

C 语言函数的参数和返回值

函数的参数

函数的参数分为形参和实参。形参出现在函数定义中，在整个函数体内都可以使用，离开该函数则不能使用。实参出现在函数调用中。

形参和实参的功能是作数据传送，发生函数调用时，实参的值会传送给形参。

形参和实参有以下几个特点：

- 1) 形参变量只有在函数被调用时才分配内存单元，在调用结束时，立刻释放所分配的内存单元。因此，形参只有在函数内部有效，不能在函数外部使用。
- 2) 实参可以是常量、变量、表达式、函数等，无论实参是何种类型的数据，在进行函数调用时，它们都必须有确定的值，以便把这些值传送给形参。因此应预先用赋值，输入等办法使实参获得确定值。
- 3) 实参和形参在数量上、类型上、顺序上必须严格一致，否则会发生“类型不匹配”的错误。

函数调用中发生的数据传送是单向的，只能把实参的值传送给形参，而不能把形参的值反向地传送给实参。因此在函数调用过程中，形参的值发生改变，而实参中的值不会变化。

【示例】计算 $1+2+3+\dots+(n-1)+n$ 的值。

```
1. #include <stdio.h>
2.
3. int sum(int n){
4.     int i;
5.     for(i=n-1; i>=1; i--){
6.         n+=i;
7.     }
8.     printf("The inner n = %d\n",n);
9.
10.    return n;
```

```

11. }
12.
13. int main() {
14.     int n, total;
15.     printf("Input a number: ");
16.     scanf("%d",&n);
17.     total = sum(n);
18.     printf("The outer n = %d \n", n);
19.     printf("1+2+3+...+(n-1)+n = %d\n", total);
20.
21.     return 0;
22. }

```

运行结果：

Input a number : 100 ✓

The inner n = 5050

The outer n = 100

1+2+3+...+(n-1)+n = 5050

通过 scanf 输入 n 的值，作为实参，在调用 sum 时传送给形参 n。

注意：本例中形参变量和实参变量的名称都是 n，但这是两个不同的量，各自的作用域不同，下节将会讲解。

在 main 函数中用 printf 语句输出一次 n 值，这个 n 值是实参 n 的值。在函数 sum 中也用 printf 语句输出了一次 n 值，这个 n 值是形参最后取得的 n 值。

从运行情况看，输入 n 值为 100，即实参 n 的值为 100，把此值传给函数 sum 时，形参 n 的初值也为 100，在执行函数过程中，形参 n 的值变为 5050。函数运行结束后，输出实参 n 的值仍为 100。可见实参的值不随形参的变化而变化。

函数的返回值

函数的值(或称函数返回值)是指函数被调用之后，执行函数体中的程序段所取得的值，

可以通过 `return` 语句返回。

`return` 语句的一般形式为：

```
return 表达式;
```

或者：

```
return (表达式);
```

例如：

```
return max;  
return a+b;  
return (100+200);
```

函数中可以有多条 `return` 语句，但每次调用只能有一个 `return` 语句被执行，所以只有一个返回值。

一旦遇到 `return` 语句，不管后面有没有代码，函数立即运行结束，将值返回。例如：

```
1. int func() {  
2.     int a=100, b=200, c;  
3.     return a+b;  
4.     return a*b;  
5.     return b/a;  
6. }
```

返回值始终 `a+b` 的值，也就是 300。

没有返回值的函数为空类型，用 `void` 进行说明。例如：

```
1. void func() {  
2.     printf("Hello world!\n");  
3. }
```

一旦函数的返回值类型被定义为 `void`，就不能再接收它的值了。例如，下面的语句是错误的：

```
int a = func();
```

为了使程序有良好的可读性并减少出错，凡不要求返回值的函数都应定义为空类型。