

# Chapter 59. Boost.MinMax

[Boost.MinMax](#) provides an algorithm to find the minimum and the maximum of two values using only one function call, which is more efficient than calling `std::min()` and `std::max()`.

Boost.MinMax is part of C++11. You find the algorithms from this Boost library in the header file `algorithm` if your development environment supports C++11.

Example 59.1. Using `boost::minmax()`

```
#include <boost/algorithm/minmax.hpp>
#include <boost/tuple/tuple.hpp>
#include <iostream>

int main()
{
    int i = 2;
    int j = 1;

    boost::tuples::tuple<const int&, const int&> t = boost::minmax(i, j);

    std::cout << t.get<0>() << '\n';
    std::cout << t.get<1>() << '\n';
}
```

`boost::minmax()` computes the minimum and maximum of two objects. While both `std::min()` and `std::max()` return only one value, `boost::minmax()` returns two values as a tuple. The first reference in the tuple points to the minimum and the second to the maximum.

[Example 59.1](#) writes **1** and **2** to the standard output stream.

`boost::minmax()` is defined in `boost/algorithm/minmax.hpp`.

Example 59.2. Using `boost::minmax_element()`

```
#include <boost/algorithm/minmax_element.hpp>
#include <array>
#include <utility>
#include <iostream>

int main()
{
    typedef std::array<int, 4> array;
    array a{{2, 3, 0, 1}};

    std::pair<array::iterator, array::iterator> p =
        boost::minmax_element(a.begin(), a.end());

    std::cout << *p.first << '\n';
    std::cout << *p.second << '\n';
}
```

Just as the standard library offers algorithms to find the minimum and maximum values in a container, Boost.MinMax offers the same functionality with only one call to the function `boost::minmax_element()`.

Unlike `boost::minmax()`, `boost::minmax_element()` returns a `std::pair` containing two iterators. The first iterator points to the minimum and the second points to the maximum. Thus,

[Example 59.2](#) writes `0` and `3` to the standard output stream.

`boost::minmax_element()` is defined in `boost/algorithm/minmax_element.hpp`.

Both `boost::minmax()` and `boost::minmax_element()` can be called with a third parameter that specifies how objects should be compared. Thus, these functions can be used like the algorithms from the standard library.