1 .CSS3中的变形 transform（不兼容 ie9-）

translate/translateX/translateY/translateZ
沿着某一个轴进行位移

```
transform: scale(1.2); 缩放比例，在原有的基础上放大1.2倍
transform: skew(45deg);倾斜45度  skewX(45deg)|skewY(45deg)|skewZ(45deg);
transform: rotate(35deg); rotateX(35deg)| rotateY(35deg)| rotateZ(35deg)
```

transform 不是动画，他是给元素设置变形的效果，在不考虑兼容的情况下，我们尽量使用transform来实现一个元素的变形和位移（尤其是移动端开发），因为他开启了设备的硬件加速，性能和体验都是强于传统的css样式改变的

2 . CSS3中的动画 transition

> transition :过渡动画： 我们设置一个过渡的规则和效果，当元素的样式发生改变的时候，就会按照这个规则来运动
> 语法：
> transition：[property] [duration] [time-duration] [delay]
> property:设置元素的哪些样式在改变后执行这个过渡的规则，默认值是all
> duration:设置动画运动的总时间
> time-function：设置动画运动的方式，默认值是linear，除此之外还有ease，ease-in，ease-out

animation: 帧动画(不能通过js来设置的)

> -webkit-animation:
> name:动画名称，运动轨迹（帧)的名字
> duration:运动时间
> time-function:运动的方式
> delay：延迟时间
> count：运动次数 infinite无限次运动
> fill-mode:
> forwards:当元素按照轨迹运动完成后停留在最后一帧的位置，不设置默认动画完成后，元素立马回到第一帧的位置
> backwards：如果本次动画有延迟，那么在延迟等到的时间，元素始终处于第一帧的位置，不设置的话是处于当前位置
> both：同时具备以上两个特点

```
@ -webkit-key-frames[name]{
        from{
            => 0%  开始帧位置
            top:10px

            }
        to{
            top:50px
            }

}
```

> 总结：
> 能用CSS搞定的不要用JS
> 能用transform实现变形的不用之前的传统样式属性(因为TF开启了硬件加速)
> 动画分为两种：
> transition
> ->目标位置和具体的运动轨迹可以不固定,我们可以在JS中动态设置运动的目标位置,只要把元素的样式发生改变，都会执行对应的过渡效果
> ->只是一个简单的从A到B的过程，不能设置复杂的运动轨迹

```
animation
   ->可以设置复杂的运动轨迹，但是需要提前指定好轨迹公式，不能在JS中动态的修改
   ->我们可以依托Animate.css中提供的强大轨迹，完成我们的业务需求
```

3. 盒子模型

> 1.box-sizing:border-box

实现四列布局

> -webkit-columns: 200px 4;
> columns: 200px 4;

浏览器根据每一列的宽度自动安排列数

> -webkit-column-count: 3;
> column-count: 3;

浏览器会自动计算列数

```
umn-count: auto，则浏览器会从新的计算列数
-webkit-column-gap: 20px;
column-gap: 20px;
```

设置列和列之间的边框

```
-webkit-column-rule: 1px dashed #ccc;
column-rule: 1px dashed #ccc;
```

CSS样式的书写顺序

```
display
position(z-index) / float
margin
border
padding
width / height
font-xxx
background-xxx
…
```