

动画

关键帧：

语法：

- 在 style 中定义，在元素选择器中调用

```
@keyframes 名字{  
  from{  
    起始位置的状态  
  }  
  to{  
    终点位置的状态  
  }  
}
```

```
eg: @keyframes move{  
  from{  
    transform: rotate(0deg);  
  }  
  to{  
    transform: rotate(360deg);  
  }  
}
```

默认：先从初始位置到 **from** 到 **to** 的位置，再回到初始位置

- 可以百分数

```
@keyframes 名字{  
  0%{  
    起始位置的状态  
  }  
  50%{ 动作}  
  100%{ 动作 }
```

百分比是时间的百分比，动画持续时间的百分比不指定就是默认值

animation-name: 定义关键帧的名字

在style调用： animation-name: 名字；

可加时间过渡： animation-duration，在css中用可以直接上来就会动

animation-timing-function:可取值如下

- 定义动画将如何完成一个周期
- 对于关键帧动画来说，timing function 作用于一个关键帧周期（就一个 **from{}**）而非整个动画周期（停顿的时间是每个{ }的时间，不是一步从头到尾），即从关键帧开始开始，到关键帧结束结束。
- 动画的默认效果：由慢变快再变慢
 - linear:从头到尾都一样
 - ease:动漫由低俗开始，然后加快，在结束前变慢
 - ease-in:由慢到快，由低速开始
 - ease-out:由快到慢，由低速结束
 - ease-in-out:由慢到快再到慢
 - cubic-bezier(1,1,2,3)

`steps(n,[start|end])`

传入一到两个参数，第一个参数意思是把动画分成 `n` 等分，然后动画就会平均地运行。
第二个参数 `start` 表示从动画的开头开始运行，相反，`end` 就表示从动画的结尾开始运行，默认值为 `end`。

animation-delay: 时间+s

定义动画开始前等待的时间,以秒或毫秒计(属于动画外的范畴)

animation-iteration-count :数字 循环多少次动画

定义了动画执行的次数（属于动画内的范畴）只执行内部的，一次完到一次 不理睬外部延迟什么之类的

animation-direction : 可取值如下

normal 按照动画自己动

reverse 决定关键帧的执行顺序和执行速度，都相反

alternate 从from到to 在从to到from 往前播放在倒带从头再播放

alternate-revers 从to到from 绑定timing-function也是相反变化

animation-fill-mode: 可取值如下

控制动画在动画外的位置

默认是：先从初始位置到 `from` 到 `to` 的位置，再回到初始位置

backward 开始就是from的位置，再到to的位置 再回到初始值

forward 开始在初始位置 到from位置再到头的位置，再到头的位置

both 就开始在from 停止在to位置

animation-play-state: paused/running

定义了动画执行的运行和暂停