

C 语言自定义函数

函数是一段可以重复使用的代码，用来独立地完成某个功能，它可以接收用户传递的数据，也可以不接收。

无参函数的定义

如果函数不接收用户传递的数据，那么定义时可以不带参数。如下所示：

```
返回值类型 函数名(){  
  
    函数体  
  
}
```

说明：

返回值类型可以是 C 语言中的任意数据类型，例如 int、float、char 等。

函数名是标识符的一种，命名规则和标识符相同。函数名后面的括号()不能少。

函数体是函数需要执行的代码。即使只有一个语句，也要由{ }包围。

在函数体中使用 return 语句返回数据。

例如，定义一个函数，计算 1 加到 100 的结果：

```
1. int sum() {  
2.     int i, sum=0;  
3.     for (i=1; i<=100; i++) {  
4.         sum+=i;  
5.     }  
6.     return sum;  
7. }
```

计算结果保存在变量 `sum` 中，通过 `return` 语句返回。`sum` 为 `int` 型，所以返回值类型也必须为 `int`，要一一对应。

`return` 是 C 语言中的一个关键字，只能用在函数中，用来返回处理结果。

将上面的代码补充完整：

```
1. #include <stdio.h>
2.
3. int sum() {
4.     int i, sum=0;
5.     for (i=1; i<=100; i++) {
6.         sum+=i;
7.     }
8.     return sum;
9. }
10.
11. int main()
12. {
13.     int a = sum();
14.     printf("The sum is %d\n", a);
15.     return 0;
16. }
```

运行结果：

```
The sum is 5050
```

函数不能嵌套定义，main 也是一个函数定义，要将 sum 放在 main 外面。函数必须先定义后使用，所以 sum 只能在 main 前面。

注意：main 是函数定义，不是函数调用。当可执行文件加载到内存后，系统从 main 函数开始执行，也就是说，系统会调用我们定义的 main 函数。

无返回值函数

有的函数不需要有返回值，或者返回值类型不确定（很少见），那么用 **void** 表示，例如：

```
1. void hello() {
2.     printf ("Hello,world \n");
3.     //没有返回值就不需要 return 语句
4. }
```

`void` 是 C 语言中的一个关键字，表示空类型或无类型，绝大部分情况下也就意味着没有 `return` 语句。

有参函数的定义

如果函数需要接收用户传递的数据，那么定义时就要带参数。如下所示：

```
返回值类型 函数名(参数列表){  
  
    函数体  
  
}
```

用户数据通过“参数列表”传递给函数，供函数处理。例如，定义一个函数求两个数中的最大值：

```
1. int max(int a, int b){  
2.     if (a>b){  
3.         return a;  
4.     }else{  
5.         return b;  
6.     }  
7. }
```

参数 (Parameter) 本质上也是变量，定义时要指明参数类型和参数名称。参数列表中
可以定义一个或多个参数，多个参数之间用逗号分隔。参数列表中给出的参数可以在函数
体中使用。

调用 `max()` 函数时可以直接传递整数：

```
int n = max(10, 20);
```

也可以传递变量：

```
int a = 10, b = 20;  
int n = max(a, b);
```

也可以整数和变量一起传递：

```
int a = 10;
int n = max(a, 20);
```

变量 n 得到的值都是 20。

函数定义时给出的参数称为**形式参数**，简称**形参**，函数调用时给出的参数（传递的数据）

称为**实际参数**，简称**实参**。函数调用时，将实参的值传递给形参，相当于一次赋值操作。**注**

意：实参和形参的类型、数目必须一致。

将上面的代码补充完整：

```
1. #include <stdio.h>
2.
3. int max(int a, int b){
4.     if (a>b){
5.         return a;
6.     }else{
7.         return b;
8.     }
9. }
10.
11. int main() {
12.     int num1, num2, maxVal;
13.     printf("Input two numbers: ");
14.     scanf("%d %d", &num1, &num2);
15.     maxVal = max(num1, num2);
16.     printf("The max number: %d\n", maxVal);
17.
18.     return 0;
19. }
```

运行结果：

```
Input two numbers: 100 200
```

```
The max number: 200
```

定义 max 时，变量 a、b 的值都是未知的；调用 max 时，分别将 num1、num2 的值传递给 a、b，类似于：

```
a=num1;  
b=num2;
```

return 语句可以有多个，可以在函数体的任意位置。在 max 中，根据 if 的判断结果来执行不同的 return 语句。

函数一旦遇到 return 语句就返回（停止执行），后面的所有语句都不会被执行到，例如：

```
1. int max(int a, int b){  
2.     int n = (a>b) ? a : b;  
3.     return n;  
4.     printf("Function is performed\n");  
5. }
```

第 4 行代码是多余的，永远没有执行的机会。