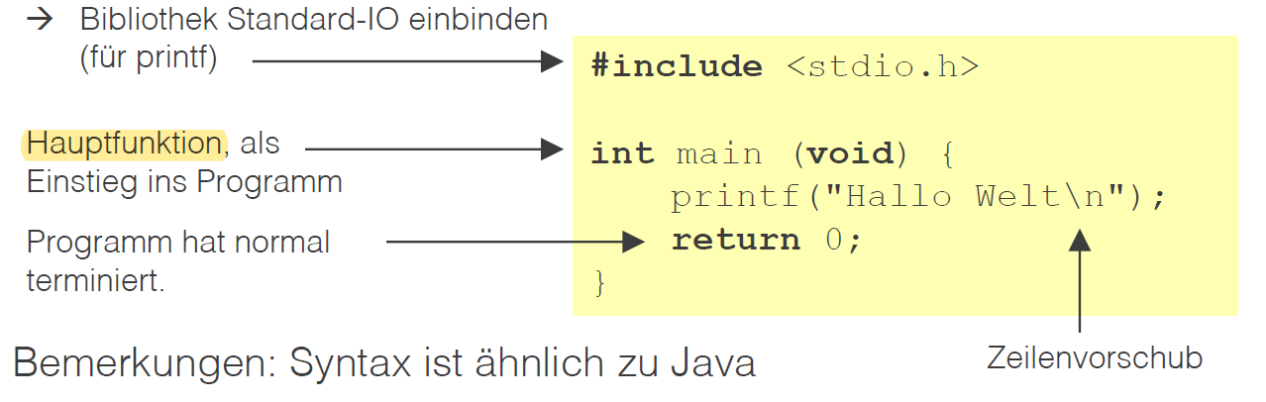
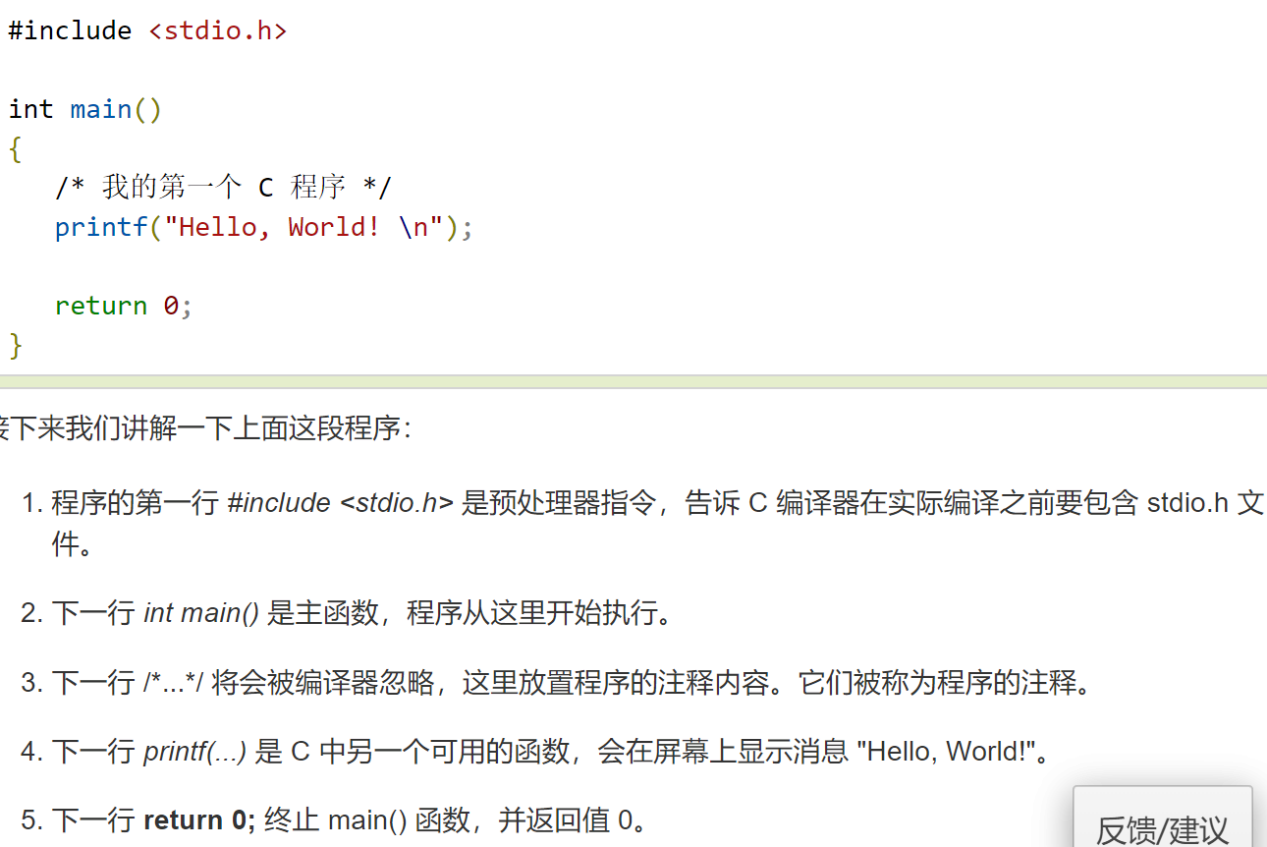
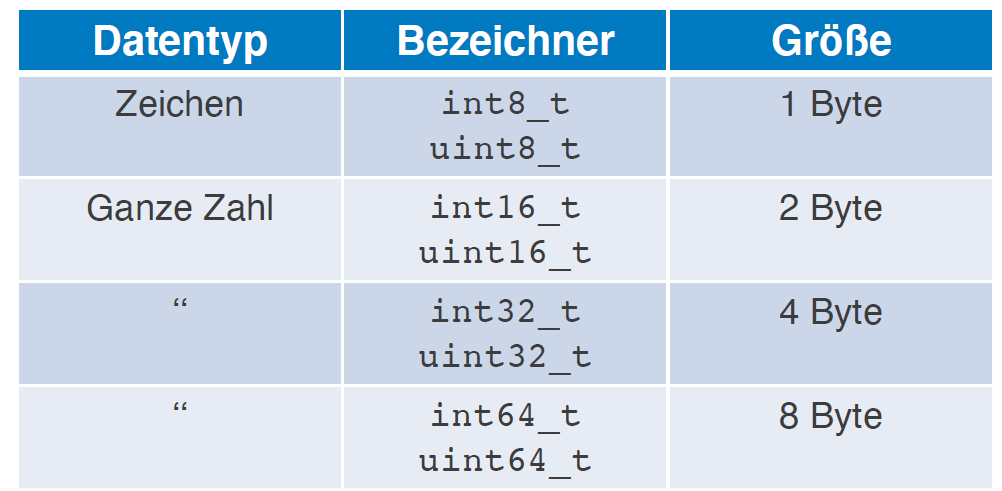
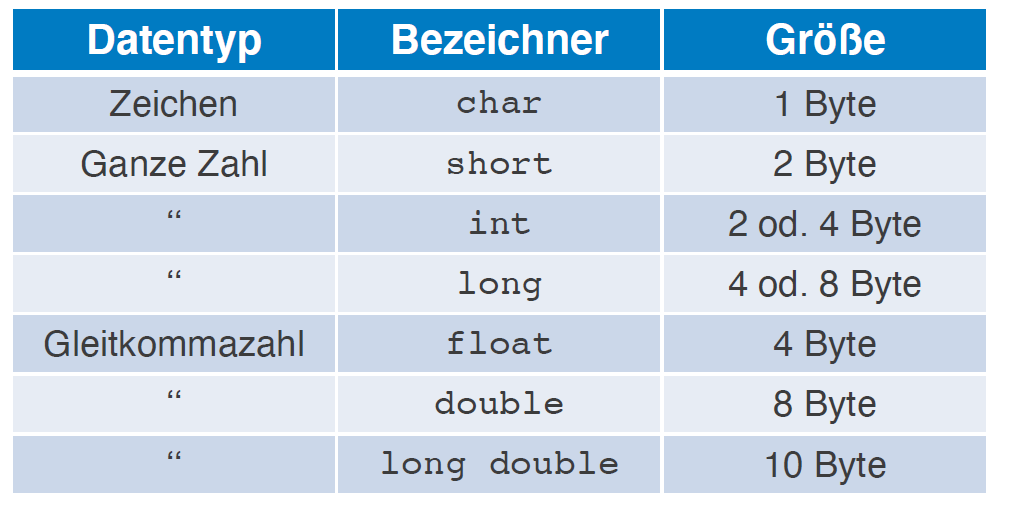
1.



2.数据类型大小



Sizeof用来获得确切大小

printf("int 存储大小 : %lu \n", sizeof(int));

**%lu** 为 32 位无符号整数，详细说明查看 [C 库函数 - printf()](https://www.runoob.com/cprogramming/c-function-printf.html)。

当您在 Linux 上编译并执行上面的程序时，它会产生下列结果：

int 存储大小 : 4

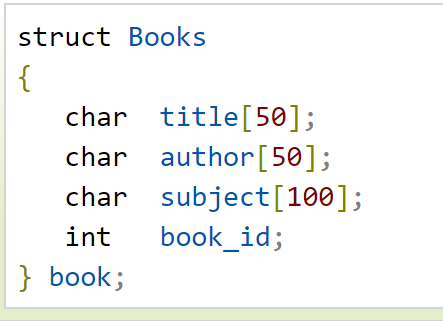
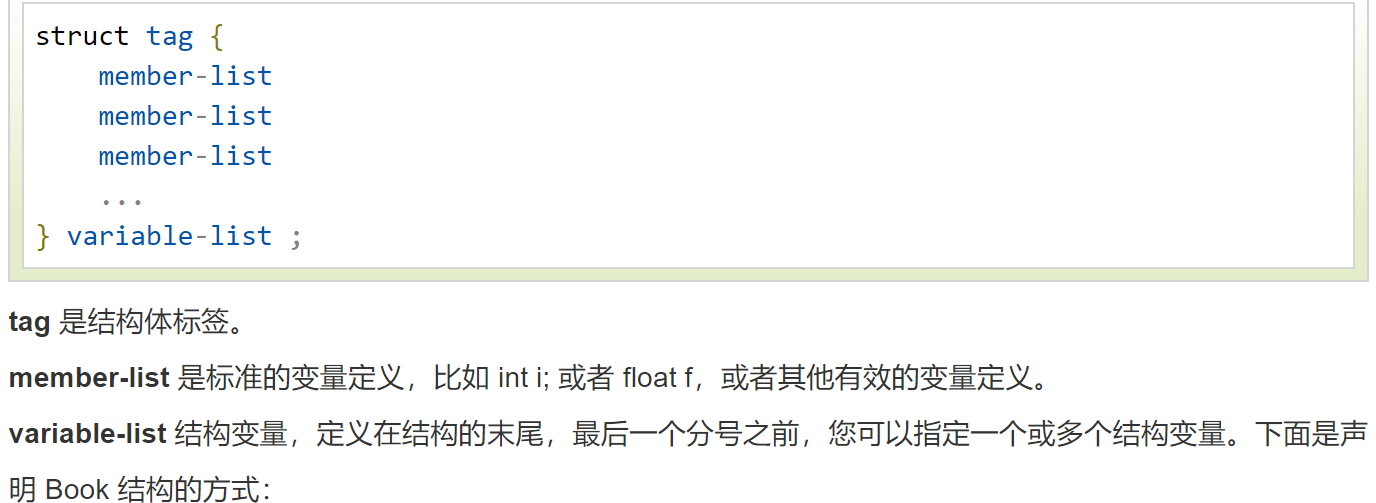
3.补:



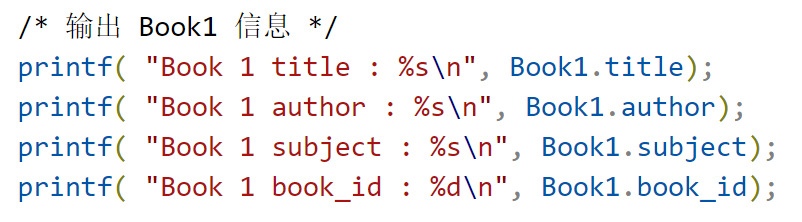
多个:

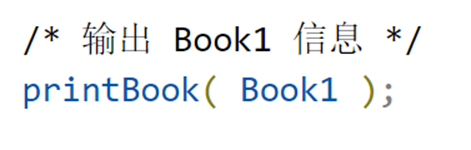


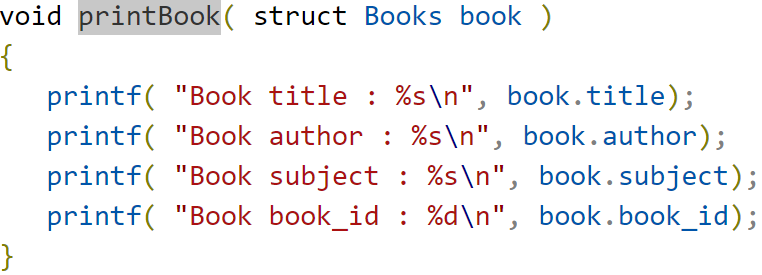
**4.结构体struct**

****

**访问结构成员**

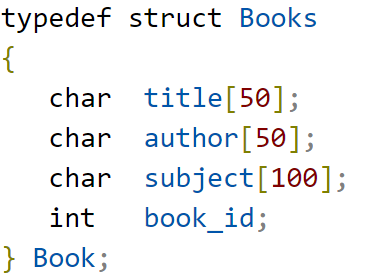
****



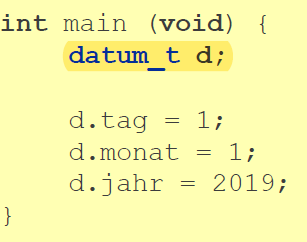


**5.typedef 为类型取一个新的名字**

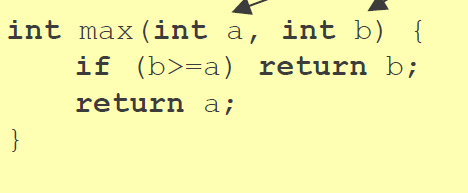
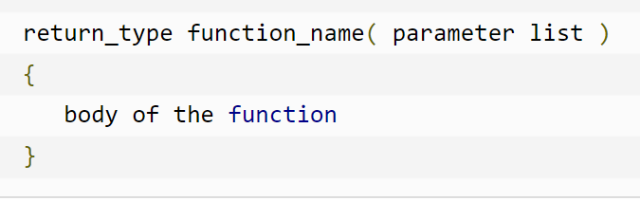
**1.将struct 改名为 Books 2.将struct datum 改名为datum\_t**

****

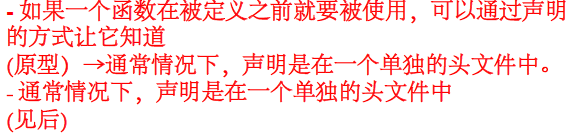
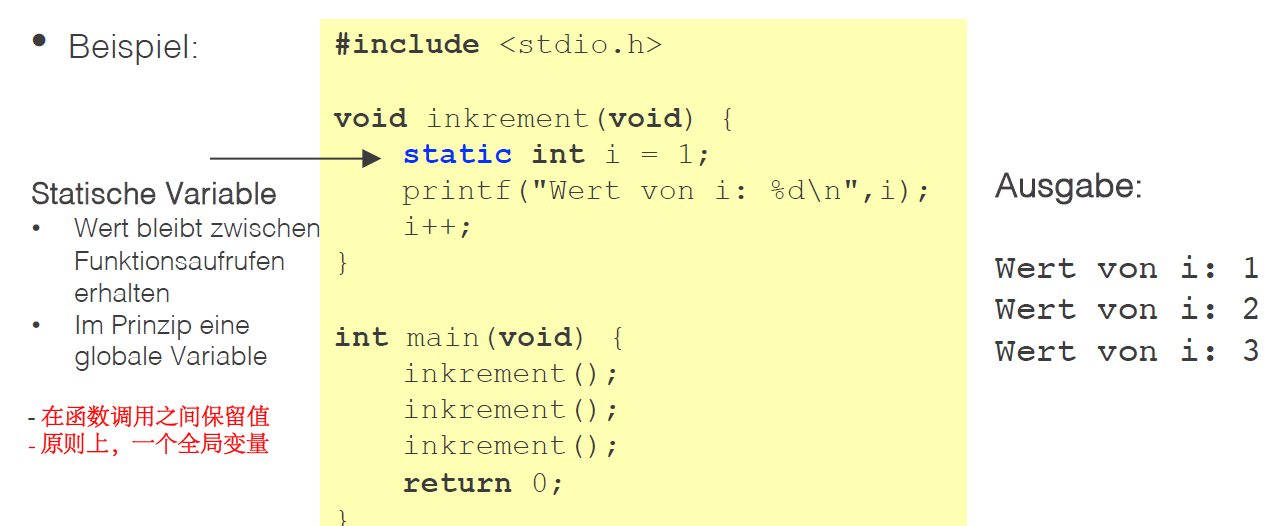
**声明**

****

**6.函数**

****

**特别的static:**

****

2.6 Präprozessor预处理

**#include** < Dateiname >

#define PI 3.1415926f

2.7 Module

% cc -c f2.c f3.c (erzeugt f2.o und f3.o )

% cc -o prog prog.o f1.o f2.o f3.o

a.头文件是扩展名为 **.h** 的文件，包含了 C 函数声明和宏定义，被多个源文件中引用共享。有两种类型的头文件：程序员编写的头文件和编译器自带的头文件。

在程序中要使用头文件，需要使用 C 预处理指令 **#include** 来引用它。前面我们已经看过 **stdio.h** 头文件，它是编译器自带的头文件。

**b..o** 在将每个源文件链接到最终可执行文件之前，编译器会为每个源文件创建一个目标文件

c.extern **int** global;

关键字extern，可以在一个文件中引用另一个文件中定义的变量或者函数

d.make <https://www.ruanyifeng.com/blog/2015/02/make.html>

2.8 Formatierte Eingabe

Scanf

scanf("%d%d%d",&a,&b,&c);

2.10 Zeiger