

The `string` Class

A string, to put it simply, is text. For example, “Welcome to my program” is a string. Naturally, it is useful to be able to manipulate strings as variables for storing names and the like. Text can be considered as an *array of characters*, but is difficult to manipulate as such. The standard library for C++ includes a string class, which permits manipulation of strings similar to other variables. A `string` can be declared just like any other variable. Take a look at this example:

```
#include <iostream>

using namespace std;

int main()
{
    string message = "Your name is ";           // Set on initialization
    string name;
    int nameLength = 0;

    cout << "Enter your name: ";
    cin >> name;                                // Read in via cin
    cout << message;                            // Displayed with cout
    nameLength = name.size();
    for(int x = 0; x < nameLength; x++)         // Loop # of characters in name
    {
        cout << name[x];                        // Display 1 character at a time
    }
    cout << endl << "Your name contains " << nameLength << " letters.";
    return 0;
}
```

Notice that quotation marks are used to surround a string when it appears in code. You can display strings directly in quotation marks called literal strings, or using a variable. You may also treat strings like *one-dimensional arrays* of characters. You may use the `size` helper function to determine the number of characters in the string and the square brackets (`[]`) to access a single letter in the string.

Select `string` Class Members

There are a number of operators and member functions available for manipulating strings, and some of them are summarized in the following table. For additional member functions of the `string` class, please consult additional on-line sources.

Member	Usage Examples
<code>=</code>	Assigns one string to another string: <pre>string S = "Hello", t; t = s;</pre>
<code>==</code> <code>></code> <code>>=</code> <code>!=</code> <code><</code> <code><=</code>	Compare two strings alphabetically: <pre>if (t >= s) cout << "s comes first!"; if (s == t) cout << "Same string!";</pre>
<code>[n]</code>	Like vector notations; Returns the (n+1)th character of the string. <pre>cout << s[0] << s[1];</pre>
<code>+= (str)</code> <code>+= (char)</code>	Appends string <code>str</code> <pre>string name = "Lexington High"; name += " School";</pre> Appends character <code>char</code> <pre>string name = "LH"; name += 'S';</pre>
<code>+</code>	Concatenates (puts together) two strings (or a char and an string): <pre>string name = "Joe"; string t = "Hello " + name; string s = 'a' + name; string w = name + 's';</pre>
<code>size()</code>	Returns the size (number of characters) of the string: <pre>int len = s.size();</pre>

<pre>resize(num) resize(num, ch)</pre>	<p>Changes the size(number of characters) of the string and fills the spots with null or the character stored in ch:</p> <pre>string s = "The"; s.resize(s.size()+5); // s is "The " s.resize(s.size()+2,' '); // s is "The "</pre>
<pre>substr(pos, len)</pre>	<p>Returns len characters of the string beginning at index pos.</p> <pre>string t = "I am king of the world!"; string s = t.substr(5, 5); //king</pre>
<pre>find(str) find(ch)</pre>	<p>Returns the index of the first occurrence of string str in the string, -1 if nothing is found.</p> <p>Can also be used to find first occurrence of a character ch in the string.</p>
<pre>getline(infile, s)</pre>	<p>Reads one line from infile into string s</p>

Select `stdlib` class Members

A list of a few `stdlib` member functions than can be useful for string manipulation

Member	Usage Examples
<pre>atoi(s)</pre>	<p>Returns <code>char * string s</code> converted to an integer. To use with string, use <code>c_str()</code> function. For example:</p> <pre>string nstr = "12345"; int num = atoi(nstr.c_str()); //num is 12345 an int</pre>
<pre>atof(s)</pre>	<p>Returns <code>char * string s</code> converted to an float. To use with string, use <code>c_str()</code> function. For example:</p> <pre>string nstr = "45.23"; float num = atof(nstr.c_str()); //num is 45.23 a decimal</pre>