

函数【C 语言的第二个重点】

为什么需要函数

避免了重复性操作;
有利于程序的模块化。

什么叫做函数

逻辑上: 能够完成特定功能的独立的代码块;

物理上: 【详情看函数举例_2.cpp】

能够接收数据; [当然也可以不接受数据]

能够对接收的数据进行处理;

能够将数据处理的结果返回。[当然也可以不返回任何值]

总结: 函数是一个工具, 他是为了解决大量类似的问题而设计的;
函数可以当作一个黑匣子。

如何定义函数

函数的返回值 函数的名字(函数的形参列表)

```
{  
    函数的执行体;  
}
```

①、函数定义的本质是详细描述函数之所以能够实现某个特定功能的具体方法。

②、return 表达式; 的含义

1>. 终止被调函数, 向主调函数返回表达式的值

2>. 如果表达式为空则之终止函数, 不向被调函数返回任何值

3>. break 是用来终止循环的, return 是用来终止函数的

例子:

```
void f()  
{  
    return; //return 只用来终止函数, 不向被调函数返回任何值。  
}  
  
int f()  
{  
    return 10; //1、终止函数; 2、向主调函数返回 10  
}
```

③、函数返回值的类型也称为函数的类型, 因为如果函数名前的返回值类型和函数执行体中的 return 表达式; 中表达式的类型不同的话, 则最终函数返回值的类型以函数名前的返回值类型为准。

void 也是一个返回值, 只不过返回值为空。

```

#include <stdio.h>

int f()
{
    return 10; // 因为函数的返回值类型是int，所以最后f返回的是10而不是10.5
}

int main(void)
{
    int i = 99;
    double x = 6.6;
    x = f();
    printf("%lf\n", x);
    return 0;
}

/*
在dev c++ 中的输出结果是:
-----
10.000000
-----
*/

```

函数的分类

有参函数和无参函数

有返回值函数和无返回值函数

库定义函数和用户自定义函数

值传递函数和地址传递函数

普通函数和主函数(main)

一个程序有且只有一个主函数

主函数可以调用普通函数，普通函数不能调用主函数

普通函数可以相互调用

主函数是程序的入口，也是程序的出口

注意的问题

函数调用和函数定义的顺序

如果函数调用写在了函数定义的前面，则必须加函数前置声明。

函数前置声明： 函数类型 函数名(形参)；

- 1、告诉编译器即将可能出现的若干个字母代表的是一个函数；
- 2、告诉编译器可能出现的字母所代表的函数的形参和返回值的具体情况；
- 3、函数声明是一个语句，末尾必须加分号；
- 4、对库函数的声明通过 #include<库函数所在的文件的名字.h>来实现。

形参和实参

个数必须相同 位置一一对应 数据类型必须相互兼容

C语言函数的参数会出现在两个地方，分别是函数定义处和函数调用处，这两个地方的参数是有区别的。

形参(形式参数):在函数定义中出现的参数可以看做是一个占位符，它没有数据，只能等到函数被调用时接收传递进来的数据，所以称为**形式参数**，简称**形参**。

实参(实际参数):函数被调用时给出的参数包含了实实在在的数据，会被函数内部的代码使用，所以称为**实际参数**，简称**实参**。

形参和实参的功能是传递数据，发生函数调用时，实参的值会传递给形参。

如何在软件开发中合理的设计函数来解决实际问题

一个函数的功能尽量独立，单一；

多写写，多模仿牛人的代码；

函数是 C 语言的基本单位，类是 Java、C#、C++的基本单位。

常用的系统函数

`double sqrt(double x);` ——求 x 的平方根。

`int abs(int x);` ——求 x 的绝对值。

`double fabs(double x);` ——求 x 的绝对值。

专题：

递归【数据结构[弄懂栈]】

变量的作用域和存储方式：

按照作用域分：

全局变量

在所有函数外部定义的变量叫全局变量。

全局变量使用范围：从定义位置开始到整个程序结束。

局部变量

在一个函数内部定义的变量或者是函数的形参，都统称为局部变量。

```
#include<stdio.h>
void f(int i)
{
    int j = 10;
}
int main (void)
{
    int i = 10;

    return 0;
}
```

//i 和 j 都属于局部变量。

局部变量使用范围：只能在本函数内使用。

注意的问题：

全局变量和局部变量命名相冲突的问题：

在一个函数内部，如果定义的局部变量名和全局变量名一样时，局部变量会屏蔽掉全局变量。

按变量的存储方式:

静态标量

自动变量

寄存器[cpu 内部可以用来存东西的]变量