**C String**

# C Strings

<cstring> (string.h)

'string.h' header file defines several functions to manipulate C strings and arrays.

# Macros

NULL Null pointer (macro )

# Types

size\_t Unsigned integral type (type )

# Functions

## Copying

|  |  |
| --- | --- |
| memcpy | Copy block of memory (function )  void \* memcpy  ( void \* destination, const void \* source, size\_t num ); |
| memmove | Move block of memory (function )  void \* memmove  ( void \* destination, const void \* source, size\_t num ); |
| strcpy | Copy string (function )  char \* strcpy  ( char \* destination, const char \* source ); |
| strncpy | Copy characters from string (function )  char \* strncpy  ( char \* destination, const char \* source, size\_t num ); |

## Concatenation

|  |  |
| --- | --- |
| strcat | Concatenate strings (function )  char \* strcat  ( char \* destination, const char \* source ); |
| strncat | Append characters from string (function )  char \* strncat  ( char \* destination, const char \* source, size\_t num ); |

## Comparison

|  |  |
| --- | --- |
| memcmp | Compare two blocks of memory (function )  int memcmp ( const void \* ptr1, const void \* ptr2, size\_t num ); |
| strcmp | Compare two strings (function )  int strcmp ( const char \* str1, const char \* str2 ); |
| strcoll | Compare two strings using locale (function )  int strcoll ( const char \* str1, const char \* str2 ); |
| strncmp | Compare characters of two strings (function )  int strncmp ( const char \* str1, const char \* str2, size\_t num ); |
| strxfrm | Transform string using locale (function )  size\_t strxfrm ( char \* destination, const char \* source, size\_t num ); |

## Searching

|  |  |
| --- | --- |
| memchr | Locate character in block of memory (function )  const void \* memchr ( const void \* ptr, int value, size\_t num );  void \* memchr ( void \* ptr, int value, size\_t num );    void \* memchr ( const void \*, int, size\_t ); // C |
| strchr | Locate first occurrence of character in string (function )  const char \* strchr ( const char \* str, int character );  char \* strchr ( char \* str, int character );  char \* strchr ( const char \*, int ); // C |
| strcspn | Get span until character in string (function )  size\_t strcspn ( const char \* str1, const char \* str2 ); |
| strpbrk | Locate characters in string (function )  const char \* strpbrk ( const char \* str1, const char \* str2 );  char \* strpbrk ( char \* str1, const char \* str2 );    char \* strpbrk ( const char \*, const char \* ); // C |
| strrchr | Locate last occurrence of character in string (function )  const char \* strrchr ( const char \* str, int character );  char \* strrchr ( char \* str, int character );    char \* strrchr ( const char \*, int ); // C |
| strspn | Get span of character set in string (function )  size\_t strspn ( const char \* str1, const char \* str2 ); |
| strstr | Locate substring (function )  const char \* strstr ( const char \* str1, const char \* str2 );  char \* strstr ( char \* str1, const char \* str2 );    char \* strstr ( const char \*, const char \* ); // C |
| strtok | Split string into tokens (function )  char \* strtok ( char \* str, const char \* delimiters ); |

## Other:

|  |  |
| --- | --- |
| memset | Fill block of memory (function )  void \* memset ( void \* ptr, int value, size\_t num ); |
| strerror | Get pointer to error message string (function )  char \* strerror ( int errnum ); |
| strlen | Get string length (function )  size\_t strlen ( const char \* str ); |

# Macros

NULL Null pointer (macro )

This macro expands to a null pointer constant.

A null pointer constant can be converted to any pointer type (or pointer-to-member type), which acquires a null pointer value.

This is a special value that indicates that the pointer is not pointing to any object.

* In C, A null-pointer constant is an integral constant expression that evaluates to zero (like 0 or 0L), or the cast of such value to type void\* (like **(void\*)0**).
* In C++, A null-pointer constant is an integral constant expression that evaluates to zero (such as **0 or 0L**).
* In C++11, A null-pointer constant is either an integral constant expression that evaluates to zero (such as **0 or 0L**), or a value of type nullptr\_t (such as **nullptr**).

# Types

size\_t Unsigned integral type (type )

**Unsigned integral type**

Alias of one of the fundamental unsigned integer types

It is a type able to represent the size of any object in bytes:

size\_t is the type returned by the sizeof operator and is widely used in the standard library to represent sizes and counts.

In <cstring>, it is used as the type of the parameter num in the functions memchr, memcmp, memcpy, memmove, memset, strncat, strncmp, strncpy and strxfrm, which in all cases it is used to specify the maximum number of bytes or characters the function has to affect.

It is also used as the return type for strcspn, strlen, strspn and strxfrm to return sizes and lengths.

# References

<https://en.cppreference.com/w/c/string/byte>

# END