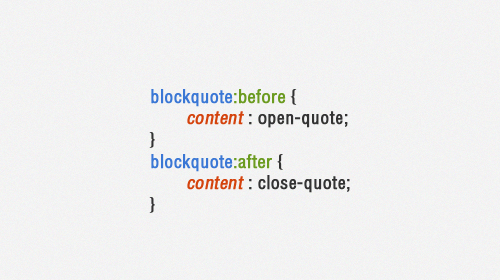
层叠样式表（CSS）的主要目的是给HTML元素添加样式，然而，在一些案例中给文档添加额外的元素是多余的或是不可能的。事实上CSS中有一个特性允许我们添加额外元素而不扰乱文档本身，这就是“伪元素”。



本文由听海根据Thoriq Firdaus的《Understanding Pseudo-Element :Before And :After》所译，整个译文带有自己的理解与思想，水平有限，还希望各位朋友多多指点。如需转载此译文，需注明英文出处：http://www.hongkiat.com/blog/pseudo-element-before-after/，  
以及作者相关信息。

作者：Thoriq Firdaus

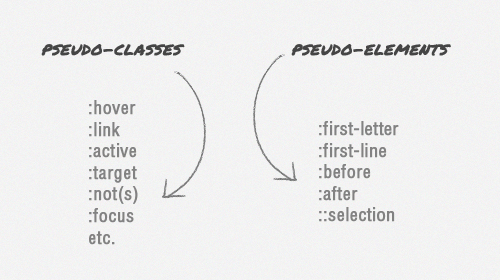
译者：听海JamiE

你一定听说过这个词，尤其是当你一直关注着我们的教程。[点此浏览原作者的其他文章](http://www.hongkiat.com/blog/author/thoriq/" \t "_blank)

事实上，的确有一些CSS家族的成员（CSS选择器）被分类为伪元素比如：:first-line, :first-letter, ::selection, :before 和 :after。但是，就本文而言，我们将把我们探讨的范围限制在:before 和 :after这两个元素上。因此，本文中的“伪元素”将特指这两个伪元素（:before和 :after），我们将从基础入手，来研究这个独特的主题。

**关于语法和浏览器支持**

伪元素实际上在CSS1中就存在了，但是我们现在所讨论的:before和:after则发布于CSS2.1中。在最初，伪元素的语法是使用“:”（一个冒号），随着web的发展，在CSS3中修订后的伪元素使用“::”（两个冒号），也就是::before和::after—以区分伪元素和伪类（比如:hover，:active等）



然而，无论你使用单冒号还是双冒号，浏览器都将能识别它们。由于IE8只支持单冒号的格式，安全起见如果你想要更广泛的浏览器兼容性那么还是使用单冒号的格式吧！

**它是做什么的**

简而言之，伪元素将会在内容元素的前后插入额外的元素，因此当我们添加它们时，使用以下的标记方式，他们在技术上是平等的。

|  |
| --- |
| <p>    <span>:before</span>      This the main content.    <span>:after</span>  </p> |

但是这些元素实际上并不在文档中生成。它们将在外部可见，但是你将不会在文档的源代码中找到它们，因此，实际上它们是“虚假”的元素。

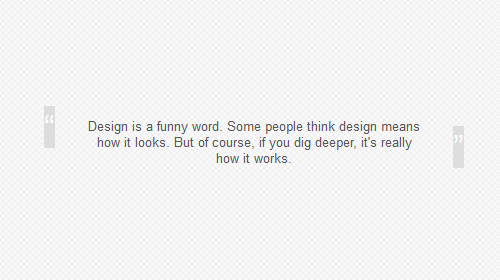
**使用伪元素**

使用伪元素是相对容易的，:before将会在内容之前“添加”一个元素而:after将会在内容后“添加”一个元素。在它们之中添加内容我们可以使用content属性。

举例来说，下面的代码片段将在引用的之前和之后分别添加添加一个引号。  

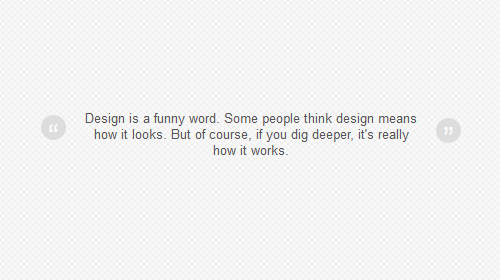

|  |
| --- |
| blockquote:before {   content: open-quote;  }  blockquote:after {   content: close-quote;  } |

**伪元素样式**

尽管作为“虚假”的元素，事实上伪元素表现上就像是“真正”的元素，我们能够给它们添加任何样式，比如改变它们的颜色、添加背景色、调整字体大小、调整它们中的文本等等。  


|  |
| --- |
| blockquote:before {   content: open-quote;   font-size: 24pt;   text-align: center;   line-height: 42px;   color: #fff;   background: #ddd;   float: left;   position: relative;   top: 30px;  }  blockquote:after {   content: close-quote;   font-size: 24pt;   text-align: center;   line-height: 42px;   color: #fff;   background: #ddd;   float: right;   position: relative;   bottom: 40px;  } |

**指定伪元素尺寸**

默认生成的元素是一个内联元素，于是当我们想要指定它们的高度和宽度的是偶，我们首先不得不使用display: block把它们声明为块级元素。  
由于已经设置float，所以无需设置display:black。  


|  |
| --- |
| blockquote:before {   content: open-quote;   font-size: 24pt;   text-align: center;   line-height: 42px;   color: #fff;   background: #ddd;   float: left;   position: relative;   top: 30px;   border-radius: 25px;   height: 25px;   width: 25px;  }  blockquote:after {   content: close-quote;   font-size: 24pt;   text-align: center;   line-height: 42px;   color: #fff;   background: #ddd;   float: right;   position: relative;   bottom: 40px;   border-radius: 25px;   height: 25px;   width: 25px;  } |

**关联背景图像**

我们也可以替换用图片替换内容而不是只有纯文本。尽管content属性提供了 url()来插入图片, 但是在更多的实例中，我更倾向于使用背景（background）属性从而更好的控制图片。  


|  |
| --- |
| blockquote:before {   content: " ";   font-size: 24pt;   text-align: center;   line-height: 42px;   color: #fff;   float: left;   position: relative;   top: 30px;   border-radius: 25px;   background: url(images/quotationmark.png) -3px -3px #ddd;   display: block;   height: 25px;   width: 25px;  }  blockquote:after {   content: " ";   font-size: 24pt;   text-align: center;   line-height: 42px;   color: #fff;   float: right;   position: relative;   bottom: 40px;   border-radius: 25px;   background: url(images/quotationmark.png) -1px -32px #ddd;   display: block;   height: 25px;   width: 25px;  } |

然而，正如你能够从上面的代码片段中看到的，我们仍旧声明了content属性，而且此时使用了空字符串。content属性是必须的而且应该经常被应用。否则，伪元素无论如何都无法正常工作。

**结合伪类**

尽管有不同类型的伪X（伪元素、伪类），我们可以使用伪类连同伪元素一起放入一个CSS规则中，例如，如果我们希望当我们的鼠标移到blockqoute上时，引号的背景色能够略微变深。

|  |
| --- |
| blockquote:hover:after, blockquote:hover:before {   background-color: #555;  } |

**添加过渡效果**

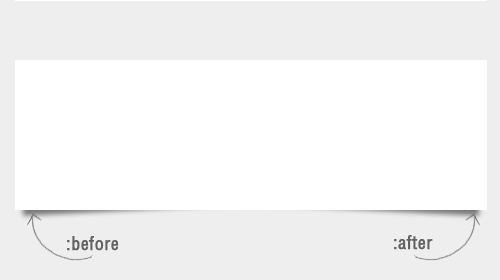
我们甚至可以在伪元素上应用transition属性来创建优美的颜色过渡效果。

|  |
| --- |
| transition: all 350ms;  -o-transition: all 350ms;  -moz-transition: all 350ms;  -webkit-transition: all 350ms; |

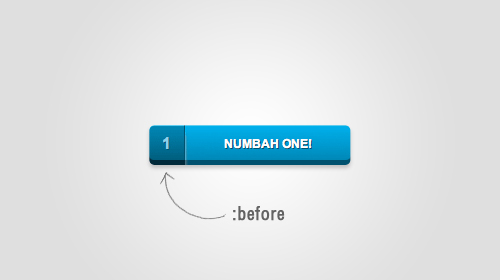
[Demo](http://media02.hongkiat.com/before-after-elements/demo/index.html)  
[Download Source](http://www.jqhtml.com/wp-content/uploads/2017/08/wz/source.zip)

**更多的灵感**

[**迷人的阴影**](https://paulund.co.uk/learn-css-box-shadow)

在这个教程中 Paul Underwood 解释了如何创建更加逼真和吸引人的阴影效果。  
使用 伪元素:before 和 :after 。它们两个都是绝对定位，而且使用负z-index来放置到图片后方实现阴影效果。  


[**3D按钮**](http://www.esecamalich.com/blog/before-&-after/)

这是一个非常聪明的实现，利用伪元素结合CSS3 box-shadow 来绘制一个令人吃惊的3D按钮，仅仅使用了CSS和单一的锚文本。伪元素:before 被用来在按钮的左侧添加数字“1”。  


[**叠加图像效果**](http://designshack.net/articles/css/use-pseudo-elements-to-create-an-image-stack-illusion/)

使用伪元素来仅仅依靠一个图片标签创建一个“凌乱的”叠加图像效果也是可能的。伪元素用于建立一个图片叠加的错觉，通过使用z-index负值使“叠加”的图片在真正的图片后面来实现。  


**结论**

伪元素很酷同时也是可应用到实际工作中的，基本上，每一个我们所添加的元素都不会干扰现有的HTML结构，而且伪元素可以做到 [几乎所有我们能想到的事情](http://css-tricks.com/9516-pseudo-element-roundup/)。  
实际上有一些[伪元素的改进工作](http://www.w3.org/TR/css3-content/),目前逐步进行，比如伪元素嵌套div::before::before { content: ”; }以及多重伪元素div::before(3) { content: ”; }。很显然，在未来的web设计中，这些改进会让我们的设计有更多的形式（更多的可能性）。然而，他们将会在最新的浏览器中得到支持，让我们现在耐心的等待吧！

原文地址：<https://tinghaige.com/2013/09/17/understanding-pseudo-element-before-and-after/>