

## #define 定义宏

#define 机制包括了一个规定，允许把参数替换到文本中

#define name(paramant-list) stuff 其中 paramant-list是一个用逗号隔开的符号标

```
1 #define SQUARE(x) x*x
2
3 int main()
4 {
5     int ret = SQUARE(5);
6     return 0;
7 }
```

**注意\*\*\*\*\*：宏是用来替换的 而不是用来传参的**

**如下代码：**

```
1 #define SQUARE(x) x*x
2
3 int main()
4 {
5     int ret = SQUARE(5 + 1);
6     return 0;
7 }
```

**输出结果是ret = 11 相当于  $5+1*1+5 = 11$**

## #define 替换规则：

- 1.首先对参数进行检查，看是否包含任何由#define定义的符号，如果是，则首先被替换
- 2.替换文本随后被插入到程序中原来文本的位置，对于宏 参数名被他们的值替换
- 3.最后，再次对结果文件进行扫描，看是否包含任何#define定义的符号，如果是则重复以上过程

**注意：**

- 1.宏不能递归
- 2.字符串中出现的 宏 不被搜索

## 如何把参数插入字符串中？ # 和 ##

#X 在字符串中会将X所对应的内容传入

X##Y 将X和Y所代表的值 合并成一个 XY

## 宏有哪些缺点？

- 1.使用宏的时候（预编译） 每一份都会插入程序中 会增加代码的长度
- 2.宏没有办法调试
- 3.宏由于和类型无关，也不够严谨
- 4.宏会带来运算符优先级的问题