class template

<stack>

**std::stack**

template <class T, class Container = deque<T> > class stack;

**LIFO stack**

Stacks are a type of container adaptor, specifically designed to operate in a LIFO context (last-in first-out), where elements are inserted and extracted only from one end of the container.  
  
**stack**s are implemented as *containers adaptors*, which are classes that use an encapsulated object of a specific container class as its *underlying container*, providing a specific set of member functions to access its elements. Elements are *pushed*/*popped* from the *"back"* of the specific container, which is known as the *top* of the stack.  
  
The underlying container may be any of the standard container class templates or some other specifically designed container class. The container shall support the following operations:

* back
* push\_back
* pop\_back

The standard container classes [vector](http://www.cplusplus.com/vector), [deque](http://www.cplusplus.com/deque) and [list](http://www.cplusplus.com/list) fulfill these requirements. By default, if no container class is specified for a particular stack class instantiation, the standard container [deque](http://www.cplusplus.com/deque) is used.

**Template parameters**

T

Type of the elements.  
Aliased as member type stack::value\_type.

Container

Type of the internal *underlying container* object where the elements are stored.  
Its value\_type shall be T.  
Aliased as member type stack::container\_type.

**Member types**

* [C++98](javascript:switch1.select(1))
* [C++11](javascript:switch1.select(2))

|  |  |  |
| --- | --- | --- |
| **member type** | **definition** | **notes** |
| value\_type | The first template parameter (T) | Type of the elements |
| container\_type | The second template parameter (Container) | Type of the *underlying container* |
| size\_type | an unsigned integral type | usually the same as [size\_t](http://www.cplusplus.com/size_t) |

**Member functions**

[**(constructor)**](http://www.cplusplus.com/reference/stack/stack/stack/)

Construct stack (public member function )

[**empty**](http://www.cplusplus.com/reference/stack/stack/empty/)

Test whether container is empty (public member function )

[**size**](http://www.cplusplus.com/reference/stack/stack/size/)

Return size (public member function )

[**top**](http://www.cplusplus.com/reference/stack/stack/top/)

Access next element (public member function )

[**push**](http://www.cplusplus.com/reference/stack/stack/push/)

Insert element (public member function )

[**emplace**](http://www.cplusplus.com/reference/stack/stack/emplace/)

Construct and insert element (public member function )

[**pop**](http://www.cplusplus.com/reference/stack/stack/pop/)

Remove top element (public member function )

[**swap**](http://www.cplusplus.com/reference/stack/stack/swap/)

Swap contents (public member function )

**Non-member function overloads**

[**relational operators**](http://www.cplusplus.com/reference/stack/stack/operators/)

Relational operators for stack (function )

[**swap (stack)**](http://www.cplusplus.com/reference/stack/stack/swap-free/)

Exchange contents of stacks (public member function )

**Non-member class specializations**

[**uses\_allocator<stack>**](http://www.cplusplus.com/reference/stack/stack/uses_allocator/)

Uses allocator for stack (class template )

**::棧的**

模板<T類，容器類deque的<T>>類棧;

**後進先出棧**

棧是一種類型的容器適配器，專門設計工作在一個LIFO上下文（後進先出），其中的元素只從容器的一端插入和提取。**堆疊** s被實現為*容器的適配器*，該適配器是類使用一個特定的容器類作為其*基礎的容器*封裝對象，提供一組特定的成員函數來訪問它的元素。*推* / *彈出*元素從特定的容器，它被稱為堆棧的*頂部“回”*。 基礎容器可以是任何標準容器類模板或其他一些專門設計的容器類。容器應支持以下操作：

* 後面
* push\_back的
* pop\_back

標準集裝箱班 [向量](http://www.cplusplus.com/vector)， [雙端隊列](http://www.cplusplus.com/deque) 和 [表](http://www.cplusplus.com/list)滿足這些要求。默認情況下，如果沒有容器類指定為特定堆 類的實例化，標準容器 [雙端隊列](http://www.cplusplus.com/deque) 被使用。

**模板參數**

Ť

別名為成員類型的元素類型。堆棧:: value\_type的。

集裝箱

不同的內部存儲元素的*基礎容器*對象。  
value\_type的 應 Ť  
別名為成員類型。堆棧:: CONTAINER\_TYPE的。

**會員類型**

* [C + +98](javascript:switch1.select(1))
* [C + +11](javascript:switch1.select(2))

|  |  |  |
| --- | --- | --- |
| **會員類型** | **定義** | **筆記** |
| value\_type的 | 第一個模板參數（Ť） | 的元素的類型 |
| CONTAINER\_TYPE | 第二個模板參數（集裝箱） | *基礎容器*類型 |
| SIZE\_TYPE | 一個無符號整型 | 通常是一樣 [為size\_t](http://www.cplusplus.com/size_t) |

**會員功能**

[**（構造函數）**](http://www.cplusplus.com/reference/stack/stack/stack/)

構建堆棧（公有成員函數）

[**空的**](http://www.cplusplus.com/reference/stack/stack/empty/)

測試是否是空的容器（公有成員函數）

[**大小**](http://www.cplusplus.com/reference/stack/stack/size/)

返回大小（公眾成員函數）

[**頂部**](http://www.cplusplus.com/reference/stack/stack/top/)

訪問下一個元素（公有成員函數）

[**推**](http://www.cplusplus.com/reference/stack/stack/push/)

插入元素（公有成員函數）

[**放列**](http://www.cplusplus.com/reference/stack/stack/emplace/)

構造和插入元素（公有成員函數）

[**彈出**](http://www.cplusplus.com/reference/stack/stack/pop/)

卸下頂部的元素（公有成員函數）

[**交換**](http://www.cplusplus.com/reference/stack/stack/swap/)

內容交換（公有成員函數）

**非成員函數重載**

[**關係運算符**](http://www.cplusplus.com/reference/stack/stack/operators/)

關係運算符棧（函數）

[**交換（堆棧）**](http://www.cplusplus.com/reference/stack/stack/swap-free/)

堆疊的交換內容（公有成員函數）

**非會員類專業**

[**uses\_allocator <stack>**](http://www.cplusplus.com/reference/stack/stack/uses_allocator/)

使用分配器棧（類模板）