

[\[关闭\]](#)

@RabbitHu 2017-09-19 15:33 字数 2006 阅读 2259

# 十个你一定会用到的 C++ string 函数!

笔记

## 目录

- |                       |                        |
|-----------------------|------------------------|
| 1. <a href="#">赋值</a> | 6. <a href="#">交换</a>  |
| 2. <a href="#">长度</a> | 7. <a href="#">子串</a>  |
| 3. <a href="#">比较</a> | 8. <a href="#">替换</a>  |
| 4. <a href="#">查找</a> | 9. <a href="#">插入</a>  |
| 5. <a href="#">连接</a> | 10. <a href="#">删除</a> |

## 赋值

将字符串2赋值给字符串1:

1. 字符串1 = 字符串2;
2. 字符串1.assign(字符串2);

```
1. string s1 = "I'm ";
2. string s2;
3. s2 = "1234Juruo1234";
4. s1.assign(s2);
5. cout << s1;
6. //输出结果: 1234Juruo1234
```

将字符串2从第m个字符开始的n个字符赋值给字符串1:

字符串1.assign(字符串2, m, n);

```
1. string s1 = "I'm ";
2. string s2 = "1234Juruo1234";
3. s1.assign(s2, 4, 5);
4. cout << s1;
5. //输出结果:Juruo
```

## 长度

返回字符串长度:

1. 字符串1.length();
2. 字符串1.size();

```
1. string s1 = "1234Juruo1234";
2. cout << s1.length() << endl;
3. cout << s1.size() << endl;
4. /*
5. 输出结果:
6. 13
7. 13
```

```
8.    */
```

## 比较

">", "<", "==" , ">=", "<="均可以用于字符串比较。

```
1.    string PPAP[] = {"I", "have", "a", "pen", "an", "apple", "um", "apple-pen"};
2.    sort(PPAP, PPAP + 8);
3.    for(int i = 0; i < 8; i++){
4.        cout << PPAP[i] << endl;
5.    }
6.    /*
7.    输出结果:
8.    I
9.    a
10.   an
11.   apple
12.   apple-pen
13.   have
14.   pen
15.   um
16.   */
```

## 查找

在字符串1中从第m个字符开始查找字符串2，返回第一次出现的首字母位置，失败时返回-1：

字符串1.find(字符串2, m);

```
1.    string s1 = "ggabcdabcbgggabcdefg";
2.    string s2 = "gg";
3.    int pos = -1;
4.    while(1){
5.        pos = s1.find(s2, pos+1);
6.        if(pos == -1) break;
7.        cout << pos << ' ';
8.    }
9.    //输出结果: 0 9 10
```

在字符串1中从第m个字符开始**从后向前**查找字符串2，返回第一次出现的首字母位置，失败时返回-1：

字符串1.rfind(字符串2, m);

```
1.    string s1 = "ggabcdabcbgggabcdefg";
2.    string s2 = "gg";
3.    int pos = s1.length();
4.    while(pos > 0){
5.        pos = s1.rfind(s2, pos-1);
6.        if(pos < 0) break;
7.        cout << pos << ' ';
8.    }
9.    //输出结果: 10 9 0;
```

## 连接

将字符串2接到字符串1尾部:

1. 字符串1.append(字符串2); //字符不可
2. 字符串1 += 字符串2; //字符亦可

```
1. string s1 = "I'm ";
2. string s2 = "Juruo";
3. s1.append(s2);
4. // 或 s1 += s2;
5. cout << s1;
6. //输出结果: I'm Juruo
```

将字符串2从第m个字符开始的n个字符接到字符串1尾部:

字符串1.append(字符串2, m, n);

```
1. string s1 = "I'm ";
2. string s2 = "1234Juruo1234";
3. s1.append(s2, 4, 5);
4. cout << s1;
5. //输出结果: I'm Juruo
```

## 交换

字符串1.swap(字符串2);

```
1. string s1 = "I'm ";
2. string s2 = "Juruo";
3. s1.swap(s2);
4. cout << s1 << endl;
5. //输出结果: Juruo
```

## 子串

返回字符串1从第m个字符开始的n个字符所组成的子串:

字符串1.substr(m, n);

```
1. string s1 = "I'm ";
2. string s2 = "1234Juruo1234";
3. s1 = s2.substr(4, 5);
4. cout << s1 << endl;
5. //输出结果: Juruo
```

## 替换

在字符串1中删除从m开始的n个字符, 然后在m处插入串s2

字符串1.replace(m, n, s2);

```
1. string s1 = "I'm Juruo";
2. string s2 = "Juruo";
3. string s3 = "Dalao";
4. int pos = s1.find(s2);
5. s1.replace(pos, s2.length(), s3);
6. cout << s1;
7. //输出结果: I'm Dalao
```

# 插入

在字符串1的第m个字符处插入字符串2:

字符串1.insert(m, 字符串2);

```
1. string s1 = "I'm Juruo";
2. string s2 = "not ";
3. s1.insert(s1.find("Juruo", 0), s2 );
4. cout << s1 << endl;
5. //输出结果: I'm not Juruo
```

# 删除

从字符串1的第m个字符开始, 删除n个字符:

字符串1.erase(m, n);

```
1. string s1 = "I'm not Dalao";
2. s1.erase(s1.find("not"), 4);
3. cout << s1 << endl;
```

作者: [胡小兔](#) //←这个网站经常崩溃

批评指正敬请联系: [yuanxiaodidl@163.com](mailto:yuanxiaodidl@163.com)

打赏一根胡萝卜:



## • 内容目录

- [十个你一定会用到的 C++ string 函数!](#)
  - [目录](#)
  - [赋值](#)
  - [长度](#)
  - [比较](#)
  - [查找](#)
  - [连接](#)
  - [交换](#)
  - [子串](#)
  - [替换](#)
  - [插入](#)
  - [删除](#)

- 
- - - 作业 5
    - [DL24袁小迪 第一次七校模拟题解](#)
    - [一份题解](#)
    - [一份题解 DL24袁小迪](#)
    - [DL24胡小兔--记一次极端悲催的模拟 \(模拟赛9\)](#)
    - [模拟7题解 DL24袁小迪](#)
  - - 参考 1
    - [24OI小组成员如何拥有一个自己的博客?](#)
  - - 学习-化学 1
    - [化学期末复习错题集](#)
  - - 学习-数学 2
    - [三角函数值域问题 \(一题多解\)](#)
    - [题解! 变态的国大班三角函数测试](#)
  - - 学习-物理 1
    - [完全弹性碰撞速度公式的推导](#)
  - - 干活 1
    - [编程社招新](#)
  - - 日记 4
    - [胡小兔的 OI 日志 3 \(2017.9.1 ~ \)](#)
    - [2017余姚集训总结](#)
    - [胡小兔的 OI 日志 2 \(2017.7.19 ~ 2017.8.31\)](#)
    - [胡小兔的OI日志](#)
  - - 笔记 7
    - [“高级” 数据结构——树状数组!](#)
    - [单调栈、单调队列专题](#)
    - [矩阵乘法的4个有趣应用](#)
    - [树链剖分——你为什么这么树链呢?](#)
    - [AC 自动机 —— WA 自动机? TLE 自动机?](#)
    - [十个你一定会用到的 C++ string 函数!](#)
    - [紫书太难了! 简单易懂的GCD和扩展GCD](#)
  - - 题目 3
    - [借书](#)
    - [最小森林](#)
    - [烷基计数](#)
  - - 题解 6
    - [最大M子段和 | 神奇的DP](#)
  - 搜索 RabbitHu 的文稿标题,  [法 | 单调队列](#)
  - 以下 [\[标签\]](#) 将用于标记这篇文稿:
    - [背包问题V3 | 升数规划](#)
    - [暑假集训D4T6](#)
    - [线段树 x 约瑟夫 —— CodeVS 1282](#)
    - [CodeVS1039 数的划分 插图题解!](#)
  - [下载客户端](#)
  - [关注开发者](#)
  - [报告问题, 建议](#)
  - [联系我们](#)
- 
- 
- 
- 
-

添加新批注



保存 取消

在作者公开此批注前，只有你和作者可见。



保存 取消



修改 保存 取消 删除

- 私有
- 公开
- 删除

查看更早的 5 条回复

回复批注



通知

取消 确认

- ☐
- ☐