

# C cơ bản

*Dây*

“  
Strings  
”



- Chuỗi trong C
- Chuỗi và con trỏ
- Chức năng chung
- Thoát dấu gạch chéo ngược

## Phần 1

# CHUỖI TRONG C

- Chuỗi là tập hợp các ký tự
- Trong lập trình C, tập hợp các ký tự được lưu trữ dưới dạng mảng. Do đó nó được gọi là dây C.
- Chuỗi C là mảng kiểu char được kết thúc bằng ký tự null, nghĩa là '\0' (Giá trị ASCII của ký tự null là 0).

**Xác định chuỗi C:**

**`char str[] = "aString";`**

- Trong đoạn mã trên, str là một chuỗi và chứa 4 ký tự.
- Mặc dù "C++" có 3 ký tự nhưng ký tự null \0 sẽ tự động được thêm vào cuối chuỗi.

- Trong lập trình C, chuỗi là một chuỗi các ký tự được kết thúc với một ký tự null `\0`. Ví dụ:

**`char str[] = "chuỗi c";`**

- Khi trình biên dịch gặp một chuỗi ký tự kèm theo trong dấu ngoặc kép, nó thêm ký tự null `\0` tại

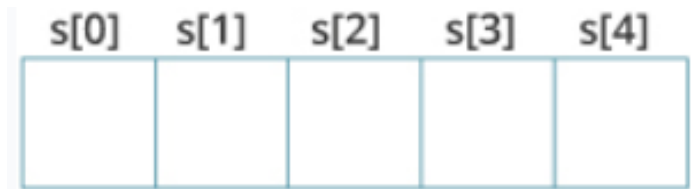
kết thúc theo mặc định.

c		s	t	r	i	n	g	\0
---	--	---	---	---	---	---	---	----

- Khai báo một chuỗi:

```
char s[5];
```

- Và chúng ta đã khai báo một chuỗi gồm 5 ký tự.



**Khởi tạo chuỗi:**

```
char c[] = "abcd";  
  
char c[50] = "abcd";  
  
char c[] = {'a', 'b', 'c', 'd', '\0'};  
  
char c[5] = {'a', 'b', 'c', 'd', '\0'};
```

**Và chúng ta đã khởi tạo một chuỗi gồm 5 ký tự.**

c[0]	c[1]	c[2]	c[3]	c[4]
a	b	c	d	\0

- Gán giá trị cho chuỗi
- ✓ Mảng và chuỗi là công dân hạng hai trong C, họ làm như vậy không hỗ trợ toán tử gán một khi nó được khai báo.

Ví dụ:

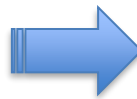
```
char c[100];  
c = "C programming"; // Error! array type is not assignable.
```



## Đọc một chuỗi:

Sử dụng **quét()** chức năng đọc một chuỗi. Các **quét()** Hàm đọc chuỗi ký tự cho đến khi gặp khoảng trắng (dấu cách, dòng mới, tab, v.v.).

```
#include <stdio.h>
int main()
{
    char name[20];
    printf("Enter name: ");
    scanf("%s", name);
    printf("Your name is %s.", name);
    return 0;
}
```



Nhập tên: Joe Biden  
Tên bạn là Joe.

Mặc dù **joe Biden** đã được nhập vào chương trình trên, chỉ **"Joe"** được lưu trữ trong **tên** sợi dây. Đó là bởi vì có một khoảng trống sau **Joe**.

## Để đọc một dòng văn bản:

Sử dụng **fgetc()** hàm đọc một dòng trong chuỗi. Và sử dụng **đặt()** để hiển thị chuỗi.

```
#include <stdio.h>
int main()
{
    char name[30];
    printf("Enter name: ");
    fgets(name, sizeof(name), stdin); // read string
    printf("Name: ");
    puts(name); // display string
    return 0;
}
```



Nhập tên: Joe Biden  
Tên: Joe Biden

*Ở đây, chúng ta đã sử dụng hàm fgets() để đọc một chuỗi từ người dùng.*

- Trong ví dụ trên về cách lấy một dòng văn bản. Chúng tôi đã sử dụng **fgetc()** chức năng đọc một chuỗi từ người dùng:

***fgets(name, sizeof(name), stdin); // đọc chuỗi***

- Các **kích thước(tên)** kết quả là 30. Do đó, chúng ta có thể lấy tối đa 30 ký tự làm đầu vào là kích thước của chuỗi tên.
- Để in chuỗi, chúng tôi đã sử dụng **đặt(tên);**

Phần 2

# CHUỖI VÀ CON TRỎ

Giống như con trỏ và

mảng, mỗi mảng

ký tự trong một chuỗi

tương đương với mỗi

phần tử trong một mảng

```
#include <stdio.h>

int main(void) {
    char name[] = "Harry Potter";

    printf("%c", *name);        // Output: H
    printf("%c", *(name+1));    // Output: a
    printf("%c", *(name+7));    // Output: o

    char *namePtr;

    namePtr = name;
    printf("%c", *namePtr);     // Output: H
    printf("%c", *(namePtr+1)); // Output: a
    printf("%c", *(namePtr+7)); // Output: o
}
```

Có nhiều loại  
lợi ích của  
sử dụng con trỏ để  
các chuỗi điểm.

Ví dụ sau  
để truy cập chuỗi  
thông qua con trỏ:

```
#include<stdio.h>

void main ()
{
    char s[11] = "javatpoint";
    char *p = s; // pointer p is pointing to string s.
    printf("%s",p); // the string javatpoint is printed if we print p.
}
```

javatpoint

...Program finished with exit code 0  
Press ENTER to exit console.

# Chuỗi và con trỏ - 2

char s[11] = "javatpoint"

Nhìn Tại các  
nhân vật vì các  
chi tiết của các  
ví dụ trên

Index	0	1	2	3	4	5	6	7	8	9	10
values	j	a	v	a	t	p	o	i	n	t	\0
Address	20	21	22	23	24	25	26	27	28	29	30
Variable	ptr	char *ptr = s									
Value	20										
Address	10										

## Sao chép nội dung chuỗi bằng con trỏ:

```
#include <stdio.h>

void main ()
{
    char *p = "hello FSofters";
    printf("String p: %s\n",p);
    char *q;
    printf("copying the content of p into q...\n");
    q = p;
    printf("String q: %s\n",q);
}
```



```
String p: hello FSofters
copying the content of p into q...
String q: hello FSofters
```



Khi một chuỗi được xác định, nó không thể được chỉ định lại cho một tập ký tự khác.


Tuy nhiên, bằng cách sử dụng con trỏ, chúng ta có thể chỉ định tập hợp các ký tự vào chuỗi.

Coi như các tiếp theo ví dụ:

```
#include <stdio.h>

void main ()
{
    char *p = "hello FSofters";
    printf("Before assigning: %s\n",p);
    p = "hello";
    printf("After assigning: %s\n",p);
}
```

```
Before assigning: hello FSofters
After assigning: hello
```



# Chuỗi và con trỏ - 5

Sử dụng độ dài của chuỗi. Hãy xem một ví dụ về đếm số nguyên âm trong một chuỗi:

```
#include <stdio.h>

void main ()
{
    char s[14] = "FSoft Academy";
    int count = 0;
    for(int i = 0; i < 14; i++)
    {
        if(*(s+i)=='a' || *(s+i) == 'e' || *(s+i) == 'i' || *(s+i) == 'u' || *(s+i) == 'o')
        {
            count ++;
        }
    }
    printf("The number of vowels %d", count);
}
```



```
The number of vowels 3
...Program finished with exit code 0
Press ENTER to exit console.□
```

## Và sử dụng ký tự null. Đếm số nguyên âm trong một chuỗi:

```
#include <stdio.h>

void main ()
{
    char s[14] = "FSoft Academy";
    int i = 0;
    int count = 0;
    while(*(s+i) != NULL)
    {
        if(*(s+i) == 'a' || *(s+i) == 'e' || *(s+i) == 'i' || *(s+i) == 'u' || *(s+i) == 'o')
        {
            count ++;
        }
        i++;
    }
    printf("The number of vowels %d", count);
}
```



```
The number of vowels 3
...Program finished with exit code 0
Press ENTER to exit console.
```

## Phần 3

# CHỨC NĂNG CHUNG

# Chuỗi trong C: Các hàm thông dụng

KHÔNG.	Chức năng	Sự miêu tả
1)	<u><a href="#">strlen(string_name)</a></u>	trả về độ dài của tên chuỗi.
2)	<u><a href="#">strcpy(đích, nguồn)</a></u>	sao chép nội dung của chuỗi nguồn tới chuỗi đích.
3)	<u><a href="#">strcat(chuỗi đầu tiên, chuỗi thứ hai)</a></u>	concatenates hoặc nối chuỗi đầu tiên với chuỗi thứ hai. Kết quả của chuỗi được lưu trữ trong chuỗi đầu tiên.
4)	<u><a href="#">strcmp(chuỗi đầu tiên, chuỗi thứ hai)</a></u>	so sánh chuỗi đầu tiên với chuỗi thứ hai sợi dây. Nếu cả hai chuỗi đều giống nhau thì trả về 0.

- Hàm strlen() lấy một chuỗi làm đối số và trả về độ dài của nó. Các giá trị trả về có kiểu size\_t (kiểu số nguyên không dấu).

```
# bao gồm <stdio.h>
# bao gồm <string.h>
int chính()
{
    ký tự a[20]="Chương trình";
    ký tự b[20]={'P','r','o','g','r','a','m','\0'};

    // sử dụng định dạng định dạng %zu để in size_t printf("Độ
    dài của a = %zu \n",strlen(a)); printf("Độ dài của b = %zu
    \n",strlen(b));

    trả về 0;
}
```

## ĐẦU RA:

Độ dài của a = 7

Độ dài của b = 7

- Nguyên mẫu hàm của strcpy() là:
- `char*strcpy(char*điểm đến, const char*nguồn);`
- Hàm strcpy() sao chép chuỗi được trả bởi nguồn (bao gồm cả ký tự null) tới đích.
- Hàm strcpy() cũng trả về chuỗi đã sao chép.
- Hàm strcpy() được xác định trong tệp tiêu đề string.h.

```
# bao gồm <stdio.h>
# bao gồm <string.h>

int chính() {
    char str1[20] = "Lập trình C"; char
    str2[20];

    // sao chép str1 sang str2
    strcpy(str2, str1);
    // in str2:
    đặt(str2); // Lập trình C

    trả về 0;
}
```



- Nguyên mẫu:

```
int strcmp (const char* str1, const char* str2);
```

- ✓ strcmp() so sánh hai chuỗi ký tự theo ký tự.
- ✓ Hàm strcmp() nhận vào hai chuỗi và trả về một số nguyên.

Return Value	Remarks
0	if both strings are identical (equal)
negative	if the ASCII value of the first unmatched character is less than the second.
positive integer	if the ASCII value of the first unmatched character is greater than the second.

```
# bao gồm <stdio.h>
# bao gồm <string.h>
int chính()
{
    char str1[] = "abcd", str2[] = "abCd", str3[] = "abcd"; kết quả int;

    // so sánh chuỗi str1 và str2 kết quả = strcmp(str1,
    str2); printf("strcmp(str1, str2) = %d\n", result); // so
    sánh chuỗi str1 và str3 kết quả = strcmp(str1, str3);
    printf("strcmp(str1, str3) = %d\n", result);

    trả về 0;
}
```

## ĐẦU RA (\*):

strcmp(str1, str2) = 32

strcmp(str1, str3) = 0

- Hàm strcat() nối (nối) hai chuỗi.
- Hàm strcat() nối chuỗi đích và chuỗi nguồn, kết quả được lưu trữ trong chuỗi đích.
- Định nghĩa hàm của strcat() là:

`char *strcat(char *destination, const char *source)`

- Hàm strcat() nối (nối) hai chuỗi.
- Định nghĩa hàm của strcat() là:

```
# bao gồm <stdio.h>
# bao gồm <string.h>

int chính() {
    char str1[100] = "Đây là", str2[] = "programiz.com";

    // nối str1 và str2
    // chuỗi kết quả được lưu trữ trong str1. strcat(str1,
    str2);

    đặt(str1);
    đặt(str2);

    trả về 0;
}
```

## ĐẦU RA:

Đây là chương trình.com  
lập trình.com

- *strcpy()* là một hàm thư viện chuẩn trong C/C++ và được sử dụng để sao chép chuỗi này sang chuỗi khác.
- Trong C, nó có trong tệp tiêu đề `string.h` và trong C++, nó có hiện diện trong tệp tin tiêu đề `cstring`.
- **Cú pháp:**  
`char* strcpy(char* dest, const char* src);`

- **Thông số:** Phương thức này chấp nhận các tham số sau:
- **số phận:** Con trỏ tới mảng đích nơi nội dung sẽ được sao chép.
- **src:** chuỗi sẽ được sao chép.
- **Giá trị trả về:** Sau khi sao chép chuỗi nguồn vào chuỗi đích, *strcpy()* hàm trả về một con trỏ tới chuỗi đích.
- Chương trình dưới đây giải thích các cách sử dụng khác nhau của chức năng thư viện này:
- **Cú pháp:**

```
char* strcpy(char* dest, const char* src);
```

# Chuỗi trong C: Các hàm thông dụng - strcpy() (3)

```
// Chương trình C minh họa //  
hàm strcpy() ic C/C++  
# bao gồm<stdio.h>  
# bao gồm<string.h>  
int chủ yếu () {  
    ký tự str1[]="Xin chào các Geek!"; ký  
    tự str2[] ="GeeksforGeeks"; ký tự  
    str3[40];  
    ký tự str4[40];  
    ký tự str5[] ="GfG";  
  
    strcpy(str2, str1);  
    strcpy(str3,"Sao chép thành công");  
    strcpy(str4, str5);  
    printf ("str1: %s\nstr2: %s\nstr3: %s\nstr4:  
           %s\n", str1, str2, str3, str4);  
    trở lại0;  
}
```



ĐẦU RA

str1: Xin chào các chuyên viên máy tính!

str2: Xin chào các chuyên viên máy tính!

str3: Sao chép thành công

str4: GfG

- **strcat**-nối hai chuỗi
- **strchr**-hoạt động quét chuỗi
- **strcmp**-so sánh hai chuỗi
- **strcpy**-sao chép một chuỗi
- **căng thẳng**-lấy độ dài chuỗi
- **strncat**-nối một chuỗi với một phần của chuỗi khác
- **strncmp**-so sánh các phần của hai chuỗi
- **strncpy**-sao chép một phần của chuỗi
- **strchr**-hoạt động quét chuỗi



phần 4

# TRẢ LỜI XUỐNG

Chuỗi ký tự có thể không trực tiếp trong mã nguồn chứa nhúng dòng mới hoặc các ký tự điều khiển khác, hoặc một số ký tự đặc biệt khác nghĩa trong chuỗi.

Để bao gồm các ký tự như vậy trong một chuỗi, dấu gạch chéo ngược thoát ra có thể được dùng.

- `\\` - Dấu gạch chéo ngược theo nghĩa đen
- `\"` - Trích dẫn kép
- `\'` - Trích dẫn đơn
- `\N` - Dòng mới (nguồn cấp dữ liệu dòng)
- `\r` - Vận chuyển trở lại
- `\b` - Phím lùi
- `\t` - Tab ngang
- `\f` - Thước ăn dạng
- `\MỘt` Cảnh báo (chuông)
- `\v` - Tab dọc
- `\0` - Ký tự rỗng

# Dấu gạch chéo ngược thoát: Ví dụ

```
# bao gồm<stdio.h>
int chủ yếu(){
    int số = 2021;
    printf("Bạn\nda\ndang học\nc' ngôn ngữ\n" "Bạn có biết ngôn ngữ C không?\\");
    trở lại 0;
}
```

ĐẦU RA:

Bạn  
là  
học hỏi  
'ngôn ngữ C  
"Bạn có biết ngôn ngữ C không"

# Cảm ơn

*Hỏi đáp*

