[C++] STL 容器 (二) - Iterator

06 Jun 2016

這篇是 STL 容器介紹的第二篇
還沒看過第一篇的可以先看這裡

這篇要介紹的主要是 iterator
iterator 像是一個比較聰明的 pointer
可以指到容器內任何一個位置
然後操作那個位置的資料

要印出整個陣列有兩個方法：

```
// 假如有一個陣列長這樣
// len = 5
int arr[] = {1, 2, 3, 4, 5};
int len = sizeof(arr) / sizeof(int);


/*========== 用 index 印出整個陣列 ==========*/

// 很簡單，應該也是大家最常用的方法
for(int i=0 ; i<len ; i++){
    cout << arr[i] << endl;
}
```

Larry Lu

I am a developer and I love working with people.

About Me
分類文章
最新文章

Larry Lu

I am a developer and I
love working with people.

About Me

分類文章

最新文章


```
/*========== 用 pointer 印出整個陣列 ==========*/

// begin 指向的是 1 那個位置
// end 指向的是 5 後面沒有東西的那個位置
int *begin = arr + 0;
int *end = arr + len;

int *ptr;
for(ptr=begin ; ptr!=end ; ptr++){
    cout << *ptr << endl;   // 1, 2, 3, 4, 5
}
```

同樣要印出整個 **Vector** 也有兩個方法：

```
int arr[] = {1, 2, 3, 4, 5};
vector<int> vec(arr, arr+5);     // vec = [1, 2, 3, 4, 5]
int len = vec.size();            // len = 5


/*========== 用 index 印出整個 Vector ==========*/

// 很簡單跟陣列一樣
for(int i=0 ; i<len ; i++){
    cout << vec[i] << endl;
}


/*========== 用 iterator 印出整個 Vector ==========*/


// begin 指向的是 1 那個位置
// end 指向的是 5 後面沒有東西的那個位置
vector<int>::iterator begin = vec.begin();
vector<int>::iterator end = vec.end();

vector<int>::iterator it;
for(it=begin ; it!=end ; it++){
    cout << *it << endl;
```

```
}

// 跟上面的指標比較一下
// 真的很像吧～



/*
    用 vector<int>::iterator 宣告出來的是 vector<int> 的 iterator
    就像用 int* 宣告出來的是一個指向 int 的指標
    雖然 iterator 的宣告比較麻煩
    但概念 iterator 上跟 pointer 是很接近的
*/
```

如果只是要印出 vector 用 vec[index] 就可以了
但不是每個 STL Container 都像 vector
可以用 vec[index] 取得任意位置的值
而且還有很多操作也會需要 iterator

Vector 的 insert 跟 erase :

```
int arr[] = {1, 2, 3, 4, 5};
vector<int> vec(arr, arr+5);     // vec = [1, 2, 3, 4, 5]


// 把 0 放在 vec.begin() 的位置 -> [0, 1, 2, 3, 4, 5]
vec.insert(vec.begin(), 0);


// 在尾巴加三個 100 -> [0, 1, 2, 3, 4, 5, 100, 100, 100]
vec.insert(vec.end(), 3, 100);


// 移除第 0 個元素 -> [1, 2, 3, 4, 5, 100, 100, 100]
vec.erase(vec.begin());
```


Larry Lu

I am a developer and I
love working with people.

About Me

分類文章

最新文章

```
// 移除最後一個元素 -> [1, 2, 3, 4, 5, 100, 100]
vec.erase(vec.end() - 1);


// 移除前五個元素 -> [100, 100]
vec.erase(vec.begin(), vec.begin() + 5);

cout << vec.size() << endl;     // size = 2
```

Algorithm 內的一些功能

```
#include<algorithm>

int main(){
    int arr[] = {3, 1, 4, 2, 5};
    vector<int> vec(arr, arr+5);    // vec = [3, 1, 4, 2, 5]


    // 排序前三個 -> [1, 3, 4, 2, 5]
    sort(vec.begin(), vec.begin() + 3);


    // 全部排序 -> [1, 2, 3, 4, 5]
    sort(vec.begin(), vec.end());


    // 反轉 -> [5, 4, 3, 2, 1]
    reverse(vec.begin(), vec.end());


    // 找找看 3 有沒有在裡面
    // 找不到就會回傳 vec.end()
    vector<int>::iterator it;
    it = find(vec.begin(), vec.end(), 3);

    if(it != vec.end()){
        cout << "3 在裡面" << endl;
```


Larry Lu

I am a developer and I love working with people.

```
    } else {
        cout << "3 不在裡面" << endl;
    }
}
```

第二篇就到這裡結束
希望大家看完這篇能稍微了解什麼是 iterator
還有怎麼用它
之後應該會再繼續寫下一篇
感覺要講的東西還有好多
只能說 C++ 真是博大精深阿～～

Larry Lu

I am a developer and I love working with people.

GitHub：@Larry850806
FaceBook 粉專：賴瑞的程式筆記
如果有新文章或是看到好的文章也會分享在粉專

About Me
分類文章
最新文章

Related Posts

[實用] 新一代的編輯器 - VSCode 17 Aug 2017

[React.js] 用 @decorator 來裝飾你的 Component 吧！ 08 Apr 2017

[實用] 終端機 session 管理神器 - tmux 14 Feb 2017