目录

[一、C 语言的字符串输入输出函数 2](#_Toc207224330)

[**1. gets(char \*str)（不推荐使用）** 2](#_Toc207224331)

[**2. fgets(char \*str, int n, FILE \*stream)** 2](#_Toc207224332)

[**3. puts(const char \*str)** 2](#_Toc207224333)

[**4. printf(const char \*format, ...)** 2](#_Toc207224334)

[**5. scanf(const char \*format, ...)** 3](#_Toc207224335)

[**6. sprintf(char \*str, const char \*format, ...)** 3](#_Toc207224336)

[**7. sscanf(const char \*str, const char \*format, ...)** 3](#_Toc207224337)

[**C语言字符串I/O使用建议：** 4](#_Toc207224338)

[二、C++ 字符串输入输出函数（基于 std::string） 4](#_Toc207224339)

[**基本输入输出函数** 4](#_Toc207224340)

[**1. std::cin >> str** 4](#_Toc207224341)

[**2. std::getline(std::istream&, std::string&)** 4](#_Toc207224342)

[**3. std::cout << str** 4](#_Toc207224343)

[**常用字符串处理函数（成员函数）** 4](#_Toc207224344)

[**示例：** 5](#_Toc207224345)

[**字符串格式化输出（ostringstream）** 5](#_Toc207224346)

[**类：std::ostringstream** 5](#_Toc207224347)

[**从字符串中提取数据（istringstream）** 5](#_Toc207224348)

[**类：std::istringstream** 5](#_Toc207224349)

[**字符串与数字的转换** 6](#_Toc207224350)

[**1. std::to\_string(value)（C++11 起）** 6](#_Toc207224351)

[**2. std::stoi(str), std::stod(str) 等（C++11 起）** 6](#_Toc207224352)

[**C++ 字符串输入输出常见用法对比总结** 6](#_Toc207224353)

[**总结：推荐使用哪些函数？** 6](#_Toc207224354)

# 一、C 语言的字符串输入输出函数

C 语言处理字符串的输入输出主要使用 stdio.h 中的函数。字符串本质是字符数组（以 '\0' 结尾）。

**1. gets(char \*str)（不推荐使用）**

⚠️ 在 C11 中被正式移除，原因是**不安全**，无法限制长度，容易造成缓冲区溢出。

char str[100];

gets(str); // 不安全，危险！建议用 fgets

**2. fgets(char \*str, int n, FILE \*stream)**

**功能：**

从 stream 中读取至多 n-1 个字符到 str，直到遇到换行、EOF 或读满为止，并自动加上 '\0'。

**示例：**

char str[100];

fgets(str, 100, stdin); // 安全替代 gets

**特点：**

* 读取到换行符会保留 \n
* 比 gets 安全
* 可用于文件读取

**3. puts(const char \*str)**

**功能：**

输出一个字符串到标准输出（结尾自动添加 \n）。

**示例：**

puts("Hello");

**4. printf(const char \*format, ...)**

**功能：**

格式化输出字符串。

**示例：**

char name[] = "Tom";

int age = 20;

printf("Name: %s, Age: %d\n", name, age);

**5. scanf(const char \*format, ...)**

**功能：**

格式化读取字符串。

**示例：**

char name[100];

scanf("%s", name); // 读到空格或回车停止

**注意：**

* %s 不会读取空格后的内容
* 无法读取整行，建议使用 fgets + sscanf

**6. sprintf(char \*str, const char \*format, ...)**

**功能：**

把格式化内容写入字符串中（类似 printf，但输出到 str）。

**示例：**

char buffer[100];

int x = 10;

sprintf(buffer, "Value is %d", x); // buffer = "Value is 10"

**7. sscanf(const char \*str, const char \*format, ...)**

**功能：**

从字符串中读取数据，类似 scanf 但输入是字符串。

**示例：**

char str[] = "Tom 18";

char name[50];

int age;

sscanf(str, "%s %d", name, &age); // name = "Tom", age = 18

**C语言字符串I/O使用建议：**

|  |  |
| --- | --- |
| 任务 | 推荐函数 |
| 读取整行输入 | fgets() |
| 输出字符串 | puts() / printf() |
| 从字符串中读取内容 | sscanf() |
| 把内容写入字符串 | sprintf()（更安全：snprintf()） |

# 二、C++ 字符串输入输出函数（基于 std::string）

**基本输入输出函数**

**1. std::cin >> str**

* 从标准输入读取一个“单词”到字符串（遇空格或换行结束）
* 类型：std::istream& operator>>(std::istream&, std::string&)

**2. std::getline(std::istream&, std::string&)**

* 从输入流读取整行（包括空格），直到 \n。
* 替代 gets()、fgets() 的推荐方式。

std::string line;

std::getline(std::cin, line); // 可读入 "Tom Smith"

**3. std::cout << str**

* 输出字符串到标准输出流。

**常用字符串处理函数（成员函数）**

std::string 提供了很多操作函数，以下是常用的字符串输入输出相关方法：

|  |  |
| --- | --- |
| 函数 | 功能 |
| length() / size() | 获取字符串长度 |
| empty() | 判断字符串是否为空 |
| clear() | 清空字符串内容 |
| append(str) | 追加字符串 |
| insert(pos, str) | 在 pos 处插入子串 |
| erase(pos, len) | 删除从 pos 开始的 len 个字符 |
| substr(pos, len) | 获取子字符串 |
| find(str) | 查找子串第一次出现位置 |
| rfind(str) | 查找子串最后一次出现 |
| replace(pos, len, str) | 替换子串 |
| c\_str() | 获取 C 风格字符串（const char\*） |

**示例：**

std::string s = "hello";

s.append(" world"); // "hello world"

std::string sub = s.substr(0, 5); // "hello"

size\_t pos = s.find("world"); // 6

s.replace(6, 5, "C++"); // "hello C++"

**字符串格式化输出（ostringstream）**

**类：std::ostringstream**

将各种类型格式化为字符串（类似 sprintf）：

#include <sstream>

std::ostringstream oss;

int age = 20;

oss << "Name: " << "Tom" << ", Age: " << age;

std::string result = oss.str(); // "Name: Tom, Age: 20"

**从字符串中提取数据（istringstream）**

**类：std::istringstream**

从字符串中按格式提取数据（类似 sscanf）：

#include <sstream>

std::string line = "Tom 25";

std::istringstream iss(line);

std::string name;

int age;

iss >> name >> age; // name = "Tom", age = 25

**字符串与数字的转换**

**1. std::to\_string(value)（C++11 起）**

将数字转为字符串：

std::string s = std::to\_string(3.14159); // "3.141590"

**2. std::stoi(str), std::stod(str) 等（C++11 起）**

将字符串转为数字：

int n = std::stoi("123"); // 123

double d = std::stod("3.14"); // 3.14

|  |  |
| --- | --- |
| 函数 | 类型转换 |
| std::stoi() | string → int |
| std::stol() | string → long |
| std::stof() | string → float |
| std::stod() | string → double |

**C++ 字符串输入输出常见用法对比总结**

|  |  |  |
| --- | --- | --- |
| 功能 | 函数或类 | 示例 |
| 单词输入 | cin >> str | cin >> name; |
| 整行输入 | getline(cin, str) | getline(cin, line); |
| 输出 | cout << str | cout << s; |
| 字符串转数字 | stoi, stod | int n = stoi("123"); |
| 数字转字符串 | to\_string | to\_string(3.14) |
| 格式化输出到字符串 | ostringstream | oss << "ID: " << id; |
| 从字符串中读取 | istringstream | iss >> name >> age; |
| 文件读写字符串 | ifstream, ofstream | getline(in, str); |

**总结：推荐使用哪些函数？**

|  |  |
| --- | --- |
| 使用场景 | 推荐函数或类 |
| 控制台读入单词 | cin >> str |
| 控制台读入整行 | getline(cin, str) |
| 格式化输出 | ostringstream |
| 从字符串解析值 | istringstream |
| 字符串转数字 | stoi, stod |
| 数字转字符串 | to\_string |
| 读写文件 | ifstream, ofstream |