# 一、<>与“”区别

1、<> 搜索 系统 指定的头文件库。

2、“” 优先检查 当前目录 ，找不到检查系统指定的头文件库。

# 保存字符串

## 方法一

定义变量

Char \*info;

赋值

Info = “sssss”;

## 方法二

声明的同时赋值

Char info[10] = “12345676”;

## 方法三

声明

Char info[10];

赋值

Info[0] = ‘1’;

Info[1] = ‘d’;

Info[2] = ‘\0’;

## 方法四

声明

Char a[10];

赋值

strcpy(a, "hello");

# 三、指向函数的指针

1、int \*p()是函数

返回值是指针

2、int (\*p)()是指针

指向一个函数

# 指针

定义

Int \*a;

//a是一个 **指针** 类型的变量，只能用于存放 地址。

//同时\*a也指向内存中的一块区域，并指代这块区域中存放的值。

当定义a为一个指针后：

变量a中存放的就是 地址（0x1074836604）

\*a 是获取变量a 中存放的地址 所指向的值。

变量a 主要用于 指针运算，例如 a++，a--等

变量a 本质上代表一个存储单元。CPU通过该存储单元的地址访问该存储单元中的数据。所以a本来代表两个值：存储单元的地址和储单元中的数据。于是就有了二异性。为了消除这种二义性，C语言规定a表示存储单元中的数据，&a表示存储单元的地址。  
a存储单元中的数据可以是一个普通数值，也可以是另一个存储单元的地址，比如：a = &b; 语句就是将b的存储单元的地址存入a存储单元中。C语言规定\*a代表a中存储的地址对应的存储单元中的数据，也就是访问\*a就等于访问b，于是**\*a提供了通过a访问b中的数据的手段**。  
  
**a表示a对应的存储单元中的数据。  
&a表示a对应的存储单元的地址。  
\*a表示：首先，要求a对应的存储单元中的数据一定是另一个存储单元的地址。于是，\*a表示另一个存储单元中的数据。**

当a声明的类型是int时，a中存储的是一个整数数值，通过a可以访问（读取或修改）这个数值。  
 当a声明的类型是int\*时，a（不是\*a）中存储的是一个存储单元的地址，而该存储单元中存储的数据是一个整数数值；通过\*a可以访问（读取或修改）这个数值。a == &\*a 都是该存储单元的地址。  
 当a声明的类型是int\*\*时，a中存储的是一个存储单元的地址，而该存储单元中存储的数据是另外一个存储单元的地址，另外这个存储单元中存储的是一个整数数值；通过\*\*a可以访问（读取或修改）这个数值。

# 五、定义

Int Char 等空间大小是不同的；

# 六、结构体

1、只有结构体定义：

struct stuff{ //定义结构体的名称（stuff）

char job[20];

int age;

};

2、附加该结构体类型的“结构体变量”的初始化的结构体定义：

//直接带变量名Huqinwei

struct stuff{  // 定义结构体的名称（stuff）

        char job[20];

        int age;

        float height;

}Huqinwei;   //使用stuff结构体定义 Huqinwei

---------------------------------------------------------------------

//以上相当于

struct stuff{

        char job[20];

        int age;

        float height;

};

struct stuff Huqinwei;

---------------------------------------------------------------------------------------------