

# C 语言基础语法整理

2026 年 1 月 9 日

## 目录

## 1 程序结构

### 1.1 主函数（程序入口）

```
1 #include <stdio.h>
2
3 int main() {
4     // 程序代码
5     printf("Hello, World!\n");
6     return 0; // 返回0表示程序正常结束
7 }
```

主函数示例

- `int`: 返回类型，表示返回整数
- `main`: 函数名，C 程序执行的入口点
- `()`: 参数列表（可为空）
- `{}`: 函数体，包含要执行的语句
- `return 0;`: 返回 0，通常表示程序成功执行

## 2 变量与数据类型

### 2.1 变量定义

```
1 // 定义变量（未初始化）
2 变量类型 变量名;
3
4 // 定义并初始化变量
5 变量类型 变量名 = 值;
```

变量定义示例

### 2.2 基本数据类型

### 2.3 类型修饰符

表 1: C 语言基本数据类型

类型	说明	示例
char	字符型	char c = 'A';
int	整型	int a = 10;
short	短整型	short s = 5;
long	长整型	long l = 100000;
float	单精度浮点型	float f = 3.14;
double	双精度浮点型	double d = 3.14159;

```
1 const int a = 10;           // 常量，值不可修改
2 unsigned int b = 10;        // 无符号整数，只能表示非负数
3 signed int c = -5;          // 有符号整数（默认）
```

类型修饰符示例

### 3 数组

#### 3.1 数组定义

```
1 // 定义数组
2 元素类型 数组名[数组长度];
3
4 // 定义并初始化数组
5 元素类型 数组名[数组长度] = {元素1, 元素2, ...};
```

数组定义示例

#### 3.2 示例

```
1 int numbers[3] = {1, 2, 3}; // 整数数组
2 float scores[5];           // 浮点数数组
3 char name[10] = "John";    // 字符数组（字符串）
```

数组使用示例

注意：数组只能存储相同类型的数据。

4 运算符

4.1 算术运算符

表 2: 算术运算符

运算符	说明	示例
+	加法	$a + b$
-	减法	$a - b$
*	乘法	$a * b$
/	除法	$a / b$
%	取模（求余）	$a \% b$

4.2 关系运算符

表 3: 关系运算符

运算符	说明	示例
>	大于	$a > b$
<	小于	$a < b$
>=	大于等于	$a >= b$
<=	小于等于	$a <= b$
==	等于	$a == b$
!=	不等于	$a != b$

4.3 逻辑运算符

表 4: 逻辑运算符

运算符	说明	示例
&&	逻辑与	$a \&\& b$
	逻辑或	$a \ \  b$
!	逻辑非	$!a$

4.4 赋值运算符

表 5: 赋值运算符

运算符	说明	示例
=	赋值	a = 5
+=	加后赋值	a += 3 (等价于 a = a + 3)
-=	减后赋值	a -= 3
*=	乘后赋值	a *= 3
/=	除后赋值	a /= 3
%=	取模后赋值	a %= 3

4.5 自增自减运算符

表 6: 自增自减运算符

运算符	说明	示例
++	自增	i++ 或 ++i
--	自减	i- 或 -i

5 控制语句

5.1 条件语句 - if/else if/else

```
1 if (条件表达式1) {
2     // 条件1为真时执行的代码
3 } else if (条件表达式2) {
4     // 条件2为真时执行的代码
5 } else {
6     // 所有条件都为假时执行的代码
7 }
```

if-else 语句示例

示例:

```
1 int a = 10;
```

```
2
3 if (a > 50) {
4     // a大于50的时候执行这里的语句
5 } else if (a > 10) {
6     // a小于50且大于10的时候执行这里的语句
7 } else {
8     // a小于等于10的时候执行这里的语句
9 }
```

if-else 具体示例

## 5.2 条件语句 - switch

```
1 switch (表达式) {
2     case 值1:
3         // 表达式等于值1时执行的代码
4         break;
5     case 值2:
6         // 表达式等于值2时执行的代码
7         break;
8     case 值3:
9         // 表达式等于值3时执行的代码
10        break;
11    default:
12        // 所有case都不匹配时执行的代码
13        break;
14 }
```

switch 语句结构

示例:

```
1 switch (a) {
2     case 1:
3         // a为1执行这个
4         break;
5     case 2:
6         // a为2执行这个
7         break;
8     case 3:
9         // a为3执行这个
```

```
10     break;
11 default:
12     // a不是1、2、3时执行
13     break;
14 }
```

switch 具体示例

重要提示：每个 case 后必须加 break；，否则会继续执行下一个 case（称为”穿透”现象）。

## 6 循环语句

### 6.1 for 循环

```
1 for (初始化表达式; 条件表达式; 更新表达式) {
2     // 循环体
3 }
```

for 循环结构

示例：

```
1 for (int i = 1; i <= 10; i++) {
2     // 循环10次，每次i自增1
3     // i的值从1到10
4 }
```

for 循环示例

### 6.2 while 循环

```
1 while (条件表达式) {
2     // 循环体
3 }
```

while 循环结构

示例：

```
1 int i = 1;
2 while (i <= 10) {
3     // 循环10次
```

```
4     i++;    // 每次循环i自增1
5 }
```

while 循环示例

### 6.3 do-while 循环

```
1 do {
2     // 循环体（至少执行一次）
3 } while (条件表达式);
```

do-while 循环结构

示例：

```
1 int i = 1;
2 do {
3     // 循环体
4     i++;
5 } while (i <= 10);
```

do-while 循环示例

## 7 语句结束符

- 每个语句以分号；结尾
- 代码块（如函数体、循环体、条件分支体）用花括号 {} 包裹，不需要分号

## 8 补充说明

### 8.1 布尔表达式

- 条件判断中使用布尔表达式，结果为真（非 0）或假（0）
- C 语言中，0 表示假，任何非 0 值表示真

### 8.2 变量作用域

- 变量只在定义它的代码块内有效
- 循环中定义的变量（如 `for(int i=0; ...)`）通常只在循环内有效



### 8.3 指针类型（补充）

每个基本类型都有对应的指针类型：

```
1 int *p;        // 整型指针
2 char *c;       // 字符指针
3 float *f;      // 浮点指针
4 double *d;     // 双精度浮点指针
```

指针类型示例

### 8.4 复合类型（补充）

- 结构体：可存储不同类型的数据
- 结构体数组：数组的元素是结构体类型

## 9 完整示例程序

```
1 #include <stdio.h>
2
3 int main() {
4     // 变量定义
5     int score = 85;
6     const int PASS_SCORE = 60;
7
8     // 条件语句
9     if (score >= PASS_SCORE) {
10         printf("恭喜，你通过了考试！\n");
11     } else {
12         printf("很遗憾，你需要补考。\n");
13     }
14
15     // 数组和循环
16     int grades[5] = {85, 90, 78, 92, 88};
17     int sum = 0;
18
19     for (int i = 0; i < 5; i++) {
20         sum += grades[i];
21     }
22 }
```

```
23     printf("平均分: %.2f\n", sum / 5.0);
24
25     // switch 示例
26     int day = 3;
27     switch (day) {
28         case 1: printf("星期一\n"); break;
29         case 2: printf("星期二\n"); break;
30         case 3: printf("星期三\n"); break;
31         default: printf("其他天\n"); break;
32     }
33
34     return 0;
35 }
```

Listing 1: C 语言基础示例程序

## 10 常见问题与解答

### 10.1 编译错误常见原因

- 缺少分号：每个语句必须以分号结尾
- 括号不匹配：确保每个左括号都有对应的右括号
- 变量未声明：使用变量前必须先声明
- 类型不匹配：注意赋值和运算时的类型一致性

### 10.2 编程建议

- 使用有意义的变量名
- 添加适当的注释
- 保持代码缩进一致
- 测试边界条件
- 避免使用“魔术数字”，使用常量代替