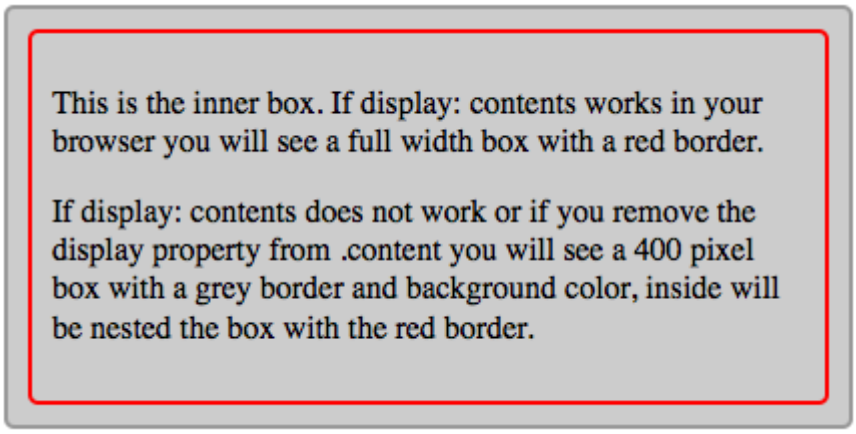
display：contents 与消失的盒模型

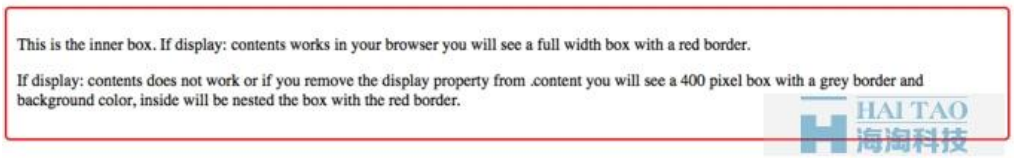
今天海淘科技为你带来的是 **display：contents 与消失的盒模型。**

在 CSS3 可视化模块中，display: contents 是 display 属性一个新的属性值。FireFox 已经实现了对该属性的支持，这里将简单介绍该属性的作用，及其被浏览器广泛支持后存在实用价值的原因。"元素本身不能生成任何盒模型，但是它的子元素或者伪元素可以正常生成。为了盒模型的生成与布局，该元素就好像在 Dom 树中被子元素与伪元素所替代一样"。上述规范说明了我们可以在文档中添加一个 HTML 元素，并在该元素的选择器中添加 display:contents 样式，那么该元素就会像是不存在一样，它的子元素会替代它在 Dom 树中的位置。

使用示例理解起来更加容易，因此我们需要使用 FireFox。假设现在我们有两个 Div 元素，外层的 Div 元素拥有类 content，内层 Div 元素拥有类 inner。分别为它们添加背景色与边界，如下所示：



如果为外层 Div 元素添加 display:contents 属性，示例变为

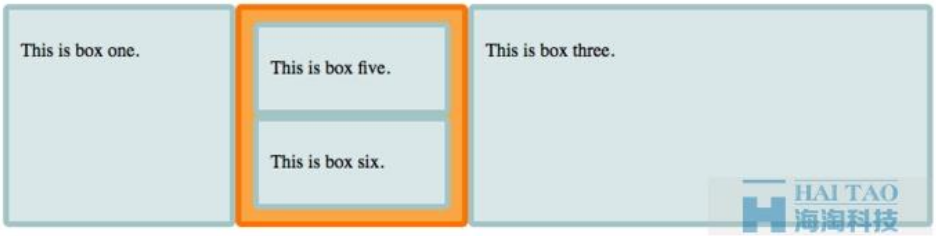


外层 Div 消失了，我们看不到外层 Div 的背景，边框 ，甚至应用到该元素的宽度也已经失效了，内层 Div 元素则占据了整个视口。display:contents 便是这样起作用的。可以使用 FireFox 打开 CodePen 中的该示例查看效果。在不支持该属性的浏览器中，display:contents 将会被忽略。需要注意的是，内部 Div 元素不能继承的属性仅仅是那些与盒模型生成或布局相关的。我们可以在外层 Div 元素上设置 font-size 属性，其子元素则会继承该属性。

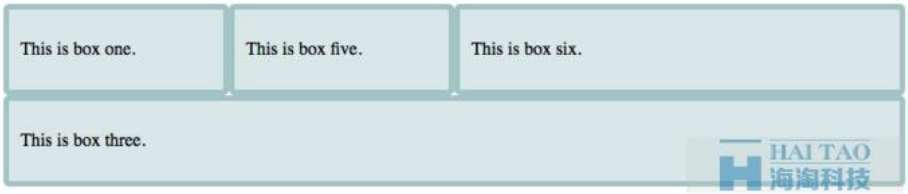
使用 display:contents 来实现 Flex 项目的"继承"

这可能有用吗?如果你正在使用弹性布局，那么你应该知道只有 Flex 容器的直接子元素可以成为 Flex 项目。而 Flex 项目的子元素是不能够使用父元素的规则来进行布局的。我们可以通过一个简单的示例来印证。有一个 Div 包含快，内部直接嵌套了三个类名分别为 box1、box2、box3 的 Div 元素，这三个 Div 均是 Flex 项目，可以对它们使用弹性布局。然而 box4、box5 是 box2 元素的子元素，因此尽管我们已经应用 Flex 项目属性到 box5，但由于其父元素 display 属性没有被设置为 flex，因此会被忽略。

我们给第二个 Flex 项目添加一个明显的背景和边界来帮助我们看清楚发生了什么。



如果为 box2 添加属性 displsy:content，则在 FireFox 中我们将看到 box2 已经消失了，我们也不能看到橘色的边界和背景。不仅仅是这样，它好像已经在 Dom 树中完全消失不见了。而 box2 的子元素则像 Flex 项目一样，应用到 box5 上面的样式也生效了。



这是站点 CodePen 上的一个示例。

如果想添加一些有语义但又不显示的元素，那么该属性是非常有用的。如果有一些内容在语义上适合被标记为 article，然而 article 元素在布局上是 Flex 项目，但是你真正希望的是使 article 内部嵌套的元素作为 Flex 项目存在。那么为了使 article 嵌套的元素成为 Flex 项目，比起去除 article 元素，你更应该使用 display:contents 来移除 article 产生的盒模型。这样做你将会在语义化以及显示效果上都能得到很好地效果。听起来还不错。

今天的 **display：contents 与消失的盒模型**，就到这里了，还有搜索引擎 seo 推广的文章。文章下载，点击：display：contents 与消失的盒模型