

Member Function Example	Description
<code>mystring.append(n, 'z');</code>	Appends <code>n</code> copies of 'z' to <code>mystring</code> .
<code>mystring.append(str);</code>	Appends <code>str</code> to <code>mystring</code> . <code>str</code> can be a string object or character array.
<code>mystring.append(str, n);</code>	The first <code>n</code> characters of the character array <code>str</code> are appended to <code>mystring</code> .
<code>mystring.append(str, x, n);</code>	<code>n</code> number of characters from <code>str</code> , starting at position <code>x</code> , are appended to <code>mystring</code> . If <code>mystring</code> is too small, the function will copy as many characters as possible.
<code>mystring.assign(n, 'z');</code>	Assigns <code>n</code> copies of 'z' to <code>mystring</code> .
<code>mystring.assign(str);</code>	Assigns <code>str</code> to <code>mystring</code> . <code>str</code> can be a string object or character array.
<code>mystring.assign(str, n);</code>	The first <code>n</code> characters of the character array <code>str</code> are assigned to <code>mystring</code> .
<code>mystring.assign(str, x, n);</code>	<code>n</code> number of characters from <code>str</code> , starting at position <code>x</code> , are assigned to <code>mystring</code> . If <code>mystring</code> is too small, the function will copy as many characters as possible.
<code>mystring.at(x);</code>	Returns the character at position <code>x</code> in the string.
<code>mystring.back();</code>	Returns the last character in the string. (This member function was introduced in C++ 11.)
<code>mystring.begin();</code>	Returns an iterator pointing to the first character in the string. (For more information on iterators, see Chapter 16.)
<code>mystring.c_str();</code>	Converts the contents of <code>mystring</code> to a C-string, and returns a pointer to the C-string.
<code>mystring.capacity();</code>	Returns the size of the storage allocated for the string.
<code>mystring.clear();</code>	Clears the string by deleting all the characters stored in it.
<code>mystring.compare(str);</code>	Performs a comparison like the <code>strcmp</code> function (see Chapter 4), with the same return values. <code>str</code> can be a string object or a character array.
<code>mystring.compare(x, n, str);</code>	Compares <code>mystring</code> and <code>str</code> , starting at position <code>x</code> , and continuing for <code>n</code> characters. The return value is like <code>strcmp</code> . <code>str</code> can be a string object or character array.
<code>mystring.copy(str, x, n);</code>	Copies the character array <code>str</code> to <code>mystring</code> , beginning at position <code>x</code> , for <code>n</code> characters. If <code>mystring</code> is too small, the function will copy as many characters as possible.
<code>mystring.empty();</code>	Returns true if <code>mystring</code> is empty.
<code>mystring.end();</code>	Returns an iterator pointing to the last character of the string in <code>mystring</code> . (For more information on iterators, see Chapter 16.)
<code>mystring.erase(x, n);</code>	Erases <code>n</code> characters from <code>mystring</code> , beginning at position <code>x</code> .
<code>mystring.find(str, x);</code>	Returns the first position at or beyond position <code>x</code> where the string <code>str</code> is found in <code>mystring</code> . <code>str</code> may be either a string object or a character array.
<code>mystring.find('z', x);</code>	Returns the first position at or beyond position <code>x</code> where 'z' is found in <code>mystring</code> .
<code>mystring.front();</code>	Returns the first character in the string. (This member function was introduced in C++ 11.)
<code>mystring.insert(x, n, 'z');</code>	Inserts 'z' <code>n</code> times into <code>mystring</code> at position <code>x</code> .
<code>mystring.insert(x, str);</code>	Inserts a copy of <code>str</code> into <code>mystring</code> , beginning at position <code>x</code> . <code>str</code> may be either a string object or a character array.
<code>mystring.length();</code>	Returns the length of the string in <code>mystring</code> .
<code>mystring.replace(x, n, str);</code>	Replaces the <code>n</code> characters in <code>mystring</code> beginning at position <code>x</code> with the characters in string object <code>str</code> .
<code>mystring.resize(n, 'z');</code>	Changes the size of the allocation in <code>mystring</code> to <code>n</code> . If <code>n</code> is less than the current size of the string, the string is truncated to <code>n</code> characters. If <code>n</code> is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.
<code>mystring.size();</code>	Returns the length of the string in <code>mystring</code> .
<code>mystring.substr(x, n);</code>	Returns a copy of a substring. The substring is <code>n</code> characters long and begins at position <code>x</code> of <code>mystring</code> .
<code>mystring.swap(str);</code>	Swaps the contents of <code>mystring</code> with <code>str</code> .