### 1.1 基本用法

在CSS 3引入Transition（过渡）这个概念之前，CSS是没有时间轴的。也就是说，所有的状态变化，都是即时完成。

**transition.html**

上面是一个演示，当鼠标放置于缩略图之上，缩略图会迅速变大。注意，缩略图的变大是瞬间实现的。下面是代码，相当简单。

img{

height:15px;

width:15px;

}

img:hover{

height: 450px;

width: 450px;

}

transition的作用在于，指定状态变化所需要的时间。

img{

transition: 1s;

}

我们还可以指定transition适用的属性，比如只适用于height。

img{

transition: 1s height;

}

### 1.2 transition-delay

在同一行transition语句中，可以分别指定多个属性。

img{

transition: 1s height, 1s width;

}

但是，这样一来，height和width的变化是同时进行的，跟不指定它们没有差别.

我们希望，让height先发生变化，等结束以后，再让width发生变化。实现这一点很容易，就是为width指定一个delay参数。

img{

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

}

delay的真正意义在于，它指定了动画发生的顺序，使得多个不同的transition可以连在一起，形成复杂效果。

### 1.3 transition-timing-function

transition的状态变化速度（又称timing function），默认不是匀速的，而是逐渐放慢，这叫做ease。

**transition-timing-function.html**

img{

transition: 1s ease;

}

除了ease以外，其他模式还包括

（1）linear：匀速（等于 cubic-bezier(0,0,1,1)）。

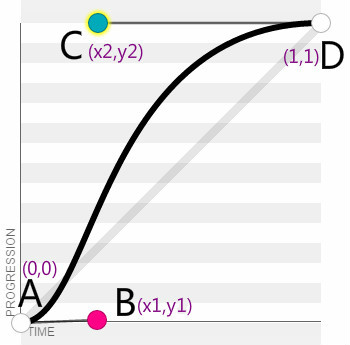
（2）ease：开始和结束慢，中间快（等于cubic-bezier(0.25,0.1,0.25,1)）。

（3）ease-in：加速（等于 cubic-bezier(0.42,0,1,1)）

（4）ease-out：减速（等于 cubic-bezier(0,0,0.58,1)）。

（5）ease-in-out：和ease类似，但比ease幅度大（等于 cubic-bezier(0.42,0,0.58,1)）。

（6）cubic-bezier函数：自定义速度模式。可能的值是 0 至 1 之间的数值。



**cubic-bezier (x1,y1,x2,y2)**

分别用A,B,C,D表示这四个点，其中起始点固定值为**A(0,0)**,终止点固定为**D(1,1)**剩下的中间点**B(x1,y1),C(x2,y2)**也就是所要动态操控的两个点了,对应**cubic-bezier (x1,y1,x2,y2)**中的四个参数,通过改变B,C两点的坐标值来动态生成一条贝塞尔曲线表示动画中的速度变化。

最后那个cubic-bezier，可以使用[工具网站](http://cubic-bezier.com/" \t "_blank)来定制。

img{

transition: 1s height cubic-bezier(.83,.97,.05,1.44);

}

上面的代码会产生一个最后阶段放大过度、然后回缩的效果。

### 1.4 transition的各项属性

transition的完整写法如下。

img{

transition: 1s 1s height ease;

}

这其实是一个简写形式，可以单独定义成各个属性。

img{

transition-property: height;

transition-duration: 1s;

transition-delay: 1s;

transition-timing-function: ease;

}

### 1.5 transition的使用注意

（1）目前，各大浏览器（包括IE 10）都已经支持无前缀的transition，所以transition已经可以很安全地不加浏览器前缀。

（2）不是所有的CSS属性都支持transition，完整的列表查看[这里](http://oli.jp/2010/css-animatable-properties/" \t "_blank)，以及具体的[效果](http://leaverou.github.io/animatable/)。

（3）transition需要明确知道，开始状态和结束状态的具体数值，才能计算出中间状态。比如，height从0px变化到100px，transition可以算出中间状态。但是，transition没法算出0px到auto的中间状态，也就是说，如果开始或结束的设置是height: auto，那么就不会产生动画效果。类似的情况还有，display: none到block，background: url(foo.jpg)到url(bar.jpg)等等。

### 1.6 transition的局限

transition的优点在于简单易用，但是它有几个很大的局限。

（1）transition需要事件触发，所以没法在网页加载时自动发生。

（2）transition是一次性的，不能重复发生，除非一再触发。

（3）transition只能定义开始状态和结束状态，不能定义中间状态，也就是说只有两个状态。

（4）一条transition规则，只能定义一个属性的变化，不能涉及多个属性。

CSS Animation就是为了解决这些问题而提出的。