# 第十二章 CSS3过渡

多年来，web前端开发人员一直在寻求通过HTML和CSS实现一些动画交互效果，而不再使用JavaScript或Flash。现在他们的愿望实现了。CSS3除了给我们带来前面介绍的一些特殊功能模块之外，还为Web设计师添加了一些动画功能模块。

可以通过:hover、:focus、:active、:checked或者JavaScript触发一个元素，这样，外观变化会显得更细腻，而不会让人感觉“一触即发”。例如悬浮修改链接色，在CSS2.1中鼠标悬浮时，立即从一个颜色变成另一个颜色。而在CSS3中使用过渡功能，鼠标悬浮时，颜色在一定的时间内，从一个颜色过渡到另一个颜色，给用户更好、更细腻的体验。

本章主要介绍如何使用CSS3的过渡属性，并帮助读者进一步掌握CSS3中transition的使用方法，实现属性值的开始值与属性值的结束值之间平滑过渡的动画。

## 12.1 CSS3过渡简介

CSS3的过渡功能像是一种黄油，通过一些CSS的简单动作触发样式平滑过渡。

W3C标准中描述的transition功能很简单：CSS3的transition允许CSS的属性值在一定的时间区间内平滑地过渡。这种效果可以在鼠标单击、获取焦点、被点击或对元素任何改变中触发，并平滑地以动画效果改变CSS的属性值。

### 12.1.1 如何创建简单的过渡

以往web中的动画都是依赖于JavaScript和flash实现，但原生CSS过渡在客户端上需要处理的资源少得多，从而显得更加平滑。

CSS3过渡与元素上的常规样式一起声明。只要目标属性更改，浏览器就会应用过渡。除了使用JavaScript触发动作外，在CSS中也可以通过一些伪类来触发，如:hover、:focus、:active、target和:checked等。这很重要：无须在JavaScript中编写动画，只需要更改一个属性值并依赖浏览器来执行所有重要工作。

以下是使用CSS创建简单过渡的步骤。

1. 在默认样式中声明元素的初始状态样式。
2. 声明过渡元素最终样式，比如悬浮状态。
3. 在默认样式中通过添加过渡函数，添加一些不同的样式。

### 12.1.3 CSS3过渡属性

Transition属性是一个复合属性，出于简洁性和便于维护考虑，过渡语法通常以简化的形式表达。

Transition:[<’transition-property’>]||<’transition-duration’>||<’transition-timing-function’>||<’transition-delay’>][,<’transition-property’>]||<’transition-duration’>||<’transition-timing-function’>||<’transition-delay’>]

Transition属性主要包含四个属性值。

1. Transition-property：指定过渡或动态模拟的CSS属性
2. Transition-duration：指定完成过渡所需的时间。
3. Transition-timing-function：指定过渡函数。
4. Transition-delay：指定过渡开始出现的延迟时间。

如果单独声明transition属性会变得非常密集，特别是在添加各个浏览器的私有前缀的时候。值得庆幸的是，transition属性可以像border、margin、padding和font这样的属性，将上面介绍的trainsition的四个子属性（transition-property、transition-duration、transition-timing-function和transition-delay）简写在一起。需要注意的是，这四个子属性之间不能使用逗号分隔，而是使用空格分隔。Transition属性的简写语法如下。

Transition:<single-transition>[‘,’<single-transition>\*]

其中<single-transition>等于：

Single-transition=[transition-property]||[transition-duration]||[transition-timing-function]||[transition-delay]

值得注意的一点是，transition属性简写中，transition-duration和transition-delay取值都是<time>，所以在简写中要区分它们。一般浏览器会根据先后顺序决定：第一个<time>会解析为transition-duration，第二个<time>会解析为transition-delay。

例如上面的实例，可以按transition简写的语法规则如下形式：

.transition{

background: #8ec63f;

width: 100px;

height: 100px;

border: 0.5em;

-webkit-transition: background 2s linear 2s ,border-radius 3s ease-in 4s;

-moz-transition: background 2s linear 2s ,border-radius 3s ease-in 4s;

-o-transition: background 2s linear 2s ,border-radius 3s ease-in 4s;

-ms-transition: background 2s linear 2s ,border-radius 3s ease-in 4s;

transition: background 2s linear 2s ,border-radius 3s ease-in 4s;

}

.transition:hover{

background: #f7941d;

border-radius: 50%;

}

综合上述，可以给transition一个速记法。

Transition：<property><duration><animation type><delay>

也就是：

Transition:<过渡属性><过渡所需时间><过渡动画函数><过渡延迟时间>

## 12.2 CSS3过渡子属性详解

在transition属性中包括四个子属性：transition-property、transition-duration、transition-timing-function和transition-delay。在使用transition时不是每个子属性都使用，其中前三个子属性是最受欢迎的。

在下面的例子中，鼠标悬浮在盒子上时，其背景色在1秒内以线性的方式从“#8ec63f”色过渡到“#f7941d”色。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

<style>

.transition{

background:#8ec63f;

width:100px;

height: 100px;

border-radius: .5em;

transition-property: background;

transition-duration: 1s;

transition-timing-function: linear;

}

.transition:hover{

background: #f7941d;

}

</style>

</head>

<body>

<div class="transition">

</div>

</body>

</html>

鼠标悬浮在盒子上时，盒子背景色慢慢过渡到另一个色。

先从简单的例子中体验一下transition的效果，接下来针对每个子属性做一定的介绍。

### 12.2.1 指定过渡属性transition-property

要让transition属性能正常工作，需要给元素设置两套样式用于用户与页面的交互。在transition中主要通过子属性transition-property来完成。简单来说，就是通过transition-property属性来指定过渡动画的CSS属性名称。该属性的基本语法如下。

Transition-property:none | all | <single-transition-property>[,<single-transition-property>]\*

取值简单说明如下。

1. None：没有指定任何样式。
2. All:默认值，表示指定元素所有支持transition-property属性的样式。
3. <single-transition-property>:指定样式，其等于all或者<IDENT>

需要特别注意，用transition-property来指定过渡属性并不是所有属性都可以过渡，只有属性具有一个中点值的属性才能具备过渡效果，其对应的类型属性主要如下。

1. 颜色属性：通过红、绿、蓝和透明度组合的过渡（每个数值处理），如background-color、border-color、color和outline-color等样式属性。
2. 具有长度值（length）、百分比（percentage）的属性：真实的数字，如word-spacing、width、vertical-align、top、right、bottom、left、min-width、min-height、max-width、max-height、line-height、height、background-position等。
3. Interger：离散步骤（整个数字），在真实的数字空间，以及使用floor（转换为整数时发生），如outline-offset、z-index等。
4. Number真实的数值：如zoom、opacity、font-weight等。
5. 变形系列属性：如rotate()、rotate3d()、scale()、scale3d()、skew()、translate()、translate3d()等。

此外还有一些其他属性，这里就不一一介绍。

在设置过渡属性时，不限于单一的一套样式转换，可以同时设置过个过渡样式，只是在样式之间需要使用逗号隔开。基于上面的实例，来做一个变化。

当鼠标悬浮在盒子上时，盒子先从高度100px过渡到50px，然后宽度从50px过渡到300px，同时盒子背景色从“#8ec63f”色过渡到“#f7941d”色。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

<style>

.transition{

background:#8ec63f;

width:100px;

height: 100px;

border-radius: .5em;

transition-property: background,width,height;

transition-duration: 1s,1s,1s;

transition-timing-function: linear,linear,linear;

transition-delay: 0s,1s,2s;

}

.transition:hover{

background: #f7941d;

width:300px;

height:50px;

}

</style>

</head>

<body>

<div class="transition">

</div>

</body>

</html>

### 12.2.2 指定过渡所需时间transition-duration

Transition-duration属性主要用来设置一个属性过渡到另一个属性所需要的时间，即从旧属性过渡到新属性花费的时间。该属性的基本语法如下。

Transition-duration:<time>[,<time>]\*

<time>为数值，单位为s（秒）或ms（毫秒）。可以作用于所有元素，包括:before和:after伪元素。其默认值为0，也就是变换是即使的。换句话说，当transition-duration取值为0时，指定元素样式过渡时，看不到过渡效果，直接看到结果。

和transition-property属性一样，设置多个过渡属性时，也可以设置多个transition-duration，每个值之间同样使用逗号分隔。而且每个值按顺序对应transition-property的属性值。基于上例，稍作修改，将正方形过渡到圆形。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

<style>

.transition{

background:#8ec63f;

width:100px;

height: 100px;

border-radius: .5em;

transition-property: background,border-radius;

transition-duration: .5s,1s;

transition-timing-function: linear,ease-in;

transition-delay: 1s;

}

.transition:hover{

background: #f7941d;

border-radius: 50%;

}

</style>

</head>

<body>

<div class="transition">

</div>

</body>

</html>

### 12.2.3 指定过渡函数transition-timing-function

Transition-timing-function属性指定某种指代过渡“缓动函数”的属性。此属性可以指定浏览器的过渡速度，以及过渡期间的操作进展情况，可以将某个值定为预定义函数、阶梯函数或者三次贝塞尔曲线。该属性的基本语法如下。

Transition-timing-function：<single-transition-time-function>[‘,’<single-transition-timing-function>]

1. 单一过渡函数

<single-transition-timing-function>指单一的过渡函数，主要包括下表所示几个。

|  |  |
| --- | --- |
| 函数 | 功能描述 |
| Ease | 默认值，元素样式从初始状态过渡到终止状态时速度由快变慢，逐渐变慢 |
| Linear | 元素样式从初始状态过渡到终止状态速度是恒速 |
| Ease-in | 元素样式从初始状态过渡到终止状态时，速度越来越快，呈一种加速状态。这种效果称为渐显效果。 |
| Ease-out | 元素样式从初始状态过渡到终止状态时，速度越来越慢，呈一种减速状态。这种效果称为渐隐效果 |
| Ease-in-out | 元素样式从初始状态到终止状态时，先加速再减速。这种效果称为渐显渐隐效果 |

2.三次贝赛尔曲线

到目前为止，看到的ease、linear、ease-in、ease-out和ease-in-out等曲线函数非常一般，用于过渡动画中也不是十分精确。而现在制作一些动画需求越来越精确，需要定义一些更精确的函数。

CSS3的transition中，定义三次贝塞尔曲线的语法如下。

Cubic-bezier(P0,P1,P2,P3)

注意，三次贝塞尔曲线中的每个点值只允许0~1的值。

3.steps()函数

Steps()函数用于把整个操作领域划成同样大小的间隔，每个间隔都是对等的。该函数还指定发生在开始或结束的时间间隔是否另外输出百分比。看看来自w3c官网对steps()函数的剖析图。

Steps()函数非常独特，它允许在固定的间隔播放动画。例如，在steps()函数上可以看出，动画属性比在0%处开始，涨到50%，动画结束时，属性值达到100%。并且在各个steps()函数中每个函数的过渡动画不够平滑。

该函数语法如下。

Steps()

Steps()函数主要包括两个参数。

1. 第一个参数是一个数值<integer>,主要用来指定step()函数间隔的数量，此值必须是一个大于0的正整数。
2. 第二个值是可选的，是start或end，如果第二个参数忽略，则默认为end值。

当使用多个过渡属性transition-property时，可以为每个过渡属性指定对应的函数，当指定多个过渡函数时，需要为用逗号将它们分隔开。如果有多个过渡属性，但只指定一个过渡函数时，这个过渡函数将应用于所有的过渡属性。

### 12.2.4 指定过渡延迟时间transition-delay

前面介绍了三个属性，过渡属性transition还有一个过渡延迟时间属性transition-delay，用来定义过渡延迟时间。该属性的基本语法如下。

Transition-delay:<time>[,<time>]\*

Transition-delay用来指定一个动画开始执行的时间，也就是说当改变元素属性值后多长时间开始执行过渡效果，其取值为<time>,它可以是正整数、负整数和零，非零的时候必须将单位设置为s（秒）或者ms（微妙）。

1. 正整数：元素的过渡动作不会立即触发，当过了设置的时间值之后才触发。
2. 负整数：元素的过渡动作会从该时间点开始显示，之前的动作被截断。
3. 0：元素的过渡动作会立即触发，没有任何延迟。在默认情况下，transition-delay取值为0。

Transition-delay属性和transition其他子属性一样，可以同时设置多个属性，也是使用逗号来分隔多个属性。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

<style>

.transition{

background:#8ec63f;

width:100px;

height: 100px;

border-radius: .5em;

transition-property: background,border-radius;

transition-duration: 2s,3s;

transition-timing-function: linear,ease-in;

transition-delay: 2s,4s;

}

.transition:hover{

background: #f7941d;

border-radius: 50%;

}

</style>

</head>

<body>

<div class="transition">

</div>

</body>

</html>

### 12.2.5 多个CSS3过渡效果

很多时候，不只想改变一个CSS的属性效果，还想改变两个或者多个CSS属性效果。这在transition过渡中并不是很复杂的一件事情。可以使用上面讨论的transition-property普通书写语法，用逗号分隔要过渡的各个CSS属性，从而声明多个过渡。反过来，其他普通书写值允许包含逗号分隔值，其对应于transition-property值。

简写语法进一步简化此操作，能够同时声明多项属性过渡，只需逗号分隔声明。

.transition{

Transition:background 2s linear 2s,border-radius 3s ease-in 4s;

}

## 12.3 CSS3触发过渡

掌握了CSS3的过渡transition语法，在实际应用中相当如何触发过渡呢？上面所述的示例定义了过渡操作的各个层面，但却没有提及触发过渡的时间和方法。因此，就目前的状况而言，单纯通过代码不会触发任何过渡操作。这一点和javascript中的过渡效果类似，需要通过用户的行为（如点击、悬浮）触发。

### 12.3.1 伪元素触发

使用transition时，常用鼠标悬浮（:hover）来触发过渡。加上“:hover”触发器后，代码如下所示。

.transition{

Background-color:green;

Width:100px;

Height:100px;

Transition:width 2s ease .5s,

Background-color 2s ease .5s,

Height 2s ease .5s;

}

.transition:hover{

Background-color:orange;

Width:200px;

Height:50px;

}

有了这个代码，当用户将鼠标悬浮在该元素上时，元素的宽度、背景色和高度会在经过0.5秒的初始延迟后，与2秒内动态从100px宽度变成200px，100px高变成50px，背景色从绿色变成橙色。

事实上，触发过渡与触发器本身没有太大关系。实际触发过渡的是元素状态变化。

在上面的示例中，状态变化是指宽度由窄边框、由高变矮，同时绿色背景颜色变为橙色背景颜色。凑巧的是，状态变化作为“:hover”事件的结果发生。

为了说明这与触发过渡的实际事件毫无关联，也为强调状态变化的重要性，下面提供了一些触发过渡的其他方法。

1. 使用:active

“:active”伪类表示用户单击某个元素并按住鼠标按钮时显示的状态。在以下示例中，当用户单击并按住元素时，发生宽度、高度和背景色过渡，因此该元素保持“活动”状态。

1. 使用:focus

“:focus”伪类通常会在元素接受键盘焦点时出现。文本输入框元素上将发生过渡，并且该元素得到焦点时会执行文本输入框元素宽度和背景色过渡。

3. 使用:checked

“:checked”伪类在发生以下状况时触发过渡。文本框选中时，“span”元素宽度发生过渡。

12.3.2 媒体查询触发

触发元素状态变化的另一种方法是使用CSS3媒体查询。如果学习过媒体查询，那么一定知道它能够根据某些元素更改应用于元素的样式。媒体查询过渡如下所示。

.transition{

background:#8ec63f;

width:100px;

height: 100px;

border-radius: .5em;

transition-property: background,border-radius;

transition-duration: 2s,3s;

transition-timing-function: linear,ease-in;

transition-delay: 0s,2s;

}

@media only screen and (max-width:960px ) {

.transition{

background: #f7941d;

border-radius: 50%;

}

}

如果网页加载时用户的窗口大小是960px或以下，浏览器会在该部分应用这些样式，但是由于不会出现状态变化，因此不会发生过渡。

只要更改的是动画属性，就会发生过渡。

12.3.3 javaScript触发

如果可以基于CSS的状态更改触发过渡，自然可以通过JavaScript做到这一点。在下面的示例中，使用纯粹CSS示例时同样也会发生过渡。也就是说，它是CSS状态变化的结果，但这次是通过JavaScript触发的过渡。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

<style>

.box{

background-color: #CCC;

width:200px;

height:200px;

transition: width 1s linear 0s,height 1s linear 0s;

}

.box.on{

width:300px;

height:300px;

}

</style>

</head>

<body>

<div class="box"></div>

<button>click</button>

<script type="text/javascript" src="jquery-1.7.2.min.js" ></script>

<script>

$("button").click(function(){

$(".box").toggleClass("on");

})

</script>

</body>

</html>

在考虑是否要使用CSS过渡替换JavaScript时需要牢记：事件通常应用JavaScript触发，简单动画或过渡则应使用CSS触发。当然，这只是一般性的指导原则，不一定是最佳选择，具体应用视条件而定。

## 12.4 CSS3过渡技巧

现在，已经对过渡语法及其使用有了很好的掌握，接下来介绍transition中一些技巧，或许在将来的项目中可以排上用场。

### 12.4.1 一个完整的过渡

CSS3过渡transition是异步运行的，该规范提供了TransitionEnd事件，允许JavaScript来同步一种过渡的结束属性。遗憾的是，该规范并没有很具体地阐述这个事件。事实上，它只是说明每个过渡属性的事件状态。如果用一个词来描述这种情况，你的“噩梦”就要开始了。

Transition规范中说transition运行简写属性（如padding），会覆盖所有属性（padding-top、padding-right等），但并没有说TransitionEnd事件是哪个属性。在Gecko、Trident和Presto内核浏览器中，即使过渡指定的是简写属性（例如padding），但事件触发的普通写法的子属性，可是在Webkit中会抓住机会将事情搞砸。如果指定transition-property:padding,webkit将触发一个padding事件，但指定transition-property:all时，Webkit将触发的事件为padding-left等。因为某些原因，当指定的过渡属性为padding时，在iPhone的Safari6.0.1浏览器中除了触发padding属性之外还会触发font-size和line-height属性。

.example{

Padding:1px;

Transition-property:padding;

Transition-duration:1s;

}

.example:hover{

Padding:10px;

}

上面的CSS在不同的浏览器中将触发不同的TransitionEnd事件。

A.在Gecko、Trident和Presto浏览器中触发padding-top、padding-right、padding-bottom和padding-left事件。

B. 在Webkit浏览器中触发padding事件。

看另外一个例子。

.example{

Padding:1px;

Transition-property:all,padding;

Transition-duration:1s;

}

.example:hover{

Padding:10px;

}

上面的CSS在不同的浏览器中将触发不同的TransitionEnd事件。

A.在Gecko、Trident和presto和webKit内核浏览器中触发padding-top、padding-right、padding-bottom和padding-left事件。

B.在iphone的Safari6.0.1浏览器中将触发padding-top、padding-right、padding-bottom、padding-left、font-size和line-height属性。

### 12.4.2 可过渡的属性

规范中列出了很多能让浏览器支持动画过渡行为的CSS属性。正如前面介绍transition-property属性时所列的属性清单。但这个列表包含的属性都是CSS2.1版本的CSS属性。任何更新的、可支持动画过渡行为的属性都没有列入规范中，如伸缩性盒模型中的order属性。

属性值的类型是一个很重要的因素。例如，margin-top同时可以接受<length>和<percentage>的属性值，但根据可过渡的CSS属性列表，只有<length>值才具有动画效果。但是无论如何也没有办法让浏览器厂商使用transition支持取值为<percentage>的CSS属性具有动画效果。该规范包括了<percentage>的值，但在撰写本书的时候，还没有浏览器能够支持它的过渡动画。

### 12.4.3 优先的过渡属性

该规范在定义transition-property属性状态时，允许同时给一个属性定义多次。所以，可以定义padding的过渡属性为1秒，同时可以定义padding-left的过渡需要2秒；或者通过transition-property:all来定义一个默认的过渡风格，或者用来覆盖一些特定的过渡属性。

在Filefox和IE浏览器中，这个工作做得很好。但在Opera浏览器下有一个优先顺序，而不是简单使用多个过渡列表中的最后一个试用属性，其中padding-left属性优先于padding和all。

真正麻烦的是在Webkit内核的浏览器下，如果一个属性设置了多次过渡，那么不知道怎么多次执行一个过渡。

### 12.4.4 过渡的开始和结束为auto

在CSS中，有些属性可以取值为auto。当属性为auto时，浏览器会自动计算出一个合理的值。例如段落“<p>”或者任何块级元素，如果设置了width：auto时，它们的宽度将等于它们的父元素宽度。有时候可能会通过transition属性将width:auto值过渡到一个指定的宽度值。在可过渡属性的规范中，auto值既不会执行也不会说不执行。

在Filefox、IE和Opera浏览器下过渡不会执行属性值为auto的属性。另一方面，在Webkit内核浏览器下几乎所有具有属性值为auto的属性都可以过渡。

### 12.4.5 隐式过渡

“隐式过渡”指的是，当一个属性改变时引起另一个属性到一个属性的过渡。

### 12.4.6 开关状态的不同过渡方式

在下面的代码示例中，开关状态均会导致背景颜色的过渡。

.example{

Background-color:blue;

Transition:background-color 2s linear 1s;

}

.example:hover{

Background-color:green;

}

更改关闭状态的持续时间和技术函数，如下所示。

.example{

Background-color:blue;

Transition:background-color 1s ease-out;

}

.example:hover{

Background-color:green;

Transition:background-color 2s linear;

}

## 12.5 综合案例：纯CSS3制作CSS Dock 导航效果