



`std::string`

A Classe String em C++

Prof. Mauro Miazaki

Declaração

```
#include <string>
using namespace std;
```

```
int main(){
    string s;
    return 0;
}
```

OU

```
#include <string>
```

```
int main(){
    std::string s;
    return 0;
}
```

Especificando um valor inicial

```
#include <string>
using namespace std;

int main(){
    string s1 = "texto inicial";
    string s2("texto inicial");

    return 0;
}
```

Leitura

- Lendo do terminal:

```
string s;
```

```
cin >> s;
```

- Lendo do terminal uma linha inteira:

```
getline(cin, s, '\n');
```

Standard input
(Entrada padrão):
teclado

Caractere
terminal da linha

Concatenação

```
string s1 = "Exemplo de strings";  
string s2 = " concatenadas."  
string s3 = s1 + s2;  
  
// Saida impressa:  
//     Exemplo de strings concatenadas.  
cout << s3 << endl;
```

Comparação

```
string senha;  
  
getline(cin, senha, '\n');  
if(senha == "xyzyzy")  
{  
    cout << "Acesso permitido!" << endl;  
}
```

Comparação

```
string s1 = "abcd";  
string s2 = "abdd";
```

```
if(s1 < s2) {  
    cout << "s1 < s2" << endl;  
} else if(s1 > s2) {  
    cout << "s1 > s2" << endl;  
} else if(s1 == s2) {  
    cout << "s1 == s2" << endl;  
}
```

Tamanho

```
string s = "ten chars.";
int tam1 = s.length();
int tam2 = s.size();
```

- Obs.:
 - C++ strings não usam ' $\backslash 0$ ' como elemento terminal;
 - Por isso, deve-se usar `length()` ou `size()` para determinar onde a string termina.

Acesso a elementos individuais (indexação)

```
string s = "Uma string";
```

```
for(int i = 0; i < s.length(); i++)
```

```
{
```

```
    cout << s[i] << endl;
```

```
}
```

Busca

```
int n;  
string s = "Esta é uma string."  
  
// Busca a partir do início da string  
n = s.find("uma");  
cout << n << " >> " << s.substr(n) << endl;  
  
// String não encontrada  
n = s.find("zzz");  
cout << n << endl; // Imprime -1
```

Busca

```
int n;  
string s = "Esta é uma string."  
  
// Busca a partir da posição 5  
n = s.find("uma", 5);  
cout << n << " : " << s.substr(n) << endl;
```

Substring

```
string s = "Hello World!";  
cout << s.substr(0, 5) << endl; // Saída: Hello
```

Posição inicial

Posição final

```
cout << s.substr(s.find("W", 0), 6) << endl;
```

```
cout << s.substr(5) << endl;
```

Posição 5 em diante

Usando uma string como separador

```
string s = "scott>=tiger>=mushroom";
string delimiter = ">=";
string token;
int pos = 0;

while ((pos = s.find(delimiter)) != string::npos)
{
    token = s.substr(0, pos);
    cout << token << endl;
    s.erase(0, pos + delimiter.length());
}
cout << s << endl;
```

Referência

- <http://www.cprogramming.com/tutorial/string.html>
- Outras funções:
<http://www.cplusplus.com/reference/string/string/>