xutility提供了许多常用的接口，下面列举了一些常用的接口及其功能：

std::pair：用于存储两个值的组合，可以方便地创建和操作键值对。

std::make\_pair：创建一个pair对象。

std::tie：将多个变量绑定到一个tuple中。

std::swap：用于交换两个对象的值。

std::move：用于将一个对象的资源所有权转移到另一个对象。

std::forward：用于将函数参数以原始类型转发。

std::get：用于从tuple中获取指定位置的元素。

std::tuple\_size：用于获取tuple中元素的数量。

std::tuple\_element：用于获取tuple中指定位置的元素类型。

std::begin 和 std::end：用于获取容器的起始迭代器和结束迭代器。

std::find 和 std::find\_if：在容器中查找指定元素或满足指定条件的元素。

std::sort 和 std::stable\_sort：对容器中的元素进行排序。

std::copy 和 std::copy\_if：将容器中的元素复制到另一个容器中。

std::fill：将容器中的元素设置为指定的值。

std::accumulate：对容器中的元素进行累加或累积操作。

std::unique：移除容器中的重复元素。

std::vector, std::list, std::map 等容器类：提供了各种容器类型，方便存储和管理数据。

这只是xutility提供的一部分接口，实际上xutility还包含了许多其他的实用函数和模板类，用于方便地进行常用的编程操作。你可以参考C++标准库的相关文档来了解更多详细的接口和用法。

std::max\_element 是 <algorithm> 头文件中的一个函数模板，用于查找范围内的最大元素。它接受两个迭代器作为参数，表示范围的起始和结束位置，返回指向范围内最大元素的迭代器。

函数模板的定义如下：

cpp

template <class ForwardIt>

ForwardIt max\_element(ForwardIt first, ForwardIt last);

使用示例：

cpp

#include <iostream>

#include <algorithm>

#include <vector>

int main() {

std::vector<int> numbers = {3, 7, 1, 9, 4, 2, 8};

auto maxIt = std::max\_element(numbers.begin(), numbers.end());

std::cout << "The maximum element is: " << \*maxIt << std::endl;

return 0;

}

输出结果为：

The maximum element is: 9