C 基础

C Reference

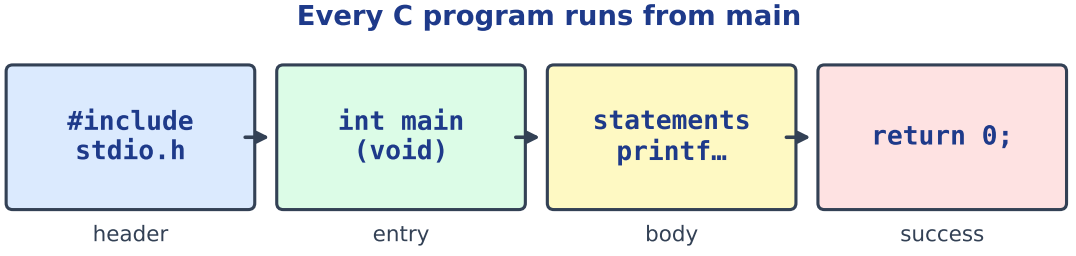
main, printf & functions

每个 C 程序都从 `main` 开始。`#include <stdio.h>` 引入一个头文件 (header), 这样你就能用 `printf` 来打印。函数 (function) 是一个有名字、可以调用的代码块;`\n` 表示换行。`main` 返回 `0` 表示” 成功”。

```c
#include <stdio.h>

void greet(void) {
    printf("Hello, world!\n");
}

int main(void) {
    greet();
    printf("I am learning C.\n");
    return 0;
}
```



Exam tip: #include for libraries; main returns 0 for success; end statements with ;

Every C program runs from main after including headers

Variables, types & arithmetic

C 要求每个变量都有类型:`int`(整数,integer)、`double`(浮点数,floating-point)、`char`(单个字符)。`printf` 为每个值使用一个格式说明符 (format specifier)—`%d`、`%f`、`%c`。两个整数之间的 `/` 是整数除法;`%` 是余数 (remainder)。

```c
#include <stdio.h>

int main(void) {
    int n = 17;
    double pi = 3.14;
    char grade = 'A';
    printf("%d %.2f %c\n", n, pi, grade);   // 17 3.14 A
    printf("%d %d\n", 7 / 2, 7 % 2);        // 3 1
    return 0;
}
```

格式串同时驱动打印和读取:

说明符	类型	示例输出
%d	int	17
%f	double(%.2f = 保留 2 位小数)	3.14
%c	char	A
%s	字符串	Mei
%zu	大小 (来自 sizeof / strlen)	3
%x	unsigned int, 按 16 进制打印	ff

`scanf("%d", &n)` 通过地址读取 `int`(注意使用 `&`)。输入和输出格式并不相同:`printf("%f", x)` 打印一个 `double`,但 `scanf("%lf", &x)` 读取一个 `double`;输入 `%f` 需要一个 `float *`。请检查成功分配项目的返回计数。其兄弟函数 `sscanf` 可从字符串中解析值:

```c
#include <stdio.h>

int main(void) {
    char line[] = "Mei 88";
    char name[20];
    int score;
    if (sscanf(line, "%19s %d", name, &score) != 2) return 1;
    printf("%s scored %d\n", name, score);   // Mei scored 88
    return 0;
}
```

Comments & style

注释 (comment) 是给人看的说明; 编译器会忽略它。单行用 `//`, 整块用 `/* ... */`。良好的缩进 (indentation) 和清晰的命名让代码易读。

```c
#include <stdio.h>

int main(void) {
    // a single-line comment
    /* a block
       comment */
    int total = 3 + 4;        // clear names help
    printf("%d\n", total);    // 7
    return 0;
}
```

常见错误

- 声明和表达式语句以 `;` 结尾; 复合块 `{ ... }` 不需要额外的分号。`main` 应返回 `int` (`return 0;`)。
- `printf` 需要格式串:`%d` 对应 `int`,`%f` 对应 `double`,`\n` 换行。
- 错误的格式如将 `%d` 用于 `double` 会导致未定义行为; 它并不保证仅打印垃圾数据。请匹配参数类型。