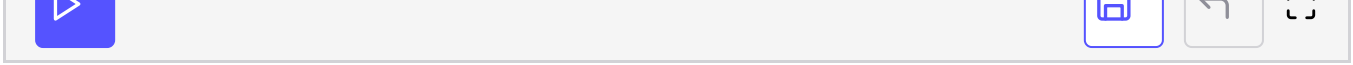
Compare

All the general comparison operators work on containers.

Containers support the comparison operators `==`, `!=`, `<`, `>`, `<=`, and `>=`. The comparison of two containers happens on the elements of the containers. When associative containers are compared, their keys are compared. Unordered associative containers support only the comparison operator `==` and `!=`.

```
// containerComparison.cpp
#include <iostream>
#include <array>
#include <set>
#include <unordered_map>
#include <vector>

using namespace std;

//output 1 represents true and 0 represents false
int main(){
  vector<int> vec1{1, 2, 3, 4};
  vector<int> vec2{1, 2, 3, 4};
  cout << (vec1 == vec2) << endl;       // 1

  array<int, 4> arr1{1, 2, 3, 4};
  array<int, 4> arr2{1, 2, 3, 4};
  cout << (arr1 == arr2) << endl;       // 1

  set<int> set1{1, 2, 3, 4};
  set<int> set2{4, 3, 2, 1};
  cout << (set1 == set2) << endl;       // 1

  set<int> set3{1, 2, 3, 4, 5};
  cout << (set1 < set3) << endl;        // 1

  set<int> set4{1, 2, 3, -3};
  cout << (set1 > set4) << endl;        // 1

  unordered_map<int, string> uSet1{{1, "one"}, {2, "two"}};
  unordered_map<int, string> uSet2{{1, "one"}, {2, "Two"}};
  cout << (uSet1 == uSet2) << endl;     // 0

  return 0;
}
```



Comparison of a container

This concludes the basic operations on containers. In the next chapter, we'll discuss sequential containers in detail.