**Funções de String**

**Erase:** Apaga parte de uma string. Exemplo:  
nomeVarial.erase(índice inicial, alcance);

**Back:** Insere caracteres no final de uma String. Exemplo:

nomeVariavel.back() = ‘caracteres a ser inserido’;

**Empty:** Verifica se a string é vazia. Exemplo:

nomeVariavel.empty();

**Clear:** apaga o conteúdo de uma string. Exemplo:

noveVariavel.clear();

**Substr:** Cria uma nova string a partir de uma anterior. Exemplo:

nomeVariavel2 = nomeVariavel.substr(índice inicial, alcance);

**find:** Retorna o primeiro índice da primeira ocorrência passada por parâmetro. Exemplo:

nomeVariavel.find(“palavra a ser encontrada”);

**rfind:** Retorna o primeiro índice da última ocorrência passada por parâmetro. Exemplo:

nomeVariavel.rfind(“palavra a ser encontrada”);

**pop\_back:** Apaga o último caracter da string. Exemplo:

nomeVariavel.pop\_back();

**push\_back:** Insere caracter(es) no final da string. Exemplo:

nomeVariavel.push\_back(“caracteres a ser inseridos”);

**swap:** Troca o conteúdo de duas strings. Exemplo:

nomeVariavel1.swap(nomeVariavel2);

**front:** Referencia primeiro caractere de uma string. Exemplo:

nomeVariavel.front();

**back:** Referencia ultimo caractere de uma string. Exemplo:

nomeVariavel.back();

**at:** Retorna caracter q está no índice passado por argumento Exemplo:

nomeVariavel.at(indice);

**insert:** Insere caracteres em uma string. Exemplo:

*#include <iostream>*

*#include <string>*

*int* main ()

{

std::string str="to be question";

std::string str2="the ";

std::string str3="or not to be";

std::string::iterator it;

*// used in the same order as described above:*

str.insert(6,str2); *// to be (the )question*

str.insert(6,str3,3,4); *// to be (not )the question*

str.insert(10,"that is cool",8); *// to be not (that is )the question*

str.insert(10,"to be "); *// to be not (to be )that is the question*

str.insert(15,1,':'); *// to be not to be(:) that is the question*

it = str.insert(str.begin()+5,','); *// to be(,) not to be: that is the question*

str.insert (str.end(),3,'.'); *// to be, not to be: that is the question(...)*

str.insert (it+2,str3.begin(),str3.begin()+3); *// (or )*

std::cout << str << '\n';

*return* 0;

}

Output:

|  |
| --- |
| to be, or not to be: that is the question... |