字符串

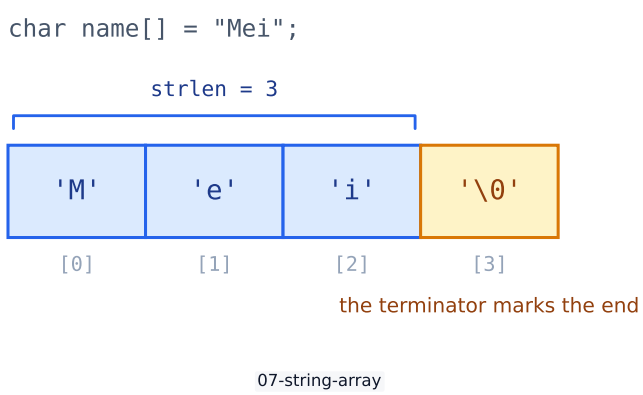
C Reference

字符数组与空终止符

C 的字符串 (string) 是一个 `char` 数组。每个字符串都以一个隐藏的空终止符 (null terminator)`'\0'` 结尾, 它标记字符串在哪里结束。`strlen`(来自 `<string.h>`) 统计到那个 `'\0'` 为止的字符数。用 `%s` 打印整个字符串, 用 `%c` 打印单个字符 (character)。

```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 (到 '\0' 为止)
    printf("%c\n", name[0]);        // M
    return 0;
}
```



07-string-array

C 字符串是一个 *char* 数组, 以隐藏的空终止符结尾

exercises.lol exercises.com.cn

你可以通过循环到 `'\0'` 为止来遍历一个字符串。

```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;
}
```

string.h 函数库

`#include <string.h>` 提供日常要用的字符串工具:

函数	作用
`strlen(s)`	长度, 不算结束符
`strcpy(dst, src)`	复制 —`dst` 必须够大
`strcat(dst, src)`	把 `src` 接到 `dst` 末尾
`strcmp(a, b)`	比较: 相等为 0, 否则为负 / 正

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

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

- `strcmp` 相等时返回 0, 所以判断写成 `strcmp(a, b) == 0`— 绝不能写 `a == b`。
- 目标数组要装得下结果, **再加上**结束符。

常见错误

- C 字符串要多留一个字节给结束符 `\0`:5 个字母的词需要 `char[6]`。
- 复制和比较字符串用 `strcpy` / `strcmp`, 不要用 `=` / `==`。
- 少了 `\0`,`printf("%s", ...)` 会读过文本末尾。