

# typedef关键字

---

## 简介

typedef工具是一个高级数据特性，利用typedef可以为某一类型自定义名称，这方面与#define有类似也有不同。

- typedef创建的符号名只受限于类型，不能用于值
- typedef由编辑器解释，而不是预处理器
- 在其受限范围，typedef比define更加灵活

## 用法

假设要用BYTE表示一个字节的数组。只需像定义个char类型变量一样定义BYTE即可。随后即可通过BYTE来定义变量。

```
typedef unsigned char BYTE;  
BYTE x, y[10], *z;
```

通常，typedef定义中用大写字母表示被定义的名称，以提醒用户注意这个类型名实际上是一个符号缩写。

## 使用场景

### 为经常出现的类型创建方便简洁的别名

用typedef来命名一个结构类型时，可以省略该结构的标签。

```
typedef struct {  
    double x;  
    double y;  
}rect,  
rect r1={3.0,6.0};  
rect r1;
```

以上代码将被翻译成为

```
struct {double x; double y;} r1 = {3.0,6.0};  
struct {double x; double y;} r2;  
r2 = r1;
```

这两个结构在声明时都没有标记，他们的成员完全相同，C认为这两个结构相同，所以r1和r2间的赋值是有效操作。

## typedef常用于给复杂的类型命名

下面的声明吧FRPTC声明为一个函数类型，该函数放回一个指针，该指针指向内含五个char类型元素的数组。

```
typedef char * (*FRPTC ()) [5];
```

## 总结

使用typedef是要注意，typedef并没有创建任何新的类型，他只是为某个已经存在的类型增加了一个方便使用的标签。通过结构、联合和typedef，C语言提供了有效处理数据的工具和可移植数据的工具。