

Bài 14: CƠ BẢN VỀ CHUỖI KÝ TỰ TRONG C++ (AN INTRODUCTION TO STD::STRING)

Xem bài học trên website để ủng hộ Kteam: [Cơ bản về chuỗi ký tự trong C++ \(An introduction to std::string\)](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Ở bài học trước, bạn đã nắm được [TOÁN TỬ QUAN HỆ, LOGIC, BITWISE, MISC VÀ ĐỘ ƯU TIÊN TOÁN TỬ TRONG C++ \(Operators\)](#).

Hôm nay, mình sẽ hướng dẫn về phần **Cơ bản về chuỗi ký tự trong C++ (String)**.

Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

- [NHẬP & XUẤT DỮ LIỆU TRONG C++ \(Input and Output\)](#)

Trong bài ta sẽ cùng tìm hiểu các vấn đề:

- Tổng quan về chuỗi ký tự (std::string)
- Khai báo, khởi tạo và gán giá trị một chuỗi ký tự
- Xuất một chuỗi ký tự (string output)
- Nhập một chuỗi ký tự (string input)
- Một số thao tác cơ bản với chuỗi ký tự

Tổng quan về chuỗi ký tự (std::string)

Bạn đã được tiếp xúc với chuỗi ký tự (string) ngay từ bài học C++ đầu tiên qua chương trình kinh điển bên dưới:

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

int main()
{
    cout << "Hello, HowKteam.com!" << endl;
    return 0;
}
```

Chuỗi ký tự là tập hợp các ký tự được đặt trong dấu ngoặc kép. Dùng để biểu diễn những thông báo, văn bản, ... trong chương trình. Trong chương trình trên, "Hello, HowKteam.com!" chính là một chuỗi ký tự.

Trong C++, kiểu **chuỗi ký tự không được xây dựng sẵn** (không phải là "built-in string") mà được cài đặt trong một lớp của **thư viện chuẩn STL (C++ Standard Template Library)**.

Khai báo, khởi tạo và gán giá trị một chuỗi ký tự

Để sử dụng chuỗi trong C ++, bạn cần phải **#include** thư viện **string** thuộc **namespace std**.

```
#include <string>
// ...
std::string strMyName;
```

Bạn có thể sử dụng **namespace std** để có thể sử dụng string một cách ngắn gọn hơn:

```
#include <string>
using namespace std;
// ...
string strMyName;
```

Tương tự như các biến thông thường, bạn có thể khởi tạo hoặc gán giá trị cho biến string theo nhiều cách:

```
/* Khai báo một chuỗi ký tự */
std::string strString;

/* Khởi tạo một chuỗi theo nhiều cách */
string strString0("");           // Khởi tạo chuỗi strString0 rỗng
string strString1("Kteam");       // Khởi tạo chuỗi "Kteam" cho strString1
string strString2{ "Free Education!" }; // Khởi tạo chuỗi "Kteam" cho strString2
string strString3 = "HowKteam.com"; // Khởi tạo chuỗi "HowKteam.com" cho strString3
string strString4 = strString1;   // Gán giá trị chuỗi strString1 cho strString4

/* Khởi tạo một chuỗi số */
string strNumber = "88";          // Khởi tạo chuỗi "88" cho strNumber, không phải số
```

Chú ý: Khi khởi tạo giá trị là số cho một chuỗi, chuỗi đó không được coi là một số, và không có những thao tác như một biến số học (cộng, trừ, nhân, chia ...). **C++ không tự động chuyển một chuỗi số về giá trị số nguyên (integer) hoặc số chấm động (floating point).**

Xuất một chuỗi ký tự (string output):

Ở bài [NHẬP & XUẤT DỮ LIỆU TRONG C++ \(Input and Output\)](#), bạn đã biết cách nhập xuất dữ liệu trong C++. Đối với chuỗi ký tự, thao tác xuất cũng tương tự:

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

int main()
{
    /* Khởi tạo một chuỗi HowKteam.com cho biến strString */
    string strString("HowKteam.com");

    /* Xuất chuỗi HowKteam.com lên console */
    cout << strString << endl;

    return 0;
}
```

Outputs:



Nhập một chuỗi ký tự (string input)

Đối với các kiểu dữ liệu cơ bản, bạn sử dụng đối tượng **std::cin** để **đọc một thông tin nào đó từ thiết bị nhập chuẩn** (mặc định là bàn phím), **sau đó lưu thông tin đó vào một biến.**

Cùng thử với kiểu dữ liệu std::string:

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

int main()
{
    cout << "Enter your full name: ";
    string strName;
    cin >> strName;

    cout << "Enter your age: ";
    string strAge;
    cin >> strAge;

    cout << "Your name is " << strName << endl;
    cout << "Your age is " << strAge << endl;

    return 0;
}
```

Outputs:



Trong chương trình trên, khi nhập chuỗi "Hello HowKteam.com! Free Education!" và nhấn enter, chương trình đã bỏ qua lần nhập tiếp theo và xuất ra kết quả.

Chú ý: Toán tử >> (**extraction operator**) được dùng chung với **std::cin** chỉ cho phép đọc một dãy gồm các ký tự liền nhau (không có khoảng trắng), nên nó sẽ trả về các ký tự đến khoảng trắng đầu tiên trong chuỗi vừa nhập.

Vì vậy, chương trình có 2 lần yêu cầu nhập chuỗi, nhưng khi bạn nhập một chuỗi có 3 khoảng trắng như trên, nó chỉ cho phép bạn nhập vào 1 lần.

Nhập văn bản bằng std::getline()

Để đọc đầy đủ chuỗi **có khoảng trắng** từ đối tượng nhập của lớp **istream** (ví dụ cin), bạn nên sử dụng hàm **std::getline()** (**trong namespace std**)

Ví dụ:

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

int main()
{
    cout << "Enter your full name: ";
```

```
string strName;  
getline(cin, strName);  
  
cout << "Enter your age: ";  
string strAge;  
getline(cin, strAge);  
  
cout << "Your name is " << strName << endl;  
cout << "Your age is " << strAge << endl;  
  
return 0;  
}
```

Outputs:



Ở chương trình trên, bạn đã nhập được chuỗi ký tự bao gồm cả khoảng trắng với hàm **std::getline()**.

Kết hợp giữ std::cin và std::getline() sẽ gây ra kết quả không mong muốn

Ví dụ:

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

```
int main()
{
    cout << "Enter your age: ";
    int nAge;
    cin >> nAge;

    cout << "Enter your name: ";
    string strName;
    getline(cin, strName);

    cout << "Hello, " << strName << endl;
    cout << "Your age " << nAge << endl;

    return 0;
}
```

Outputs:



Khi bạn nhập một thông tin bất kỳ, mọi ký tự bạn gõ vào bàn phím (kể cả ký tự Enter **\n**) đều được đẩy vào bộ nhớ đệm trước khi được gán vào biến.

Trong chương trình trên, bạn nhập số bằng **std::cin**, chúng chỉ nhận số chứ không nhận được ký tự **Enter** (**\n**), và ký tự **Enter** vẫn còn trong bộ nhớ đệm. Đến khi nhập chuỗi, hàm **std::getline()** nhận được ký tự **Enter** từ bộ nhớ đệm thì kết thúc nhập và chương trình vẫn chạy tiếp. Điều này khiến kết quả bị sai.

Bạn có thể xóa ký tự **Enter** **\n** sau khi sử dụng **std::cin** bằng cách sử dụng phương thức **cin.ignore()** thuộc **namespace std**:

```
// Xóa khỏi bộ nhớ đệm 32767 ký tự, hoặc đến khi gặp ký tự '\n'
```



```
std::cin.ignore(32767, '\n');
```

Ví dụ:

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

int main()
{
    cout << "Enter your age: ";
    int nAge;
    cin >> nAge;

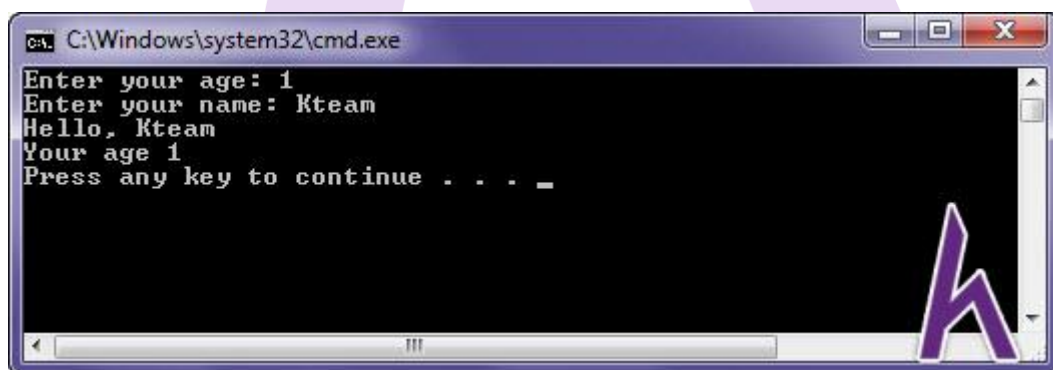
    // Xóa khỏi bộ nhớ đệm 32767 ký tự, hoặc đến khi gặp ký tự '\n'
    std::cin.ignore(32767, '\n');

    cout << "Enter your name: ";
    string strName;
    getline(cin, strName);

    cout << "Hello, " << strName << endl;
    cout << "Your age " << nAge << endl;

    return 0;
}
```

Outputs:



Chú ý: Nên xóa ký tự **Enter** '\n' trong bộ nhớ đệm trước khi nhập chuỗi bất kỳ.

Nhập một văn bản bao gồm ký tự xuống dòng

Mặc định, hàm **std::getline()** sử dụng ký tự '\n' khi nhấn phím Enter là ký tự **báo hiệu kết thúc việc nhập chuỗi**.

Nếu muốn nhập nhiều dòng văn bản vào một biến string, bạn có thể thay đổi nó, ví dụ:

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

int main()
{
    cout << "Enter your text: ";
    string strText;
    getline(cin, strText, '_');

    cout << "Your text: " << endl;
    cout << strText << endl;

    return 0;
}
```

Outputs:



Trong chương trình trên, bạn có thể nhập chuỗi ký tự cho đến khi chương trình nhận vào ký tự gạch dưới '_'. Kết quả nhận được là một chuỗi ký tự gồm nhiều dòng.

Một số thao tác cơ bản với chuỗi ký tự

Nối chuỗi (Appending strings)

Bạn có thể sử dụng **toán tử +** để nối hai chuỗi với nhau, hoặc **toán tử +=** để nối thêm một chuỗi khác.

Ví dụ:

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

int main()
{
    string a("69");
    string b("96");
    string c = a + b; // a and b will be appended, not added

    cout << c << endl;
    a += " string";
```

```
cout << a << endl;  
  
return 0;  
}
```

Outputs:



Chú ý: Khi nối chuỗi, lưu ý là 2 chuỗi số được nối lại với nhau, không phải cộng 2 giá trị số.

Độ dài chuỗi ký tự (String length)

Lớp string định nghĩa cho chúng ta 2 phương thức để thực hiện việc lấy ra độ dài của chuỗi ký tự.

Ví dụ:

```
#include <iostream>  
#include <string>  
using namespace std;  
  
int main()  
{  
    string myName("Hello HowKteam.com!");  
  
    cout << myName << " has " << myName.length() << " characters\n";  
}
```

```
cout << myName << " has " << myName.size() << " characters\n";  
  
return 0;  
}
```

Outputs:



Kết luận

Qua bài học này, bạn đã nắm được [Cơ bản về chuỗi ký tự trong C++ \(std::string\)](#). Std::string là tương đối phức tạp, và còn nhiều tính năng khác mà bạn chưa cần phải biết đến tại thời điểm này. Bạn sẽ được hướng dẫn kỹ hơn ở những bài học sau.

Ở bài tiếp theo, bạn sẽ được học về [BIẾN CỤC BÔ & TẦM VỰC CỦA BIẾN TRONG C++ \(Variable scope\)](#)

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.