

# 嵌入式C语言之- typedef的用法

讲师：叶大鹏

助力你成为优秀的电子工程师！



# typedef的用法

- typedef用来为已有的数据类型定义一个别名，语法格式为：

**typedef           数据类型           名字**

**typedef enum QualityLevel QualityLevel;**

- ✓ 好处是，便于程序的阅读和维护：

举个例子：一套程序，需要运行在C51单片机和ARM32位单片机的平台上，在代码里经常需要用到32位的整形数据类型，我们知道C51的int是16位，long是32位，ARM32的int是32位，为了保证程序的兼容性，可以这样实现：

```
#define __C51__
```

```
#ifdef __C51__
```

```
typedef signed long INT32;
```

```
#else
```

```
typedef signed int INT32;
```

```
#endif
```

# typedef和#define的区别

- #define 与typedef功能都是给一个对象取一个别名，增强程序的可读性，但它们在使用时有以下几点区别：

## 1. 应用场景不同：

- 1) typedef用来给一种数据类型定义别名；
- 2) #define用来给数字、表达式、代码语句定义别名；

## 2. 执行时机不同：

- 1) typedef在编译阶段执行；
- 2) #define在预编译阶段执行；

## 3. 定义方法不同：

- 1) #define别名在替换对象的前面，并且定义后面不用加分号；

```
#define PI 3.14159
```

- 2) typedef的别名在替换对象的后面，并且定义后面需要加分号；

```
typedef signed char int8_t;
```

# typedef的用法

```
typedef struct {  
    unsigned int  consoleID;  
    ...  
    char  shellBuf[SHOW_MAX_LEN];  
    char  shellWorkingDirectory[PATH_MAX];  
} ShellCB;
```

```
typedef enum {  
    CMD_TYPE_SHOW = 0,  
    CMD_TYPE_STD = 1,  
    CMD_TYPE_EX = 2,  
    CMD_TYPE_BUTT  
} CmdType;
```

# typedef和#define的区别

- typedef不同于#define，不是简单的字符串替换：

```
typedef int * PINT1;
```

```
#define PINT2 int *
```

```
// i1、i2 都为int *，typedef为int *引入了一个新的别名
```

```
PINT1 i1, i2;
```

```
// 相当于int * i3, i4; i3是int *, 而i4是int
```

```
PINT2 i3, i4;
```

**THANK YOU!**