2.1 生活中的数据_物联网 / 嵌入式工程师 - 慕课网

> 慕课网慕课教程 2.1 生活中的数据涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

我们在现实生活中，在对具体的对象进行描述的时候，发现对象是比较复杂的。一般是由不同的类型组合在一起的。例如：我们描述一个人的时候，习惯性会描述他的姓名，年龄，分数等。这些不同类型的数据是互相联系组成了一个有机的整体。此时，就要用到一种新的构造类型数据——结构体（struct）。

现实生活中，我们定义一个职工worker结构体，在这个结构体中包括职工编号、姓名、性别、年龄、工资、家庭住址、联系电话。这样就可以用一个结构体数据类型的变量来存放某个职工的所有相关信息。并且，用户自定义的数据类型worker也可以与int、double等基本数据类型一样，用来作为定义其他变量的数据类型

```
char   name[20];
int    age;
double offer;
int    id;
```

方法 1：先声明，在定义。

```
(1)结构体类型的声明
struct 结构体名
{
        数据类型   变量名1;
        数据类型   变量名2;
        数据类型   变量名3;
        ...
};

(2)结构体变量的定义

变量定义：struct   结构体名   变量名;

访问结构体对象内部成员变量的方法：
 A.结构体普通变量通过"."来访问内存的成员属性。
 B.结构体指针变量通过"->"来访问内存的成员属性。

 例如：



 struct worker
 {
        char name[20];
        int  age;
        double offer;
        int id;
};

struct worker wk;
struct worker *p = &wk;

wk.name                 ===>char [20]; 类型
wk.id                   ===>int类型
wk.offer                ===>double类型

p->name                 ===>char [20]; 类型
p->id                   ===>int类型
p->offer                ===>double类型
```

示例代码：

```
include <stdio.h>
#include <string.h>


struct student{
        char name[20];
        int id;
        int score;
};


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



        struct student st;
        struct student *sp = &st;



        strcpy(st.name,"jack");
        (&st)->id = 1;
        st.score = 100;

        printf("NAME\tID\tSCORE\n");
        printf("%s\t%d\t%d\n",st.name,st.id,st.score);

        return 0;
}
```

运行结果：

```
AME        ID        SCORE
jack        1        100
```

方法 2：在声明类型的同时定义变量

```
struct 结构体名
{
        数据类型  变量名1;
        数据类型  变量名2;
        数据类型  变量名3;
        ...
}变量名列表;
==============================
例如：
struct student
{
        char name[20];
        int id;
        int score;
}st,*sp;



struct student s3;

s3.name
st.id
sp->score
```

示例代码

```
#include <stdio.h>
#include <string.h>

struct student{
        char name[20];
        int id;
        int score;
}st;



int main()
{
```

```
        struct student *sp = &st;

        strcpy(sp->name,"jack");
        sp->id = 1;
        sp->score = 80;

        printf("NAME\tID\tSCORE\n");
        printf("%s\t%d\t%d\n",st.name,st.id,st.score);

        return 0;
}
```

运行结果

```
NAME        ID        SCORE
jack        1        80
```

方法 3：省略结构体名，直接定义变量。

```
struct
{
        数据类型  变量名1;
        数据类型  变量名2;
        数据类型  变量名3;
        ...
}变量名1，变量2,变量3...;

注：此种方法，只能在变量名列表的位置定义变量，其他位置不能定义变量。
================================
例如：
struct
{
        char name[20];
        int id;
        int score;
}st,*sp;

struct {
        int m;
        int n;

}s2;
```

示例代码

```
#include <stdio.h>
#include <string.h>

struct
{
    char name[20];
    int id;
    int score;
}st1 = {"rose",2,100};


int main()
{
        printf("NAME\tID\tSCORE\n");
        printf("%s\t%d\t%d\n",st1.name,st1.id,st1.score);

        return 0;
}
```

运行结果

```
NAME        ID        SCORE
rose        2        100
```

```
struct student{
        char name[20];
        int id;
        int score;
}st1 = {"jack",1,100};
```

```
int main()
{


        struct student st[3] = {{"rose",2,70},
                                {"lilei",3,60},
                                {"hmm",4,50}};
        int id;









}
```

全文完