

C++
Information
Tutorials
Reference
Articles
Forum
Reference
C library:
Containers:
Input/Output:
Multi-threading:
Other:
<algorithm>
<bitset>
<chrono>
<codecvt>
<complex>
<exception>
<functional>
<initializer_list>
<iterator>
<limits>
<locale>
<memory>
<new>
<numeric>
<random>
<ratio>
<regex>
<stdexcept>
<string>
<system_error>
<tuple>
<typeindex>
<typeinfo>
<type_traits>
<utility>
<valarray>
<string>
class templates:
basic_string
char_traits
classes:
string
u16string
u32string
wstring
functions:
stod
stof
stoi
stol
stold
stoll
stoul
stoull
to_string
to_wstring
basic_string
basic_string::basic_string
basic_string::~basic_string
member functions:
basic_string::append
basic_string::assign
basic_string::at
basic_string::back
basic_string::begin
basic_string::capacity
basic_string::cbegin
basic_string::cend
basic_string::clear
basic_string::compare
basic_string::copy
basic_string::crbegin
basic_string::crend
basic_string::c_str
basic_string::data
basic_string::empty
basic_string::end
basic_string::erase
basic_string::find
basic_string::find_first_not_of

class template

std::basic\_string

<string>

```
template < class charT,
           class traits = char_traits<charT>,      // basic_string::traits_type
           class Alloc = allocator<charT>         // basic_string::allocator_type
       > class basic_string;
```

Generic string class

The basic\_string is the generalization of class string for any character type (see string for a description).

Template parameters

charT

Character type.  
The string is formed by a sequence of characters of this type.  
This shall be a non-array POD type.

traits

Character traits class that defines essential properties of the characters used by basic\_string objects (see char\_traits).  
traits::char\_type shall be the same as charT.  
Aliased as member type basic\_string::traits\_type.

Alloc

Type of the allocator object used to define the storage allocation model. By default, the allocator class template is used, which defines the simplest memory allocation model and is value-independent.  
Aliased as member type basic\_string::allocator\_type.

Note: Because the first template parameter is not aliased as any member type, charT is used throughout this reference to refer to this type.

Template instantiations

string	String class (class )
wstring	Wide string (class )
u16string	String of 16-bit characters (class )
u32string	String of 32-bit characters (class )

Member types

C++98 C++11

member type	definition	notes
traits_type	The second template parameter (traits)	defaults to: char_traits<charT>
allocator_type	The third template parameter (Alloc)	defaults to: allocator<charT>
value_type	traits_type::char_type	shall be the same as charT
reference	value_type&	
const_reference	const value_type&	
pointer	allocator_traits<allocator_type>::pointer	for the default allocator: charT*
const_pointer	allocator_traits<allocator_type>::const_pointer	for the default allocator: const charT*
iterator	a random access iterator to charT	convertible to const_iterator
const_iterator	a random access iterator to const charT	
reverse_iterator	reverse_iterator<iterator>	
const_reverse_iterator	reverse_iterator<const_iterator>	
difference_type	allocator_traits<allocator_type>::difference_type	usually the same as ptrdiff_t
size_type	allocator_traits<allocator_type>::size_type	usually the same as size_t

Member functions

(constructor)	Construct basic_string object (public member function )
(destructor)	String destructor (public member function )
operator=	String assignment (public member function )

Iterators:

begin	Return iterator to beginning (public member function )
end	Return iterator to end (public member function )
rbegin	Return reverse iterator to reverse beginning (public member function )
rend	Return reverse iterator to reverse end (public member function )
cbegin	Return const_iterator to beginning (public member function )
cend	Return const_iterator to end (public member function )

[basic\\_string::find\\_first\\_of](#)  
[basic\\_string::find\\_last\\_not\\_of](#)  
[basic\\_string::find\\_last\\_of](#)  
[basic\\_string::front](#)  
[basic\\_string::get\\_allocator](#)  
[basic\\_string::insert](#)  
[basic\\_string::length](#)  
[basic\\_string::max\\_size](#)  
[basic\\_string::operator+=](#)  
[basic\\_string::operator=](#)  
[basic\\_string::operator\[\]](#)  
[basic\\_string::pop\\_back](#)  
[basic\\_string::push\\_back](#)  
[basic\\_string::rbegin](#)  
[basic\\_string::rend](#)  
[basic\\_string::replace](#)  
[basic\\_string::reserve](#)  
[basic\\_string::resize](#)  
[basic\\_string::rfind](#)  
[basic\\_string::shrink\\_to\\_fit](#)  
[basic\\_string::size](#)  
[basic\\_string::substr](#)  
[basic\\_string::swan](#)  
**member constants:**  
[basic\\_string::npos](#)  
**non-member overloads:**  
[getline \(basic\\_string\)](#)  
[operator+ \(basic\\_string\)](#)  
[operator<< \(basic\\_string\)](#)  
[operator>> \(basic\\_string\)](#)  
[relational operators \(basic\\_string\)](#)  
[swap \(basic\\_string\)](#)

<b>crbegin</b>	Return const_reverse_iterator to reverse beginning (public member function )
<b>crend</b>	Return const_reverse_iterator to reverse end (public member function )
<b>Capacity:</b>	
<b>size</b>	Return size (public member function )
<b>length</b>	Return length of string (public member function )
<b>max_size</b>	Return maximum size (public member function )
<b>resize</b>	Resize string (public member function )
<b>capacity</b>	Return size of allocated storage (public member function )
<b>reserve</b>	Request a change in capacity (public member function )
<b>clear</b>	Clear string (public member function )
<b>empty</b>	Test whether string is empty (public member function )
<b>shrink_to_fit</b>	Shrink to fit (public member function )
<b>Element access:</b>	
<b>operator[]</b>	Get character of string (public member function )
<b>at</b>	Get character of string (public member function )
<b>back</b>	Access last character (public member function )
<b>front</b>	Access first character (public member function )
<b>Modifiers:</b>	
<b>operator+=</b>	Append to string (public member function )
<b>append</b>	Append to string (public member function )
<b>push_back</b>	Append character to string (public member function )
<b>assign</b>	Assign content to string (public member function )
<b>insert</b>	Insert into string (public member function )
<b>erase</b>	Erase characters from string (public member function )
<b>replace</b>	Replace portion of string (public member function )
<b>swap</b>	Swap string values (public member function )
<b>pop_back</b>	Delete last character (public member function )
<b>String operations:</b>	
<b>c_str</b>	Get C-string equivalent
<b>data</b>	Get string data (public member function )
<b>get_allocator</b>	Get allocator (public member function )
<b>copy</b>	Copy sequence of characters from string (public member function )
<b>find</b>	Find first occurrence in string (public member function )
<b>rfind</b>	Find last occurrence in string (public member function )
<b>find_first_of</b>	Find character in string (public member function )
<b>find_last_of</b>	Find character in string from the end (public member function )
<b>find_first_not_of</b>	Find non-matching character in string (public member function )
<b>find_last_not_of</b>	Find non-matching character in string from the end (public member function )
<b>substr</b>	Generate substring (public member function )
<b>compare</b>	Compare strings (public member function )

### Non-member function overloads

<b>operator+</b>	Concatenate strings (function template )
<b>relational operators</b>	Relational operators for basic_string (function template )
<b>swap</b>	Exchanges the values of two strings (function template )
<b>operator&gt;&gt;</b>	Extract string from stream (function template )
<b>operator&lt;&lt;</b>	Insert string into stream (function template )
<b>getline</b>	Get line from stream into string (function template )

### Member constants

<b>npos</b>	Maximum value of size_type (public static member constant )
-------------	---



中国还有很多孩子  
吃不饱 穿不暖 >>> 立即