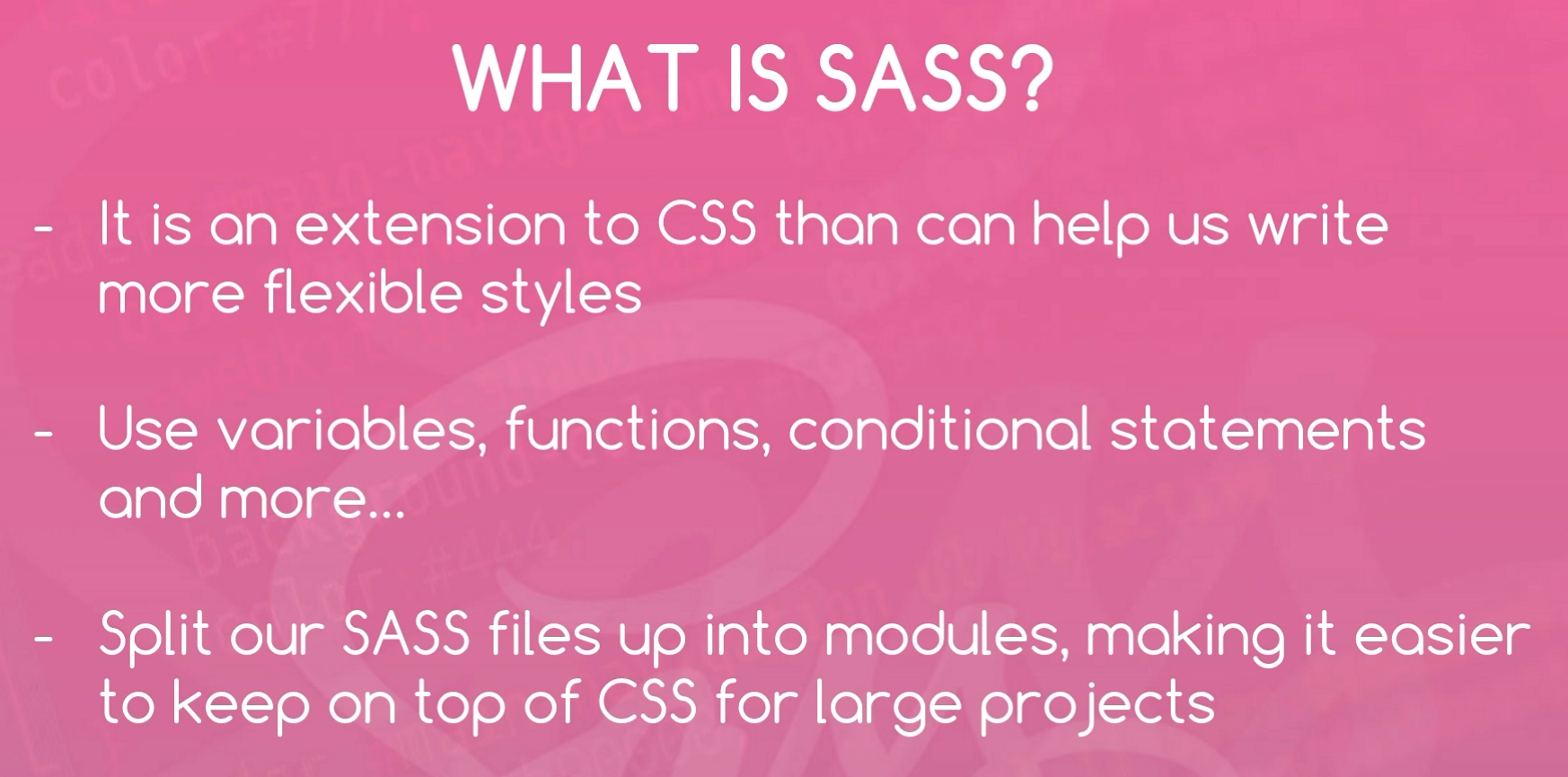
01\_what\_is\_sass



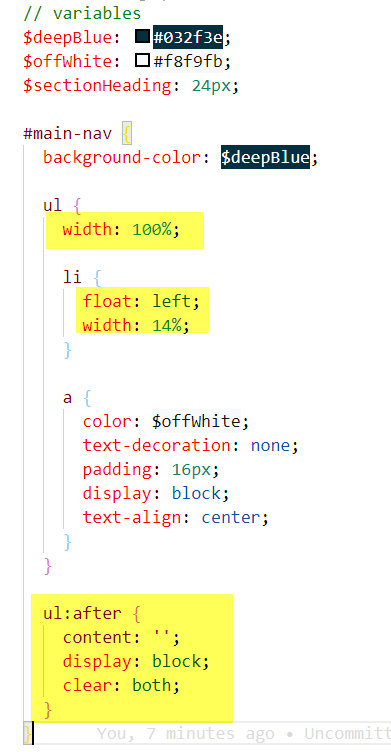


03\_sass\_variables

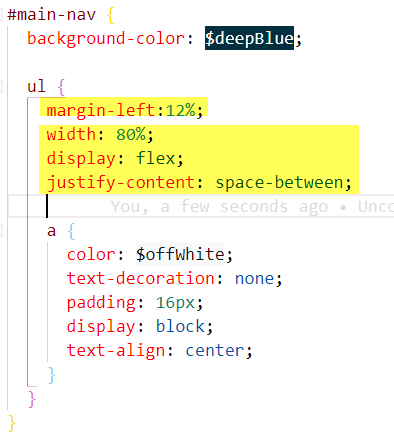


注意：

Sass 的variables可以储存color以及font-size

另外视频对于header的ul用的是比较老的float:left, 然后每个li 占14%，最后还必须清除ul的float，   


我们可以修改为flex

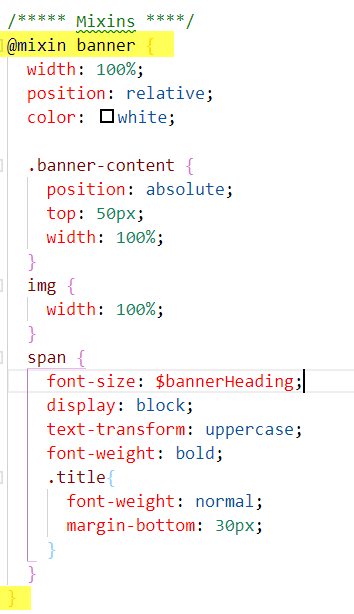


05\_mixins

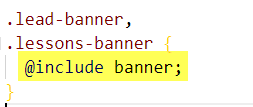
A mixin is a chunk of reusable css/sass that we can inject into any element that we want

比如在我们的项目当中，有两个banner,它们有一些相同的基本属性，这些属性就可以写成为一个mixin，然后include到这两个banner的样式当中去

1. 定义mixins, 使用@mixin <name> { }



1. 在不同的element调用mixin, 使用@include <mixin\_name>



1. 然后在不同的element使用自己独有的css



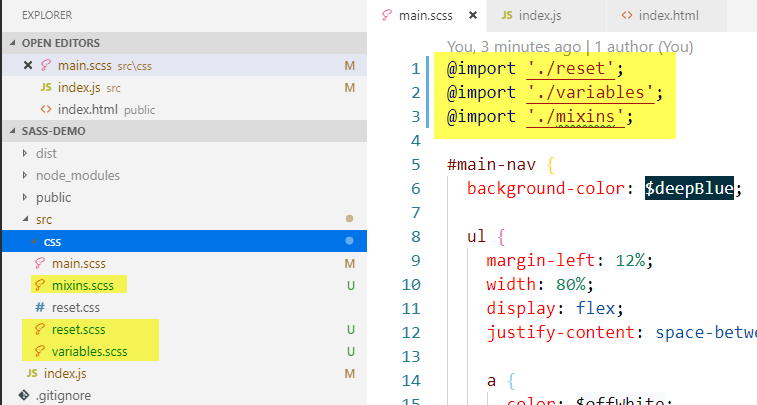
06\_importing\_files

对于大的项目，这个sass文件可能太长了，我我们可以将它模块化，然后插入需要的sass文件

我们将刚才的文件分拆成为variables.scss 和 mixins.scss

然后使用@import 将不同的scss文件import进来。

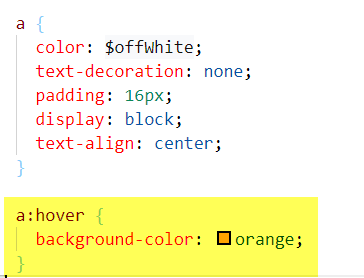
注意！import的顺序是有意义的，比如先import variables,这样mixins当中如果使用variables可以找到



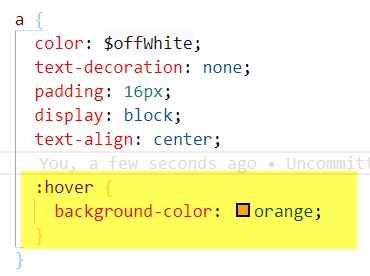
07\_pseudo\_classes 以及 &

& concatenates the parent class

如果要给按钮加上hover effect,我们一般是这样写



如果在sass里面使用nested property的写法，下面这样是不行的

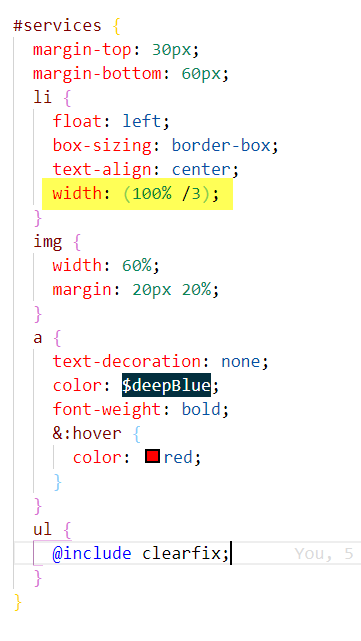


必须在pseudo class前面加上&号，表示<parent>:<pseudo class>

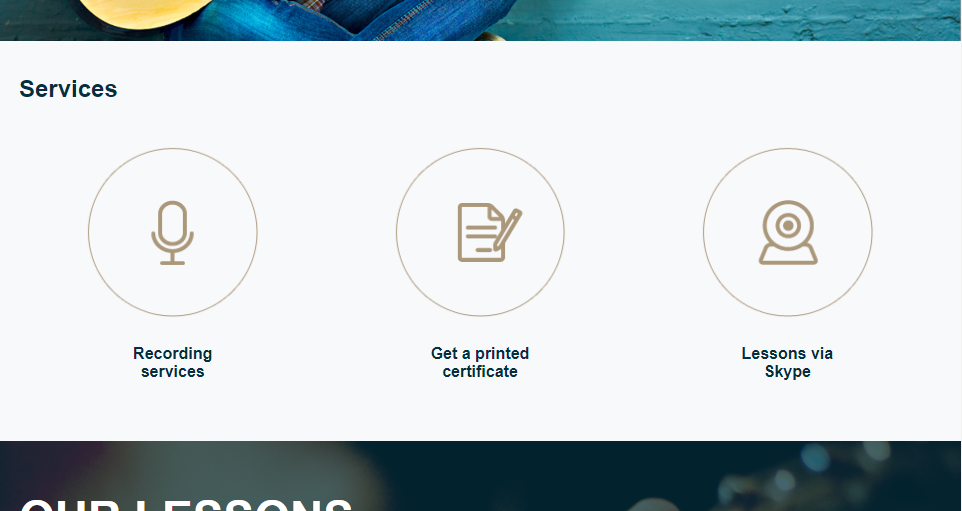


08\_mathematical\_operators

即 sass支持 加减乘除

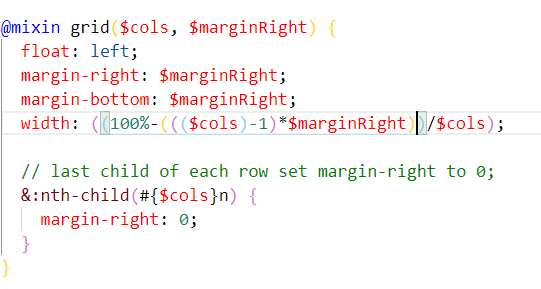


即取它的parent element (ul)的1/3



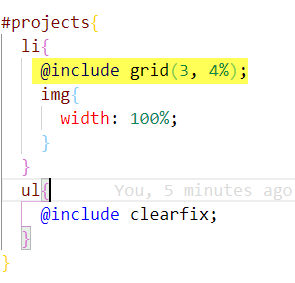
09\_creating\_a\_grid\_with\_sass\_math

我们创建一个简单的mixin,它会根据你传入的一行有几个columns，几个每个column的margin right的值，来自动算出每个column的width

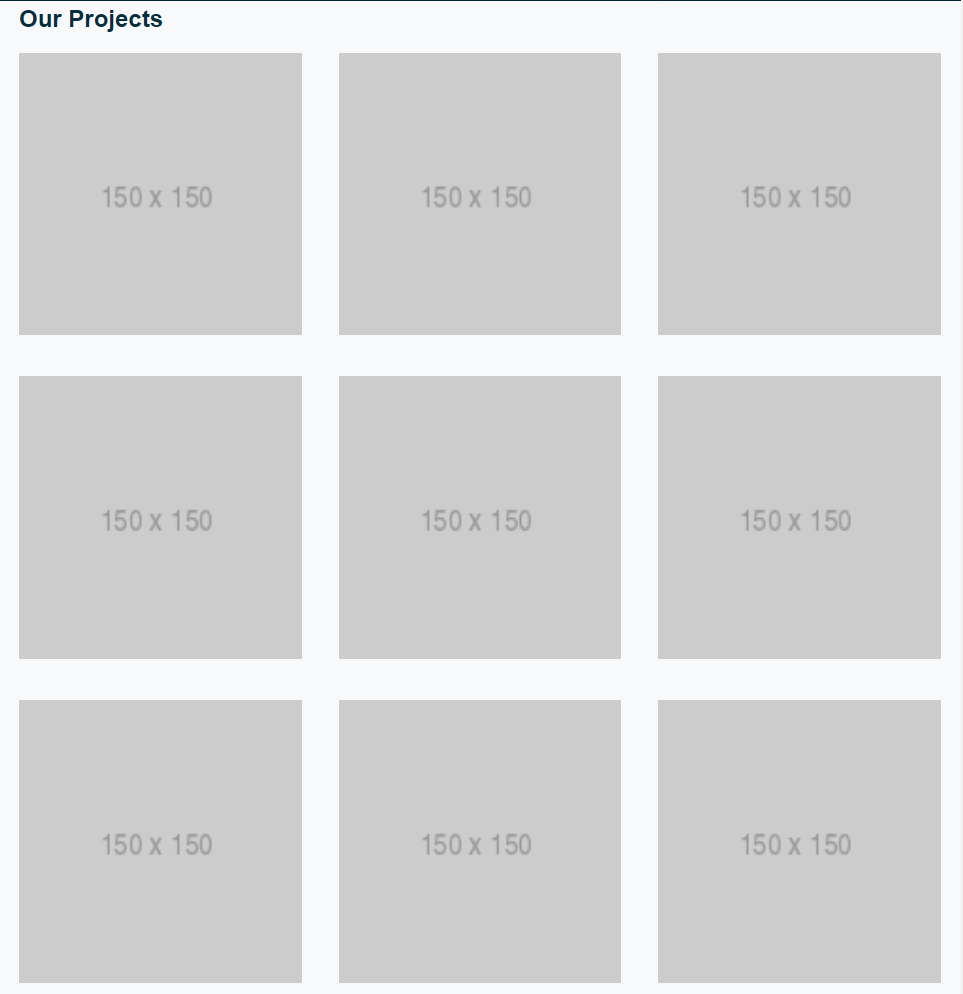


1. 每行的最后一个child element不应该有margin-right, 所以使用nth-child来消除每行最后一个child element的margin-right
2. 每个column的宽度，是由全部columns的个数-1(每行最后一个column没有margin-right) \* marginRight, 来得出全部的margin, 然后用100%-总的margin，得到目前available的宽度，再除以columns的个数，得到每个column的宽度
3. 计算公式当中不能有空格，不然报错
4. 计算公式最外面要用()包裹起来

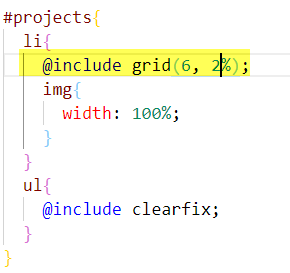
使用：



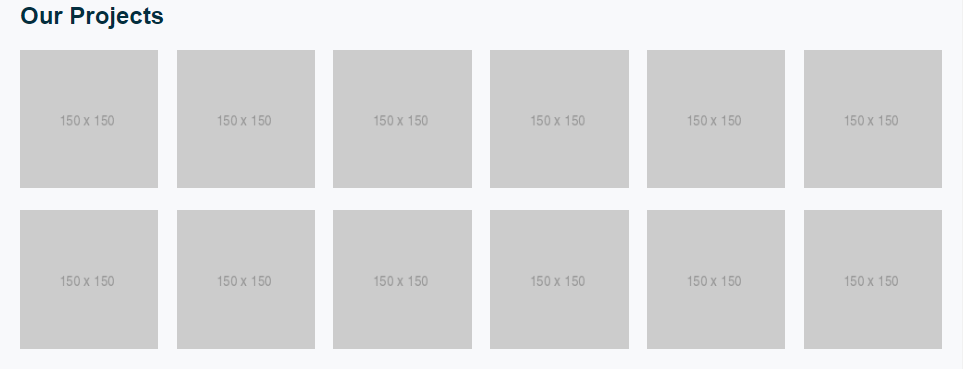
效果是：



如果



效果变为：



我们使用sass的mixin和计算功能，建立了一个自动计算宽度的简单的grid

10\_colour\_functions

<https://sass-lang.com/documentation/functions/color>

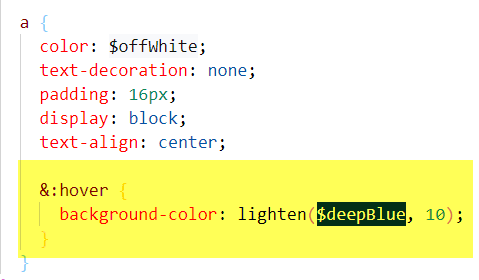
有很多，我们只看几个：

1. $lighten()

**lighten**($color**,** $amount) *//=>* [*color*](https://sass-lang.com/documentation/values/colors)

The $amount must be a number between 0% and 100% (inclusive). Increases the HSL lightness of $color by that amount.

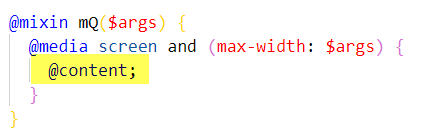
使用：



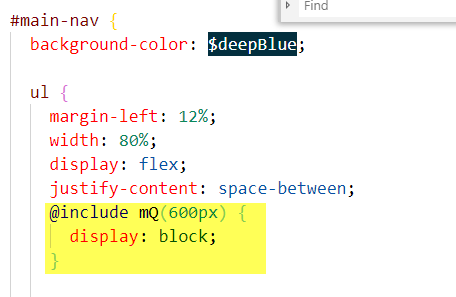
11\_the\_content\_keyword

Content keyword像是一个placeholder,它等待用户传入的具体内容

我们创建一个mixin

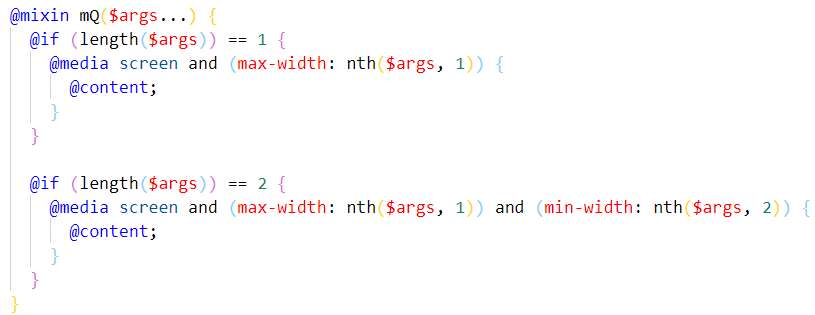


这里使用@content 关键字等待user调用时传入的具体样式，并在mixin当中显示出来

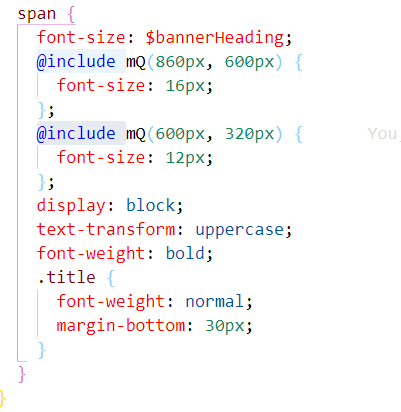


这个例子，我们在调用这个mixin的时候，传入的样式为display:block, 这样就在600px以下的时候，@content采用这个display:block

那么问题来了，如果我想同时传入并且使用max-width和min-width，应该怎么处理呢？



调用：



效果：

