

C 基础

C Reference

main、printf 与函数

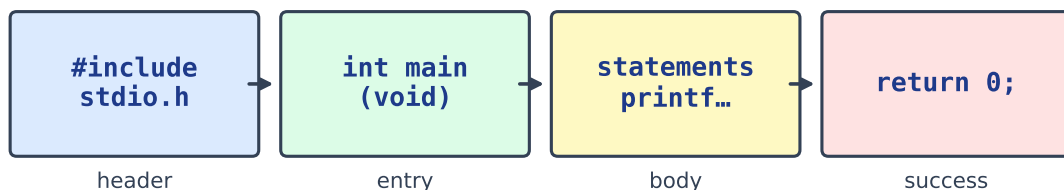
每个 C 程序都从 main 开始。`#include <stdio.h>` 引入一个头文件 (header), 这样你就能用 `printf` 来打印。函数 (function) 是一个有名字、可以调用的代码块;\n 表示换行。main 返回 0 表示“成功”。

```
#include <stdio.h>

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

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

Every C program runs from main



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

Every C program runs from main after including headers

变量、类型与算术

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)。

```
#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	int, 按 16 进制打印	ff

scanf("%d", &n) 用同样的说明符读取键盘输入 (注意 &)。它的同伴 sscanf 从字符串里解析值, 所以在哪里都能运行:

```
#include <stdio.h>

int main(void) {
    char line[] = "Mei 88";
    char name[20];
    int score;
    sscanf(line, "%19s %d", name, &score);    // 从文本解析出值
    printf("%s scored %d\n", name, score);    // Mei scored 88
    return 0;
}
```

注释与风格

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。

```
#include <stdio.h>

int main(void) {
```

```
// 单行注释
/* 一段
   块注释 */
int total = 3 + 4;      // 清晰的命名有帮助
printf("%d\n", total);  // 7
return 0;
}
```

常见错误

- 每条语句以 ; 结尾,main 应返回 int(return 0;)。
- printf 需要格式串:%d 对应 int,%f 对应 double,\n 换行。
- 格式不匹配,比如对 double 用 %d,会打印乱码——类型要对上。