C basics

C Reference

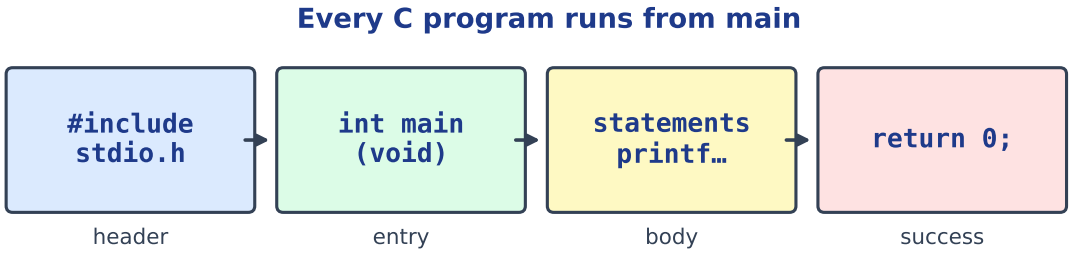
main, printf & functions

Every C program starts in `main`. `#include <stdio.h>` pulls in a header 头文件 so you can use `printf` to print. A function 函数 is a named block you can call; `\n` starts a new line. `main` returns `0` to mean "success".

```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 needs a type for every variable: `int` (integer 整数), `double` (floating-point 浮点数), `char` (one character). `printf` uses a format specifier 格式说明符 — `%d`, `%f`, `%c` — for each value. `/` between two ints is integer division; `%` is the remainder 余数.

```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;
}
```

The format string drives both printing and reading:

Specifier	Type	Example output
%d	int	17
%f	double (%.2f = 2 decimal places)	3.14
%c	char	A
%s	a string	Mei
%zu	a size (from sizeof / strlen)	3
%x	int, printed in base 16	ff

scanf("%d", &n) reads keyboard input with the same specifiers (note the &). Its sibling sscanf parses values out of a string, so it runs anywhere:

```c
#include <stdio.h>

int main(void) {
    char line[] = "Mei 88";
    char name[20];
    int score;
    sscanf(line, "%19s %d", name, &score);   // parse text -> values
    printf("%s scored %d\n", name, score);   // Mei scored 88
    return 0;
}
```

Comments & style

A comment 注释 is a note for humans; the compiler ignores it. Use // for one line and /* ... */ for a block. Good indentation 缩进 and clear names make code easy to read.

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

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

Common mistakes

- Every statement ends with `;`, and `main` should return `int` (`return 0;`).
- `printf` needs a format string: `%d` for `int`, `%f` for `double`, and `\n` for a new line.
- A wrong format like `%d` for a `double` prints garbage — match the type.