尽管人们期望在屏幕上有些改变，但是**CSS**和**HTML**对页面中的交互能做的实在太少了，而那些还需要用代码来实现。

比如一个链接要么是这个颜色，要么是那个颜色；一个文本区域要么这么大，要么那么大；一张图片要么是透明的要么是不透明的；它们是从一个状态直接变到另一个状态——**中间并没有过渡**。

这导致大部分网页有些生硬，因为元素只会很死板的切换或改变。

是的，你可以使用DHTML、jQuery或者自己编写JS来实现过度，但是这需要更多的代码来实现本应该非常简单的功能。

我们需要的是快速而简单的方法来给页面添加简单的变换(transition)效果。在本文中，你会发现很有用的关于**CSS变换(transition)**以及如何使用它们的信息。

几个月之前，我就[建议设计师应该开始使用新的CSS 3 技术](http://www.webdesignerdepot.com/2009/08/5-css3-design-enhancements-that-you-can-use-today/)来实现一些它们渴求已久的基本的功能了——唯一的问题就是，这些技术没有一个能在IE中可用，包括IE8。

一些读者认为的那些技术将会有75%的用户看不到的观点是不靠谱的。

对那些读者我想说，“坐稳了”，因为我将要向你介绍另一个新的CSS属性，它允许你通过简单的几行代码来为任意元素添加很酷的变换效果。

CSS 变换刚刚在CSS 3中被引入，但是已经被添加为webkit的扩展了。也就是说，现在它们只能用于基于webkit内核的浏览器，包括Apple Safari和Google Chrome。不过从Opera 10.5 pre-Alpha版本来看，Opera将在下一个10.5中支持CSS 3变换。所以**要看到本文中提到的实际效果，强烈建议你使用Chrome或者Safari 4来查看本文**。

**CSS变换从哪里来？**

变换曾经只是Webkit的一部分，而且是Safari UI能做而其它浏览器不能实现的一些很酷的东东的基础。

但是W3C CSS工作组曾经拒绝将变换添加到它的官方特性里面，一些成员坚持认为变换并非CSS属性而通过脚本来处理会更好。(内牛满面啊，我前一段也有类似的观点，也和[高温](http://gaowhen.com)讨论过，我认为CSS动画超出了表现的范围，交互的东西应该有脚本来实现，不过后来在[鬼哥](http://www.cssforest.org)的点拨下，才开始有了新的认识——神飞)

但是很多设计师和开发人员，包括我自己，坚持认为这确实是样式——只是**动态**样式，而不是我们日常使用的传统的静态样式。

幸运的是，关于动态样式的争论已经成为过去。去年三月份，来自Apple和Mozilla的代表们开始[将CSS变换模块添加到CSS 3特性里面](http://www.w3.org/TR/css3-transitions/)，非常接近Apple已经添加到webkit中的表现。

**关于设计增强的一个简要介绍**

在我们继续之前，让我强调一点：永远不要让网站的功能依赖样式，如果该样式不是浏览器通用的话(也就是说，所有的常用浏览器都支持)。

对错过的同学再一次强调：**永远不要让网站的功能依赖样式，如果该样式不是浏览器通用的话**。

这也就是说，你可以使用样式，比如变换，作为设计增强以提高用户体验——在不牺牲看不到它们的用户的可用性的前提下。如果你不用CSS变换照样能用而且用户依然能够完成他们的任务，就没问题，你就可以使用CSS变换。

**首先，一些变换的想法**

CSS变换将不会替代所有的DHTML，不过它会提供一些支持过渡的方法来提高您在浏览器中的设计。

你需要到下载[Apple Safari 3+](http://www.apple.com/safari/) 或[Google Chrome](http://www.google.com/chrome)浏览器来查看这些变换效果。这两个浏览器都支持Mac和PC系统。

**滚动效果**

变换最常见的用法就是当用户的鼠标悬放到元素上的时候将元素高亮(无论是链接、表格、表单还是其它的什么元素)，变换是为页面添加平滑的界面的非常棒的方法。

**例一**

[Link](http://www.qianduan.net/css-transitions-101.html)

**下拉菜单**

纯CSS菜单比较容易实现，而变换还可以让你给菜单添加下拉和高亮效果。

**例二**

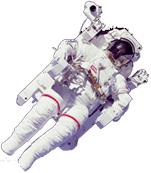
* About Us

**动画**

你可以在页面上的两点之间移动一个对象，然后使用变换给它添加动画效果。

**例三**

点击一下，不要放开!



**变换、状态与动作**

但是请稍等一下。在深入了解变换之前，我们需要理解一个元素能变换的不同的状态。

状态定义当前页面中相应的元素如何与用户进行交互，它们在CSS中通过伪类来定义，比如当用户的鼠标经过一个元素的时候，那个元素就会被hover伪类控制。

|  |  |  |
| --- | --- | --- |
| **动态伪类** | **起作用的元素** | **描述** |
| :link | 只有链接 | 未访问的链接 |
| :visited | 只有链接 | 访问过的链接 |
| :hover | 所有元素 | 鼠标经过元素 |
| :active | 所有元素 | 鼠标点击元素 |
| :focus | 所有可被选中的元素 | 元素被选中 |
| None | 所有元素 | 所有元素的默认状态 |

变换通过改变不同元素状态之间的一个时间段内的样式来起作用。比如，一个元素的默认状态的颜色值将会在呈现hover状态的色彩值之前逐渐显示色盘中的中间颜色。

**一个简单的变换**

让我们假设一个简单的变换，在用户的鼠标经过一个链接的时候，将颜色从一个变换成另外一个。与其它CSS属性一样，变换也是直接添加到要使用它的选择器中。该属性可以采用下面的4个值。

**CSS property**

被变换的属性(比如, color)。看一下下面的表格了解所有可以被变换的CSS属性列表。

**Duration**

变换持续的时间，通常以秒来计算(比如, .25s).

**Timing function**

允许你控制持续的时间的计算方式。与其使用一个简单的线性计算，你可以使变换加速(渐入)或者减速(淡出)，或者甚至specify a beat or count (比如，linear). More on this later in the article.

**Delay**

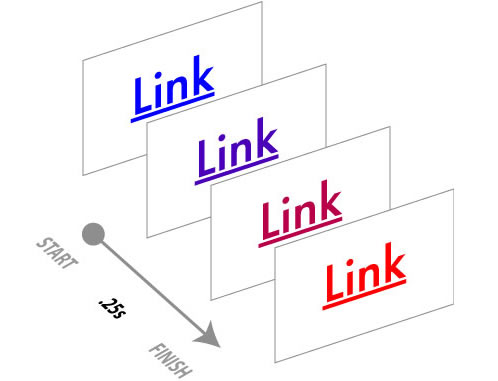
在动作和变换开始之间等待多久，通常用秒来表示(比如, .1s)。如果你不想延迟，该值可省略。

因为变换属性始于Webkit扩展，我们不得不同时使用transition 和-webkit-transition 属性以向后兼容。

让我们首先提阿贾这这些属性到:hover 伪类中:

|  |  |
| --- | --- |
| 1  2  3  4  5 | a:hover {  **color**: red;  -webkit-transition: **color** .25s linear;  transition: **color** .25s linear;  } |

那么现在，当鼠标经过一个链接，不会直接从蓝色跳转到红色，而是用四分之一秒的时间逐渐变换它们的中间颜色(过渡颜色)。



当然，我们也希望变换回到默认的链接颜色，那么我们可以添加一个变换到:link (以及:visited)伪类上，并在它褪去之前添加一个简单的延迟(十分之一秒) :

|  |  |
| --- | --- |
| 1  2  3  4  5 | a:link, a:visited {  **color**: **blue**;  -webkit-transition: **color** .25s linear .1s;  transition: **color** .25s linear .1s;  } |

**添加多重变换**

因为一个变换就是一个CSS属性，如果你在同一个CSS规则中添加多个变换的实例，那么最后的那个将会覆盖前面的，而不是添加它们。所以在下面的规则中，唯一的变换将是背景色彩：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | a:hover {  **color**: red;  **background-color**: rgb(235,235,185);  -webkit-transition: **color** .25s linear;  transition: **color** .25s linear;  -webkit-transition: **background-color** .15s linear .1;  transition: **background-color** .15s linear .1;  } |

当然，这并不是说，不能添加多重变换——多重变换可以在同一个变换属性定义中用逗号隔开：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | a:hover {  **color**: red;  **background-color**: rgb(235,235,185);  -webkit-transition: **color** .25s linear, **background-color** .15s linear .1s;  transition: **color** .25s linear, **background-color** .15s linear .1s;  } |

这条定义将产生色彩和背景色的双重变换。

**什么可以被变换？**

几乎所有的有色彩、大小或位置等组件的CSS属性，包括许多新添加的CSS 3属性， 都可以应用变换。一个值得注意的例外是box-shadow。

来自W3C的变换的说明，这里是一个可以赋予变换的CSS属性的列表，附带转变的对象，我也高亮了一些比较有用的属性。

|  |  |
| --- | --- |
| **CSS属性** | **改变的对象** |
| **background-color** | **色彩** |
| background-image | 只是渐变 |
| background-position | 百分比，长度 |
| border-bottom-color | 色彩 |
| border-bottom-width | 长度 |
| **border-color** | **色彩** |
| border-left-color | 色彩 |
| border-left-width | 长度 |
| border-right-color | 色彩 |
| border-right-width | 长度 |
| border-spacing | 长度 |
| border-top-color | 色彩 |
| border-top-width | 长度 |
| border-width | 长度 |
| **bottom** | **百分比，长度** |
| **color** | **色彩** |
| crop | 百分比 |
| **font-size** | **百分比，长度** |
| **font-weight** | **数字** |
| grid-\* | 数量 |
| **height** | **百分比，长度** |
| **left** | **百分比，长度** |
| **letter-spacing** | **长度** |
| line-height | 百分比，长度，数字 |
| margin-bottom | 长度 |
| margin-left | 长度 |
| margin-right | 长度 |
| margin-top | 长度 |
| max-height | 百分比，长度 |
| max-width | 百分比，长度 |
| min-height | 百分比，长度 |
| min-width | 百分比，长度 |
| **opacity** | **数字** |
| outline-color | 色彩 |
| outline-offset | 整数 |
| outline-width | 长度 |
| padding-bottom | 长度 |
| padding-left | 长度 |
| padding-right | 长度 |
| padding-top | 长度 |
| **right** | **百分比，长度** |
| text-indent | 百分比，长度 |
| **text-shadow** | **阴影** |
| **top** | **百分比，长度** |
| vertical-align | 百分比，长度，关键词 |
| visibility | 可见性 |
| **width** | **百分比，长度** |
| word-spacing | 百分比，长度 |
| z-index | 正整数 |
| zoom | 数字 |

**变换计时与延迟**

使用变换，你可以改变变换的速率，在开始的时候较慢然后在结束的时候加速，反之亦然，或者之间的任何事情。CSS变换有5个计时的关键词，同时也允许你自己定义你自己的计时曲线。

|  |  |
| --- | --- |
| **名称** | **如何工作** |
| cubic-bezier(x1, y1, x2, y2) | X 和 Y 值在0到1之间，以定义用于Time function的贝塞尔曲线的形状。 |
| linear | 均速 |
| ease | 逐渐慢下来 |
| ease-in | 加速(渐入) |
| ease-out | 减速(淡出) |
| ease-in-out | 加速然后减速 |

**全部变换？**

变换将很快成为所有网站的标准操作方式，从而增强用户界面的体验。

为了给你的整个网站添加一个普遍的变换，一个做法就是添加一个变换到全体选择器，类似CSS reset。下面的代码会给页面中的所有元素添加一个默认的变换，从而允许你保持一个统一的变换效果：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | \*:link, \*:visited, \*:hover, \*:active, \*:focus {  -webkit-transition:  **color** .25s linear,  **background-color** .25s linear,  **border-color** .25s linear;  transition:  **color** .25s linear,  **background-color** .25s linear,  **border-color** .25s linear;  } |

一个反对全部变换，同时明确反对使用全体选择器作为CSS reset的争论是，将一个样式用到页面的所有元素会减缓页面的渲染。然而，我并没有发现任何有关与此的证据。有人知道吗？

OK，其实CSS 变换就这么简单，它并不难理解，而且你也不用些大量的JS脚本来实现它，这很方便实用，多做几次练习，你就可以熟练的使用它了。

译注：本文原文题目为 CSS transitions 101，101这个数字比较不好理解，其实美国大学第一门课程通常编号为101，所以101一般表示入门、初级的意思。另外关于 transition 这个词的翻译，我之前是翻译成“转换”，但是这个翻译很别扭，在twitter上和[@gaowhen](http://twitter.com/gaowhen)、[@ghostzhang](http://twitter.com/ghostzhang)、[@ivane](http://twitter.com/ivane)、[@hiwanz](http://twitter.com/hiwanz)、[@cnjoel](http://twitter.com/cnjoel)等人讨论后，觉得翻译成“变换”更合适一些，多谢各位。——神飞