

知识点 1【函数的概述】

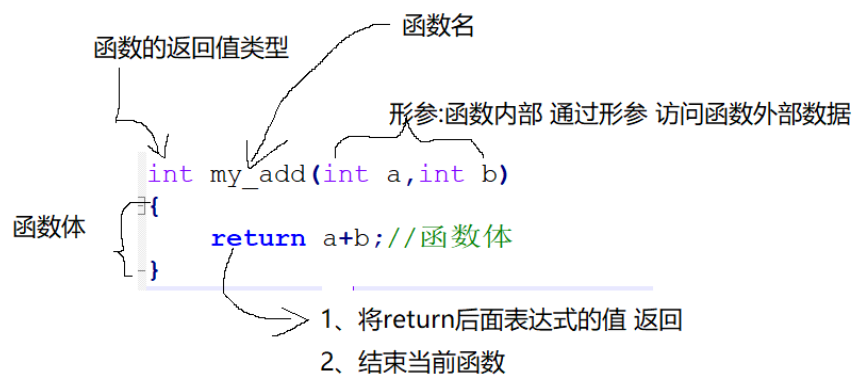
函数是 c 语言的**功能**单位，实现一个功能可以封装一个函数来实现。

1、函数的分类

用户自定义函数、**库函数**（c/c++库、第三方库）、**系统调用**（linux 内核提供的可访问内核资源的函数接口）（**记**）

无参、有参的函数。 有无返回值的函数。

2、函数的定义



函数的**形参**：在函数定义的时候 不开辟空间 只有在**函数调用**的时候 才为形参开辟空间，

形参是**局部变量**。（**背**）

```
#include<stdio.h>

//函数声明：告知编译函数名、函数的参数以及返回值类型 不能实现函数体

int my_add(int a,int b);

void test01()
```

```

{

//函数调用:函数名+(实参) 执行函数体

int ret = my_add(100,200);

printf("ret=%d\n",ret);


//data1 data2 100 200 叫做函数的实参

int data1=10,data2=20;

ret = my_add(data1,data2);

printf("ret=%d\n",ret);

}

int main(int argc,char *argv[])

{

test01();

return 0;

}

//函数的定义：定义函数名、参数、返回值类型以及函数体

int my_add(int a,int b)

{

return a+b;//函数体

}

```

函数名代表的是函数的入口地址（了解 建议不看）

```

18 void test02()
19 {
20     printf("%p\n", my_add);
21
22     int ret = ((int (*)(int,int)) (0x4005f2)) (300,400);
23     printf("ret=%d\n", ret);
24 }
25 int main(int argc,char *argv[])
26 {
27     test02();
28     return 0;
29 }
30 //函数的定义：定义函数名、参数、返回值类型以及函数体
31 int my_add(int a,int b)
32 {
33     return a+b;//函数体
34 }

```

```

2. 10.9.42.114
Re-attach Fullscreen Stay on top Duplicate
edu@edu:~/work/c/day03$ ./a.out
0x4005f2
段错误 (核心已转储)
edu@edu:~/work/c/day03$ gcc 00_code.c
edu@edu:~/work/c/day03$ ./a.out
0x4005f2
ret=700
edu@edu:~/work/c/day03$

```

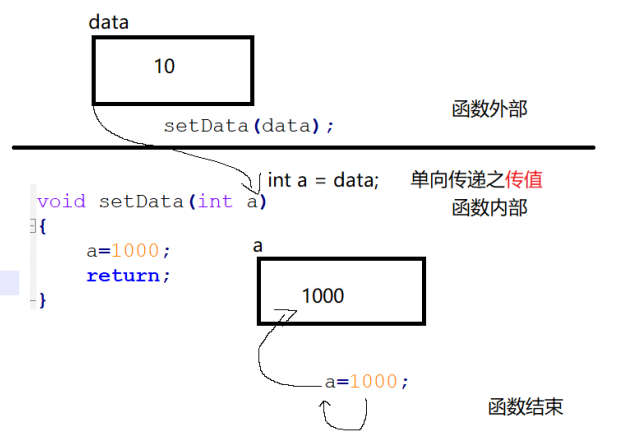
知识点 2 【函数的调用】

1、如果**普通变量** 作为函数的**形参** 函数**内部** 无法**修改外部实参** 值 (记)

```

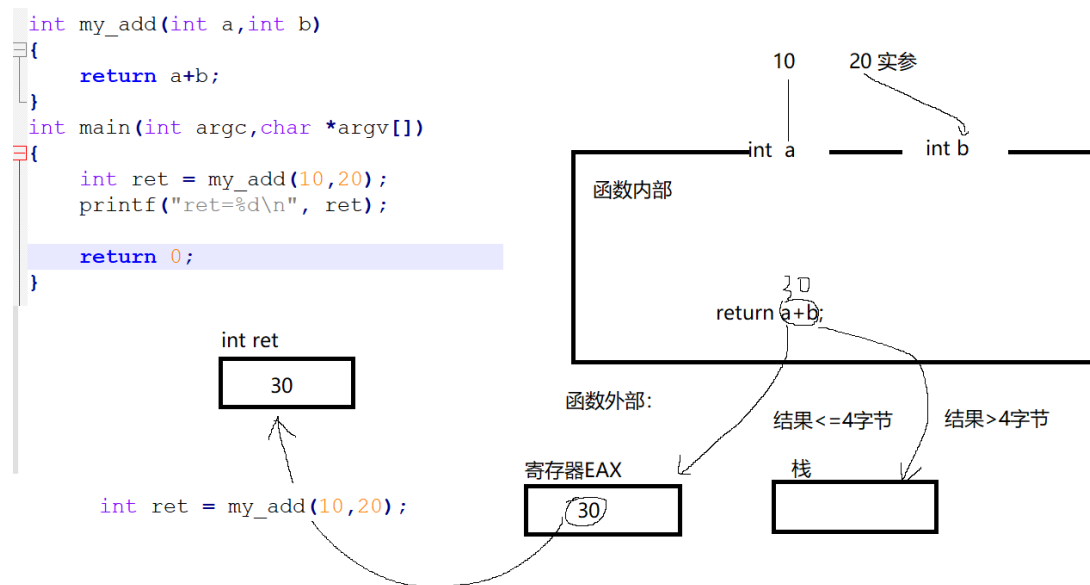
void setData(int a)
{
    a=1000;
    return;
}
int main(int argc,char *argv[])
{
    int data=10;//实参
    setData(data);
    printf("data=%d\n", data);
    return 0;
}

```



2、函数的返回值

将函数执行过程中产生的**必要的结果** 通过返回值 返回给**函数外部**。 (记)



3、一维数值数组作为函数的形参 函数内部 可以通过形参 对外部数组的元素进行读写。（记）

函数内部的形参名 等价 外部的数组名。

```

3 void print_int_array(int array[5], int n)
4 {
5     //操作array等价 操作外部数组arr
6     int i=0;
7     for(i=0;i<n;i++)
8     {
9         printf("%d ",array[i]);
10    }
11    printf("\n");
12    array[2]=1000;
13    return;
14 }
15 void test01()
16 {
17     int arr[5]={10,20,30,40,50};
18     int n = sizeof(arr)/sizeof(arr[0]);
19
20     //实现函数完成int数组的遍历
21     print_int_array(arr, n);
22
23     printf("函数外部arr[2]=%d\n",arr[2]);
24 }

```

```

2.10.9.42.114
Re-attach Fullscreen Stay on top Duplicate
edu@edu:~/work/c/day03$ gcc 02_code.c
edu@edu:~/work/c/day03$ ./a.out
10 20 30 40 50
函数外部arr[2]=1000
edu@edu:~/work/c/day03$

```

案例 1：键盘上输入 10 个数求数组的最大值

```

void input_int_array(int arr[10], int n)
{

    printf("请输入%d 个 int 数值:", n);

    int i=0;

```

```
    for(i=0;i<n;i++)
    {
        scanf("%d", &arr[i]);
    }

    return;
}

int get_max_from_array(int arr[10], int n)
{
    //假设第 0 个元素为最大值

    int max = arr[0];

    int i=0;

    for(i=1;i<n;i++)
    {
        if(max<arr[i])
            max = arr[i];
    }

    return max;
}

void test02()
{
    int arr[5]={0};

    int n = sizeof(arr)/sizeof(arr[0]);
```

```

//实现函数完成 数组的输入

input_int_array(arr, n);


//实现函数完成 数组的最大值

int max = get_max_from_array(arr, n);

printf("max=%d\n",max);

}

```

4、**二维数值数组**作为函数的**形参**，函数**内部** 可以通过**形参** 操作（读写）**外部二维数值数组的元素**。

```

void input_int_two_array(int arr[3][4], int row, int col)

{

    printf("请输入%d 个 int 数值:",row*col);

    int i=0;

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

    {

        int j = 0;

        for(j=0;j<col;j++)

        {

            scanf("%d", &arr[i][j]);

        }

    }

}

```

```
    return;
}

void print_int_two_array(int arr[3][4], int row, int col)
{
    int i=0;

    for(i=0;i<row;i++)
    {
        int j = 0;

        for(j=0;j<col;j++)
        {
            printf("%d ", arr[i][j]);
        }

        printf("\n");
    }
}

void test03()
{
    int arr[3][4]={0};

    int row = sizeof(arr)/sizeof(arr[0]);

    int col = sizeof(arr[0])/sizeof(arr[0][0]);

    //实现函数完成 数组的输入
```

```

input_int_two_array(arr, row, col);

//实现函数完成遍历

print_int_two_array(arr, row, col);
}

```

5、一维字符数组作为函数的形参 函数内部 可以通过形参 操作外部一维字符数组的元素。

```

#include<stdio.h>

#include<string.h>

void input_char_array(char buf[128], int size)

//void input_char_array(char *buf)

{

    //在 64 位平台 任意指针变量的大小为 8 字节

    //printf("内部 sizeof(buf)=%ld\n",sizeof(buf));//8

    //尽量在函数内部不要使用 sizeof 测量 形参的数组名（被优化成指针变量 得到的 4 或
8 字节）

    printf("请输入一个字符串:");

    //fgets(buf,sizeof(buf),stdin);

    fgets(buf,size,stdin);

    buf[strlen(buf)-1]='\0';//去掉输入的回车符

    return;
}

```



```
}

int my_strlen(char buf[128])
{
    int len = 0;

    while(buf[len] && ++len);

    return len;
}

void my_strcpy(char dst[128],char src[128])
{
    //逐个元素拷贝

    int i=0;

    while((dst[i]=src[i]) && ++i);

    return;
}

int my_strcmp(char dst[128],char src[128])
{
    int flag = 0;

    int i=0;

    while( !(flag=dst[i]-src[i]) && dst[i] && ++i);
```

```
    return flag;
}

void test01()
{
    char buf[128]="";

    printf("sizeof(buf)=%ld\n",sizeof(buf));//128

    //定义函数 实现字符数组的键盘输入

    input_char_array(buf, sizeof(buf));


    printf("%s\n", buf);


    //定义一个函数测量字符数组 buf 的长度（遇到'\0'结束）

    int len = my_strlen(buf);

    printf("len=%d\n",len);


    char buf2[128]="";

    //定义一个函数完成将 buf 的字符 拷贝到 buf2 中（遇到'\0'）

    my_strcpy(buf2, buf);

    printf("buf2=%s\n",buf2);


    //定义一个函数完成 buf 和 buf2 的比较（遇到'\0'结束 返回值>0 大于 <0 小于 ==0
    等于）
```

```
int ret= my_strcmp(buf2,buf);

if(ret == 0)

{

    printf("%s 等于%s\n", buf2,buf);

}

else if(ret > 0)

{

    printf("%s 大于%s\n", buf2,buf);

}

else if(ret < 0)

{

    printf("%s 小于%s\n", buf2,buf);

}

}

int main(int argc,char *argv[])

{

test01();

return 0;

}
```

6、二维字符数组作为函数的形参 函数内部 可以通过形参 操作外部二维字符数组的元素。

```
void input_two_char_array(char buf[5][128], int row, int col)

{

printf("请输入%d 个字符串\n", row);

int i=0;

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

{

//scanf("%s", buf[i]);

fgets(buf[i], col, stdin);

}

}

void print_two_char_array(char buf[5][128], int row)

{

int i=0;

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

{

printf("%s\n", buf[i]);

}

}

void test02()

{
```

```

char buf[5][128]={""};

int row = sizeof(buf)/sizeof(buf[0]);

int col = sizeof(buf[0])/sizeof(buf[0][0]);

//获取键盘输入

input_two_char_array(buf, row, col);

//遍历

print_two_char_array(buf, row);

}

```

知识点 3 【函数的调用流程】

