## 注释的区别。

/\*\*/这种注释，压缩成css文件之后，仍然会保留

// 形式的注释，压缩成css文件之后，不会保留。

既然维护的是less文件，可以//来注释，不必再css文件中保留注释内容。

## Less里面定义变量必须用@

@less-width：300px；//声明less变量

.box{width：@less-width；}//使用less变量

编译成css后，.box{width：300px；}

## 混合

@less-width：300px；//声明less变量

.box{

width：@less-width；

height：@less-width；

.text-border；//less的混合，直接用名称 调用

}

.text-border{border:1px solid red};//less中声明，后被混合调用

.box2{.box；margin-left：10px；}//编译成css时，“.box”属性会自动补全box里面的所有属性，并且加上后来的margin-left。

//真正编译成css的时候 .box里面的.text-border属性会自动变成（.text-border{border:1px solid red};）

## 带参数的混合

.border-02(@border-width){//带less变量参数

border:solid 1px @border-width;

}

.text-hunhe{.border-02（30px）；}//less混合的使用，输出其实是 border:solid 1px 30px；//传入的30px的参数

## 默认带值得混合

.border-03（@border-width：1px）{

border：solid green @border-width；

}//“：1px”表示默认的宽度是1px，@border-width这个参数名是可以重复的，即其他的border也可以定义一个@border-width，[border-03的@border-width的作用域只是在.border-03里面](mailto:border-03的@border-width的作用域只是在.border-03里面)。

使用： .border-hunhe-03{

.border-03（）；//则默认输出1px

}

## 混合在CSS3里面的使用

.border-raidus(@raidus:5px){

border-raidus:@rdidus;

-moz-border-raidus:@rdidus;

-webkit-border-raidus:@rdidus;

}//声明带less参数的border-raidus

//使用less混合

.border-test{

height：100px；

width：50px；

.border-raidus（）；//使用less混合。就不用每次都打css3的兼容代码了

}

## 匹配模式

.pos(r){positions:relative;}

.pos(f){position:fixed;}

.pos(a){position:absolute;}

.piper{

height:10px;

width:10px;

.pos(a); 或者.pos（r）；

}//匹配的使用。传的哪个字符用的就是哪个相对应的css样式

## 运算

@text-01：300px；

.box-01：{width：@text-01+20；}//注意：没有强制要求带px,只要@text-01或者20，其中一个带px就行。

## 嵌套规则-重要

.list{//其他用处，写层级的结构，如导航

…

…

li{…}

a{…}

}//就像是html里面标签的嵌套，less写的时候就可以按照html来写嵌套

相当于

.list li{}

.list a{}



第二种情况：当要写

.list a{…}

.list a:hover{…}

用less 借助嵌套可以写成：

.list{

…

a{

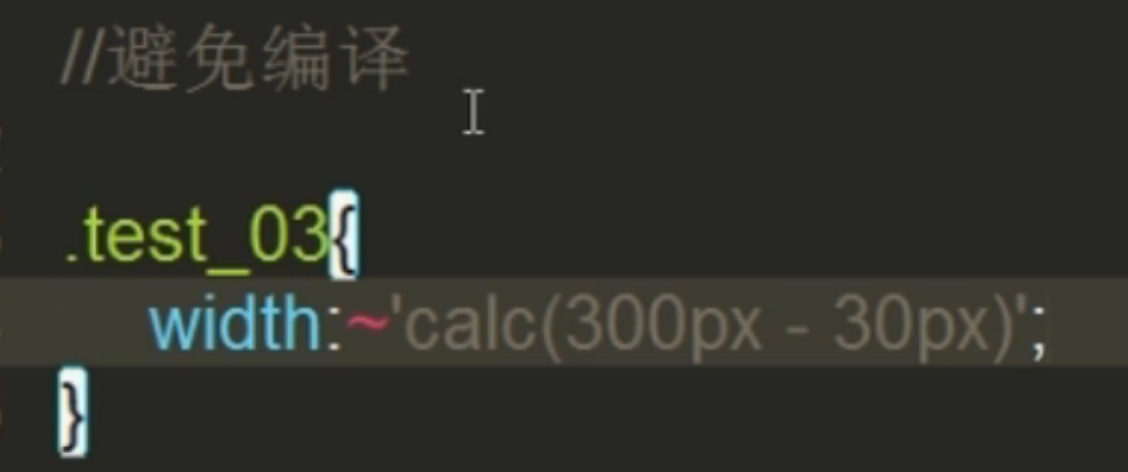
…

&:hover{color:red;} //&代表他的上一层选择器

}

}

## 避免编译



像这种不希望less帮我们编译的代码，用~‘’把相关代码包进去，编译的时候就会原封不动的放进less里面。不会进行操作。