



# C 语言光速入门

next page →

# 课程介绍

C 语言入门 +

- qq : 2820047809 (Trdthg)



For IO Club 成员



Presenter 软件 20 阎明铸



QQ 2820047809



Github trdthg



# 资源

C 语言学习资料推荐 /

- 视频 : 翁恺 C 语言

- 文档 :

阮一峰 C 语言入门教程

C++ 风格指南

各类培训机构的课程文档

- 书籍 : <https://github.com/search?q=书籍>

- 算法 :

- 力扣、洛谷

- OI-Wiki、代码随想录、杭电 OJ、浙大 OJ



编程2021C语言零基础视  
C语言入门初学者小白...  
!大学计算机... · 2021-8-24



【C语言】《带你学C带你飞》  
鱼C-小甲鱼 · 2018-7-26



【清华大学】C语言课程，  
通C语言程序设计！C语言  
秃头美少年坤坤 · 9-14

言学那么多语法有用吗？  
一个小球的编程之旅



球，27分钟，讲清楚所  
语言语法知识  
老师 · 5-19



翁恺C语言  
carrot-boer · 2018-1-6



【C语言教程600集】目前最  
完整的C语言教程，包含所  
编程不加班 · 9-24



《C#语言入门详解》全



嵌入式C语言



100道  
C语言经典100题（手把手  
）

# 课程内容

- 🎥 C 语言简介 - 应用
- 💻 开发环境 - IDE / 插件
- 📋 C 语言基本语法 - 数据类型 / 流程控制
- 🚀 项目实战

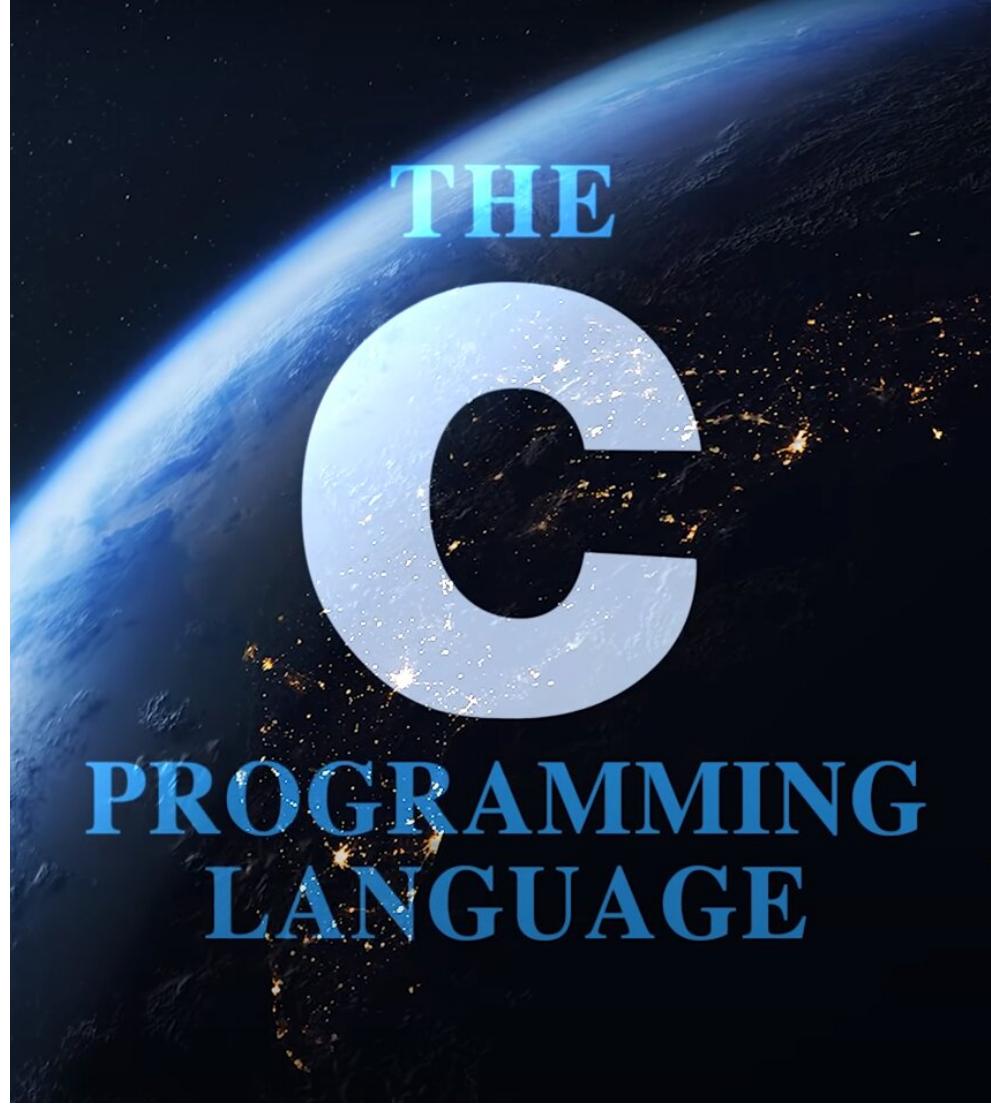
```
#include <stdio.h>

int main() {
    printf("Hello World!");
    return 0;
}
```



# C 语言简介

1. 操作系统一般都是由 C 语言编写
  - C 语言可以直接使用操作系统提供的功能
2. C 语言是嵌入式系统的首选编程语言
3. C 语言几乎成为事实标准
4. 高性能 / 可移植 / 手动管理内存…



# 开发环境搭建

运行 C 语言编写的程序需要的工具

## 1. 需要什么？

- 任意文本编辑器
- gcc 等编译器
- CMD 等命令行

## 2. 如何使用？

- 编写代码
- 把 原始的文本文件 编译成 可执行文件
- 运行

使用 nano + 命令行 演示

The screenshot shows a terminal window with three tabs: '输出' (Output), '调试控制台' (Debug Console), and '终端' (Terminal). The '终端' tab is active, displaying the following sequence of commands and output:

```
project/club/ppt/code/c)
55:33)→ gcc hello.c -o hello
project/club/ppt/code/c)
55:39)→ ./hello
world
project/club/ppt/code/c)
55:43)→ |
```

The code editor at the top shows a C file named 'hello.c' containing the following code:

```
hello.c > { main(int argc, char const *argv[])
    include <stdio.h>
    int main(int argc, char const *argv[])
        printf("hello world\n");
        return 0;
}
```

# 区别

## 2. 疑问

- 什么是 Dev C++ ?



- 什么是 VS ?



Visual C++ 6

- 它们有什么区别 ?
- 我应该如何选择 ?
- gcc 和它们有什么关系 ?

# 编译器

C / Cpp 编译器 & 工具链.<sup>[1]</sup>

支持平台	编译器	工具链	Logo	安装
Linux	gcc	GNU		<code>apt install gcc</code>
Windows	gcc	MinGW <sup>[2]</sup>	 MinGW-w64 A complete runtime environment for GCC & LLVM for 32 and 64 bit Windows	<code>直接下载解压</code> or <i>Dev C++</i>
Windows	CL	MSVC		下载 VS
All	clang	Clang & LLVM		<code>apt install clang</code> 使用 VS 安装

1. 工具链是什么？包含 Binutils、GCC、Glibc

2. Minimalist GNU for Windows

# 回答

1. 什么是 Dev C++ ? 什么是 VS ?



代码编辑器 + 编译器 + (代码高亮 + 调试 + 一键编译运行等)

2. 它们有什么区别 ?

使用的编译器不同 , MinGW (gcc) , MSVC (cl)

3. gcc 和它们有什么关系 ?

■ 什么是 IDE ?

■ 什么是 vscode ?



■ 代码编辑器

■ (代码高亮 + 基本的代码补全等)

# 推荐

- cph
- c/c++

# 目录

## C 语言语法

- 数据类型
- 控制流
- 函数
- 数组
- 结构体 & 指针
- 文件 & IO & 标准库
- malloc 内存布局

## 通识

- 环境变量
- 编译（器）
- 包管理器 | 第三方库
- IDE / 常用插件
- 格式化 / 补全 / 高亮
- Git / Github
- Markdown / Editor

A close-up photograph of a traditional Chinese wooden gong. A black mallet is shown in motion, creating a blurred circular effect at its tip as it strikes the gong. The gong itself has a warm, reddish-brown finish. Below the gong, concentric ripples are visible in the dark, reflective water, suggesting the sound is being projected into the water.

# 数据类型

整数类型

浮动数（小数）

字符型

## 其他语言的类型定义

## Rust

Go

```
package main

func main() {
    var a8 int8 = 8
    var a16 int16 = 8
    var a32 int32 = 8
    var a64 int64 = 8

    var u8 uint8 = 0
    var u16 uint16 = 0
    var u32 uint32 = 0
    var u64 uint64 = 0

    // byte 和 uint8
    var c2 uint8 = 'a'
    var c1 byte = 'a' // byte 是 uint8 的别名，和 uint8 效果相同

    print(a8, a16, a32, a64, u8, u16, u32, u64)
}
```

# 数据类型

# C 语言基本数据类型

## 整数类型

## 浮点类型

```
float c1 = 1.0;          // 32  
double c2 = 1.0;         // 64  
long double c3 = 1.0;   // 128
```

## 字符型

```
char c1 = 'a';  
char c2 = '\0';
```

## ASCII 码对照表

## 转义字符

A close-up photograph of a traditional wooden boat's bow and hull. The wood is a warm, weathered brown. A black compass rose is mounted on the bow. The boat is on dark, rippling water.

# 流程控制

判断 (if, switch)

循环 (for, while)

# if

## 格式

```
if (xxx) {  
    ...  
} else if (xxx) {  
    ...  
} else {  
    ...  
}
```

## 作用域

```
#include <stdio.h>  
int main() { // scope 0  
    int i = 0;  
  
    { // scope 1  
        int a = 0;  
        printf("%d", i);  
    }  
  
    printf("%d", a);  
  
    { // scope 2  
        int b = 0;  
        printf("%d", i);  
    }  
  
    return 0;  
}
```

当语句只有一条时，可以省略大括号

```
if (i == 0)  
    printf("只有一条语句\n");  
else if (i == 1)  
    printf("也只有一条语句\n");  
else  
    printf("还只有一条语句\n");
```

# switch

格式

```
switch (i)
{
    case 1:
        statement1;
        statement2;
        break;
    case 2:
        statement;
        break;
    ...
    default:
        statement;
}
```

小练习：FSM（有限状态机）

```
#include <stdio.h>
int main() {
    return 0;
}
```

# 循环

for

```
for ( 初始化 ; 终止条件 ; 变量更新 ) {  
    ...  
}  
  
for (;;) {  
    // 无限循环  
}
```

while

```
while (终止条件) {  
    ...  
}
```

Break 和 Continue

```
#include<stdio.h>  
int main() {  
    for(int i = 0; i < 5; i++) {  
        if (i == 3) {  
            continue; //  
            break; //  
        }  
        printf("%d\n", i); // 0 1 2  
    }  
}
```

# 函数

## 1. 什么是函数？

f(x)

- 函数就好像有输入和输出的机器。



## 2. 函数的格式

```
int add(int a, int b) {  
    return a + b;  
}
```

练习：分段函数 `y = |x|`

```
#include <stdio.h>  
  
int main() {  
  
    return 0;  
}
```

# void 是什么类型？

void 在不同地方有不同的含义

## 1. 用作函数返回值

```
void printSomeSentence()
{
    printf("=====\\n");
    printf("Company *****\\n");
    printf("Fax ***** \\n");
    printf("Email ***** \\n");
    printf("=====\\n");
}
```

## 2. 用作函数参数

```
void printSomeSentence(void)
{
    printf("=====\\n");
    printf("Company *****\\n");
    printf("Fax ***** \\n");
    printf("Email ***** \\n");
    printf("=====\\n");
}
```

## 3. 用作变量类型 (x)

# main 函数的参数

```
#include<stdio.h>
int main(int argc, char* argv[]) {
    // argc: 参数数量
    // argv: 参数数组

    return 0;
}
```

初中生也能看懂的 C/C++ 类型声明规则教学，很简单的！

# 数组

固定长度，编译时就知道大小

一维数组

```
int a[6];  
  
char b[6];  
  
double c[6];
```

使用下标（索引）访问

```
#include <stdio.h>  
int main() {  
    char chars[5] = {'a', 'b', 'c', 'd', 'e'};  
    //           0   1   2   3   4  
  
    char chars_first = chars[0];  
    char chars_last = chars[4];  
  
    for (int i = 0; i < 5; i++) {  
        printf("%c\n", chars[i]);
```





# 字符串

字符串

```
char chars[10] = {'a', 'b', 'c', 'd', 'e', '\0'};  
printf("%s", chars); // abcde
```

字符串以 `'\0` 结尾

```
char chars[10] = {'a', 'b', 'c', 'd', 'e', '\0', 'a'  
printf("%s", chars); // abcde
```

字符串的简化写法

```
char chars[] = {"abcde"};  
char chars[] = "abcde";
```



# C 标准库 string.h 文档

## 1. 获取字符串长度

```
#include <string.h>
char str[] = "abcde";
int len = strlen(str);
```

## 2. 字符串复制

```
char str1[] = "abcde";
char str2[10];
strcpy(str2, str1);
```

## 3. 比较两个字符串是否相等

```
char str1[] = "aaa";
char str2[] = "bbb";

int res = strcmp(str1, str2);

if (res == 0)
    printf("str1 等于 str2\n");
else
    printf("str1 和 str2 不相等\n");
```

# 二维数组

```
#include <stdio.h>
int main(int argc, char *argv[]) {
    int arr[10][10] = {};
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

# main 函数的参数

```
#include<stdio.h>
int main(int argc, char* argv[]) {
    return 0;
}
```

# `#define`

```
#define LIGHT 1
#define DARK 1

enum Week {
    Monday = 0,
    Tuesday = 1,
    Wednesday = 2,
};

int main(int argc, char const *argv[]) {
    printf("%d\n", LIGHT);
    return 0;
}
```

## 1. 预处理 (Preprocessing)

- 预处理用于将所有的#include 头文件以及宏定义替换为其实现内容

## 2. 编译 (Compilation)

- 将经过预处理之后的程序转换成特定汇编代码

## 3. 汇编 (Assemble)

- 汇编过程将上一步的汇编代码转换成机器码

## 4. 链接 (Linking)

- 链接过程将多个目标文以及所需的库文件 (.so / .dll) 链接成最终的可执行文件。

templete