# Less语法规则

# 一、less简介

Less 是一门 CSS 预处理语言，它扩展了 CSS 语言，增加了变量、Mixin、函数等特性，使 CSS 更易维护和扩展。

本质上，Less 包含一套自定义的语法及一个解析器，用户根据这些语法定义自己的样式规则，这些规则最终会通过解析器，编译生成对应的 CSS 文件。Less并没有裁剪 CSS 原有的特性，更不是用来取代 CSS 的，而是在现有 CSS 语法的基础上，为 CSS 加入程序式语言的特性。

Less 可以LESS文件编译生成静态 CSS 文件，并在 HTML文档中应用。

less中。

Less有两种注释：

/\*\*/会被编译，

//注释不会被编译，即编译后不会被保留

# 二、less变量

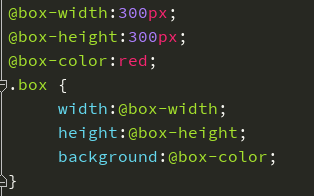
LESS 允许开发者自定义变量，变量可以在全局样式中使用，变量使得样式修改起来更加简单。

定义：@变量：值；

使用：属性：变量；

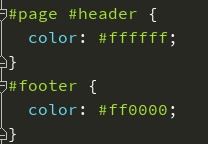
LESS中定义变量必须以“@”开头，定义之后可以当作属性值使用。变量可以将相同的值定义成变量统一管理起来。

该特性适用于定义主题，我们可以将背景颜色、字体颜色、边框属性等常规样式进行统一定义，这样不同的主题只需要定义不同的变量文件就可以了。当然该特性也同样适用于 CSS RESET（重置样式表），在 Web 开发中，我们往往需要屏蔽浏览器默认的样式行为而需要重新定义样式表来覆盖浏览器的默认行为，这里可以使用 LESS 的变量特性，这样就可以在不同的项目间重用样式表，我们仅需要在不同的项目样式表中，根据需求重新给变量赋值即可。

例如：定义一个长宽为300px的div，背景颜色为红色：

注意：变量也存在作用域问题，@变量 会首先在当前作用域寻找，然后再逐层往父作用域中寻找，一直到顶层的全局作用域中为止。

例如：编译后，#page的颜色为白色：

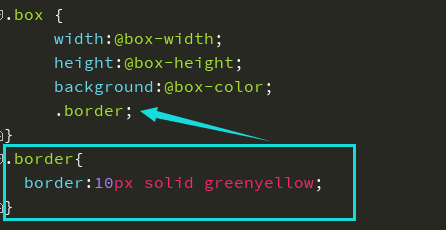


# 三、less混合

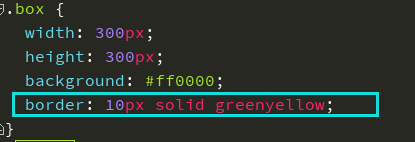
LESS 中，混入是指在一个 CLASS 中引入另外一个已经定义的 CLASS，就像在当前 CLASS 中增加一个属性一样。

### 1、基础混合

Mixins 其实是一种嵌套，它允许将一个类嵌入到另外一个类中使用，直接将引用的类名写在该类里，例如：



编译后：



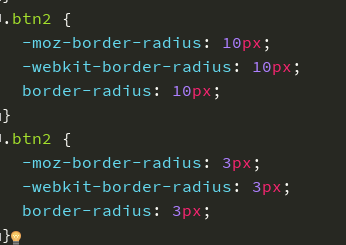
### 2、带参数混合

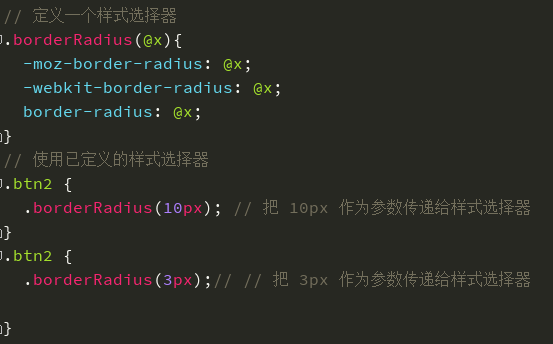
可以给CLASS类传递参数，内部属性值便可直接使用，这样更加灵活和具有通用性。

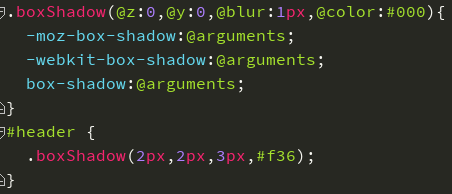
定义：.类名（变量形参）{ 属性：变量形参 }

使用：选择器{ .类名（实参）}

例如：我们定义一个按钮类“.borderRadius”，希望所有的按钮都应用该类的结构，只是根据需要设置尺寸的大小：

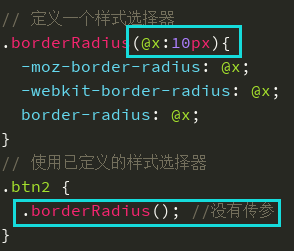
编译后即为右图：

 像 JavaScript 中 arguments一样，Mixins 也有这样一个变量：@arguments。@arguments 在 Mixins 中具是一个很特别的参数，当 Mixins 引用这个参数时，该参数表示所有的变量：

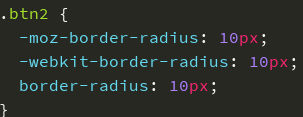
注意：显示定义不带参数和带参数的样式库(mixin库)，不会输出到最终输出中，仅供调用

### 3、带默认值混合

还可以给变量形参设置默认值：

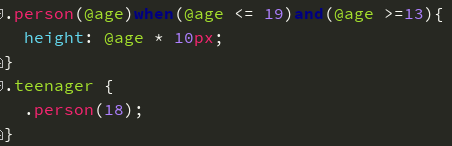


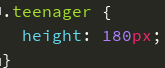
编译后：



# 四、选择与循环控制

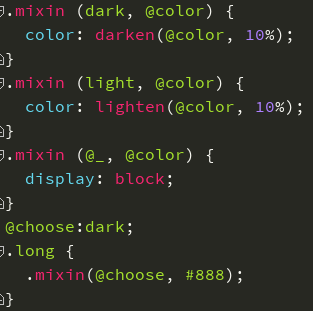
Less中通过混合（Mixin）后的when关键字来提供选择控制，通过递归来实现循环控制：

例如：

编译后：

# 五、匹配模式

### 1、一般匹配模式

 模式匹配的语法，类似于switch case的思维，只有满足匹配要求的混合才会被使用。混合中的变量可以匹配任何值，非变量形式的值只有与传入的值完全相等时才可以匹配成功。

注意：最后需要跟一个匹配是@\_的模式，常用于放置公共部分，匹配不出时也会执行

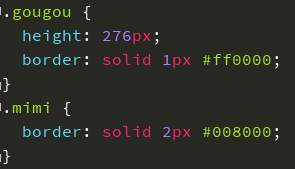
### 2、参数数量匹配模式

我们也可以根据参数的数量进行匹配，mixin的重载可定义多个同名mixin，调用时只要参数数量匹配则会执行相应的mixin。例如：

### 3、@arguments和@rest...

mixin的中内置两个特殊的对象@arguments和@rest...。@arguments代表mixin的所有传入参数，而@rest代表mixin的剩余入参数组。如果你希望你的mixin接受数量不定的参数，你可以使用...。在变量名后面使用它，它会将这些参数分配给变量。

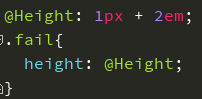
 编译后：

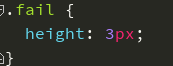


# 六、运算符

Less还支持+、-、\*、/运算符。但对单位不一致的运算数进行运算要注意以下两点：

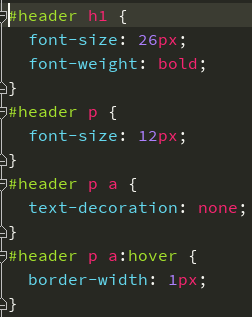
1. 运算数与运算符间必须用空格分隔；
2. 以第一个运算数的单位作为运算结果的单位；

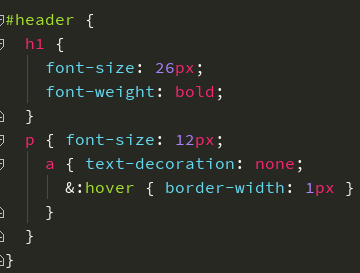
 编译后：



# 七、嵌套

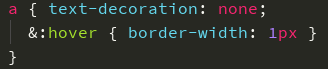
less提供嵌套功能，这种嵌套不仅格式清晰，而且代码量少。

 嵌套，其实在我们HTML代码中一直都在使用，如div中的p元素等，对于我们样式的书写，需要使用div p {} 进行样式的指定，这也就是我们的后代选择器。为了防止样式的覆盖，每种选择器都是一大长串，写起来很麻烦。less利用嵌套的书写方式，让我们在写后代选择器的时候有了新的方法。

编译后：

### 1、&父选择器

采用&引用上一层选择器，例如：



通过&::after等方式添加伪元素、伪类样式规则集合

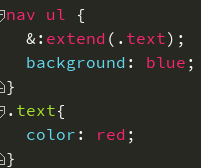
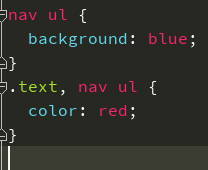
# 八、继承

扩展选择器是less的伪类选择器，他会复制当前选择器，定义新的样式， 有两种语法形式：

<selector>:extend(<parentSelector>){}

 <selector>{ &:extend(<parentSelector>); }

例如：当文本为红色，列表也想设置为红色，并且额外有自己样式时：

编译后：

# 九、函数

Less为我们提供了一个功能强大的内置函数库，其中绝大部分为颜色处理函数。主要说明default函数和颜色处理函数。

### 1、default函数

default函数用于条件控制当中，当参数不满足条件时调用，充当else的角色：



注意：default函数必须在条件控制语句当中使用

### 2、颜色处理函数

颜色会先被转化成 HSL 色彩空间, 然后在通道级别操作

颜色处理函数又分为四大类：

颜色定义函数(Color Definition)

颜色通道值获取函数(Color Channel)

颜色通道值修改函数(Color Operation Function)

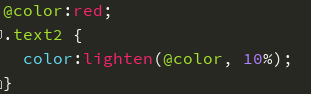
混色函数(Color Blending)

常用的lighten和darken函数：

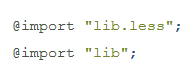
   lighten(color, amount) ，color为颜色，amount为增加的亮度值，取值范围为0-100%。

   darken(color, amount) ，color为颜色，amount为减少的亮度值，取值范围为0-100%。

例如：



# 十、导入指令(Import)

 less样式文件可通过 @import '文件路径'; 引入外部的less文件。

其中.less 后缀可带可不带:如果你想导入一个CSS文件而且不想LESS对它进行处理，只需要使用.css后缀就可以

@import可出现在任何位置，而不像css的@import那样只能放在文件第一行。

另外@import还提供了6个可选配置项(分别为reference, inline, less, css, once, multiple)，用来改变引入文件的特性。

语法为： @import (reference) '文件路径';

1. @import (reference) "文件路径"：

将引入的文件作为样式库使用，因此文件中样式不会被直接编译为css样式规则。当前样式文件通过extend和mixins的方式引用样式库的内容。

1. @import (inline) "文件路径";

用于引入与less不兼容的css文件，通过inline配置告知编译器不对引入的文件进行编译处理，直接输出到最终输出。注意：引入的文件和当前文件会被编译为一个样式样式

1. @import (less) "文件路径";

默认使用该配置项，表示引入的文件为less文件。

1. @import (css) "文件路径";

表示当前操作为CSS中的@import操作。当前文件会输出一个样式文件，而被引入的文件自身为一个独立的样式文件

1. @import (once) "文件路径";

默认使用该配置项，表示对同一个资源仅引入一次。

1. @import (multiple) "文件路径";

表示对同一资源可引入多次。