String Member Functions

length

length() returns the size of the string. Remember there is no **null zero**, so this is the actual length of the string. The size() member function produces the same result, although length is most commonly used.

resize

resize(**int**) allows you to increase or decrease the size of your string. If you decrease the size, then the contents from the index of resize(**int**)'s argument and beyond are deleted. Increasing the size only affects the total size of the string.

clear

clear() takes the string and turns it into an empty string. Nothing is being held in the string.

empty

empty() returns a boolean expression. True if the string is empty and false if the string has characters.

Re-initialize

To re-initialize a string, just use the equals operator just as you would for an **int** or **double** or **char**.

Element/Index

Element/Index (**[]**) allows to access individual element. You may alter this element if you desire. This is akin to array elements.

at

at(**index**) also allows you to access individual elements. It is a function, so use parentheses versus the square brackets.

front

front() accesses the very first element of the string.

back

back() accesses the very last element of the string.

append

append(string) or append(**"string"**) allows you to add onto a string. The argument may be a string literal or a string variable.

push_back

push_back(**char**) lets you add one character to the end of the string.

insert

insert(**index**, **"string"**) allows you to insert a string at the given index. This will shift over all characters to the right of the index.

erase

erase(**index start**, **number of elements to erase**) erases all characters starting at the specified index in the first argument until the number of characters to be erased has been reached from the second argument.

replace

replace(**index start**, **number of elements to override**, **string**) erases from the starting index (first argument) until the number of elements to override has been reached (second argument). The **string** (third argument) fits inside that place even if it is longer than the number of elements that were erased.

swap

swap(**string**) switches around two strings. The argument variables now holds the string from the calling object variable and vice versa.

find

find(**string**) or find(**"string"**) searches for the string in the argument. If found, it returns the starting index of the string. If it is not found, it returns **std::string::npos**. **npos** stands for **no position** and is really -1. It may search for a string variable or string literal. **using namespace std;** you can just put **npos**.

find_first_of

find_first_of(**string**) or find_first_of(**"string"**) will search the string for the first occurrence of any character in the function's argument. It will return the position at the first occurrence. If none of the characters are found, **std::string::npos** is returned.

Conclusion

There are plenty more member functions listed at **http://www.cplusplus.com/reference/string/string/**

A lot of the member functions have more than one possible combination of arguments. Full examples are given on the above website. The functions here are some of the most common you will come across.