疯狂Python精讲 Q ≥ 消息 ® ▼

原 C学习笔记之预处理指令

2014年05月18日 17:50:53 葬花桥 阅读数: 792

一、什么是预处理指令

预处理指令是<mark>告诉编译器在编译之前预先处理的一些指令</mark>,有宏定义,文件包含,条件编译。 预处理指令一般<mark>以#号开头</mark>,可以出现在文件的任何地方,作用于整个文件。

二、宏定义

宏定义分不带参数的和带参数的宏定义。

1≥不带参数的宏定义

```
#import <stdio.h>
1
2
       #define AGE 10 // 宏名用大写
3
4
5
      int main()
6
7
           int age = AGE; // AGE在编译之前会替换成 10
8
9
10
           printf("%d", age);
11
12
13
           return 0;
14
       }
```

在编译器编译之前,代码中的所有AGE会被替换成 10 ,当AGE的值要修改的时候<mark>只需要在定义宏的地方修改一次,</mark>文件中的所有AGE都会被替换 由于宏是在编译之前被替换,所以由此引起的语法错误只有<mark>等到编译的时候才会对已经被替换在宏名作语法检查。</mark>

可以在定义一个宏时,引用已经定义的宏,如:

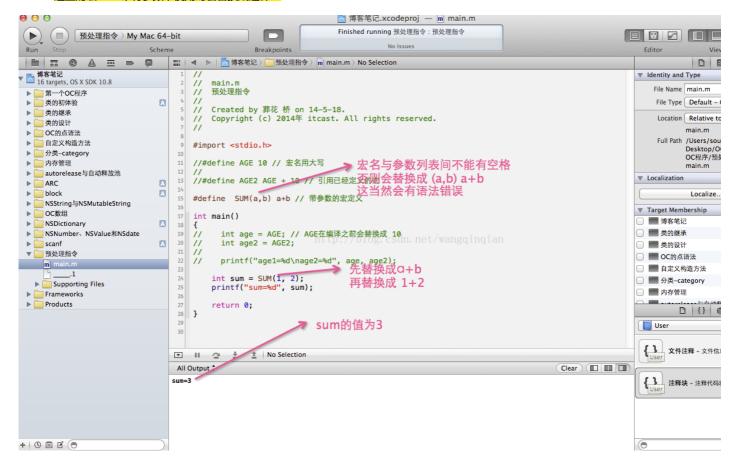
```
1 //
2 // main.m
3 // 预处理指令
4
   // Created by 葬花 桥 on 14-5-18.
5
   // Copyright (c) 2014年 itcast. All rights reserved.
6
7
   //
8
9
   #import <stdio.h>
10
11
   #define AGE 10 // 宏名用大写
12
13
   #define AGE2 AGE+10 // 引用已经定义的宏
14
15
   int main()
16
17
      int age = AGE; // AGE在编译之前会替换成 10
      int age2 = AGE2;
18
19
      printf("age1=%d\nage2=%d", age, age2);
20
21
22
23
       return 0;
24 }
```

2≥带参数的宏定义

带参数的宏定义和看起来和函数有点相似,但是它们的本质的区别,<mark>宏定义只是替换,不分配内存,没有参数的传送,更没有返回值。</mark>

#define SUM(a,b) a+b // 带参数的宏定义

这里的宏SUM带有参数,使用时会替换成这样:



三、文件包含

文件包含预编译指令我们都在用,#include<> #import 等等

它会将目标文件的内容拷贝一份到当前文件中,这很简单,没有什么可以多说的。<mark>只是要注意重复包含,循环包含造成的错误</mark>,<mark>重复定义错误或循环包</mark> 但是如果用到了<mark>条件编译就能避免重复包含引起的重复定义错误。</mark>

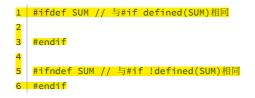
四、条件编译

```
1 #if defined(SUM)
2
       printf("SUM已经被定义! \n");
                                                              预处理
3
      #endif
这现个预处理指令表示如果SUM已经被定义就编译printf语句,与些相反的是
1 #if !defined(SUM)
                                          #pragma once
       printf("SUM没有被定义! \n");
2
       #endif
3
                                         #ifdef
                                                     cplusplus
                                         extern
                                          #endif
                                          #ifdef cplusplus
                                          #endif
```

如果SUM没有被定义,就编译printf语句 博客笔记.xcodeproj ─ m main.m Finished running 预处理指令: 预处理指令 预处理指令) My Mac 64-bit Run Scheme Breakpoints D #import <stdio.h> ▼ Identity and Type File Name main.m 第一个OC程序 //#define AGE 10 // 宏名用大写 类的初体验 File Type Default //#define AGE2 AGE + 10 // 引用已经定义的宏 类的继承 Location Relative ____类的设计 #define SUM(a, b) a+b // 带参数的宏定义 OC的点语法 16 17 18 19 20 自定义构造方法 Desktop ─ 分类-category { // OC程序/ 内存管理 int age = AGE; // AGE在编译之前会替换成 10 int age2 = AGE2; autorelease与自动释放池 **▼** Localization A ARC printf("age1=%d\nage2=%d", age, age2); i block A Localiz NSString与NSMutableString int sum = SUM(1, 2); printf("sum=%d\n", sum); > SUM是一个已经被定义的宏 ▼ Target Membership OC数组 ── 博客笔记 A NSDictionary #if defined(SUM) // 如果SUM已经被定例》,如神神间的代码语句会编译 [ian printf("SUM已经被定义! \n"); NSNumber、NSValue和NSdate □ ■ 类的继承 scanf A □ ■ 类的设计 . 预处理指令 □ ■ OC的点语法 #if !defined(SUM) // 如果SUM沒有被定义,就编译中间的代码 printf("SUM沒有被定义! \n"); 不编译这一句 □ 自定义构造方法 ☐ 分类-category Supporting Files □ 内存管理 Frameworks D {} M SUM已经被定义 return 0; User 所以中间的printf不会被编译 文件注释 - 文件 ▼ II 2 ± ± No Selection All Output ‡ Clear 🔲 🔳 注释块 - 注释代 sum=3 SUM已经被定义!

与这两个相同作用的是

+ | 0 = 7 (



那么在文件包含时,使用条件编译就可以防止重复包含同一个文件。



6