

## C 基本语法

我们已经看过 C 程序的基本结构，这将有助于我们理解 C 语言的其他基本的构建块。

### C 的令牌 ( Tokens )

C 程序由各种令牌组成，令牌可以是关键字、标识符、常量、字符串值，或者是一个符号。例如，下面的 C 语句包括五个令牌：

```
printf("Hello, World! \n");
```

这五个令牌分别是：

```
printf  
(  
"Hello, World! \n"  
)  
;
```

### 分号；

在 C 程序中，分号是语句结束符。也就是说，每个语句必须以分号结束。它表明一个逻辑实体的结束。

例如，下面是两个不同的语句：

```
printf("Hello, World! \n");  
return 0;
```

### 注释

C 语言有两种注释方式：

```
// 单行注释
```

以 `//` 开始的单行注释，这种注释可以单独占一行。

```
/* 单行注释 */  
/*  
多行注释  
多行注释  
多行注释  
*/
```

`/* */` 这种格式的注释可以单行或多行。

您不能在注释内嵌套注释，注释也不能出现在字符串或字符值中。

## 标识符

C 标识符是用来标识变量、函数，或任何其他用户自定义项目的名称。一个标识符以字母 A-Z 或 a-z 或下划线 `_` 开始，后跟零个或多个字母、下划线和数字（0-9）。

C 标识符内不允许出现标点字符，比如 `@`、`$` 和 `%`。C 是**区分大小写**的编程语言。因此，在 C 中，*Manpower*和 *manpower* 是两个不同的标识符。下面列出几个有效的标识符：

|          |       |     |           |        |
|----------|-------|-----|-----------|--------|
| mohd     | zara  | abc | move_name | a_123  |
| myname50 | _temp | j   | a23b9     | retVal |

## 关键字

下表列出了 C 中的保留字。这些保留字不能作为常量名、变量名或其他标识符名称。

| 关键字      | 说明                       |
|----------|--------------------------|
| auto     | 声明自动变量                   |
| break    | 跳出当前循环                   |
| case     | 开关语句分支                   |
| char     | 声明字符型变量或函数返回值类型          |
| const    | 声明只读变量                   |
| continue | 结束当前循环，开始下一轮循环           |
| default  | 开关语句中的"其它"分支             |
| do       | 循环语句的循环体                 |
| double   | 声明双精度浮点型变量或函数返回值类型       |
| else     | 条件语句否定分支（与 if 连用）        |
| enum     | 声明枚举类型                   |
| extern   | 声明变量或函数是在其它文件或本文件的其他位置定义 |
| float    | 声明浮点型变量或函数返回值类型          |
| for      | 一种循环语句                   |

|          |                       |
|----------|-----------------------|
| goto     | 无条件跳转语句               |
| if       | 条件语句                  |
| int      | 声明整型变量或函数             |
| long     | 声明长整型变量或函数返回值类型       |
| register | 声明寄存器变量               |
| return   | 子程序返回语句（可以带参数，也可不带参数） |
| short    | 声明短整型变量或函数            |
| signed   | 声明有符号类型变量或函数          |
| sizeof   | 计算数据类型或变量长度（即所占字节数）   |
| static   | 声明静态变量                |
| struct   | 声明结构体类型               |
| switch   | 用于开关语句                |
| typedef  | 用以给数据类型取别名            |
| unsigned | 声明无符号类型变量或函数          |
| union    | 声明共用体类型               |
| void     | 声明函数无返回值或无参数，声明无类型指针  |
| volatile | 说明变量在程序执行中可被隐含地改变     |
| while    | 循环语句的循环条件             |

C99 新增关键字

|       |          |            |        |          |
|-------|----------|------------|--------|----------|
| _Bool | _Complex | _Imaginary | inline | restrict |
|-------|----------|------------|--------|----------|

C11 新增关键字

|                |               |         |          |           |
|----------------|---------------|---------|----------|-----------|
| _Alignas       | _Alignof      | _Atomic | _Generic | _Noreturn |
| _Static_assert | _Thread_local |         |          |           |

# C 中的空格

只包含空格的行，被称为空白行，可能带有注释，C 编译器会完全忽略它。

在 C 中，空格用于描述空白符、制表符、换行符和注释。空格分隔语句的各个部分，让编译器能识别语句中的某个元素（比如 int）在哪里结束，下一个元素在哪里开始。因此，在下面的语句中：

```
int age;
```


在这里，int 和 age 之间必须至少有一个空格字符（通常是一个空白符），这样编译器才能够区分它们。另一方面，在下面的语句中：


```
fruit = apples + oranges;    // 获取水果的总数
```

fruit 和 =，或者 = 和 apples 之间的空格字符不是必需的，但是为了增强可读性，您可以根据需要适当增加一些空格。

← C 程序结构

C 数据类型 →

 **4 篇笔记**

 **写笔记**