

# 前端深入之css篇 | 2020年，彻底掌握css动画【transform】

---

## 写在前面

---

已经2020年了，不知道小伙伴们今年学习了css3动画了吗？

说起来css动画是一个很尬的事，一方面因为公司用css动画比较少，另一方面大部分开发者习惯了用JavaScript来做动画，所以就导致了許多程序员比较排斥来学习css动画（至少我是），但是一个不懂css动画的前端工程师不能称之为掌握css3，其实当你真正学习css动画之后，你会被它的魅力所吸引的，它可以减少代码量、提高性能。

这是一个系列文章，共分为四篇。

[前端深入之css篇 | 2020年，彻底掌握css动画【transition】](#)

[前端深入之css篇 | 2020年，彻底掌握css动画【animation】](#)

[前端深入之css篇 | 2020年，彻底掌握css动画【transform】](#)

[前端深入之css篇 | 初探【transform】，手把手带你实现1024程序员节动画](#)

前面几篇文章我们已经一起学习了 `animation` 和 `transition` 的使用，今天我们终于开始学习 `transform` 和 `translate` 了，其实 `translate` 只是 `transform` 的一个属性，只是很多初学者对 `transform` (变形)、`translate` (移动) 和 `transition` (过渡) 容易混淆，所以我把它们放到一块来写了，小伙伴们可不要混淆啊。

话不多说，马上跟我一起学习今天的主角**transform**吧。

## transform 语法

---

值	描述
none	定义不进行转换。
translate(x,y)	定义 2D 转换。
translate3d(x,y,z)	定义 3D 转换。
translateX(x)	定义转换，只是用 X 轴的值。
translateY(y)	定义转换，只是用 Y 轴的值。
translateZ(z)	定义 3D 转换，只是用 Z 轴的值。
scale(x[,y]?)	定义 2D 缩放转换。
scale3d(x,y,z)	定义 3D 缩放转换。
scaleX(x)	通过设置 X 轴的值来定义缩放转换。
scaleY(y)	通过设置 Y 轴的值来定义缩放转换。
scaleZ(z)	通过设置 Z 轴的值来定义 3D 缩放转换。
rotate(angle)	定义 2D 旋转，在参数中规定角度。
rotate3d(x,y,z,angle)	定义 3D 旋转。
rotateX(angle)	定义沿着 X 轴的 3D 旋转。
rotateY(angle)	定义沿着 Y 轴的 3D 旋转。
rotateZ(angle)	定义沿着 Z 轴的 3D 旋转。
skew(x-angle,y-angle)	定义沿着 X 和 Y 轴的 2D 倾斜转换。
skewX(angle)	定义沿着 X 轴的 2D 倾斜转换。
skewY(angle)	定义沿着 Y 轴的 2D 倾斜转换。
perspective(n)	为 3D 转换元素定义透视视图。

`transform` 字面上就是变形，改变的意思。看起来他有很多属性，其实我们把常用的总结起来就是下面四个属性。

- rotate(旋转)
- skew(扭曲)
- scale(缩放)
- translate(移动)

## rotate 旋转



```
<style>
img{
  margin-left: 50px;
  width:50px;
  height:50px;
  border-radius:50%;
}
@keyframes rotate{
  0% {transform:rotate(0deg);}
  100% {transform:rotate(360deg);}
}
@keyframes rotateX{
  0% {transform:rotateX(0deg);}
  100% {transform:rotateX(360deg);}
}
@keyframes rotateY{
  0% {transform:rotateY(0deg);}
  100% {transform:rotateY(360deg);}
}

.rotate{
  animation:rotate 2s infinite linear;
}
.rotateX{
  animation:rotateX 2s infinite linear;
}
.rotateY{
  animation:rotateY 2s infinite linear;
}
</style>
<body>
  
  
  
</body>
复制代码
```

这里一共有三个属性的展示 `rotate` , `rotateX` , `rotateY` 。分别代表在平面上根据指定角度进行旋转、沿着X轴进行指定角度的旋转、沿着Y轴进行制定角度的旋转。

## translate 移动



@掘金技术社区

```
<style>
img{
  margin-left: 50px;
  width:50px;
  height:50px;
  border-radius:50%;
}
@keyframes translate{
  0% {transform:translate(0px,0px);}
  100% {transform:translate(100px,100px);}
}
@keyframes translateX{
  0% {transform:translateX(0px);}
  100% {transform:translateX(100px);}
}
@keyframes translateY{
  0% {transform:translateY(0px);}
  100% {transform:translateY(100px);}
}

.translate{
  animation:translate 2s infinite linear;
}
.translateX{
  animation:translateX 2s infinite linear;
}
.translateY{
  animation:translateY 2s infinite linear;
}
</style>
```

```
<body>
  
  
  
</body>
```

复制代码

这里一共有三个属性的展示 `translate(x,y)` , `translateX(x)` , `translateY(Y)` 。分别代表水平方向和垂直方向同时移动、仅水平方向移动、仅垂直方向移动。

## scale 缩放



@掘金技术社区

```
<style>
img{
  margin-left: 50px;
  width:50px;
  height:50px;
  border-radius:50%;
}
@keyframes scale{
  0% {transform:scale(0.1,0.1);}
  100% {transform:scale(1,1);}
}
@keyframes scaleX{
  0% {transform:scaleX(0.1);}
  100% {transform:scaleX(1);}
}
@keyframes scaleY{
  0% {transform:scaleY(0.1);}
  100% {transform:scaleY(1);}
}

.scale{
  animation:scale 2s infinite linear;
}
.scaleX{
```

```

        animation:scaleX 2s infinite linear;
    }
    .scaleY{
        animation:scaleY 2s infinite linear;
    }

</style>
<body>
    
    
    
</body>
复制代码

```

这里一共有三个属性的展示 `scale(x,y)` , `scaleX(x)` , `scaleY(Y)` 。分别代表水平方向和垂直方向同时缩放、仅水平方向缩放、仅垂直方向缩放。但它们具有相同的缩放中心点和基数，其中心点就是元素的中心位置，缩放基数为1，如果其值大于1元素就放大，反之其值小于1，元素缩小。

## skew 扭曲



@掘金技术社区

```

<style>
img{
    margin-left: 50px;
    width:50px;
    height:50px;
    /* border-radius:50%; */
}
@keyframes skew{
    0% {transform:skew(0deg,0deg);}
    100% {transform:skew(25deg,25deg);}
}
@keyframes skewX{
    0% {transform:skewX(0deg);}
    100% {transform:skewX(25deg);}
}
@keyframes skewY{
    0% {transform:skewY(0deg);}
    100% {transform:skewY(25deg);}
}

```

```
}

.skew{
  animation:skew 2s infinite linear;
}
.skewX{
  animation:skewX 2s infinite linear;
}
.skewY{
  animation:skewY 2s infinite linear;
}
</style>
<body>
  
  
  
</body>
复制代码
```

这里一共有三个属性的展示 `skew(x,y)` , `skew(x)` , `skewY(Y)` 。分别代表水平方向和垂直方向同时扭曲、仅水平方向扭曲、仅垂直方向扭曲。

## 结论

今天我们一起学习了 `transform` 的常见属性，你会发现他们其实非常简单，你只需要记住 `rotate(旋转)`、`skew(扭曲)`、`scale(缩放)`、`translate(移动)` 就可以了，所代表的含义就是会像X和Y轴同时变形，而加上X或者Y就代表会向着该方向进行变形。

到现在为止我们一起学完了css3动画的所有内容，剩下的就需要你来多加练习，在其他网站上看到了有趣的动画要想办法用css动画来实现它，在你真正练习过之后，你将会真正掌握它，这样你就算做在2020年掌握了css动画！