

C++ 教程

- C++ 教程
- C++ 简介
- C++ 环境设置
- C++ 基本语法
- C++ 注释
- C++ 数据类型
- C++ 变量类型
- C++ 变量作用域
- C++ 常量
- C++ 修饰符类型
- C++ 存储类
- C++ 运算符
- C++ 循环
- C++ 判断
- C++ 函数
- C++ 数字
- C++ 数组
- C++ 字符串
- C++ 指针
- C++ 引用
- C++ 日期 & 时间
- C++ 基本的输入输出
- C++ 数据结构
- C++ 面向对象
- C++ 类 & 对象
- C++ 继承

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。

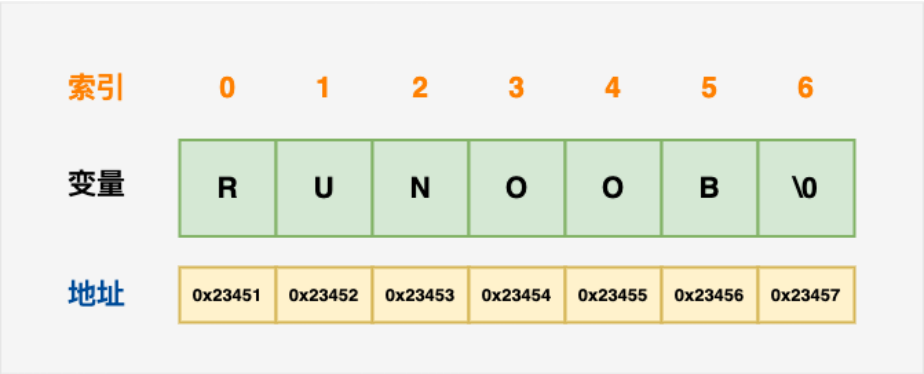
下面的声明和初始化创建了一个 **RUNOOB** 字符串。由于在数组的末尾存储了空字符，所以数组的大小比单词 **RUNOOB** 的字符数多一个。

```
char site[7] = {'R', 'U', 'N', 'O', 'O', 'B', '\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 site[] = "RUNOOB";
```

以下是 C/C++ 中定义的字符串的内存表示：



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 ()
```

分类导航

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

Advertisement



反馈/建议

C++ 重载运算符  
和重载函数

C++ 多态

C++ 数据抽象

C++ 数据封装

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

```
{  
    char site[7] = {'R', 'U', 'N', 'O', 'O', 'B', '\\0'};  
  
    cout << "菜鸟教程：";  
    cout << site << endl;  
  
    return 0;  
}
```

当上面的代码被编译和执行时，它会产生下列结果：

菜鸟教程： RUNOOB

C++ 中有大量的函数用来操作以 null 结尾的字符串：

序号	函数 & 目的
1	<b>strcpy(s1, s2);</b> 复制字符串 s2 到字符串 s1。
2	<b>strcat(s1, s2);</b> 连接字符串 s2 到字符串 s1 的末尾。连接字符串也可以用 + 号，例如： <pre>string str1 = "runoob"; string str2 = "google"; string str = str1 + str2;</pre>
3	<b>strlen(s1);</b> 返回字符串 s1 的长度。
4	<b>strcmp(s1, s2);</b> 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	<b>strchr(s1, ch);</b> 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	<b>strstr(s1, s2);</b> 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

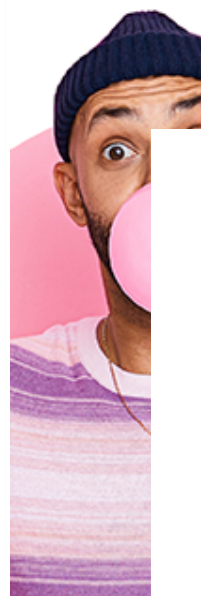
```
#include <iostream>  
#include <cstring>  
  
using namespace std;  
  
int main ()  
{  
    char str1[13] = "runoob";  
    char str2[13] = "google";  
    char str3[13];
```

SEMRL

Let's talk  
about your  
marketing  
persona

Start Quiz

#ShapeYour



反馈/建议

```

int len ;

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

// 连接 str1 和 str2
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) : runoob
strcat( str1, str2): runoobgoogle
strlen(str1) : 12

```

## 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 = "runoob";
    string str2 = "google";
    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 : runoob
str1 + str2 : runoobgoogle
str3.size() : 12
```

← C++ 数组

C++ 指针 →



13 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

- CSS clip-path 属性
- CSS @charset 规则
- CSS grid-row 属性
- CSS grid-template...
- CSS grid-template...
- CSS grid-template...
- CSS grid-template...

#### 站点信息

- 意见反馈
- 免责声明
- 关于我们
- 文章归档

#### 关注微信

