
8.6. Operations to Generate Iterators

The following member functions return iterators to iterate over the elements of the containers. Table 8.1 lists the iterator category (see Section 9.2, page 433) according to the various container types.

Table 8.1. Required Iterator Categories of Container Types

Container	Iterator Category
Array	Random access
Vector	Random access
Deque	Random access
List	Bidirectional
Forward list	Forward
Set	Bidirectional; element is constant
Multiset	Bidirectional; element is constant
Map	Bidirectional; key is constant
Multimap	Bidirectional; key is constant
Unordered set	Forward; element is constant
Unordered multiset	Forward; element is constant
Unordered map	Forward; key is constant
Unordered multimap	Forward; key is constant
String	Random access

```
iterator container::begin ()
const_iterator container::begin () const
const_iterator container::cbegin () const
```

- Return an iterator for the beginning of the container (the position of the first element).
- If the container is empty, the calls are equivalent to *container* `::end()` or *container* `::cend()`, respectively.
- Note that unordered containers also provide `begin()` and `cbegin()` for a numeric argument to provide the bucket interface (see Section 8.9.3, page 429, for details).
- `cbegin()` is available since C++11.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

```
iterator container::end ()
const_iterator container::end () const
const_iterator container::cend () const
```

- Return an iterator for the end of the container (the position after the last element).
- If the container is empty, the calls are equivalent to *container* `::begin()` or *container* `::cbegin()`, respectively.
- Note that unordered containers also provide `begin()` and `cbegin()` for a numeric argument to provide the bucket interface (see Section 8.9.3, page 430, for details).
- `cend()` is available since C++11.
- Provided by array, vector, deque, list, forward list, set, multiset, map, multimap, unordered set, unordered multiset, unordered map, unordered multimap, string.

```
reverse_iterator container::rbegin ()
const_reverse_iterator container::rbegin () const
const_reverse_iterator container::crbegin () const
```

- Return a reverse iterator for the beginning of a reverse iteration over the elements of the container (the position of the last element).
- If the container is empty, the calls are equivalent to *container* `::rend()` or *container* `::crend()`, respectively.

- For details about reverse iterators, see Section 9.4.1, page 448.
- `crbegin()` is available since C++11.
- Provided by array, vector, deque, list, set, multiset, map, multimap, string.

```
reverse_iterator container::rend ()
const_reverse_iterator container::rend () const
const_reverse_iterator container::crend () const
```

- Return a reverse iterator for the end of a reverse iteration over the elements of the container (the position before the first element).
- If the container is empty, the calls are equivalent to *container* `::rbegin()` or *container* `::crbegin()`, respectively.
- For details about reverse iterators, see Section 9.4.1, page 448.
- `crend()` is available since C++11.
- Provided by array, vector, deque, list, set, multiset, map, multimap, string.