

知识点1【函数的概述】（了解）

1、函数的概述

函数：为同一功能服务器的代码 封装成一个整体，可以多次调用。（一份代码、多次调用）

需要传哪些数据给函数？（实参）

函数的功能代码（函数体）如何实现？

函数需要返回啥类型的数据？（函数的返回值类型）

2、函数的分类（定义角度）

库函数（c语言库）、**自定义函数**、系统调用（内核提供给用户的函数接口）

3、函数的分类（有无参数）

无参的函数：不能将函数外部的数据 传递 给函数内部

有参的函数：通过参数 将函数外部的数据 传递到函数内部

（**参数**：函数**外部**数据 到 函数**内部**的桥梁）

知识点2【函数定义、声明、调用】（重要）

函数的**定义**、函数的**声明**、函数的**调用**。

1、函数定义：实现**函数体**，确定函数名，函数的形参、函数的返回值类型

```
1 //函数的定义 定义处x y叫形参 （函数定义的时候 形参不会开辟空间）
2 //只有当函数调用的时候 才会为形参 开辟空间 保存实参的值
3 int my_add(int x, int y) //x = data1, y=data2
4 {
5     //return 返回函数运算结果 结束当前函数
6     return x + y;
7 }
```

2、函数声明（先调用 后定义）

告知编译器 该函数的函数名是啥 有几个形参 返回值类型是啥

```
1 //函数声明 告知编译器 如果遇到函数名为my_add，有两个int形参，以及一个int返回值类型
2 //请通过编译
3 //int my_add(int, int);//ok 不推荐
4 int my_add(int x, int y);
```

3、函数的调用：执行**函数体**。函数名+（实参）；

```

//函数的定义 定义处x y叫形参
int my_add(int x, int y) //x = data1, y=data2
{
    //return 返回函数运算结果 结束当前函数
    return x + y;
}

void test02()
{
    int data1 = 10;
    int data2 = 20;

    //函数的调用 data1,data2实参
    int ret = 0;
    ret = my_add(data1, data2);

    printf("结果为:%d\n", ret);
}

```

```

edu@edu: ~/work/c/day04
edu@edu: ~/work/c/day04$ ./a.out
结果为:30
edu@edu: ~/work/c/day04$

```

完整代码:

```

1 //函数声明 告知编译器 如果遇到函数名为my_add, 有两个int形参, 以及一个int返回值
  类型
2 //请通过编译
3 //int my_add(int, int);//ok 不推荐
4 int my_add(int x, int y);
5
6 void test02()
7 {
8     int data1 = 10;
9     int data2 = 20;
10
11     //函数的调用 data1,data2实参
12     int ret = 0;
13     ret = my_add(data1, data2);
14
15     printf("结果为:%d\n", ret);
16 }
17
18 //函数的定义 定义处x y叫形参
19 int my_add(int x, int y) //x = data1, y=data2
20 {
21     //return 返回函数运算结果 结束当前函数
22     return x + y;
23 }

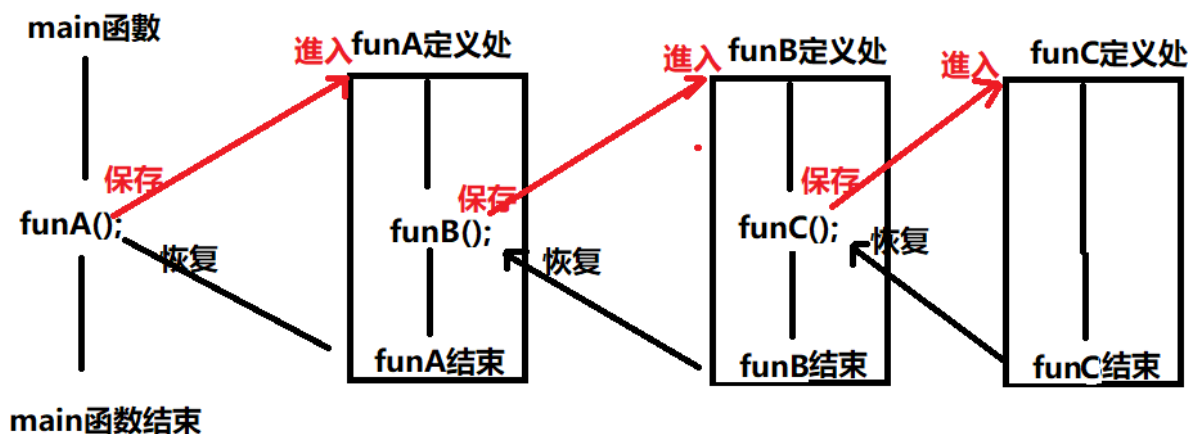
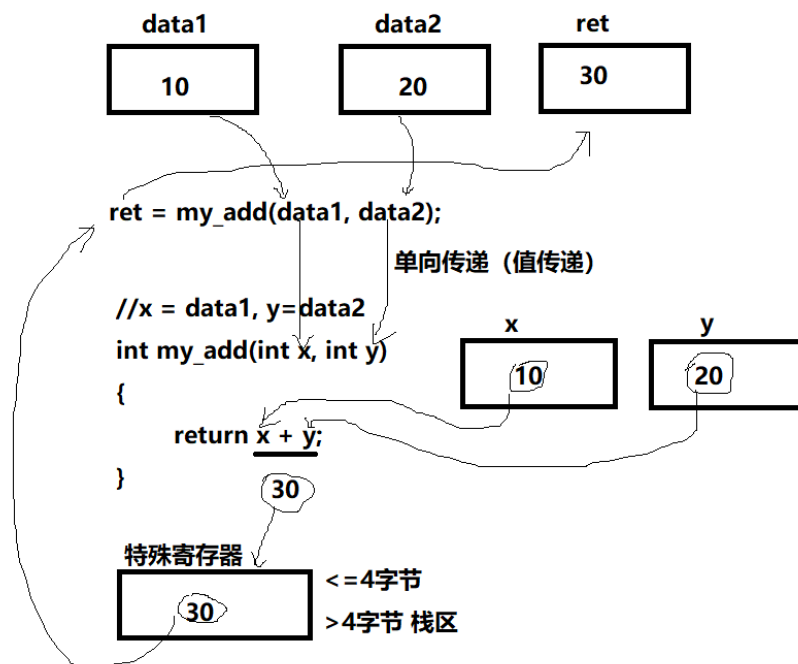
```

```
edu@edu:~/work/c/day04$ sudo gcc 00_code.c
edu@edu:~/work/c/day04$ ./a.out
结果为:30
edu@edu:~/work/c/day04$
```

知识点3 【函数调用的过程】（重要）

```
void test02()
{
    int data1 = 10;
    int data2 = 20;

    //函数的调用 data1,data2 实参
    int ret = 0;
    ret = my_add(data1, data2);
    printf("结果为:%d\n", ret);
}
```



知识点4 【函数传参】（重要）

1、普通变量作为函数的参数。

函数内部 需要使用外部变量的值 需要将外部变量的 值传递 给函数内部。这时 普通变量就要作为函数的 形参 （单向传递之值传递）

```

void set_data(int num)
{
    (int)100
    num = 100;
}

void test03()
{
    int data = 0;

    set_data(data);

    printf("data = %d\n", data);
}

```

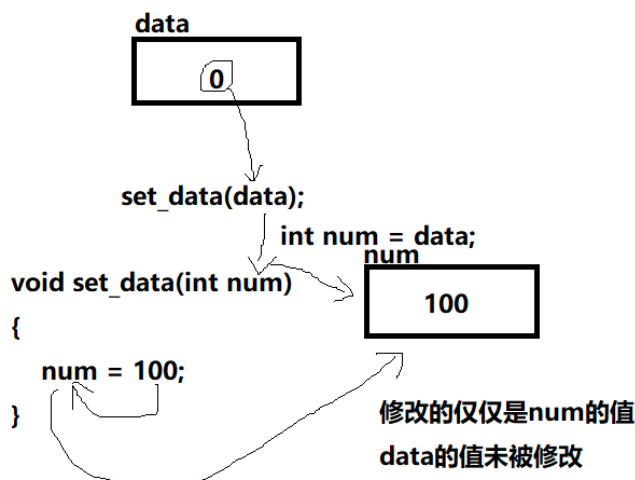
设置值失败

edu@edu: ~/work/c/day04

```

edu@edu: ~/work/c/day04$ sudo gcc 00_co
[sudo] edu 的密码:
edu@edu: ~/work/c/day04$ ./a.out
data = 0
edu@edu: ~/work/c/day04$

```



```

void test03()
{
    int data = 0;

    set_data(data);

    printf("data = %d\n", data);
}

```

2、数组作为函数的参数。

1、数值数组作为函数的参数

函数内部 可以 操作（读写） 外部数组的元素。

```

1 void inputIntArray(int arr[], int n);
2 void printfIntArray(int arr[], int n);
3 void sortIntArray(int arr[], int n);
4 void test04()
5 {
6     int arr[5] = {0};
7     int n = sizeof(arr) / sizeof(arr[0]);
8
9     //给数组获取键盘输入
10    inputIntArray(arr, n);
11
12    //对数组进行排序

```

```
13     sortIntArray(arr, n);
14
15     //遍历排序后的数组元素
16     printfIntArray(arr, n);
17
18     return;
19 }
20 int main(int argc, char const *argv[])
21 {
22     test04();
23     return 0;
24 }
25
26 void sortIntArray(int arr[], int n)
27 {
28     int i = 0;
29     for (i = 0; i < n - 1; i++)
30     {
31         int min = i;
32         int j = min + 1;
33
34         for (; j < n; j++)
35         {
36             if (arr[min] > arr[j])
37                 min = j;
38         }
39
40         if (i != min)
41         {
42             int tmp = arr[i];
43             arr[i] = arr[min];
44             arr[min] = tmp;
45         }
46     }
47
48     return;
49 }
50 void inputIntArray(int arr[], int n)
51 {
52     printf("请输入%d个int元素\n", n);
53 }
```

```

54     int i = 0;
55     for (i = 0; i < n; i++)
56     {
57         scanf("%d", &arr[i]);
58     }
59
60     return; //结束标记
61 }
62 void printfIntArray(int arr[], int n)
63 {
64     int i = 0;
65     for (i = 0; i < n; i++)
66     {
67         printf("%d ", arr[i]);
68     }
69     printf("\n");
70
71     return;
72 }

```

```

edu@edu: ~/work/c/day04$ sudo gcc 00_code.c
edu@edu: ~/work/c/day04$ ./a.out
请输入5个int元素
3 4 2 1 5
1 2 3 4 5
edu@edu: ~/work/c/day04$

```

2、字符数组作为函数的参数

```


1  #include <string.h>
2  void my_get_string(char str[], unsigned long n);
3  int my_strlen(char str[]);
4  void test05()
5  {
6      char str[128] = "";
7
8      //键盘获取字符串
9      my_get_string(str, sizeof(str));
10
11     //定义函数my_strlen测量字符串的参数

```

```

12     printf("字符串的长度为%d\n", my_strlen(str));
13 }
14 int main(int argc, char const *argv[])
15 {
16     test05();
17     return 0;
18 }
19 int my_strlen(char str[])
20 {
21     int i = 0;
22     while (str[i] != '\0')
23     {
24         i++;
25     }
26
27     return i;
28 }
29 void my_get_string(char str[], unsigned long n)
30 {
31     printf("请输入一个字符串:");
32     fgets(str, n, stdin);
33     str[strlen(str) - 1] = 0; //去掉回车
34
35     return;
36 }

```

 edu@edu: ~/work/c/day04

```

edu@edu:~/work/c/day04$ sudo gcc 00_code.c
edu@edu:~/work/c/day04$ ./a.out
请输入一个字符串:hello world
字符串的长度为11
edu@edu:~/work/c/day04$ _

```