<https://www.cnblogs.com/Griouges/p/8021708.html>

**什么是结构体？**

在C语言中，结构体(struct)指的是一种数据结构，是C语言中聚合数据类型(aggregate data type)的一类。结构体可以被声明为变量、指针或数组等，用以实现较复杂的数据结构。结构体同时也是一些元素的集合，这些元素称为结构体的成员(member)，且这些成员可以为不同的类型，成员一般用名字访问。

**结构体的定义：**

C语言结构体类型的定义模板大概为：

struct 类型名{

成员表列

} 变量;

* 在成员表列中可以是几种基本数据类型，也可以是结构体类型。
* struct 类型名{} 变量;后的分号不能漏

*下面给出定义结构体类型的几种方法*

**1. 先定义结构体类型，再定义结构体变量。**

struct student{

char no[20]; //学号

char name[20]; //姓名

char sex[5]; //性别

int age; //年龄

};

struct student stu1,stu2;

//此时stu1,stu2为student结构体变量

**2. 定义结构体类型的同时定义结构体变量。**

struct student{

char no[20]; //学号

char name[20]; //姓名

char sex[5]; //性别

int age; //年龄

} stu1,stu2;

此时还可以继续定义student结构体变量如：  
struct student stu3;

**3.直接定义结构体变量。**

struct{

char no[20]; //学号

char name[20]; //姓名

char sex[5]; //性别

int age; //年龄

} stu1,stu2;

一般不会使用第三种定义方法，因为直接定义结构体变量stu1,stu2后就不能再继续定义该类型的变量。

**注意：**

1. 在C语言中使用struct定义结构体类型后定义结构体变量时struct不能省略，在C++中允许省略struct。
2. 在c中:
3. struct student{
4. ...
5. };

struct student stu1; //struct不可省略

在c++中：

struct student{

...

};

student stu1; //struct可省略

1. 在C中定义结构体类型后每次定义变量时都要使用struct，如果嫌麻烦，我们可以这样：
2. typedef struct student{
3. ...
4. }STUDENT;

STUDENT stu1;

使用typedef给struct student取一个"别名"STUDENT

1. 在某些情况下还可以使用#define来实现更简化的结构体定义与变量的定义，但可能会牺牲部分可读性。
2. #define STUDENT struct student;
3. STUDENT{
4. ...
5. };

STUDENT stu1;

typedef和#define用法不同，甚至可以结合起来灵活使用，使用时一定要注意两者的不同之处。