

一、transform 转换属性

1. translate位移

```
1 transform : translate(50px,100px); //把元素水平移动 50 像素，垂直移动 100 像素
2
3 transform : translateX(50px); //把元素水平移动 50 像素
4
5 transform : translateY(100px); //把元素垂直移动 100 像素
```

2. scale缩放

```
1 transform : scale(2); //把元素宽高同时放大两倍
2
3 transform : scale(2,4); //把元素宽度放大2倍，高度放大4倍
4
5 transform : scaleX(0.5); //把元素宽度缩小一半
6
7 transform : scaleY(1.2); //把元素高度放大1.2倍
```

3. rotate旋转

```
1 transform : rotate(45deg); //把元素旋转45deg
2
3 transform : rotateX(45deg); //把元素沿X轴旋转45deg
4
5 transform : rotateY(45deg); //把元素沿Y轴旋转45deg
```

4. skew倾斜

```
1 transform : skew(30deg,20deg); //把元素沿水平方向倾斜30deg,沿垂直方向倾斜20deg
2
3 transform : skewX(30deg); //把元素沿水平方向倾斜30deg
4
5 transform : skewY(20deg); //把元素沿垂直方向倾斜20deg
6
```

每个元素只能设置一个transform属性，可以同时设置多个值,如:

```
1 transform: rotate(30deg) translate(50px) scale(2);
```

5. 旋转元素的基点位置

```
1 transform-origin: x y z;
```

二、transition 过渡属性

```
1 transition-property : css的属性名称 或者all
2
3 transition-duration : 动画持续的时间
4
5 transition-timing-function : 动画的执行的曲线
6     ease:平滑运动
7     linear: 线性运动
8     ease-in:逐渐加速 速度越来越快
9     ease-out:逐渐减速 速度越来越慢
10    ease-in-out:先加速再减速 速度先变快再变慢
11
12 transition-delay : 动画延迟时间
13
14 // 一般不分开设置，直接写复合属性，
```

```
15 transition: all 2s linear 2s;
```

css过渡效果结束后执行的js事件:transitionend

```
1 document.getElementById("myDIV").addEventListener("transitionend", function () {
2     // ...todo
3 });
4
```

三、animation 关键帧动画

1. 定义动画

```
1 @keyframes 动画名{
2     0%{ ... }
3     50%{ ... }
4     100%{ ... }
5 }
6 //开始状态和结束状态时，可以用from 和 to代替
```

2. 调用动画

```
1 animation-name: 动画名称
2
3 animation-duration: 动画持续的时间
4
5 animation-timing-function: 动画的执行的速度,与transition属性相同
6
7 animation-delay: 延迟时间
8
9 animation-iteration-count: 动画执行的次数,可以是数字或者infinite (无限循环)
10
11 animation-direction: 是否循环交替反向播放动画,
12     normal( 正常播放 )
13     alternate(奇数次正向, 偶数次反向)
14     alternate-reverse (奇数次反向, 偶数次正向)
15     reverse( 反向播放 )
16
17 animation-fill-mode: 动画填充的模式,
18     forwards(动画结束后停在最后的位置)
19     backwards(保留动画开始的初始状态)
20
21 animation-play-state: 动画是否暂停,
22     running( 继续 )
23     paused( 暂停 )
```

3. 请求动画帧 requestAnimationFrame

1. 屏幕刷新频率: 屏幕每秒出现图像的次數。普通笔记本为60Hz (赫兹)
2. 动画原理: 计算机每16.7ms刷新一次, 由于人眼的视觉停留, 所以看起来是流畅的移动。
3. 通过定时器达到的动画效果, 容易卡顿抖动,原因是:
 - a. setTimeout异步加载, 只有当主线程任务执行完后才会执行, 因此实际执行时间总是比设定时间要晚
 - b. setTimeout的固定时间间隔不一定与屏幕刷新时间相同, 会引起丢帧
4. 每次刷新的间隔中会执行一次requestAnimationFrame函数, 不会引起丢帧, 不会卡顿

四、calc() 函数用于动态计算长度值。

1. 需要注意的是，运算符前后都需要保留一个空格，例如：`width: calc(100% - 10px);`
2. 任何长度值都可以使用`calc()`函数进行计算；
3. `calc()`函数支持 "+", "-", "*", "/" 运算；
4. `calc()`函数使用标准的数学运算优先级规则；