# 目录

[目录 1](#_Toc520211367)

[第4章 C语言-函数 2](#_Toc520211368)

[4.1 C语言函数概述 2](#_Toc520211369)

[4.1.1 何谓函数 2](#_Toc520211370)

[4.2 自定义函数 2](#_Toc520211371)

[4.2.1 自定义函数定义 2](#_Toc520211372)

[4.2.2 自定义函数声明 3](#_Toc520211373)

[4.2.3 自定义函数使用 4](#_Toc520211374)

[4.3 多文件操作 10](#_Toc520211375)

[4.4 函数递归算法 12](#_Toc520211376)

[4.4.1 何谓递归算法 12](#_Toc520211377)

[4.4.2 递归算法如何实现 12](#_Toc520211378)

# C语言-函数

## C语言函数概述

### 何谓函数

函数：就是将具有特定功能的代码封装起来 printf函数 scanf函数

在C语言中：

1. 主函数：作为程序入口，有且只有一个
2. 子函数(自定义函数)：是程序员自己编写的，实现自己想要的功能
3. 库函数：别人写好了，只要调用即可

### 常见库函数

1. strlen函数：计算字符串的长度
2. 头文件：#include <string.h>
3. 函数原型：size\_t strlen( char \*str );//char a[10] = “sadsads”;
4. 函数形参：char \*str：需要计算的字符串的首地址
5. 返回值：计算的长度(int) 不包含：’\0’
6. strcat函数：字符串连接函数
7. 头文件：#include <string.h>
8. 函数原型：char \*strcat( char \*str1, const char \*str2 );
9. 函数形参：

char \*str1：被连接的字符串

const char \*str2：要连接的字符串

将str2连接到str1的后面

1. 返回值：返回指针str1
2. strcmp函数：比较字符串大小
3. 头文件：#include <string.h>
4. 函数原型：int strcmp( const char \*str1, const char \*str2 );
5. 函数形参

这const char \*str1、const char \*str2进行比较

1. 返回值：

大于0： str1 > str2

等于0： str1 = str2

小于0： str1 < str2

1. strcpy函数：字符串拷贝函数
2. 头文件：#include <string.h>
3. 函数原型：char \*strcpy( char \*to, const char \*from );
4. 函数形参：

char \*to：到

char \*from：从

从from字符串拷贝到to字符串里面

1. 返回值：返回值为指针to
2. menset函数：给数组设定一个值
3. 头文件：#include <string.h>
4. 函数原型：void \*memset( void \*buffer, int ch, size\_t count );
5. 函数形参：

void \*buffe：万能指针

int ch：要写入缓冲区的值

size\_t count：要写入缓冲区的个数

1. pow函数：求一个数的n次幂
2. 头文件：#include <math.h>
3. 函数原型：double pow( double base, double exp );
4. 函数形参：

double base：底

double exp：幂

1. 返回值：返回计算结果
2. abs函数：求一个数的绝对值
3. 头文件：#include <stdlib.h>
4. 函数原型：int abs( int num );
5. 函数形参：int num：求这个数的绝对值
6. 返回值：返回结果
7. sprintf函数：将信息输出到字符数组
8. 头文件：#include <stdio.h>
9. 函数原型：int sprintf( char \*buffer, const char \*format, ... );
10. 函数形参：

char \*buffer：指向字符数组的首地址

const char \*format：格式控制字符串

参数：在格式控制字符串里面有多少个格式控制字符就需要多少个参数

1. 返回值：是写入的字符数量
2. gets函数：获取从键盘输入的字符串
3. 头文件：#include <stdio.h>
4. 函数原型：char \*gets( char \*str );
5. 函数形参：char \*str：需要保存进去的缓冲区首地址
6. 与scanf的区别

scanf：遇到空格、回车、换行都会停止读取

gets：遇到回车、换行都会停止读取

1. puts函数：输出字符串
2. 头文件：#include <stdio.h>
3. 函数原型：int puts( char \*str );
4. 函数形参：char \*str：需要输出缓冲区的首地址
5. 返回值：成功返回非负值, 失败时返回EOF
6. getchar函数：获取从键盘输入的一个字符
7. 头文件：#include <stdio.h>
8. 函数原型：int getchar( void );
9. 函数形参：void
10. 返回值：返回读取到字符
11. putchar函数：输出一个字符到屏幕上
12. 头文件：#include <stdio.h>
13. 函数原型：int putchar( int ch ); ‘a’
14. 函数形参：int ch：需要输出去的字符
15. 返回值：返回值是被写的字符, 发生错误时返回EOF.

## 自定义函数

### 自定义函数定义

目的：想要实现某种功能 把函数就当成一个功能来看

格式：

|  |
| --- |
| //自定义函数的格式  返回值类型 函数名(形参列表) 🡪 函数头  {  函数体；  } |

1. 返回值类型

可以是基本数据类型、构造数据类型、指针类型、空类型

怎么确定函数的返回值类型？由函数的功能来定

1. 输出hello world函数，不需要返回任何东西，返回值类型就为void类型
2. 判断闰年，需要返回一个判断的结果，返回值类型是int类型
3. 求两个int和，需要将计算结果返回出来，返回值类是int类型
4. 函数名
5. 满足标识符的命名规则，具有含义
6. 不能与库函数重名
7. 作为函数的入口地址
8. 形参列表 形参的个数
9. 形参与实参的位置是一样，不同的场合，代表的意义不同
10. 形参：形式上参数，在函数定义和声明的时候表示形参
11. 实参：实际的参数，在函数调用时表示实参

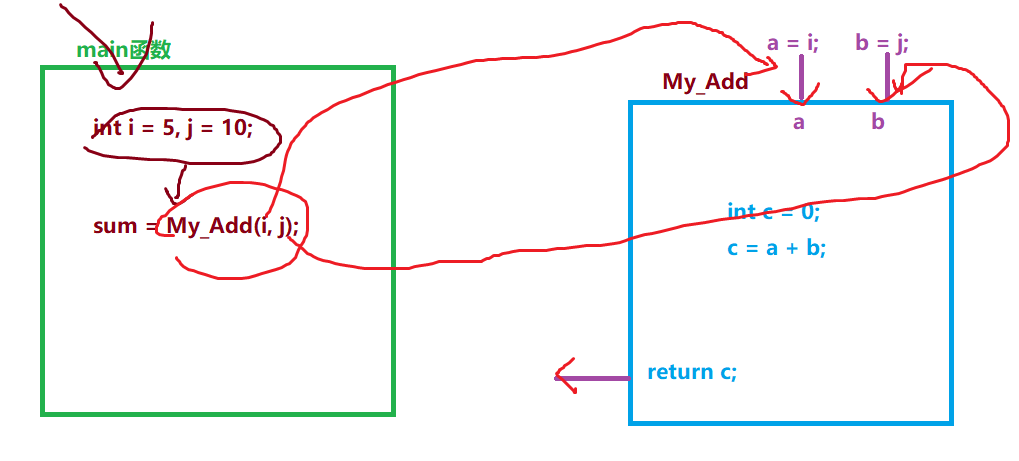
外部函数的数据不能直接到达内部函数

所以内部函数就会提供接口(形参)给外部函数的数据使用

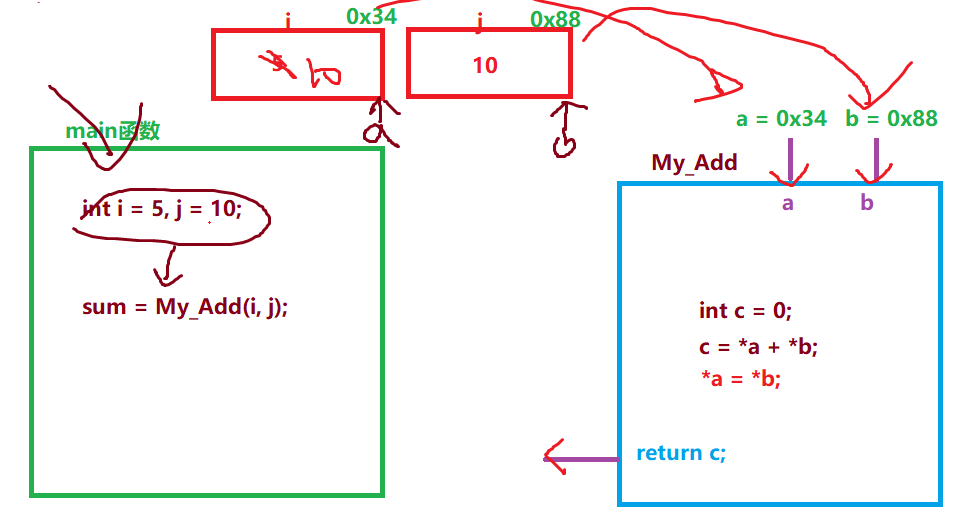
1. 实参和形参的传递方向

将实参赋值给形参的过程，方向单一(实参 🡪 形参)

1. 传递方式
2. 值传递：改变形参时，会改变实参？不会



1. 地址传递：改变形参，就会影响实参



1. 形参个数和类型怎么确定？由函数功能而定和经验

(1) 输出hello world函数，不需要给他数据，形参类型就为void类型

(2) 判断闰年，需要给他一个年份，形参的个数1个，类型为int类型

(3) 求两个int和，需要给他两个求和的数据，形参的个数为2，类型为int类型

1. 函数体：功能代码
2. 函数定义放在哪里？

可以放在主函数前面、可以放在主函数后面、也可以放在别的文件中(多文件操作)

### 自定义函数声明

目的：告诉编译器，该函数已经弄好了，你们可以去使用它了

基本格式

|  |
| --- |
| 返回值类型 函数名(形参列表);  函数头 + 分号； |

注意：

1. 在函数声明时，返回值类型要和函数定义时一样
2. 在函数声明时，形参的个数和类型要和函数定义时一样，形参名可以省略
3. 调用该函数前要进行声明
4. 如果函数定义在主函数前面，可以不用声明
5. 函数声明在多文件操作中，放在对应.h头文件里面

### 自定义函数使用

目的：实现功能

1. 带返回值

|  |
| --- |
| 变量 = 函数名(实参列表)；//变量的类型要和返回值类型一样 实参：变量、常量、表达式  变量 = 函数名( ); //如果形参列表为void类型，要把void去掉 |

1. 不带返回值

|  |
| --- |
| 函数名(实参列表)；//变量的类型要和返回值类型一样  函数名( ); //如果实参列表为void类型，要把void去掉 |

1. 注意
2. 如果返回值类型是指针类型，那么如果返回的是函数里面的变量地址，那么就会出问题(无效指针)
3. 可以子函数调用子函数，主函数调用子函数，但是子函数不能调用主函数
4. 如果子函数自己调用自己，变成递归函数

课堂练习：

判断一个素数的函数？

返回值：判断结果 1 是素数 0 不是素数 int类型

形参个数：1 整数 int类型

## 多文件操作

1. 为什么要多文件操作
2. 方便查看代码、方便修改
3. 方便移植
4. 多文件操作需要怎么做
5. 需要哪些文件

添加一个模块就需要添加需要两个文件

.c文件：源文件：主要用于该模块函数定义

.h文件：头文件：主要是存放函数声明、宏定义、包含其他头文件、声明外部变量

1. 怎么构建
2. 新建.c文件
3. 再建.h文件(名字要一样)
4. .c文件里面要包含对应.h文件
5. .h里面还有标准模板

|  |
| --- |
| 模板：  编译有四步  预处理、编译、汇编、链接  #：预处理命令  #ifndef：条件编译    #ifndef XXXX(宏名) 如果没有定义这个宏名，  那就选择将下面的代码加进去编译  #define XXXX(宏名) 保证独立性、防止不重复声明    1. 包含其它头文件  2. 函数声明  3. 宏定义  4. 声明外部变量  #endif |

1. 构造多文件
2. 新建工程
3. 新建一个main.c文件
4. 新建一个main.h文件

## 函数递归算法

### 何谓递归算法

子函数自己调用自己

怎么实现递归？

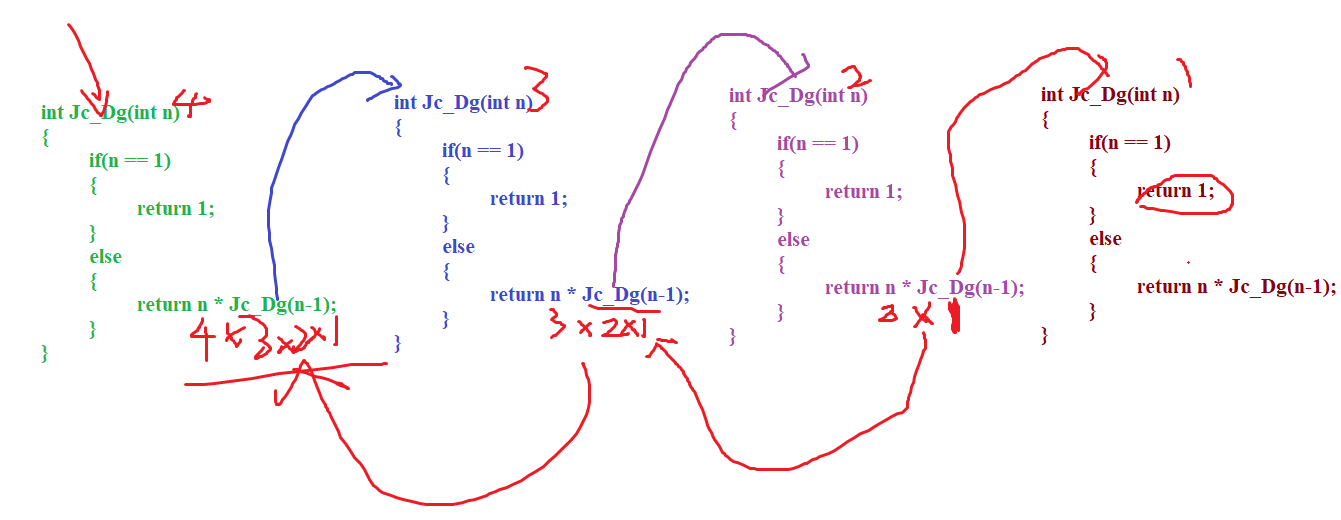
找规律：

1. 前一项和后一项的关系
2. 找出口(确定的值)

5! = 5 \* 4 \* 3 \* 2 \* 1;

1. f(n) = n \* f(n-1);
2. f(1) = 1;

4! = 4 \* 3 \* 2 \* 1;



### 递归算法如何实现

实例1：

1+2+3+….+100 5050

找关系？

f(n) = n+f(n-1);

f(1) = 1;

实例2：

1 1 2 3 5 8 13 … 用递归得到第n项的值

1. f(n) = f(n-1)+f(n-2)
2. f(1) = f(2) = 1;