

# Iterators

## 如何访问关联容器里面的数据

### Iterators

**Key question:** How do we iterate over associative containers?

**Remember:**

Assoc. containers have no notion of a sequence/indexing!

~~for(int i = umm?; i < uhh?; i++ maybe?) {~~

C++ has a solution!

## iterators

### iterators介绍

1. Iterators are something in STL that allow iteration over any container, so all of this sequence and associative containers.

Iterators allow iteration over **any** container,  
whether it is ordered or not.

### iterator的作用

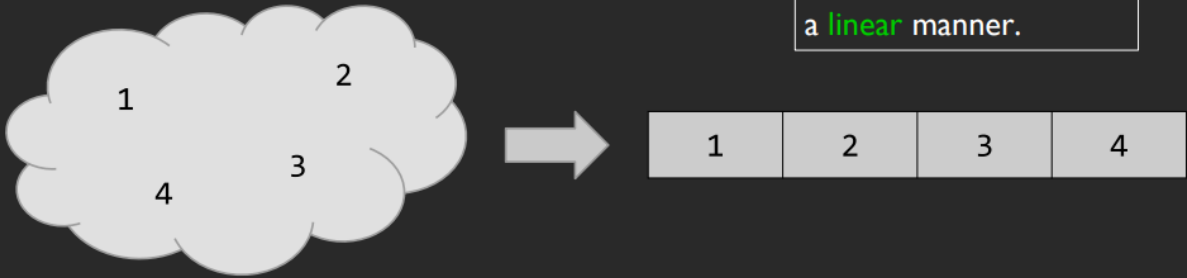
1. iterator are a device that let us view this nonlinear container in a linear manner

# Iterators

Let's try and get a mental model of iterators:

Say we have a `std::set<int> mySet`

Iterators let us view a **non-linear** collection in a **linear** manner.



## iterator的机制是如何实现的

1. the key idea is iterators are like a sort of abstraction.
2. You're not actually worrying about like what exactly how its implemented, you just know that it's pointing to some elements and you can say next or previous on that.

目前我们这个阶段, 还不用掌握 **iterator** 是如何实现的

# Iterators

How are they able to represent a non-linear collection in a “sequential” way?



**We don't care right now.\***

\* They do an in-order traversal on a binary tree.

We will just use them like any other thing - assume they just work somehow.

# iterator的底层实现

1. iterator的底层实现就是指针, 可以将 **iterator** 看作是对指针的封装  
即便 **iterator** 和指针还是有一点不同的, 但是我们仍然可以将 **iterator** 理解为指针(内存)
2. vector iterators are just pure iterators.
3. sequential containers iterators really are pointers under the hood.

## 为什么iterator那么有用

1. 为什么iterator那么有用  
任何的容器都以相同的方式使用 **iterator**

### Iterators

Our examples have used sets, but (almost) **all** C++ containers have iterators.

Why is this powerful?

- Many scenarios require looking at elements, regardless of what type of container is storing those elements.
- Iterators let us go through sequences of elements in a **standardised** way.
- **C++ is huge!**

This standard interface for looping through things is going to be really **powerful**.

## iterator相关操作(使用iterator)

### \* iterator(解引用)

1. 对iterator解引用,得到其所指向的对象

## `++iterator` vs `iterator++`

前缀后缀递增对比	效果
<code>++iterator</code>	先递增 <code>iterator</code> , 然后再使用 <code>iterator</code>
<code>iterator++</code>	先使用 <code>iterator</code> , 然后再递增 <code>iterator</code>

## begin函数

## end函数

## 示例: 对比`vector::iterator`和`vector::operator[]`的效率

1. 当我们使用使用 `for` 循环来遍历vector中的数据时  
是使用 `operator[]` 访问数据的效率高?  
还是使用 `vector::iterator` 访问数据的效率高?  
基本上来说老师觉得两者的效率其实是一样的  
because teacher believe C++ may actually under the hood , be  
using an iterator to implement that.

## 操作总结

### Iterators - Usage

A summary of the essential iterator operations:

Create iterator

Dereference iterator to read value currently pointed to

Advance iterator

Compare against another iterator (especially `.end()` iterator)