## **函数是什么**

在C语言中函数的定义：****子程序****。  
子程序是一个大型程序中的某部分代码， 由一个或多个语句块组成。它负责完成某项特定任务，而且相较于其他代码，具备相对的独立性。

## **C语言中函数的分类**

1. 库函数（使用库函数，必须包含 #include 对应的头文件）
2. 自定义函数

## **自定义函数**

自定义函数和库函数一样，有函数名，返回值类型和函数参数。

#include<stdio.h>

int Max(int x,int y)//自定义函数

{

return x>y ? x:y;

}

// int 返回类型

// Max 函数名

// int x,int y 函数参数

int main()

{

int a=10;

int b=20;

int max=Max(a,b);//自定义函数的调用

printf("%d",max);

return 0;

}

## **函数的参数**

****实际参数（实参****）：真实传给函数的参数，叫实参。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量，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 。  
****形式参数（形参）****：形式参数是指函数名后括号中的变量，因为形式参数只有在函数被调用的过程中才分配内存单元，所以叫形式参数。形式参数当函数调用完成之后就自动销毁了。因此形式参数只在函数中有效。

#include <stdio.h>

void Swap1(int x, int y)

{

int tmp = 0;

tmp = x;

x = y;

y = tmp;

}

void Swap2(int \*px, int \*py)

{

int tmp = 0;

tmp = \*px;

\*px = \*py;

\*py = tmp;

}

int main()

{

int num1 = 1;

int num2 = 2;

Swap1(num1, num2);

printf("Swap1 num1 = %d num2 = %d\n", num1, num2);

Swap2(&num1, &num2);

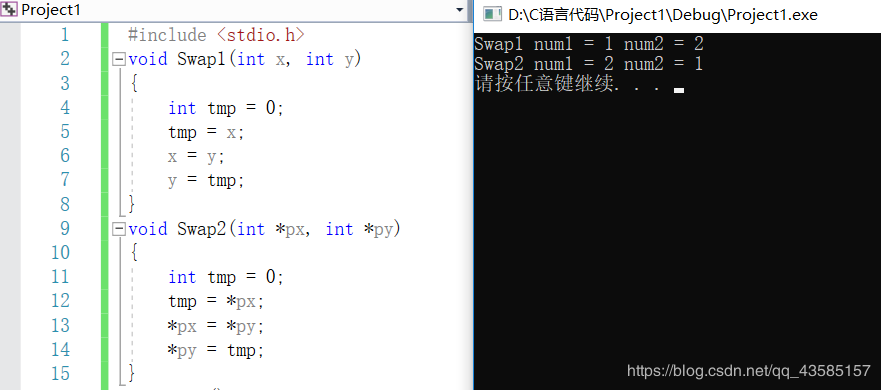
printf("Swap2 num1 = %d num2 = %d\n", num1, num2);

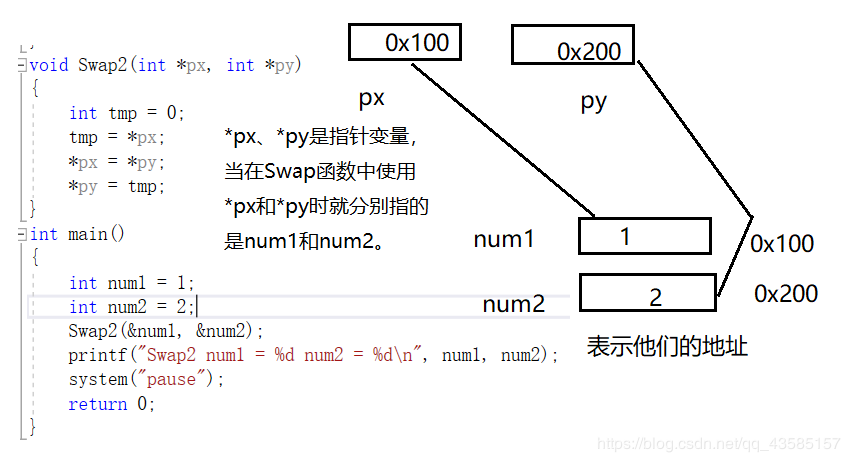
return 0;

}

上面Swap1和Swap2函数中的参数 x，y，px，py 都是形式参数。

在main函数中传给Swap1的num1，num2和传给Swap2函数的&num1，&num2是实际参数。

  
观察Swap1函数并没有交换两个数的值，原因是：\*\*当实参传给形参的时候，形参是实参的一份临时拷贝，对形参的修改并不会改变实参。\*\*通俗讲，就是x，y，num1，num2都有自己独立的空间。在函数中只是改变了x，y的值，并未对num1，num2造成影响。

****分析Swap2****为什么能交换成功  


## **函数的调用**

1.****传值调用****  
函数的形参和实参分别占有不同内存块，对形参的修改不会影响实参。  
*Swap1函数就是传值调用*  
2.****传址调用****  
传址调用是把函数外部创建变量的内存地址传递给函数参数的一种调用函数的方式。这种传参方式可以让函数和函数外边的变量建立起正真的联系，也就是函数内部可以直接操作函数外部的变量。  
*Swap2函数就是传址调用*

## **函数的嵌套调用和链式访问**

//\*\*嵌套调用\*\*

//在一个函数里面可以出现另一个函数

#include <stdio.h>

void one\_line()

{

printf("haha\n");

}

void three\_line()

{

int i = 0;

for(i=0; i<3; i++)

{

onr\_line();

}

}

int main()

{

three\_line();

return 0;

}

//\*\*链式访问\*\*

//把一个函数的返回值座位另一个函数的参数

#include<stdio.h>

#include<string.h>

int main()

{

printf("%d\n",strlen("abcdefg"));

//把strlen的返回值作为printf的参数

return 0;

}