```c
//三：用typedef来定义类型
//int ,char,float,double, 结构体，共用体，枚举类型。
//我们可以用typedef来定义新的类型名(不是用来定义变量的) 来 代替已经有的类型名;
typedef int INTEGER; //我们用INTEGER代表了int，那我们再定义整型变量时我们就可以如下定义
INTEGER a,b,c; //定义三个整型变量  int a,b,c;
```

```c
//如何定义结构体类型
typedef struct date
{
    int month;
    int day;
    int year;
}DATE;
//struct date birthday;
DATE birthday; //
DATE *p; //p为指向此结构体类型数据的指针
```

```c
//说明：
//a)typedef中用的类型别名一般都大写；
//b)typedef用来定义类型名的，不是用来定义变量的；
//c)typedef只是对已经存在的类型增加一个类型名，并没有创造新类型；
//d)typedef 是编译的时候处理的；
//可执行文件 ：编译（预处理(#define,#include,#ifdef)，编译(typedef)，汇编），链接
//e)typedef主要作用：程序的通用性和可移植性。
```

```c
//变形，大家要记一记
typedef int NUM[100]; //定义NUM为整形数组类型  typedef int NUM[100];
NUM n; //等价于 int n[100];

typedef char *PSTRING; //定义PSTRING为字符指针类型 typedef char *PSTRING;
PSTRING p,q; //char *p,*q;

typedef int (*POINTER)(); //定义POINTER为指向函数的指针类型，该函数返回的是整型值
POINTER p1,p2;

//总结一下typedef这种语句怎么写，以定义一个整型数组为例：
//第一步：先写出常规的整型数组定义方法
//int n[100];
//第二步：将变量名n替换成自己想用的类型名
//int NUM[100];
//第三步：在前面加上typedef
//typedef int NUM[100];
NUM a,b,c,d;
```

给类型区别名后

例如 tyoedef int INTENT;

只用修改这一句中的INTENT就可以修改所有的

int类型