1. 在默认样式中声明元素的初始状态样式。
2. 声明过渡元素最红状态样式，比如悬浮状态。
3. 在默认样式中通过添加过度函数，添加一些不同的样式。

来和去都会过渡。

transition: transition-property transition-duration transition-timing-function transition-delay

例如：

transition: width 0.5s ease-in 1s, border-radius 0.5s ease-out 0.5s;

1. transition-property: all || none || <single-transition-property>
2. transition-duration:单位是s或ms，默认值是0，也就是说变换是即时的。
3. transition-timing-function：ease（默认的，由快到慢）。
4. transition-delay：制定一个动画开始执行的时间，也就是说多久后开始执行动画。  
   正值：不会立即触发，而是过一段时间再触发。  
   负值：从该时间点开始显示，其实前面已经发生过渡，但是之前的动作会被截断。  
   默认值为0，立即触发，没有延迟。

如何触发过渡？

单纯通过代码是没法触发任何过渡操作的。需要用户的行为触发，如点击click、悬浮hover。

1. 伪元素触发 :active :focus :hover :checked
2. 媒体查询
3. .test{
4. width: 500px;
5. height: 500px;
6. transition: all 0.5s ease 1s;
7. }
8. @media only screen and (max-width:960px){
9. .test{
10. width:200px;
11. height: 200px;
12. }
13. }
14. JS触发

隐式过渡：比如说设置padding：2em。但是，那你过渡了font-size。

1. 开关状态的不同过渡方式
2. .father{
3. height: 100px;
4. transition: all 3s ease-in 0.5s; //到达这个状态所用的时间
5. }
6. .father:hover{
7. height: 50px;
8. transition: all 2s ease 1s;// 到达这个状态所用的时间
9. }
10. 几乎无限延迟的过渡。回来的时间很长，基本可以看作是不动。因此，结果是:hover变为永久性状态，从而使人产生JS改变元素样式的错觉。
11. .father{
12. height: 100px;
13. transition: all 999999s ease-in 999999s;
14. }
15. .father:hover{
16. height: 50px;
17. transition: all 2s ease 1s;
18. }
19. transform：translateZ（0）；可以让GPU渲染，硬件加速。