Strings

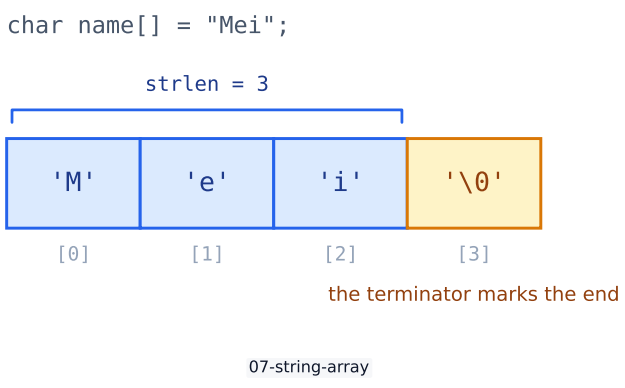
C Reference

Char arrays & the null terminator

A C string 字符串 is an array of `char`. Every string ends with a hidden null terminator 空终止符 `'\0'`, which marks where it stops. `strlen` (from `<string.h>`) counts characters up to that `'\0'`. Print a whole string with `%s` and one character 字符 with `%c`.

```c
#include <stdio.h>
#include <string.h>

int main(void) {
    char name[] = "Mei";            // 'M', 'e', 'i', '\0'
    printf("%s\n", name);           // Mei
    printf("%zu\n", strlen(name)); // 3 (stops at '\0')
    printf("%c\n", name[0]);        // M
    return 0;
}
```



A C string is a char array ending in a hidden null terminator

You can traverse a string by looping until you reach `'\0'`.

```c
#include <stdio.h>

int main(void) {
    char word[] = "banana";
    int count = 0;
    for (int i = 0; word[i] != '\0'; i++) {
        if (word[i] == 'a') count++;
    }
    printf("%d\n", count);   // 3
    return 0;
}
```

The string.h library

`#include <string.h>` gives you the everyday string tools:

Function	Does
`strlen(s)`	the length, not counting the terminator
`strcpy(dst, src)`	copy — `dst` must be big enough
`strcat(dst, src)`	append `src` onto the end of `dst`
`strcmp(a, b)`	compare: `0` when equal, else negative / positive

```c
#include <stdio.h>
#include <string.h>

int main(void) {
    char full[40];
    strcpy(full, "Mei");           // full is now "Mei"
    strcat(full, " Chen");         // append -> "Mei Chen"
    printf("%s (%zu)\n", full, strlen(full));       // Mei Chen (8)
    printf("%d\n", strcmp("apple", "apple") == 0);  // 1 (equal)
    return 0;
}
```

- `strcmp` returns `0` for equal, so the test is `strcmp(a, b) == 0` — never `a == b`.
- The destination array must have room for the result **plus** the terminator.

Common mistakes

- A C string needs one extra byte for the `\0` terminator: a 5-letter word needs `char[6]`.

- Copy and compare strings with `strcpy` / `strcmp`, not `=` / `==`.
- Forgetting the `\0` makes `printf("%s", ...)` run past the end of the text.