

typedef的四个用途和两大陷阱计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/645/2021\\_2022\\_typedef\\_E7\\_9A\\_84\\_c97\\_645923.htm](https://www.100test.com/kao_ti2020/645/2021_2022_typedef_E7_9A_84_c97_645923.htm)

用途一：定义一种类型的别名，而不只是简单的宏替换。可以用作同时声明指针型的多个对象。比如：`char* pa, pb.` // 这多数不符合我们的意图，它只声明了一个指向字符变量的指针，// 和一个字符变量；以下则可行：`typedef char* PCHAR.` // 一般用大写 PCHAR `pa, pb.` // 可行，同时声明了两个指向字符变量的指针 虽然：`char *pa, *pb.` 也可行，但相对来说没有用typedef的形式直观，尤其在需要大量指针的地方，typedef的方式更省事。用途二：用在旧的C的代码中（具体多旧没有查），帮助struct。以前的代码中，声明struct新对象时，必须要带上struct，即形式为：`struct 结构名 对象名`，如：`struct tagPOINT1 { int x. int y. }. struct tagPOINT1 p1.` 而在C中，则可以直接写：`结构名 对象名`，即：`tagPOINT1 p1.` 估计某人觉得经常多写一个struct太麻烦了，于是就发明了：`typedef struct tagPOINT { int x. int y. }POINT.`  
100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)