

C 作用域规则

任何一种编程中，作用域是程序中定义的变量所存在的区域，超过该区域变量就不能被访问。C 语言中有三个地方可以声明变量：

1. 在函数或块内部的**局部**变量
2. 在所有函数外部的**全局**变量
3. 在**形式**参数的函数参数定义中

让我们来看看什么是**局部**变量、**全局**变量和**形式**参数。

局部变量

在某个函数或块的内部声明的变量称为局部变量。它们只能被该函数或该代码块内部的语句使用。局部变量在函数外部是不可知的。下面是使用局部变量的实例。在这里，所有的变量 a、b 和 c 是 main() 函数的局部变量。

实例

```
#include <stdio.h>
int main ()
{
    /* 局部变量声明 */
    int a, b;
    int c;
    /* 实际初始化 */
    a = 10;
    b = 20;
    c = a + b;
    printf ("value of a = %d, b = %d and c = %d\n", a, b, c);
    return 0;
}
```

全局变量

全局变量是定义在函数外部，通常是在程序的顶部。全局变量在整个程序生命周期内都是有效的，在任意的函数内部能访问全局变量。

全局变量可以被任何函数访问。也就是说，全局变量在声明后整个程序中都是可用的。下面是使用全局变量和局部变量的实例：

实例

```
#include <stdio.h>
/* 全局变量声明 */
int g;
int main ()
{
    /* 局部变量声明 */
```

```
int a, b;
/* 实际初始化 */
a = 10;
b = 20;
g = a + b;
printf ("value of a = %d, b = %d and g = %d\n", a, b, g);
return 0;
}
```

在程序中，局部变量和全局变量的名称可以相同，但是在函数内，如果两个名字相同，会使用局部变量值，全局变量不会被使用。下面是一个实例：

在程序中，局部变量和全局变量的

实例

```
#include <stdio.h>
/* 全局变量声明 */
int g = 20;
int main ()
{
/* 局部变量声明 */
int g = 10;
printf ("value of g = %d\n", g);
return 0;
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
value of g = 10
```

形式参数

函数的参数，形式参数，被当作该函数内的局部变量，如果与全局变量同名它们会优先使用。下面是一个实例：

实例

```
#include <stdio.h>
/* 全局变量声明 */
int a = 20;
int main ()
{
/* 在主函数中的局部变量声明 */
int a = 10;
int b = 20;
int c = 0;
int sum(int, int);
printf ("value of a in main() = %d\n", a);
c = sum( a, b);
printf ("value of c in main() = %d\n", c);
return 0;
}
/* 添加两个整数的函数 */
int sum(int a, int b)
```

```
{
printf ("value of a in sum() = %d\n", a);
printf ("value of b in sum() = %d\n", b);
return a + b;
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
value of a in main() = 10
value of a in sum() = 10
value of b in sum() = 20
value of c in main() = 30
```

全局变量与局部变量在内存中的区别：

- 全局变量保存在内存的全局存储区中，占用静态的存储单元；
- 局部变量保存在栈中，只有在所在函数被调用时才动态地为变量分配存储单元。

更多内容可参考：[C/C++ 中 static 的用法全局变量与局部变量](#)

初始化局部变量和全局变量


当局部变量被定义时，系统不会对其初始化，您必须自行对其初始化。定义全局变量时，系统会自动对其初始化，如下所示：


| 数据类型 | 初始化默认值 |
|---------|--------|
| int | 0 |
| char | '\0' |
| float | 0 |
| double | 0 |
| pointer | NULL |

正确地初始化变量是一个良好的编程习惯，否则有时候程序可能会产生意想不到的结果，因为未初始化的变量会导致一些在内存位置中已经可用的垃圾值。

← C 函数

C 数组 →

 1 篇笔记

 写笔记



C 的形参与实参

在 C 语言中，形参与实参虽然很简单，但是，是大家比较容易混淆的一个点，这里将为大家详细的讲解。

概念：从字面上理解，所谓形式参数即只是声明了一个作为参数的变量，并未直接进行赋值使用，而实际参数则相反。

如下例

```
#include <stdio.h>

int test(int,int); // 形参，只声明

int main()
{
    int a,b;
    printf("%d",test(5,3)); // 实参，已赋值
}

int test(int a,int b) // 形参
{
    a=a+b;
    return a;
}
```

像上面在 test() 函数里只声明了最为参数的变量，而 main() 函数里则对它赋了值。

K_R 2年前 (2017-08-24)