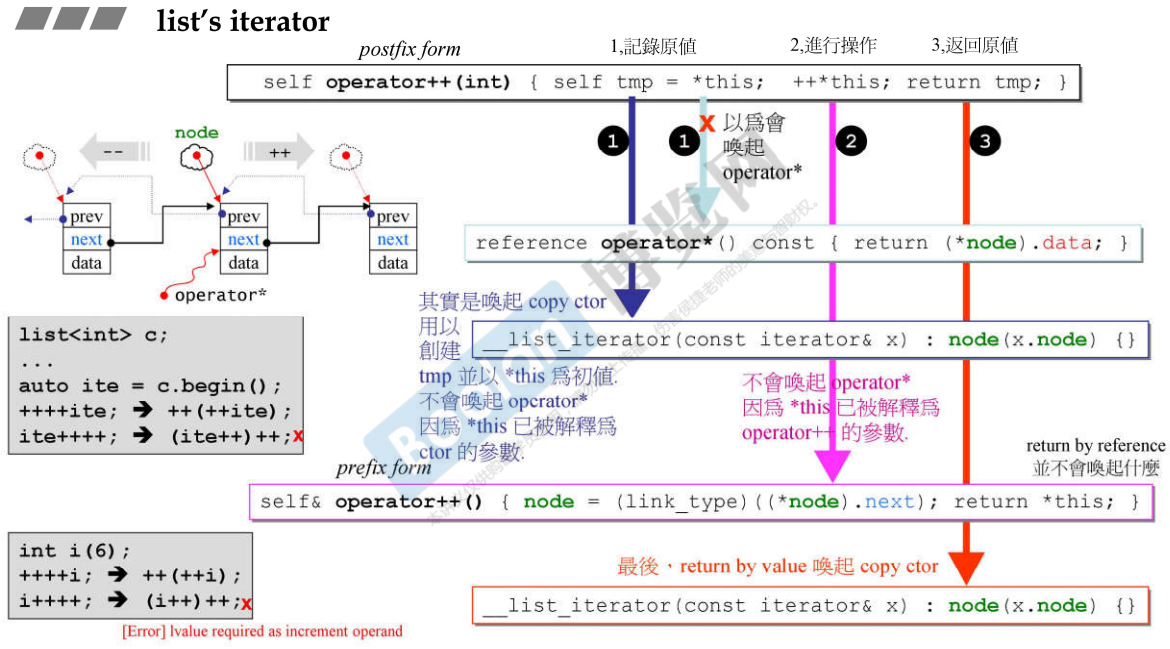
除了array和vector之外的其他所有容器的iterator都是一个class，因为要它们smart起来。

2、

这里是list的iterator设计



容器iterator的设计大同小异，首先一定要有的5个typedef，其他容器也要有，为啥？老师说后面说。其次，都要有大量的操作符重载，因为要模拟指针。



这里查看iterator的++操作，前置操作符是++i没有参数的，而后置i++是有参数(int)的，不过这个参数无意义。++的操作是把原来node指向的结点的next指针的值赋给node，这样node就移动到了下一个结点了。

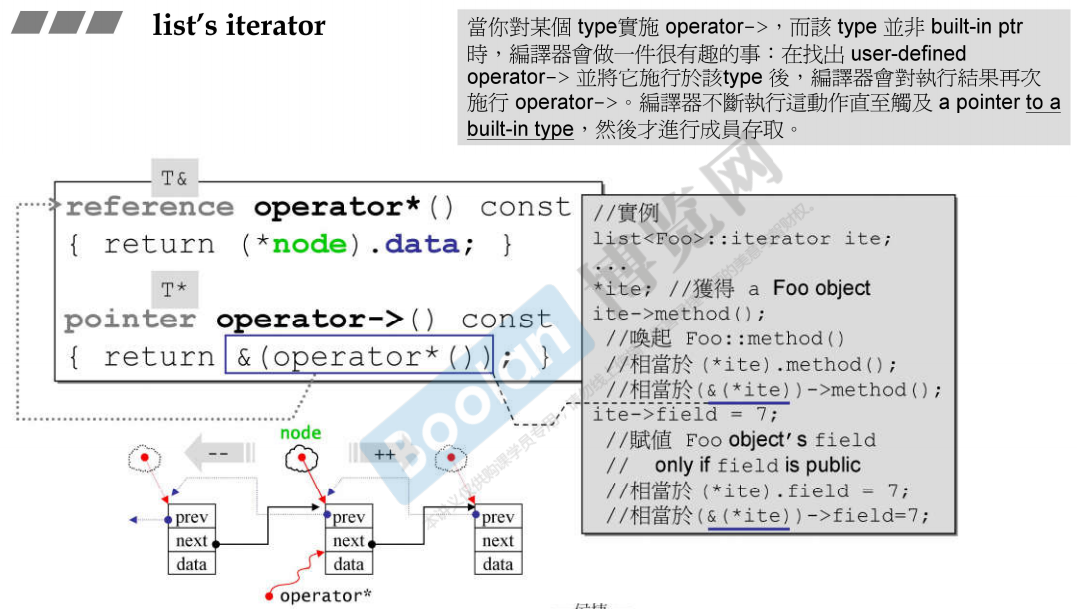
后置++的操作是，先把原来的东西记起来，然后调用前置++这个东西，然后把东西传回去。

这里误以为唤起operator\*其实是错误的，即使\*被重载了，但在遇到重载\*之前，它已经遇到了=，既拷贝构造函数这个assign。

这里的前置++和后置++的返回类型不一样，前置有引用而后置没有，因为在设置的时候要往整数int靠齐。可以连续的前置++，不能连续的后置++，因为后置++返回的不是引用。

3、

这里是list的iterator的用法



除了移动动作++或--，还有提取动作dereference