

Search:

Go

Not logged in

register

log in

Reference <string> string getline

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

string

string::string

string::~string

member functions:

string::append

string::assign

string::at

string::back

string::begin

string::capacity

string::cbegin

string::cend

string::clear

string::compare

string::copy

string::cbegin

string::crend

string::c\_str

string::data

string::empty

string::end

string::erase

string::find

string::find\_first\_not\_of

function

## std::getline (string)

<string>

C++98

C++11

(1) istream& getline (istream& is, string& str, char delim);

istream& getline (istream& is, string& str, char delim);

(2) istream& getline (istream& is, string& str);

istream& getline (istream& is, string& str);

### Get line from stream into string

Extracts characters from *is* and stores them into *str* until the delimitation character *delim* is found (or the newline character, '\n', for (2)).

The extraction also stops if the end of file is reached in *is* or if some other error occurs during the input operation.

If the delimiter is found, it is extracted and discarded (i.e. it is not stored and the next input operation will begin after it).

Note that any content in *str* before the call is replaced by the newly extracted sequence.

Each extracted character is appended to the [string](#) as if its member [push\\_back](#) was called.

### Parameters

*is*

[istream](#) object from which characters are extracted.

*str*

[string](#) object where the extracted line is stored.

The contents in the string before the call (if any) are discarded and replaced by the extracted line.

### Return Value

The same as parameter *is*.

A call to this function may set any of the internal state flags of *is* if:

flag	error
eofbit	The end of the source of characters is reached during its operations.
failbit	The input obtained could not be interpreted as a valid textual representation of an object of this type. In this case, <i>dist</i> preserves the parameters and internal data it had before the call. Notice that some eofbit cases will also set failbit.
badbit	An error other than the above happened.

(see [ios\\_base::iostate](#) for more info on these)

Additionally, in any of these cases, if the appropriate flag has been set with *is*'s member function [ios::exceptions](#), an exception of type [ios\\_base::failure](#) is thrown.

### Example

```
1 // extract to string
2 #include <iostream>
3 #include <string>
4
5 int main ()
6 {
7     std::string name;
8
9     std::cout << "Please, enter your full name: ";
10    std::getline (std::cin,name);
11    std::cout << "Hello, " << name << "!\n";
12
13    return 0;
14 }
```

### Complexity

Unspecified, but generally linear in the resulting [length](#) of *str*.

### Iterator validity

Any iterators, pointers and references related to *str* may be invalidated.

### Data races

Both objects, *is* and *str*, are modified.

### Exception safety

**Basic guarantee:** if an exception is thrown, both *is* and *str* end up in a valid state.

string::find_first_of
string::find_last_not_of
string::find_last_of
string::front
string::get_allocator
string::insert
string::length
string::max_size
string::operator+=
string::operator=
string::operator[]
string::pop_back
string::push_back
string::rbegin
string::rend
string::replace
string::reserve
string::resize
string::rfind
string::shrink_to_fit
string::size
string::substr
string::swap
<b>member constants:</b>
string::npos
<b>non-member overloads:</b>
getline (string)
operator+ (string)
operator<< (string)
operator>> (string)
relational operators (string)
swap (string)

See also	
<b>istream::getline</b>	Get line (public member function )
<b>operator&gt;&gt; (string)</b>	Extract string from stream (function )