

Categories

Iterators can be categorized into three primary types, each with its own advantages.

Their capabilities can categorize iterator. C++ has forward, bidirectional and random access iterators. With the forward iterator, you can iterate the container forward, with the bidirectional iterator, in both directions. With the random access iterator, you can directly access an arbitrary element. In particular, this means for the last one, that you can use iterator arithmetic and ordering comparisons (e.g.: `<`). The category of an iterator depends on the type of the container used.

In the table below is a representation of containers and their iterator categories. The bidirectional iterator includes the forward iterator functionalities, and the random access iterator includes the forward and the bidirectional iterator functionalities. `It` and `It2` are iterators, `n` is a natural number.

Iterator category	Properties	Container
Forward iterator	<code>++It, It++, *It</code>	unordered associative container
	<code>It == It2, It != It2</code>	<code>std::forward_list</code>
Bidirectional iterator	<code>--It, It--</code>	ordered associative container
		<code>std::list</code>
Random access iterator	<code>It[i]</code>	<code>std::array</code>

	<code>It+= n, It-= n</code>	<code>std::vector</code>
	<code>It+n, It-n</code>	<code>std::deque</code>
	<code>n+It</code>	<code>std::string</code>
	<code>It-It2</code>	
	<code>It < It2, It <= It2,</code> <code>It > It2</code>	
	<code>It >= It2</code>	

The iterator categories of the container

The input iterator and the output iterator are special forward iterators: they can read and write their pointed element only once.