



## 全局变量

在所有函数外部声明的变量，称为全局变量。

全局变量的作用域是从变量定义的位置开始到文件结束。

全局变量的值在程序的整个生命周期内都是有效的。

全局变量在程序执行的全过程中一直占用内存单元。

全局变量在定义时若没有赋初值，其默认值为 0。

在一个函数内部，既可以使用本函数定义的局部变量，也可以使用在此函数前定义的全局变量。

### 【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. //全局变量声明
4. int g;
5. int main()
6. {
7.     int a = 10;
8.     int b = 20;
9.     g = a + b;
10.    cout << g;
11.    return 0;
12. }
```

### 【全局变量的好处】

全局变量使得函数间多了一种传递信息的方式。如果在一个程序中多个函数都要对同一个变量进行处理，就可以将这个变量定义成全局变量，使用非常方便。

### 【全局变量的坏处】

过多地使用全局变量，会增加调试难度。因为多个函数都能改变全局变量的值，不易判断某个时刻全局变量的值。

过多地使用全局变量，会降低程序的通用性。如果将一个函数移植到另一个程序中，需要将全局变量一起移植过去，同时还可能出现重名问题。



## 局部变量

在函数或一个代码块内部声明的变量，称为局部变量。主函数 main 中定义的变量是局部变量。函数的形参也是局部变量。

局部变量只能被函数内部或者代码块内部的语句使用。所以，在不同的函数中变量名可以相同，它们分别代表不同的对象，在内存中占据不同的内存单元，互不干扰。

如下面代码中的 i 是在该 for 循环语句中定义的，存在时间和作用域只能被限制在该 for 循环语句中。

```
1. for(int i=0; i<=n; i++)
2. {
3.     sum += i;
4. }
```

局部变量的存储空间是临时分配的，当函数执行完毕，局部变量的空间就被释放，其中的值无法保留到下次使用。

一个局部变量和一个全局变量是可以重名的，在相同的作用域内局部变量有效时全局变量无效。即局部变量可以屏蔽全局变量。

全局变量初始全部为 0；局部变量值是随机的，必须要初始化初值。

局部变量受栈空间大小限制，无法开辟规模很大的数组。