



moonx RP 329 发布于 moonx
2016-05-06 发布

CSS3动画使用技巧与总结

构建css3动画的属性很简单，总的来说只有transform（变形），transition（过渡），和animation（动画）这三种。

所以这里，更少的从介绍语法的角度，而是从使用的角度来介绍css3动画
具体的每一个属性的用法写法，推荐见[CSS3动画手册](#)

基础知识 - 三个属性

先来回顾下这三个属性和大致功能

transform（变形）

直接改变大小和位置，显示改变的结构，没有过渡和形变时间

```
transform: rotate(360deg) scale(-1.2, 1.2) translate(100px, 50px) skew(45deg, 45deg)
```

1. 旋转rotate

利用css变形属性可以将对应的属性旋转相对应的角度，可以顺时针旋转，逆时针（负值）旋转，可以3D旋转，3D旋转可以分别绕X轴，Y轴，Z轴旋转。

2. 缩放scale

负值为先翻转，再缩放对应的倍数，可以在X,Y, Z上做缩放

3. 位移translate

可以在X, Y轴上做平移，或者同时在x,y,z轴上做平移

4. 斜切skew

斜切skew 二重轴斜切 二重轴斜切 二重轴斜切 二重轴斜切



首页



问答



专栏



讲堂



更多

transition（过渡）

会有一个形变的过程，会有过渡和形变时间

举个复杂的例子

```
.first{
left:110px;
top:0;
-webkit-transition-property:left,top,background;
-webkit-transition-duration:1s,2s,2s;
-webkit-transition-delay:0,0,2s;
-webkit-transition-timing-function:ease,ease-in,linear;
}
.first:hover{
left:310px;
top:200px;
background:red;
}
```

这个例子实现的呢就是，不同属性在不同起始时间开始发生了不同变化时间长短的变化，并且不同变化的变化速率不同

animation（动画）

****同样的也是一个过渡的过程，引入了帧的概念，定义keyframes动画，然后绑定一个或者多个动画。
transition有点像他的简化版，animation也可以实现transition变化某个属性的功能****

并且可以去设置动画的播放时间（animation-duration），播放方式（animation-timing-function，播放速度），播放次数，播放方向（在不同次数，发生不同方向的动画，比如从左到右，然后从右到左），播放后的状态（会到初始状态，还是结束状态）控制动画状态（控制某物体暂停或执行动画）

三者的关系与结合使用？

transition和animation主要区别在哪？

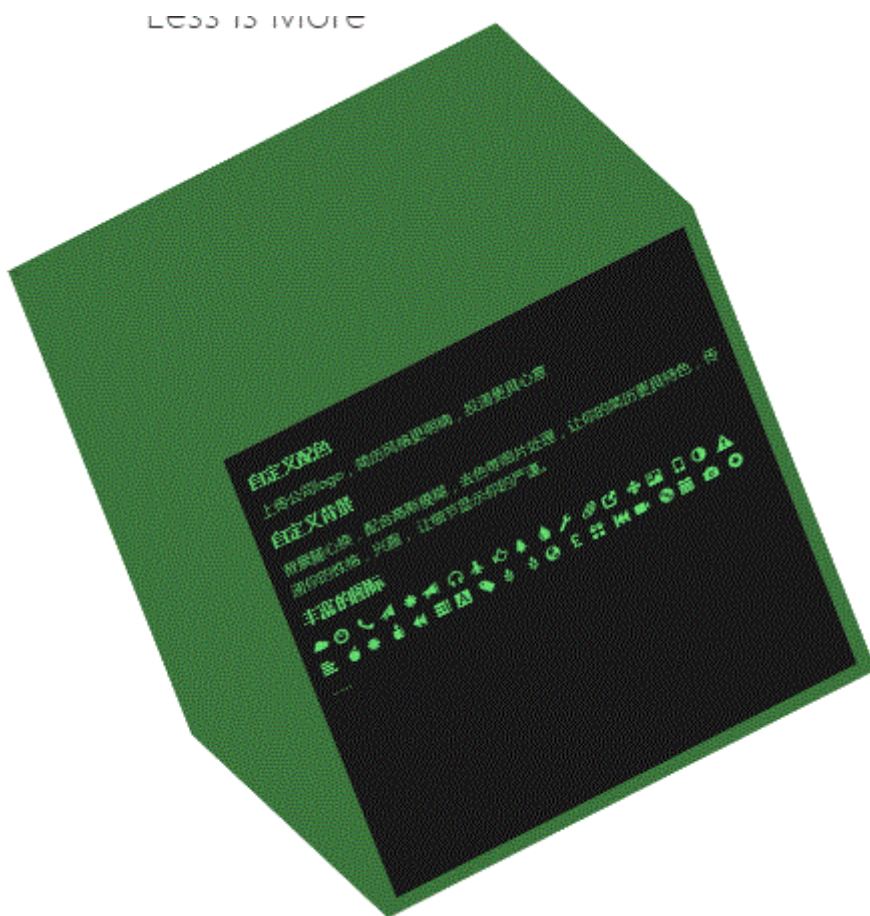
transition也可以看做animation的缩略版，他们实现的功能有重复的地方，比如说要实现某个div的宽度有30rem变到20rem,两者都可以实现。

```
//transition的方法
div{width:30rem;transition:width 1s ease}
div:hover{width:20rem}
//animation的方法
div{animation:demo 1s ease}
@keyframes mymove
{
  from {width:30rem;}
  to {width:20rem;}
}
```

虽然两者有相似的部分，但是实现的概念和出发点是不一样的，transition是着重于属性的变化，而animation重点是在时间轴和关键帧,是在于创建帧，让不同帧在不同的时间节点发生不同变化，基于animation和@keyframe 的动画一方面也是为了实现表现与行为的分离，另一方面也使得前端设计师可以专注得进行动画开发而不是冗余的代码中。

如何结合创造出更好的动画

举一个制作立方体由小变大，并且在这个过程当中翻转的例子

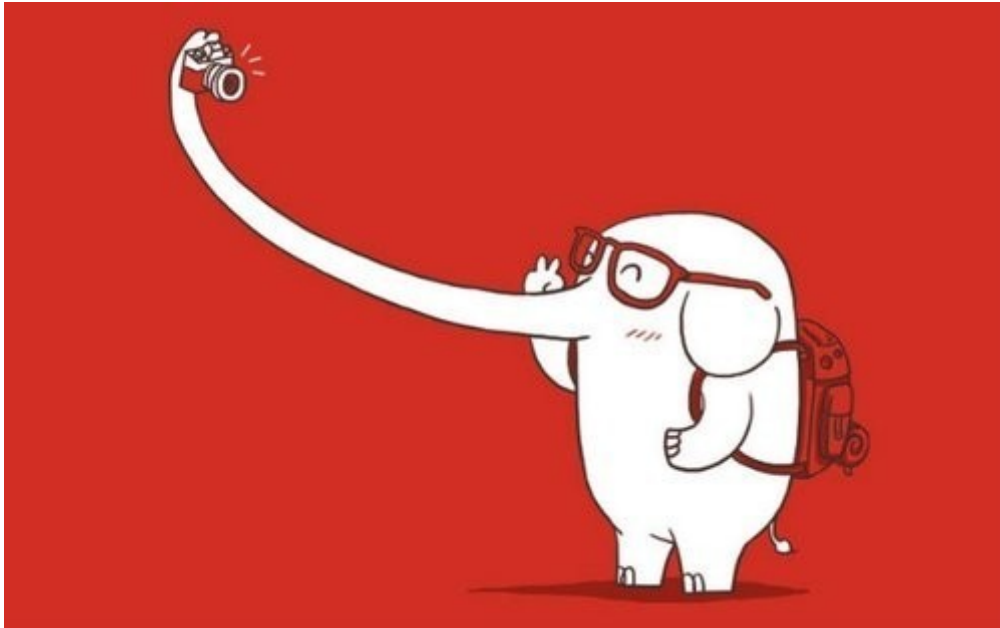


关于CSS3-3D

1.我们要使用css-3D效果，首先得去申明，我们的显示方式是要3D转换

```
//这句话要添加到要做3D转换元素的父元素上，这样他的子元素就都支持3D转换了
transform-style: preserve-3d;
```

2.设置视点



大家应该都知道透视这么个东西，往简单的说就是近大远小。

举个例子，为什么女生自拍要躲在背后呢，因为显得脸小,把手机拿的近一些，屏幕里面的脸就越大，加上自拍杆呢，脸就越小。

假设你拿着手机呢，那么你的眼睛就是视点，你设置的perspective的值就是，你眼睛到手机屏幕的水平距离就是，也就是视点到屏幕的值，当perspective的值设置为0的时候，则不产生近大远小的效果，比较像平行投影。

如果你把相机拿到45度，像这只大象一样仰拍呢，就是调整perspective-origin，相机只能在你面前上下左右的改变角度，而不能从你的背后往前拍，也就是Z坐标固定，可以修改的是相机的X，Y 坐标。

3.rotateX/Y/Z到底是怎么个转法

看到[张鑫旭-鑫空间-鑫生活](#)这篇里面讲的方式，真的感觉这辈子都忘记不了

rotateX:



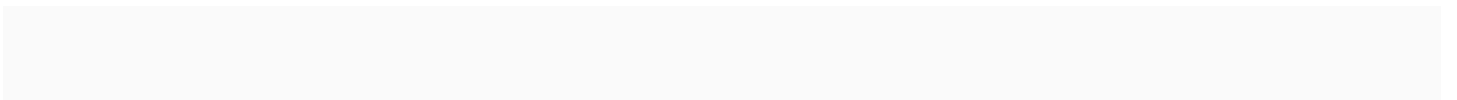
rotateY:

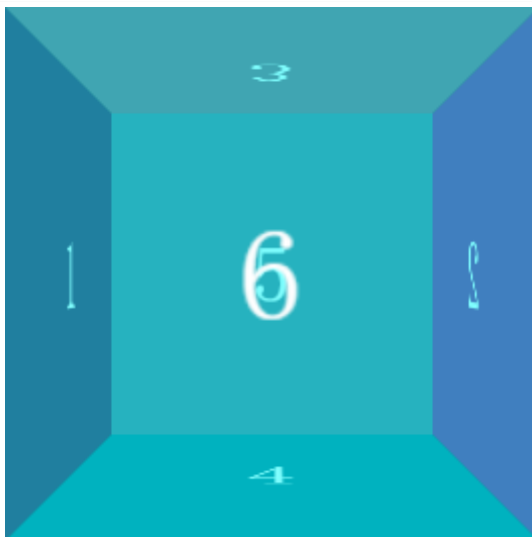


rotateZ:



构建一个立方体





```
<div class="cube-section cube-wrap">
  <div class="cube-1 cube">1</div>
  <div class="cube-2 cube">2</div>
  <div class="cube-3 cube">3</div>
  <div class="cube-4 cube">4</div>
  <div class="cube-5 cube">5</div>
  <div class="cube-6 cube">6</div>
</div>
//那么立方体的左侧面要怎么画呢？，假设左侧面是cube-1
.cube-1{
-webkit-transform:rotatey(90deg) translatez(-100px);
}
```

像钢管舞绕圈一样，先用rotateY绕到与平面垂直的这个地方，可以想象到，绕的轴心点不变的话，是两个相互垂直，且中心点相同的平面，而后再用translateZ去调整这个平面的位置关系，这个值呢相当于面和你电脑平面的距离。

通过这种方法呢，可以构造出其余六个平面

如何让正方体动起来？

```
@-webkit-keyframes animation2 {
  0% {
    -webkit-transform: scale(0.2) rotateX(0deg) rotateY(0deg) rotateZ(0deg);
    -webkit-transform-origin: center center;
  }
  100% {
    -webkit-transform: scale(0.7) rotateX(180deg) rotateY(180deg) rotateZ(180deg);
    -webkit-transform-origin: center center;
  }
}
```

```
-webkit-transform-style:preserve-3d;
-moz-transform-style:preserve-3d;
perspective: 0px;
-webkit-perspective: 0px;
-moz-perspective: 0px;
-webkit-animation:animation2 3s ease ;
-moz-animation:animation2 3s ease ;
}
```

如何控制动画的播放

CSS动画一般在页面加载完就自动播放了，也只有对应的delay属性能够推迟其发生的时间，但是我们常常需要控制动画的播放时间，播放触发点，比如在点击的时候，换屏的时候播放动画，那么我们要怎么做呢？——用Js在特定的时刻添加包含动画的Class。

```
.cube-wrap{//设置形状大小等}
.active .cube-wrap{//设置动画}
//然后再相应的触发点时，使用js方法X.classList.add("active");
```

如何让动画停止在你想要的任意效果

1.利用.active与transform 动画播放完毕

```
.active .cube-wrap {
//不管他怎么转，最后转到哪里，固定下来的位置就是transform的位置
-webkit-transform: rotatey(0deg) rotatex(0deg);
}
```

2.animation-fill-mode

这个设置成backwards可以使动画回到初始模式，设置成forwards：结束后保持动画结束时的状态，但当animation-direction为0，则动画不执行，持续保持动画开始时的状态

3.animation-play-state

检索或设置对象动画的状态，可以设置运动或者暂停，如果提供多个属性值，以逗号进行分隔。

如何每次执行到触发点的时候都执行一遍动画？


```
$(".cube-section").removeClass("click-animation");  
/*加一句这个触发reflow*/  
$('.cube-section').innerWidth($('.cube-section').innerWidth());  
$(".cube-section ").addClass("click-animation");
```

如何更有逻辑的编写动画

推荐看看这篇[ISUX](#)写的，感觉很有收获

效率速度

```
//使用  
.somecss{  
transform: translateX(10rem)  
}  
.active{  
transform: translateX(0)!important;  
}  
//使用tranform的translateX的效率比用transition的left这样来改变有效率的多
```

css动画的优缺点

CSS3动画肯定不是万能的，因为一些固有的局限性，也很难百分百的实现AE上所有的动画效果.比如：

- 实现曲线动画
- 基于物理的的动画效果

(这个[动画工具](#)的网站呢提供了大量的基础动画，我们可以基于这些动画来做扩展和调整)

但是他简单，高效，硬件加速等优点又十分惹人喜爱，至于用js还是用css,更多的是取决于你

参考资料

[Myth Busting: CSS Animations vs. JavaScript](#)

[经验分享：多屏复杂动画CSS技巧三则](#)

[CSS Animation](#)

高性能CSS3动画

好吧，CSS3 3D transform变换，不过如此！



赞 | 4

收藏 | 39

你可能感兴趣的

- 前端的3D(css3版本)--淘宝造物节3D创景的制作 QRL [html5](#) [javascript](#) [css3动画](#)
- 常用css3整理 清儿 [css3](#)
- css3 炫酷的动画transform, transition, animation 小渝人儿 [css3](#)
- css3 动画（一） transition hileix [css3](#) [css3动画](#) [transition](#)
- CSS3 Transition, transform 和 animation 介绍 MockingBird [animation](#) [css3](#) [javascript](#)
- CSS3 变换 Chinaxiang [css3](#) [transform](#) [css](#)
- 理解CSS3之`transform` geek_zwb [frontend](#) [css3](#)
- CSS3 过渡 Chinaxiang [transition](#) [css3](#)

评论

默认排序 时间排序

文明社会，理性评论

发布评论

