

# min 和 max

## 1- 头文件

包含在c++标准库中头文件中，在头文件中定义了min,max的宏，若在包含的同时包含会导致函数无法使用。

提供了\_cpp\_min等函数来代替min函数的功能。

## 2- 函数原型

C++11标准：中min函数的原型

default (1)	<code>template &lt;class T&gt; const T&amp; min (const T&amp; a, const T&amp; b);</code>
custom (2)	<code>template &lt;class T, class Compare&gt; const T&amp; min (const T&amp; a, const T&amp; b, Compare comp);</code>
initializer list (3)	<code>template &lt;class T&gt; T min (initializer_list&lt;T&gt; il);template &lt;class T, class Compare&gt; T min (initializer_list&lt;T&gt; il, Compare comp);</code>

Return the smallest Returns the smallest of a and b. If both are equivalent, a is returned. The versions for initializer lists (3) return the smallest of all the elements in the list. Returning the first of them if these are more than one. The function uses **operator< (or comp, if provided)** to compare the values. 【函数或函数对象】

```
template <class T, class Compare>
const T& min (const T& a, const T& b, Compare comp);
```

## 3- min举例

```
#include<iostream>
#include<algorithm>      /// min and max func
using namespace std;
struct var {
    char *name;
    int key;
    var(char *a,int k):name(a),key(k){}
};
```

```
bool comp(const var& l, const var& r) {           /// 重写 operator <
    return l.key < r.key;
}
int main() {
    var v1("var1", 2);
    var v2("var2", 3);
    cout << std::min(v1, v2, comp).name << endl;    /// var1
    return 0;
}
```

max函数同min. 不过多介绍了。