转载：http://blog.csdn.net/xiaoyali/article/details/4393486/

关闭

[**xiaoyali的专栏**](http://blog.csdn.net/xiaoyali)

* [http://static.blog.csdn.net/images/ico_list.gif目录视图](http://blog.csdn.net/xiaoyali?viewmode=contents)
* [http://static.blog.csdn.net/images/ico_summary.gif摘要视图](http://blog.csdn.net/xiaoyali?viewmode=list)
* [http://static.blog.csdn.net/images/ico_rss.gif订阅](http://blog.csdn.net/xiaoyali/rss/list)

[**异步赠书：Kotlin领衔10本好书**](http://blog.csdn.net/epubit17/article/details/77770966)      [**SDCC 2017之区块链技术实战线上峰会**](http://edu.csdn.net/huiyiCourse/series_detail/66)      [**程序员9月书讯**](http://blog.csdn.net/turingbooks/article/details/78017356)      [**每周荐书：Java Web、Python极客编程（评论送书）**](http://blog.csdn.net/broadview2006/article/details/77878868)

[结构体的定义以及使用](http://blog.csdn.net/xiaoyali/article/details/4393486)

标签： [struct](http://www.csdn.net/tag/struct" \t "_blank)[存储](http://www.csdn.net/tag/%e5%ad%98%e5%82%a8)[date](http://www.csdn.net/tag/date)[语言](http://www.csdn.net/tag/%e8%af%ad%e8%a8%80)[c](http://www.csdn.net/tag/c)

2009-07-30 11:56 101579人阅读 [评论](http://blog.csdn.net/xiaoyali/article/details/4393486/#comments)(8) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/xiaoyali/article/details/4393486/#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

按章总结--------2009.7 ---C谭浩强（5） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

**结构的定义**

定义一个结构的一般形式为：

**struct 结构名**

**{**

**成员表列**

**}**

成员表由若干个成员组成，每个成员都是该结构的一个组成部分。

对每个成员也必须作类型说明。

例如：

struct stu

{

int num;

char name[20];

int age;

}

**结构类型变量的说明**

结构体定义并不是定义一个变量，而是定义了一种**数据类型**，这种类型是你定义的，它可以和语言本身所自有的简单数据类型一样使用（如 int ）。

结构体本身并不会被作为数据而开辟内存，真正作为数据而在内存中存储的是这种结构体所定义的**变量**。

一个结构体变量占多大内存空间呢?这是由这种结构体类型的定义决定的，可以想象为了同时存储结构体的每一个成员，那么结构体变量的存储大小，就应该是所有分量的存储空间的总合。  
说明结构变量有以下三种方法。以上面定义的stu为例来加以说明。

1. 先定义结构，再说明结构变量。 如：

struct stu

{

int num;

char name[20];

int age;

};

struct stu boy1,boy2;

说明了两个变量boy1和boy2为stu结构类型。

也可以用宏定义使一个符号常量来表示一个结构类型， 例如：

#define STU struct stu

STU

{

int num;

char name[20];

int age;

};

STU boy1,boy2;

1. 在定义结构类型的同时说明结构变量。例如：

struct stu

{

int num;

char name[20];

int age;

}boy1,boy2;

1. 直接说明结构变量。

例如：

struct

{

int num;

char name[20];

int age;

}boy1,boy2;

第三种方法与第二种方法的区别在于第三种方法中省去了结构名，而直接给出结构变量。

说明了boy1,boy2变量为stu类型后，即可向这两个变量中的各个成员赋值。

在上述stu结构定义中，所有的成员都是基本数据类型或数组类型。成员也可以又是一个结构， 即构成了嵌套的结构。

例如：

struct date{

int month;

int day;

int year;

}

struct{

int num;

char name[20];

struct date birthday;

}boy1,boy2;

**使用结构变量成员的一般形式是：**

**结构变量名.成员名**

例如：

boy1.num

/\*即第一个人的学号　\*/

如果成员本身又是一个结构则必须逐级找到最低级的成员才能使用。

例如：boy1.birthday.month

即第一个人出生的月份成员可以在程序中单独使用，与普通变量完全相同。

**结构变量的初始化： 类似多维数组的初始化。**

**结构变量的赋值**

你可对结构体变量的成员逐个赋值，这种方式和数组相像，数组就是逐个元素逐一赋值（对于数组，除非初始化，否则你别无选择，只能这样赋值）。**不同于数组，标准C中结构体变量可以整体赋值**。

例一：

#include <stdio.h>

int main(void)

{

**struct student**

**{**

        long int num;

        int      age;

        char\*    name;

    }st1={200,18,"zhangsan"};

    struct student  st2,st3;

    printf("    NO.    age    name   /n");

    printf("   %ld    %d   %s/n",st1.num,st1.age,st1.name);

    getch();

    st2.num=199;

    st2.age=19;

    st2.name="lisi";

    printf("   %ld    %d   %s/n",st2.num,st2.age,st2.name);

    getch();

    st3=st2;

    printf("   %ld    %d   %s/n",st3.num,st3.age,st3.name);

    getch();

    printf("/n/n   struct student:%d",sizeof(struct student));

    getch();

    return 0;

}

例二：

#include <stdio.h>

#include <conio.h>

struct birth

    {

        int year;

        int month;

        int day;

    };

struct student

    {

        long int num;

        struct  birth  birthday;

        char\*    name;

    }st1={200,{1988,8,8},"zhangsan"};

int main(void)

{

    struct student  st2;

    st2=st1;

    printf("   %ld     %s  %d/n",st2.num,st2.name,sizeof(int));

    printf("   year: %d   month: %d   month: %d/n",

                st2.birthday.year,

                st2.birthday.month,

                st2.birthday.day);

    getch();

    return 0;

}

**注意**

当你的结构体变量中有存放多个字符的成员，建议你将它定义为数组（比如前面的姓名成员，**不知道数组该开辟多大时，也可以定义为指针**）。原因是指针变量不能存放实际数据，仅仅是地址。