

## 无值宏

在定义宏的时候不需要给定某一个值，对于无值来说只是用来做一个简单的判断（是否有定义）

```
1 #define    DE_BUG
```

## 条件编译

根据某一个条件来决定某一代码块是否需要编译。

语法：

形式1：

通过无值的宏来判断， 则只能判断是否有定义

```
1 #ifdef    // 判断某一个宏是否有定义
2
3         // 代码块
4
5 #endif    // 判断语句的结束
6
7 #ifdef DE_BUG // 如果定义了DE_BUG 宏则一下代码块会被编译，反之则不会被编译
8     printf("new:%s--%d--%s--%d\n" ,
9         new->Name ,new->age ,new->skill , new->udel );
10 #endif
11
12
```

```
1
2 #ifdef DE_BUG // 判断是否定义了
3     printf("__%s_%s_%d__\n", __FUNCTION__ , __FILE__ , __LINE__ );
4 #endif
5
6 #ifndef DE_BUG // 判断是否没定义
7     printf("__%s_%s_%d__\n", __FUNCTION__ , __FILE__ , __LINE__ );
8 #endif
```

形式2：

通过有值的宏来进行判断， 则可以通过值来判断（非零则真）

```
1 #define MACRO 0 // 非零则真
2 #define MACRO "Hello" // 错误的， 不允许出现字符串
3 #define MACRO 'A' // 允许
4
5 #if MACRO // 只要判断MACRO为非零值则表示条件为真
6     printf("__%s__%s__%d__\n", __FUNCTION__ , __FILE__ , __LINE__ );
7 #endif
```

```
1
2 #if MACRO
3     printf("__%s__%s__%d__\n", __FUNCTION__ , __FILE__ , __LINE__ );
4 #else
5     printf("__%s__%s__%d__\n", __FUNCTION__ , __FILE__ , __LINE__ );
6 #endif
7
8
9 #if MACRO1
10     printf("__%s__%s__%d__\n", __FUNCTION__ , __FILE__ , __LINE__ );
11 #elif MACRO2
12     printf("__%s__%s__%d__\n", __FUNCTION__ , __FILE__ , __LINE__ );
13 #else
14     printf("__%s__%s__%d__\n", __FUNCTION__ , __FILE__ , __LINE__ );
15 #endif
```

注意：

在使用有值宏进行条件编译的时候， 宏的值只允许出现整型/字符

多路分支可以根据自己的需求继续延续下去

使用条件编译必须有结束的语句 #endif 与开头进行对应

## 条件编译的实际应用场景

除了打开代码进行修改宏的值或者重新定义或删除宏的定义， 还可以通过编译命令来定义宏

```
1 $ gcc ifdef.c -DDE_BUG
```

2 `-D` --> 定义宏 `define`

3 `DE_DEBUG` --> 需要定义宏的名字为 `DE_DEBUG`