

C++ 字符串

C++ 提供了以下两种类型的字符串表示形式：

- C 风格字符串
- C++ 引入的 string 类类型

C 风格字符串

C 风格的字符串起源于 C 语言，并在 C++ 中继续得到支持。字符串实际上是使用 `null` 字符 `'\0'` 终止的一维字符数组。因此，一个以 `null` 结尾的字符串，包含了组成字符串的字符。

下面的声明和初始化创建了一个 "Hello" 字符串。由于在数组的末尾存储了空字符，所以字符数组的大小比单词 "Hello" 的字符数多一个。

```
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

依据数组初始化规则，您可以把上面的语句写成以下语句：

```
char greeting[] = "Hello";
```

以下是 C/C++ 中定义的字符串的内存表示：

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

其实，您不需要把 `null` 字符放在字符串常量的末尾。C++ 编译器会在初始化数组时，自动把 `'\0'` 放在字符串的末尾。让我们尝试输出上面的字符串：

实例

```
#include <iostream>

using namespace std;

int main ()
{
    char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

分类导航

- HTML / CSS
- JavaScript
- 服务端
- 数据库
- 移动端
- XML 教程
- ASP.NET
- Web Service
- 开发工具
- 网站建设

Advertisement



反馈/建议

C++ 接口 (抽象类)
C++ 高级教程
C++ 文件和流
C++ 异常处理
C++ 动态内存
C++ 命名空间
C++ 模板
C++ 预处理器
C++ 信号处理
C++ 多线程
C++ Web 编程
C++ 资源库
C++ STL 教程
C++ 标准库
C++ 有用的资源
C++ 实例

```
cout << "Greeting message: ";
cout << greeting << endl;

return 0;
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
Greeting message: Hello
```

C++ 中有大量的函数用来操作以 null 结尾的字符串：supports a wide range of functions that manipulate null-terminated strings:

序号	函数 & 目的
1	strcpy(s1, s2); 复制字符串 s2 到字符串 s1。
2	strcat(s1, s2); 连接字符串 s2 到字符串 s1 的末尾。
3	strlen(s1); 返回字符串 s1 的长度。
4	strcmp(s1, s2); 如果 s1 和 s2 是相同的，则返回 0；如果 s1<s2 则返回值小于 0；如果 s1>s2 则返回值大于 0。
5	strchr(s1, ch); 返回一个指针，指向字符串 s1 中字符 ch 的第一次出现的位置。
6	strstr(s1, s2); 返回一个指针，指向字符串 s1 中字符串 s2 的第一次出现的位置。

下面的实例使用了上述的一些函数：

```
实例

#include <iostream>
#include <cstring>

using namespace std;

int main ()
{
    char str1[11] = "Hello";
    char str2[11] = "World";
    char str3[11];
    int len ;

    // 复制 str1 到 str3
    strcpy( str3, str1);
    cout << "strcpy( str3, str1) : " << str3 << endl;

    // 连接 str1 和 str2
```

亿速云

10大游戏公司都在用

29

↑

☐

★

反馈/建议

```

    strcat( str1, str2);
    cout << "strcat( str1, str2): " << str1 << endl;

    // 连接后, str1 的总长度
    len = strlen(str1);
    cout << "strlen(str1) : " << len << endl;

    return 0;
}

```

当上面的代码被编译和执行时, 它会产生下列结果:

```

strcpy( str3, str1) : Hello
strcat( str1, str2): HelloWorld
strlen(str1) : 10

```

C++ 中的 String 类

C++ 标准库提供了 **string** 类类型, 支持上述所有的操作, 另外还增加了其他更多的功能。我们将学习 C++ 标准库中的这个类, 现在让我们先来看看下面这个实例:

现在您可能还无法透彻地理解这个实例, 因为到目前为止我们还没有讨论类和对象。

所以现在您可以只是粗略地看下这个实例, 等理解了面向对象的概念之后再回头来理解这个实例。

实例

```

#include <iostream>
#include <string>

using namespace std;

int main ()
{
    string str1 = "Hello";
    string str2 = "World";
    string str3;
    int len ;

    // 复制 str1 到 str3
    str3 = str1;
    cout << "str3 : " << str3 << endl;

    // 连接 str1 和 str2
    str3 = str1 + str2;
    cout << "str1 + str2 : " << str3 << endl;

    // 连接后, str3 的总长度
    len = str3.size();
    cout << "str3.size() : " << len << endl;

    return 0;
}

```

当上面的代码被编译和执行时, 它会产生下列结果:

```

str3 : Hello
str1 + str2 : HelloWorld

```

反馈/建议



```
str3.size() : 10
```

[← C++ 数组](#)[C++ 指针 →](#)**9 篇笔记**[✎ 写笔记](#)

在线实例

- [HTML 实例](#)
- [CSS 实例](#)
- [JavaScript 实例](#)
- [Ajax 实例](#)
- [jQuery 实例](#)
- [XML 实例](#)
- [Java 实例](#)

字符集&工具

- [HTML 字符集设置](#)
- [HTML ASCII 字符集](#)
- [HTML ISO-8859-1](#)
- [HTML 实体符号](#)
- [HTML 拾色器](#)
- [JSON 格式化工具](#)

最新更新

- [Python redis 使...](#)
- [Windows10 MYSQ...](#)
- [Docke 镜像加速](#)
- [Debian Docker 安装](#)
- [C 库函数](#)
- [Linux groupadd ...](#)
- [CSS var\(\) 函数](#)

站点信息

- [意见反馈](#)
- [合作联系](#)
- [免责声明](#)
- [关于我们](#)
- [文章归档](#)

关注微信



Copyright © 2013-2019 **菜鸟教程**
runoob.com All Rights Reserved.
备案号: 闽ICP备15012807号-1

[反馈/建议](#)