



# String Trong C++

---

Xâu Kí Tự

# String là gì?

String là lớp để xử lý chuỗi ký tự trong ngôn ngữ lập trình C++.  
Các bạn có thể nghĩ String như một **mảng ký tự** nhưng có thể mở rộng, thu hẹp và hỗ trợ rất nhiều **hàm xử lý chuỗi thông dụng**.

Index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0
Address	0x23451	0x23452	0x23453	0x23454	0x23455	0x23456



# 1

## Nhập xuất String.

---

# Nhập xuất String

## Tính Chất:

Khi nhập xuất string bạn cần chú ý: Nếu xâu kí tự bạn nhập không có dấu cách thì bạn có thể sử dụng **cin** để nhập, trường hợp có dấu cách thì bạn cần dùng hàm **getline**.

```
int main(){  
    string s;  
    cin >> s; // nhập s từ bàn phím  
    cout << s;  
}
```

## Khai Báo:

```
string name_string;
```





# Nhập xuất String

## Tính Chất:

Trong trường hợp xâu bạn nhập có dấu cách, nếu sử dụng `cin`, bạn chỉ nhập được từ đầu tiên của xâu đó, vì bản chất của `cin` là nó sẽ dừng nhập khi gặp khoảng trắng.

## Chú ý việc trôi lệch khi sử dụng hàm `getline`:

Hãy nhớ rằng không phải cứ trước `getline` là bạn cần xóa bộ đệm, bao giờ trước `getline` mà có câu lệnh `cin` thì mới cần phải xóa bộ đệm. Các bạn xóa đi phím enter trong bộ đệm bằng câu lệnh `cin.ignore()`

## Khai Báo:

```
int main(){
    string s;
    getline(cin, s);
    cout << s << endl;
}
```

```
int x; cin >> x; // cin sẽ để lại enter trong bộ đệm bàn phím
string s;
cin.ignore(1); // Xóa 1 kí tự khỏi bộ đệm bàn phím chính là phím enter của cin để lại
getline(cin, s); // getline đọc phải phím enter và kết thúc việc nhập
cout << s << endl;
```

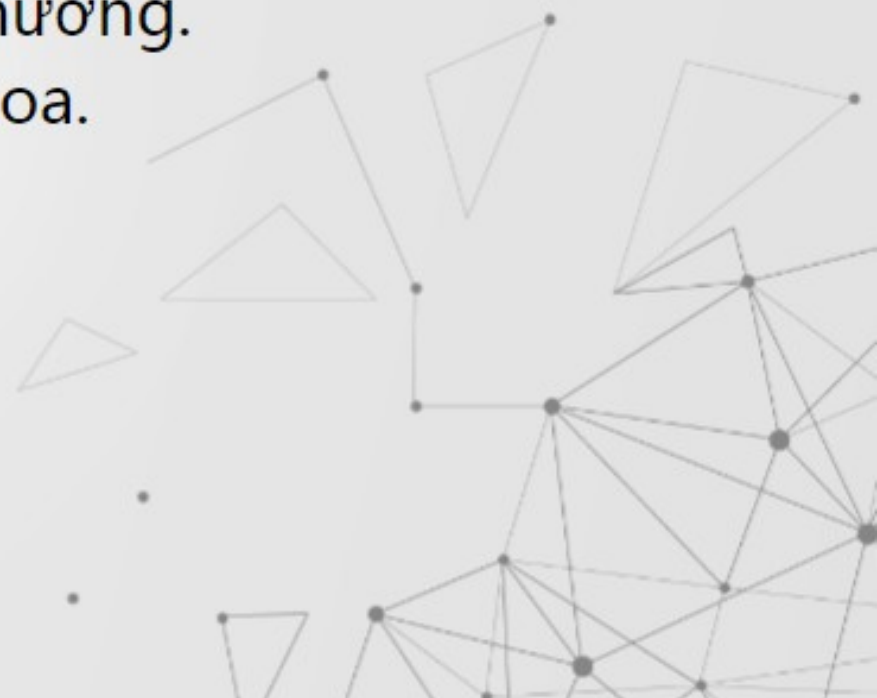
# Một số hàm thông dụng trong String:

- Hàm **insert(index,string)**: Bạn có thể chèn 1 xâu kí tự vào vị trí bất kì trong xâu ban đầu. Bạn chỉ cần cung cấp chỉ số cần chèn và xâu cần chèn cho hàm này.
- Hàm **push\_back** và **pop\_back**: Tương tự như vector thì string cũng hỗ trợ thêm 1 phần tử vào cuối hay xóa 1 phần tử ở cuối xâu. Ở đây các phần tử của string chính là kiểu char vì thế các bạn chỉ có thể push\_back từng kí tự một.
- Hàm **size** và **length**: Cả 2 hàm này đều dùng để xác định chiều dài xâu, hay nói cách khác là số những kí tự xuất hiện trong xâu.
- Hàm **erase**: Bạn có thể chỉ rõ chỉ số bắt đầu xóa kí tự và số lượng kí tự muốn xóa.
- Hàm **clear**: xóa mọi phần tử trong string.
- Hàm **find** : Giả sử bạn cần kiểm tra sự tồn tại của xâu con t trong xâu s. Hàm find trả về chỉ số đầu tiên của xâu t trong xâu s nếu s có chứa t, ngược lại hàm này trả về giá trị **string::npos**.



# Các Hàm Kiểm Tra Ký Tự

- **isdigit(char c)** : Kiểm tra chữ số
- **islower(char c)** : Kiểm tra chữ in thường
- **isupper(char c)**: Kiểm tra in hoa
- **isalpha(char c)**: Kiểm tra chữ cái
- **tolower(char c)** : Chuyển thành chữ in thường.
- **toupper(char c)** : Chuyển thành chữ in hoa.



# Các Hàm Chuyển Đổi Xâu Thành Số Và Ngược Lại

- Để chuyển xâu gồm các chữ số thành số, ta dùng hàm **stoi** để chuyển một xâu thành số **int** và hàm **stoll** để chuyển một xâu thành số

```
string s = "12345";  
int n = stoi(s);  
cout << n << endl;  
string t = "123456729293293";  
long long m = stoll(t);  
cout << m << endl;
```

- Để chuyển một số thành xâu ta dùng hàm **to\_string** (C++11 tr

```
int a = 12345;  
string s = to_string(a);  
cout << s << endl;
```





# THANKS

Full House Tất Cả Là Một Nhà!

CREDITS: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#), and infographics & images by [Freepik](#).

**Please keep this slide for attribution.**