后盾网 人人做后盾

www.houdunwang.com

CSS3动画

后盾网 2011-2015

animation:

- Animation只应用在页面上已存在的DOM元素上，我们可以使用CSS3的Animation制作动画我们可以省去复杂的js,jquery代码。

keyframes

- keyframes即关键帧。Keyframes具有其自己的语法规则，他的命名是由“@keyframes”开头，后面紧接着是这个“动画的名称”加上一对花括号“{}”，括号中就是一些不同时间段样式规则，有点像我们css的样式写法一样。对于一个“@keyframes”中的样式规则是由多个百分比构成的，如“0%”到“100%”之间，我们可以在这个规则中创建多个百分比，我们分别给每一个百分比中给需要有动画效果的元素加上不同的属性，从而让元素达到一种在不断变化的效果，比如说移动，改变元素颜色，位置，大小，形状等，不过有一点需要注意的是，我们可以使用“from”“to”来代表一个动画是从哪开始，到哪结束，也就是说这个 “from”就相当于“0%”而“to”相当于“100%”，值得一说的是，其中“0%”不能像别的属性取值一样把百分比符号省略。

animation动画

animation简化写法:

- animation: name duration timing-function delay iteration-count direction

animation-name:

- 定义一个动画的名称。

animation-duration:

- 指定动画播放时长，单位为s(秒),默认值为“0”。

animation-timing-function:

- 动画的播放方式。和transition中的transition-timing-function一样，具有以下六种变换方式：ease;ease-in;ease-in-out;linear;cubic-bezier。

animation-delay:

- 指定延迟时间。单位为s(秒)，其默认值也是0。这个属性和transition-delay使用方法是一样的。

animation-iteration-count:

- 指定元素播放动画的循环次数，其默认值为“1”；infinite为无限次数循环。

animation动画

animation-direction:

- 是否应该轮流反向播放动画
- 如果 animation-direction 值是 "alternate"，则动画会在奇数次数（1、3、5 等等）正常播放，而在偶数次数（2、4、6 等等）向后播放。
- alternate-reverse 是相反

animation-play-state:

- 控制元素动画的播放状态。有两个值，running和paused其中running为默认值。他们的作用就类似于我们的音乐播放器一样，可以通过paused将正在播放的动画停下了，也可以通过running将暂停的动画重新播放。

animation-fill-mode:

- 设置动画执行结束后的元素状态。
- forwards 动画结束时的状态
- backwards 动画的初始状态
- none 默认值，动画在动画执行之前和之后不会应用任何样式到目标元素

animation动画

```
<style type="text/css">
#box{
	width: 50px;
	height: 50px;
	border-radius: 50px;
	margin: 50px auto;
	text-align: center;
	line-height: 55px;
	/*一刷新页面就动画开始*/
	/*名称，动画时间，动画函数，延迟几秒开始，无限播放动画，1,3,5正常播放2,4,6反向播放*/
	animation: hd 1s linear 0 infinite  alternate;
}
@-webkit-keyframes hd{
	/*从正常开始*/
	from{-webkit-transform: scale(1);}
	/*放大3倍*/
	to{-webkit-transform: scale(3);}
}
/*鼠标经过停止*/
#box:hover{
	/*还有一个值是running是运行的意思*/
	animation-play-state:  paused;
	cursor: pointer;
}
</style>
<div id="box">❤</div>
```

animation动画-完整参数

```
<style type="text/css">
#box{
	width: 50px;
	height: 50px;
	border-radius: 50px;
	margin: 50px auto;
	text-align: center;
	line-height: 55px;
	/*一刷新页面就动画开始*/
	/*名称，动画时间，动画函数，延迟几秒开始，动画播放3次，1,3,5正常播放2,4,6反向播放*/
	animation: hd 0.5s linear 0 3 alternate;
	/*注释掉这样看对比*/
	animation-fill-mode: forwards;
}
@-webkit-keyframes hd{
	0%{
		/*从正常开始*/
		-webkit-transform: scale(1);
	}
	100%{
		/*放大3倍*/
		-webkit-transform: scale(3);
	}
}
</style>
<div id="box">❤️</div>
```

animation动画-fill-mode

perspective透视（没有透视，不成3D，设置在旋转的元素上）：

用在舞台元素上：

- .stage {
- perspective: 100px;
- }

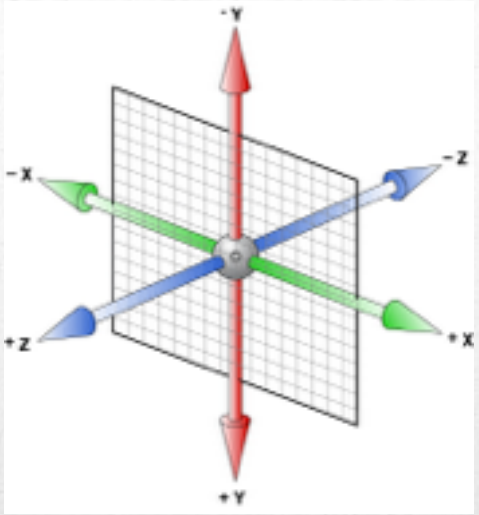
用在当前动画元素上：

- #stage .box {

perspective： perspective是视点，也就是眼睛看东西，眼睛作为视点，那么可以想象一只手在左右摆动，离眼睛越近，那么效果就越明显，如果远的话，效果就不明显

rotate：rotateX rotateY rotateZ 可以设置这3个，分别就是右图的3个轴

- transform: perspective(100px) rotateX(180deg);
- }



transform-style子元素位于3D空间中(否则进行3d旋转时没有效果)

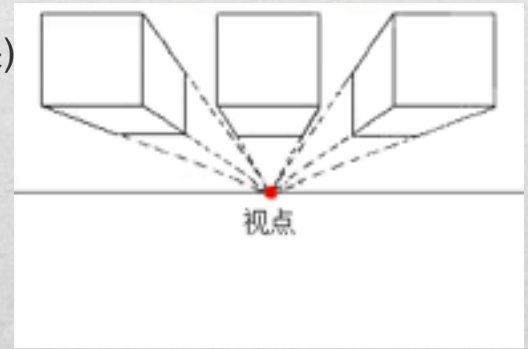
.stage{

`flat` 子元素将不保留其 3D 位置。默认参数

`preserve-3d` 子元素将保留其 3D 位置。

transform-style: preserve-3d;

}



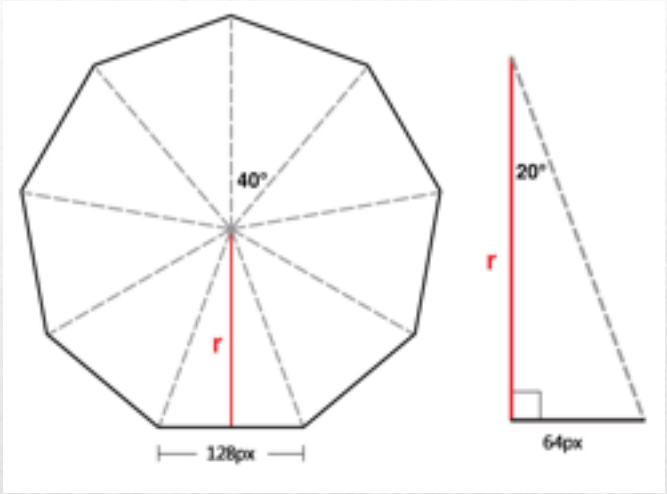
CSS3 3D transform

3d实例：

```
<style type="text/css">
	.red{
		background: red;
		width: 100px;
		height: 100px;
		animation: hd 3s linear infinite;
		transform-style: preserve-3d;
	}
	p{
		width: 50px;
		height: 50px;
		background: yellow;
		transform: rotateX(90deg);
	}
	@-webkit-keyframes hd{
		from {
			transform: rotateX(0deg);
		}
		to {
			transform: rotateX(360deg);
		}
	}
</style>
<div class="red">
	<p>hd</p>
</div>
```

CSS3 preserve-3d

正方体实例



tan(弧度) = 对边/邻边

r = (图片宽度/2) / tan(每张图片占的度 / 2 * Math.PI / 180)

3d照片墙实例