

QStack Class

```
template < typename T> class QStack
```

QStack类是一个提供堆栈的模板类。[更多的...](#)

Header:	#include < QStack>
CMake:	find_package(Qt6 REQUIRED COMPONENTS Core) target_link_libraries(mytarget PRIVATE Qt6::Core)
qmake:	QT += core
Inherits:	QList

- [所有成员的列表，包括继承的成员](#)
- QStack 是[隐式共享类](#)的一部分。

注意：该类中的所有函数都是[reentrant](#)。

公共方法

T	pop()
void	push (const T & <i>t</i>)
void	swap (QStack & <i>other</i>)
T &	top()
const T &	top() const

详细说明

QStack 是 Qt 的泛型之一[container classes](#)。它为相同类型的项目实现堆栈数据结构。

堆栈是后进先出 (LIFO) 结构。使用以下命令将项目添加到堆栈顶部[push\(\)](#) 并使用从顶部检索[pop\(\)](#)。这[top\(\)](#) 函数提供对最顶层项目的访问，而无需删除它。

例子：

```
QStack<int> stack;
stack.push(1);
stack.push(2);
stack.push(3);
while (!stack.isEmpty())
    cout << stack.pop() << Qt::endl;
```

该示例将按顺序输出 3、2、1。

QStack继承自QList。所有的QList的功能也适用于QStack。例如，您可以使用isEmpty() 测试栈是否为空，可以使用以下方式遍历QStackQList的迭代器类（例如，QListIterator）。但除此之外，QStack 还提供了三个方便的函数，可以轻松实现 LIFO 语义：push(),pop ()，和top()。

QStack 的值类型必须是assignable data type。这涵盖了大多数常用的数据类型，但是编译器不允许您存储例如QWidget作为一个值；相反，存储一个QWidget*。

也可以看看QList和QQueue。

成员函数文档

T QStack::pop()

从堆栈中删除顶部项目并将其返回。该函数假设堆栈不为空。

也可以看看top(),push ()，和isEmpty()。

void QStack::push(const T &t)

添加元素t到堆栈的顶部。

这与QList::append()。

也可以看看pop () 和top()。

void QStack::swap(QStack< T> &other)

掉期堆栈other有了这个堆栈。这个操作非常快并且永远不会失败。

T &QStack::top()

返回对堆栈顶部项目的引用。该函数假设堆栈不为空。

这与QList::last()。

也可以看看pop(),push ()，和isEmpty()。

const T &QStack::top() const

这是一个重载功能。

也可以看看[pop \(\)](#) 和[push\(\)](#)。