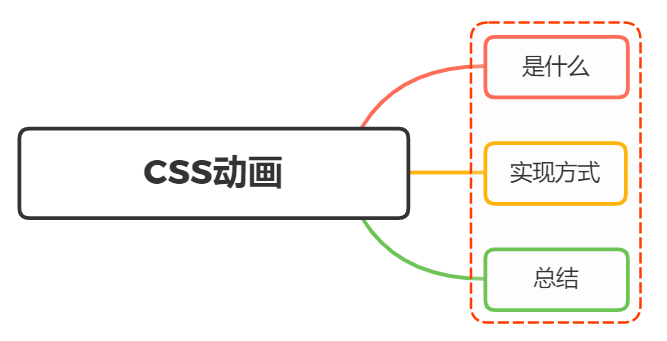
# 面试官：css3动画有哪些？



## 一、是什么

CSS动画（CSS Animations）是为层叠样式表建议的允许可扩展标记语言（XML）元素使用CSS的动画的模块

即指元素从一种样式逐渐过渡为另一种样式的过程

常见的动画效果有很多，如平移、旋转、缩放等等，复杂动画则是多个简单动画的组合

css实现动画的方式，有如下几种：

* transition 实现渐变动画
* transform 转变动画
* animation 实现自定义动画

## 二、实现方式

### transition 实现渐变动画

transition的属性如下：

* property:填写需要变化的css属性
* duration:完成过渡效果需要的时间单位(s或者ms)
* timing-function:完成效果的速度曲线
* delay: 动画效果的延迟触发时间

其中timing-function的值有如下：

|  |  |
| --- | --- |
| 值 | 描述 |
| linear | 匀速（等于 cubic-bezier(0,0,1,1)） |
| ease | 从慢到快再到慢（cubic-bezier(0.25,0.1,0.25,1)） |
| ease-in | 慢慢变快（等于 cubic-bezier(0.42,0,1,1)） |
| ease-out | 慢慢变慢（等于 cubic-bezier(0,0,0.58,1)） |
| ease-in-out | 先变快再到慢（等于 cubic-bezier(0.42,0,0.58,1)），渐显渐隐效果 |
| cubic-bezier(*n*,*n*,*n*,*n*) | 在 cubic-bezier 函数中定义自己的值。可能的值是 0 至 1 之间的数值 |

注意：并不是所有的属性都能使用过渡的，如display:none<->display:block

举个例子，实现鼠标移动上去发生变化动画效果

<style>  
 .base {  
 width: 100px;  
 height: 100px;  
 display: inline-block;  
 background-color: #0EA9FF;  
 border-width: 5px;  
 border-style: solid;  
 border-color: #5daf34;  
 transition-property: width, height, background-color, border-width;  
 transition-duration: 2s;  
 transition-timing-function: ease-in;  
 transition-delay: 500ms;  
 }  
  
 /\*简写\*/  
 /\*transition: all 2s ease-in 500ms;\*/  
 .base:hover {  
 width: 200px;  
 height: 200px;  
 background-color: #5daf34;  
 border-width: 10px;  
 border-color: #3a8ee6;  
 }  
</style>  
<div class="base"></div>

### transform 转变动画

包含四个常用的功能：

* translate：位移
* scale：缩放
* rotate：旋转
* skew：倾斜

一般配合transition过度使用

注意的是，transform不支持inline元素，使用前把它变成block

举个例子

<style>  
 .base {  
 width: 100px;  
 height: 100px;  
 display: inline-block;  
 background-color: #0EA9FF;  
 border-width: 5px;  
 border-style: solid;  
 border-color: #5daf34;  
 transition-property: width, height, background-color, border-width;  
 transition-duration: 2s;  
 transition-timing-function: ease-in;  
 transition-delay: 500ms;  
 }  
 .base2 {  
 transform: none;  
 transition-property: transform;  
 transition-delay: 5ms;  
 }  
  
 .base2:hover {  
 transform: scale(0.8, 1.5) rotate(35deg) skew(5deg) translate(15px, 25px);  
 }  
</style>  
 <div class="base base2"></div>

可以看到盒子发生了旋转，倾斜，平移，放大

### animation 实现自定义动画

animation是由 8 个属性的简写，分别如下：

|  |  |  |
| --- | --- | --- |
| 属性 | 描述 | 属性值 |
| animation-duration | 指定动画完成一个周期所需要时间，单位秒（s）或毫秒（ms），默认是 0 |  |
| animation-timing-function | 指定动画计时函数，即动画的速度曲线，默认是 "ease" | linear、ease、ease-in、ease-out、ease-in-out |
| animation-delay | 指定动画延迟时间，即动画何时开始，默认是 0 |  |
| animation-iteration-count | 指定动画播放的次数，默认是 1 |  |
| animation-direction 指定动画播放的方向 | 默认是 normal | normal、reverse、alternate、alternate-reverse |
| animation-fill-mode | 指定动画填充模式。默认是 none | forwards、backwards、both |
| animation-play-state | 指定动画播放状态，正在运行或暂停。默认是 running | running、pauser |
| animation-name | 指定 @keyframes 动画的名称 |  |

CSS 动画只需要定义一些关键的帧，而其余的帧，浏览器会根据计时函数插值计算出来，

通过 @keyframes 来定义关键帧

因此，如果我们想要让元素旋转一圈，只需要定义开始和结束两帧即可：

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

from 表示最开始的那一帧，to 表示结束时的那一帧

也可以使用百分比刻画生命周期

@keyframes rotate{  
 0%{  
 transform: rotate(0deg);  
 }  
 50%{  
 transform: rotate(180deg);  
 }  
 100%{  
 transform: rotate(360deg);  
 }  
}

定义好了关键帧后，下来就可以直接用它了：

animation: rotate 2s;

## 三、总结

|  |  |
| --- | --- |
| 属性 | 含义 |
| transition（过度） | 用于设置元素的样式过度，和animation有着类似的效果，但细节上有很大的不同 |
| transform（变形） | 用于元素进行旋转、缩放、移动或倾斜，和设置样式的动画并没有什么关系，就相当于color一样用来设置元素的“外表” |
| translate（移动） | 只是transform的一个属性值，即移动 |
| animation（动画） | 用于设置动画属性，他是一个简写的属性，包含6个属性 |

## 参考文献

* https://segmentfault.com/a/1190000022540857
* https://zh.m.wikipedia.org/wiki/CSS%E5%8A%A8%E7%94%BB
* https://vue3js.cn/interview