**译序：**本文译自Smashingmagazine，但是原文讲述的内容有些浅，也不是很完整，[前端观察](http://www.qianduan.net)在翻译的前提下，增加了更多的更系统的内容。如有不足之处，欢迎指正补充。

随着网络的发展，浏览器具有更强的渲染更高级代码的能力，我们正逐步实现跨越所有平台和浏览器的目标。我们不但可以要花费更少的时间来确保我们的盒模型在IE6中看起来正常，而且形成了鼓励创新、避免hack、重前端脚本的氛围。

网络是一个非常好的环境，也是一个有丰富的知识来分享的协作社区。我们想要有[圆角](http://ragamo.medioclick.com/jquery/corners/)，我们就实现了它；我们想要[多背景图片](http://www.protocoder.com/css/css-multiple-backgrounds-background-layering-with-jquery/)，我们实现了它；我们想要[边框图片](http://www.lrbabe.com/sdoms/borderImage/index.html)，我们也让它实现了。所以需求从来不是问题，否则，我们可能还都仍然在使用table来布局页面而且使用过多的代码呢。我们都知道，网络无所不能。

**为网络而生**

CSS 3的属性比如[border-radius](http://www.css3.info/preview/rounded-border/)、[box-shadow](http://www.css3.info/preview/box-shadow/)和 [text-shadow](http://www.css3.info/preview/text-shadow/)在webkit(Safari、Chrome等)和Gecko(Firefox)等先进的浏览器中的使用开始出现增长的势头。它们(这些CSS属性)已经为用户创建更轻量的页面和更丰富的体验，而且它们可以优雅的降级。然而，这些只是CSS 3能为我们做的众多事情中的一小部分。

在本文中，我们将走的更远，看一看 **变形(transformation)、转换(transition)和动画(animation)**等更高级的CSS3技术。我们将涉及代码本身、浏览器支持以及一些可以正确地展示这些新属性是如何即提升你的设计又增强整体用户体验的例子。

**CSS 变形(Transformation)**

CSS transformation是W3C的一个草案。但当我第一次坐下来阅读[它的全部特性](http://www.w3.org/TR/css3-3d-transforms/)以了解一些东西的时候，它并没有让我感到开窍。你可以想象的到，这个文档是用技术术语的撰写的，而且它更关注变形的图形(比如绘图)元素和矩阵。大一学习微积分之后就没有碰过矩阵了，我必须为本章节做很多好的旧浏览器测试以及猜测和检验。

在看完我能找到的每一个教程和大量的浏览器测试之后，我总结出一些大家都能从中获益的有用的关于**CSS变形**的信息。

**transform**

transform属性实现了一些可用[SVG](http://www.w3.org/Graphics/SVG/)实现的同样的功能。它可用于内联(inline)元素和块级(block)元素。它允许我们旋转、缩放和移动元素——只需要一行CSS代码。

一些前卫设计的最大诟病就是文字不可选(必须要使用切图的方法实现)。当你熟练使用transform属性来控制文字的时候，这就不再是问题了，因为这是个纯CSS方法，所以元素内的文字会保持可选。这是CSS相对于使用图片(或背景图片)的一个巨大优势。

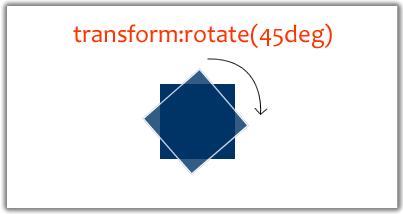
一些有趣而有用的变形功能：

* **rotate**  
  Rotate(旋转)允许你通过传递一个度数值来转动一个对象。
* **scale**  
  Scale是一个缩放功能，可以让任一元素变的更大。它使用正数和负数以及小数作为参数。
* **translate**  
  Translate就是基于X和Y 坐标重新定位元素
* **skew**  
  顾名思义，skew就是要将对象倾斜，参数是度数
* **matrix**  
  transform支持矩阵变换，就是基于X和Y 坐标重新定位元素

让我们更深入的了解每一个功能吧。

**Rotate**

transform属性有很多用法，其中一个就是**translate(旋转)**。translate就是基于角度转动一个对象并可用于内联元素和块级元素，它可以实现[很酷的效果](http://www.csskarma.com/lab/css3/text_rotation.html)。



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | #nav{  -webkit-transform: rotate(-90deg);  -moz-transform: rotate(-90deg);  -o-transform: rotate(-90deg);  filter: progid:DXImageTransform.Microsoft.BasicImage(rotation=3);  } |

请注意在IE8中(非标准模式)它需要你写成“-ms-filter”而不是“filter”。

**浏览器支持**

浏览器对translate的支持令人惊奇的广泛。在上面的CSS片段中，我们直接加上-webkit-和-moz-前缀然后将#nav元素旋转-90度。

相当简单吧？唯一的问题是对于一个相当重要的设计元素，如果IE不支持，很多设计师就会不情愿使用它。

幸运的是，**IE有这方面的滤镜**:图形旋转滤镜。它可以有4个旋转值：0, 1, 2,和3。你将不会获得和Webkit和Gecko一样的精密控制，但是至少在一定程度上保持统一(甚至IE6)。虽然这个滤镜只支持4个值，显得有些鸡肋，但是对于IE来说，聊胜于无吧。

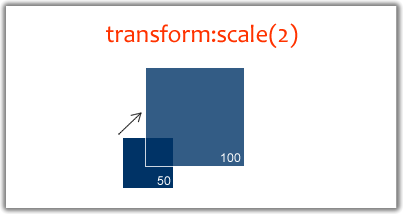
另外，非常值得一提的是，Opera在前几天发布了Opera 10.50 pre版本，声称支持CSS3 的transition和transform。只是还是需要使用私有属性，也就是要使用前缀 -o-

**scale**

scale并不像你想象的那样工作：简单的缩小和放大一个元素。缩放功能同时采用宽和高两个值，这些值可以是正数、负数和小数。

正数值放大一个元素，正如你期望的那样，基于指定的宽度和高度。

负数值并不会缩小元素，而是翻转它(比如，文字被反转)然后相应的缩放它。然而，你可以使用小于1的小数(例如.5)来收缩或者缩小一个元素。



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | #nav {  */\* nav元素的宽和高都会被放大双倍 \*/*  -webkit-transform: scale(2);  -moz-transform: scale(2);  -o-transform: scale(2); }  #nav {  */\* nav元素的宽会是双倍，而高度保持不变 \*/*  -webkit-transform: scale(2, 1);  -moz-transform: scale(2, 1);  -o-transform: scale(2, 1);  }  #nav {  */\* nav的宽度将是双倍，并且水平翻转，但是高度保持不变 \*/*  -webkit-transform: scale(-2, 1);  -moz-transform: scale(-2, 1);  -o-transform: scale(-2, 1);  } |

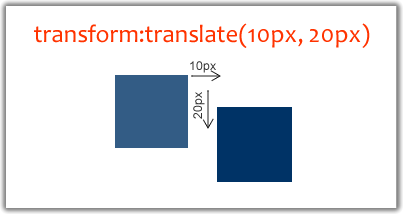
**浏览器支持**

scale属性目前只有**Firefox, Safari/Chrome以及 opera 10.50**支持，到目前为止没有一个IE版本支持。缩放一个对象是相当有意义的设计选择。它可以通过渐进增强来使用:hover，这可以在你的导航上添加一个小小的趣味。

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | #nav li a:hover{  */\* 这可以让你的导航链接在鼠标经过的时候轻微的放大 \*/*  -webkit-transform: scale(1.1);  -moz-transform: scale(1.1);  -o-transform: scale(1.1);  } |

**translate**

这个名称“translate(转化)”有点容易误解。它事实上是一种**使用X和Y坐标值定位元素**的方法。



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | #nav{  */\* 这会将nav元素向左移动10像素并向下移动20像素 \*/*  -moz-transform: translate(10px, 20px);  -webkit-transform: translate(10px, 20px);  -o-transform: translate(10px, 20px);  } |

**浏览器支持**

translate属性目前只被**Firefox, Safari/Chrome和Oprea 10.5**支持，而且还要使用浏览器的私有前缀-moz- 和-webkit-。

**Skew**

Skew也是一个很有用的transform功能，它可以将一个对象围绕着x和y轴按照一定的角度倾斜。这和rotate的旋转不一样，rotate只是旋转，而不会让元素的形状改变。skew会让一个元素的形状改变。skew有两个参数，分别代表x和y轴的倾斜**度数**。

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | #nav{  */\* 这会将nav元素向左移动10像素并向下移动20像素 \*/*  -moz-transform: skew(30deg, -10deg);  -webkit-transform: skew(30deg, -10deg);  -o-transform: skew(30deg, -10deg);  } |

**浏览器支持**

Skew属性目前只被**Firefox, Safari/Chrome和Oprea 10.5**支持，而且还要使用浏览器的私有前缀-moz- 和-webkit-。

**Matrix**

没错，Matrix就是矩阵，矩阵是高等数学中非常基础的一个东东，而在CSS 3中确实一个相当高级的功能，CSS 3引入matrix函数，可以非常灵活的实现上述的各种效果。它有6个参数(a,b,c,d,e,f)，它事实上是一个3\*3矩阵，经过该矩阵将旧的参数转换成新值：  
| a b e |  
| c d f |  
| 0 0 1 |  
如果你有兴趣深入研究，可以看一下这里<http://www.w3.org/TR/SVG/coords.html#TransformMatrixDefined>，这是SVG的一个文档，但是对于matrix变换的原理是通用的。

让我们来看一个例子吧：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | #nav{  */\* nav元素将会像右上角倾斜 \*/*  -moz-transform: matrix(1, -0.2, 0, 1, 0, 0);  -webkit-transform: matrix(1, -0.2, 0, 1, 0, 0);  -o-transform: matrix(1, -0.2, 0, 1, 0, 0);  } |

**浏览器支持**

可喜的是，IE支持[matrix滤镜](http://msdn.microsoft.com/en-us/library/ms533014(VS.85,loband).aspx)，虽然它不支持大部分CSS3变形功能，但是使用它的这个滤镜，基本也可以实现相同的效果，不过，你要先搞明白矩阵运算再说。webkit和Firefox(3.5+)、Opera 10.5都支持该功能。

**链式变形**

变形常常是单独的，但是如果你想同时用到多种变形，代码也是可以快速整合的，特别是使用私有扩展。幸运的是，我们有一些内置的缩写：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | #nav{  -moz-transform: translate(10, 25);  -webkit-transform: translate(10, 25);  -o-transform: translate(10, 25);  -moz-transform: rotate(90deg);  -webkit-transform: rotate(90deg);  -o-transform: rotate(90deg);  -moz-transform: scale(2, 1);  -webkit-transform: scale(2, 1);  -o-transform: scale(2, 1);  } |

这些变形**可以被链到一起**，从而让你的代码更高效:

|  |  |
| --- | --- |
| 1  2  3  4  5 | #nav{  -moz-transform: translate(10, 25) rotate(90deg) scale(2, 1);  -webkit-transform: translate(10, 25) rotate(90deg) scale(2, 1);  -o-transform: translate(10, 25) rotate(90deg) scale(2, 1);  } |

这些属性的真正的威力是合并到一起。你可以移动、旋转、缩放并控制任何内联元素和块级元素而不使用JavaScript。一旦这些属性的支持变的更加广泛，我们就可以创建更丰富和更轻量的界面和应用。

**transform-origin**

transform-origin不是transform的一个子功能，但是和transform关系非常密切。它可以用来指定transform的**起点**，比如，rotate变形的默认起点是其中间，你可以将起点设置在左上角，或者左下角，这样rotate变形的结果就可能大不相同了。

transform-origin接受两个参数，它们可以是百分比，em，px等具体的值，也可以是left,center,right，或者 top,middle,bottom等描述性参数。几个例子：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | .class1{-moz-transform-origin: 0 0;  -webkit-transform-origin: 0 0;  -o-transform-origin: 0 0;  ...  }  .class2{-moz-transform-origin:100% 100%;  -webkit-transform-origin:100% 100%;  -o-transform-origin:100% 100%;  ...}  .class3{-moz-transform-origin:**top** left;  -webkit-transform-origin:**top** left;  -o-transform-origin:**top** left;  ...} |

**浏览器兼容性**

该属性目前也只有webkit、firefox和Opera 10.5支持，而且需要添加各自的私有前缀。

**transition(转换)**

一个涉及到CSS属性的基本的转换就是定义和移动一个元素从它的静止状态(比如，深蓝色背景)到它的hover或者激活状态(比如，浅蓝色背景)。

转换可以和变形同时使用(以及引发诸如:hover 或:focus事件)以创建一些动画。淡出背景色彩，滑动一个元素以及让一个对象旋转都可以通过CSS转换实现。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | #nav a{ **background-color**:red; }  #nav a:hover, #nav a:focus{  **background-color**:**blue**;  */\* 告诉转换去应用到background-color 属性(看起来像一个CSS 参数，不是吗？ #foreshadowing)\*/*  -webkit-transition-property:background-color;  -o-transition-property:background-color;  */\* 让它持续两秒钟\*/*  -webkit-transition-duration:2s;  -o-transition-duration:2s;  } |

**转换的一些参数**

* **transition-property**

指定转换的CSS属性名称，比如，上面的例子中，将转换应用于background-color属性。

* **transition-duration**

定义转换花费的时间（从旧属性换到新属性花费的时间）

* **transition-timing-function**

可以理解为过度效果。指定转换过程中的中间值。可以用cubic-bezier指定。当然有几个常用的内置值：ease | linear | ease-in | ease-out | ease-in-out

* **transition-delay**

这个比较容易理解，就是转换延迟的时间。时间可以为正整数、负整数和零，非零的时候必须设置单位是s(秒)或者ms(毫秒)，为负数的时候，转换的动作会从该时间点开始显示，之前的动作被截断。

注：参数部分为[前端观察](http://www.qianduan.net)翻译时添加，原文中没有。

**浏览器支持**

像transform属性一样酷，但是目前只有**WebKit**浏览器支持。Opera 10.5 pre版本也可以通过添加-o-前缀来支持。-moz-transition已经在最近的nightly-build版本的Firefox 3.7中可用。你也可以添加-moz-transition到你的CSS中以实现将来的增强。甚至可以添加一个不用私有前缀的属性，以防万一。

**Animation**

动画是CSS 3最有用的地方。你可以将我们在上面讨论的所有的元素与动画属性比如animation-duration、animation-name 和animation-timing-function整合以创建像Flash动画一样的效果——纯CSS。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | #rotate {  **margin**: 0 auto;  **width**: 600px;  **height**: 400px;  */\* 确保我们是在一个 3-D 空间 \*/*  -webkit-transform-style: preserve-3d;  */\*让整个行使用x-轴旋转、7秒中的动画、无限次播放以及线性 \*/*  -webkit-animation-name: x-spin;  -webkit-animation-duration: 7s;  -webkit-animation-iteration-count: infinite;  -webkit-animation-timing-function: linear;  }  */\* 定义要调用的动画 \*/*  @-webkit-keyframes x-spin {  0% { -webkit-transform: rotateX(0deg); }  50% { -webkit-transform: rotateX(180deg); }  100% { -webkit-transform: rotateX(360deg); }  } |

这个梦幻的CSS动画代码和在线演示可以在[webkit](http://webkit.org/blog-files/3d-transforms/poster-circle.html)网站看到。该演示可以在任何网站看到，但是动画效果却只能在Mac os(雪豹)的[WebKit的nightly build版本](http://nightly.webkit.org/)可见。它看起来就像上面的视频中的一样，如果**你是在用mac os (雪豹版本)，**我认为安装一个webkit以亲眼看看这个效果是很值得的(它真的很酷)。Windows系统用户暂时无法欣赏这个效果。

**animation的一些参数**

animation的参数和translate有些像，所以如果你理解了translate的参数，这里就不难理解了。

* **animation-name**

动画的名称。

* **animation-duration**

定义动画播放一次需要的时间。默认为0；

* **animation-timing-function**

定义动画播放的方式，参数设置类似transition-timing-function

* **animation-delay**

定义动画开始的时间

* **animation-iteration-count**

定义循环的次数，infinite即为无限次。默认是1。

* **-webkit-animation-direction**

动画播放的方向，两个值，默认为normal，这个时候动画的每次循环都向前播放。另一个值是alternate，则第偶数次向前播放，第奇数次向反方向播放。

**浏览器支持**

不幸的是，目前，只有少数浏览器支持CSS动画。2D CSS animations 可以在Safari 4 (Leopard)、Chrome 3、Safari Mobile、Shira 以及其它Webkit 浏览器中工作。Safari 4 (Snow Leopard)支持3D动画

**总结**

现在，JavaScript可以在CSS 3完善之前弥补这个差距。遗憾的是，让所有浏览器支持这些很棒的属性可能需要一个很长的过程。不用等待那一天的到来，我们可以先使用一些严谨的**渐进增强**以及以来Javascript来增强我们的网站和应用。这不是件坏事，至少现在看起来是。

看了最近的[IE9的公告](http://www.readwriteweb.com/archives/microsoft_announces_ie9_html5_css4_javascript_performance.php)，如果IE团队添加这些属性中的一些到这个新的版本的浏览器中我不会感到惊讶，他们已经开始考虑整合CSS3了(border-radius)。

web的前景是光明的，特别是因为这些类似动画的高度实验性的属性。尽管很多属性对客户或高级产品工作还不可用，至少他们很有趣！我们都期待着有一天，我们可以支持所有平台，以建立一些真的很棒的轻量级应用。

**参考与资源**

* [A Crash Course in Advanced CSS3 Effects](http://net.tutsplus.com/videos/screencasts/a-crash-course-in-advanced-css3-effects/)  
  Net Tuts 发布的一个很酷的关于CSS3效果的视频。
* [Webkit Announces Support for CSS 3D Transforms](http://www.css3.info/webkit-announces-support-for-css-3d-transforms/)  
  CSS3.info 制作的一个使用CSS3动画的demo，目前只支持nightly build 版的WebKit.
* [Jonathan Snook on CSS Text Rotation](http://snook.ca/archives/html_and_css/css-text-rotation)  
  Jonathan Snook 讲述即将到来的CSS3 属性.
* [DEV Camp CSS3 Tricks](http://www.slideshare.net/incidentist/predevcampsf-css3-tricks-1838843)  
  一份来自Dan Kurtz的幻灯演示
* [W3C Spec on 2-D Transformations](http://www.w3.org/TR/css3-2d-transforms/)  
  关于2-D 变换的资源.
* [W3C Spec on 3-D Transformations](http://www.w3.org/TR/css3-3d-transforms/)  
  一个关于3-D 变换的资源.
* [W3C Spec on CSS3 Animations](http://www.w3.org/TR/css3-animations/)  
  CSS animations的W3C工作草案
* [Safari CSS参考](http://developer.apple.com/mac/library/documentation/AppleApplications/Reference/SafariCSSRef/Articles/StandardCSSProperties.html)
* [MDC:Using CSS transforms](https://developer.mozilla.org/En/CSS/Using_CSS_transforms)
* [MDC: -moz-transform](https://developer.mozilla.org/En/CSS/-moz-transform)

**关于原作者**

Tim Wright是一位网页设计师/开发工程师和博主。他从2004年就成为一名web标准和易用性的倡导者。你可以在[CSSKarma.com](http://www.csskarma.com/)上找到他写的更多文章，或者[follow Tim on Twitter](http://www.twitter.com/csskarma)。