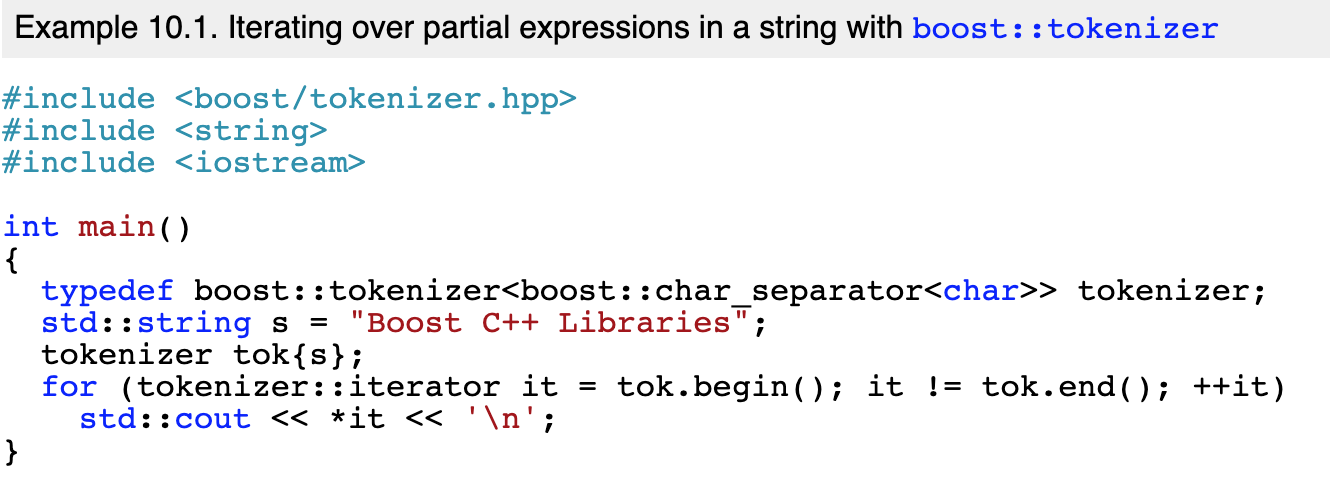
# Boost.Tokenizer

The library [Boost.Tokenizer](http://www.boost.org/libs/tokenizer" \t "_top) allows you to iterate over partial expressions in a string by interpreting certain characters as separators.



Boost.Tokenizer defines a class template called boost::tokenizer in boost/tokenizer.hpp. It expects as a template parameter a class that identifies coherent expressions.

[Example 10.1](https://theboostcpplibraries.com/boost.tokenizer#ex.tokenizer_01) uses the class boost::char\_separator, which interprets spaces and punctuation marks as separators.

A screenshot of a cell phone

Description automatically generated

To keep punctuation marks from being interpreted as separators, initialize the boost::char\_separator object before passing it to the tokenizer.

The constructor of boost::char\_separator accepts a total of three parameters, but only the first one is required. The first parameter describes the individual separators that are suppressed. [Example 10.2](https://theboostcpplibraries.com/boost.tokenizer#ex.tokenizer_02), like [Example 10.1](https://theboostcpplibraries.com/boost.tokenizer#ex.tokenizer_01), treats spaces as separators.

The program will now display Boost, C++ and Libraries.

A screenshot of a cell phone

Description automatically generated

Using the third parameter, the default behavior can be changed.

A screenshot of a cell phone

Description automatically generated

[Example 10.4](https://theboostcpplibraries.com/boost.tokenizer#ex.tokenizer_04) displays two additional empty partial expressions. The first one is found between the two plus signs, while the second one is found between the second plus sign and the following space.

A screenshot of a cell phone

Description automatically generated

[Example 10.5](https://theboostcpplibraries.com/boost.tokenizer#ex.tokenizer_05) iterates over a string of type std::wstring, he class boost::char\_separator must also be initialized with wchar\_t.

A screenshot of a social media post

Description automatically generated

boost::escaped\_list\_separator is used to read multiple values separated by commas. This format is commonly known as CSV (Comma Separated Values). boost::escaped\_list\_separator also handles double quotes and escape sequences.

A screenshot of a cell phone

Description automatically generated

boost::offset\_separator specifies the locations within the string where individual partial expressions end.

<https://theboostcpplibraries.com/boost.tokenizer>