

# 借助Web Animations API实现JS keyframes动画

这篇文章发布于 2018年03月23日, 星期五, 00:25, 归类于 [JS API](#)。阅读 7812 次, 今日 6 次 [3 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=7438>

本文可全文转载, 但需得到原作者书面许可, 同时保留原作者和出处, 摘要引流则随意。

## 一、一句话解释Web Animations API

用一句话解释Web Animations API就是把CSS3实现的animation动画变成有JS代码实现。

各有优势。CSS3 Animations动画简单, 灵活, 复用性强, 非常适合静态动画效果; Web Animations参数由JS控制, 与单个元素绑定, 因此非常适合不固定的动态动画效果, 例如随机动画。

## 二、Web Animations API使用典型示意

语法如下:

```
var myAnimation = element.animate(keyframes, options);
```

其中 `keyframes` 对应CSS3中`@keyframes`中的声明块, `options` 对应 `animation-*` 属性及属性值。  
`myAnimation`为动画的Animation对象, 包含多种方法。

所有方法以及属性, 以及当前浏览器是否支持, 可参见下面信息 (翻译自[此CodePen](#)):

## 浏览器是否原生支持Web Animations API

S 支持 | N 不支持

- N `Element.animate()`支持
    - N 第一个参数支持数组关键帧类似这样: `[{opacity: 1}, {opacity: 0}]`
    - N 第一个参数作为对象, 且属性值是数组, 类似这样: `{ opacity: [1, 0] }`
    - N 单个keyframe ([案例](#))
    - N `id`
    - N `play()`
    - N `pause()`
    - N `reverse()`
    - N `cancel()`
    - N `finish()`
    - N `playState`
    - N `playbackRate`
    - N `pending`
    - N `currentTime/startTime`
    - N `timeline`
    - N `oncancel handler`
    - N `onfinish handler`
    - N `finished Promise`
    - N `ready Promise`
    - N

## Web Animations API使用demo

拿淡出再淡入效果举例，下面是一个常见属性出动的一个JS代码示意：

```
element.animate([
  { opacity: 0 },
  { opacity: 1, offset: 0.4 },
  { opacity: 0 }
], {
  duration: 3000,
  delay: 0,
  fill: 'forwards',
  easing: 'steps(8, end)',
  iterations: Infinity
});
```

最终效果如下GIF：



若要自己亲自零距离感受效果，您可以狠狠地点击这里：[Web Animations实现元素的淡入淡出demo](#)

## Web Animations API常用属性说明

**keyframes** 通常我们使用一个数组，数组值都是存在动画关键帧数据的纯对象，里面有CSS键值对，还有其他一些参数，例如 **offset** 对应CSS3 animation中的百分比，`{ opacity: 1, offset: 0.4 }` 等同于CSS中 `40% { opacity: 1; }`，也就是动画进程 **40%** 位置透明度为 **1**，这就是为什么上面GIF截图中红色圆出现速度要比消失快的原因。

**options** 可以直接是动画时间长度，又或者是对象值。例如：

```
element.animate([
  { opacity: 0 },
  { opacity: 1 }
], 3000);
```

就是个3秒淡出效果，不过是 **'linear'** 线性的，这个和CSS3 animation默认是 **'ease'** 类型动画不同。

如果 **options** 是使用纯对象作为参数，则基本上和CSS3 animation中的几个属性想对应，例如：**duration** 表示时长；**delay** 表示动画延迟执行时间；**fill** 表示动画前后保持的状态，**easing** 表示缓动类型，指为 **steps()** 时候表示断片分步走；**iterations** 表示动画执行次数，如果无限次，则使用JavaScript中的关键字 **Infinity**，注意，不要和CSS3动画中的 **infinite** 拼写搞混，最后一个字母不一样。

## 三、Web Animations语法文档

Web Animations还提供了一些非常方便来控制动画的方法。例如：

### **myAnimation.cancel()**

取消动画，会立即清除所有的关键帧效果。

### **myAnimation.finish()**

动画直接结束。

### **myAnimation.pause()**

动画暂停。

### **myAnimation.play()**

动画继续。

### **myAnimation.reverse()**

动画反过来播放，直到播放到动画开始时停止。如果动画完成或未播放，它将从头到尾播放。

### **myAnimation.updatePlaybackRate()**

在第一次同步播放位置后设置动画的速度。如果设置的值比 **1** 大，加速，如果比 **1** 小，则减速。相比直接设置 `playbackRate` 属性值，`updatePlaybackRate()` 方法不会出现动画因为速度变化突然跳一下的情况发生，因为方法内部做了优化。

事件接口如下：

### **myAnimation.ontancel**

动画取消触发的回调

### **myAnimation.onfinish**

动画结束触发的回调

个人觉得使用上要比CSS动画触发的 `animationend` 方法要简单些。

一些属性如下：

### **myAnimation.currentTime**

动画当前时间值，单位毫秒，无论动画暂停与否又会有。如果动画缺失时间线，或者未执行，或者未激活，则返回值是 `null`。

### **myAnimation.effect**

获取或设置动画效果，通常是个关键帧效果对象。如下截图效果：

```
> myAnimation.effect
< ▼ KeyframeEffect {composite: "replace", t
  composite: "replace"
  ▼ timing: AnimationEffectTiming
    delay: 0
    direction: "normal"
    duration: 3000
    easing: "steps(8)"
    endDelay: 0
    fill: "forwards"
    iterationStart: 0
    iterations: Infinity
    ► __proto__: AnimationEffectTiming
  ► __proto__: KeyframeEffect
```

### **myAnimation.finished**

只读。返回当前动画已完成的Promise。

```
> myAnimation.finished
< ▼ Promise {<pending>} ⓘ
  ► __proto__: Promise
    [[PromiseStatus]]: "pending"
    [[PromiseValue]]: undefined
```

### **myAnimation.id**

获取或者这是动画唯一的字符标识。默认获取的值是空字符串。

### myAnimation.pending

只读。动画当前是否是等待其他异步操作完成的等待状态，例如启动播放或者暂停一个正在跑的动画。

### myAnimation.playState

只读。当前动画播放的状态，例如播放中的动画的返回值就是 `'running'`。

### myAnimation.playbackRate

获取或设置动画播放速率。默认的线性动画的 `playbackRate` 值是 `1`。

### myAnimation.ready

只读。返回当前动画准备的Promise。

### myAnimation.startTime

获取或设置动画开始播放时的预定时间。

### myAnimation.timeline

获取或设置与此动画相关联的时间线。

## 四、结束语

### 兼容性

目前仅Chrome浏览器和Firefox浏览器原生支持，Firefox已经完全支持。至于iOS Safari和桌面Safari只能作为实验性质属性使用，但是没有关系，这并不影响我们在实际项目中使用，因为有polyfill: <https://github.com/web-animations/web-animations-js>

我刚测试了下，IE11浏览器也是OK的（主要功能和属性都支持），则Safari一定没问题，意味着移动端是可以用起来。

简单介绍，简单了解。关于Web Animations更深入的使用更实际的案例，我们以后有机会在分享。

感谢阅读，欢迎交流！

《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

（本篇完） // 想要打赏？点击[这里](#)。有话要说？点击[这里](#)。



« 玩转HTML5 Video视频WebVTT字幕使用样式与制作

借助SVG文字尺寸自动缩放甚至突破Chrome 12px限制 »

### 猜你喜欢

- jQuery-innerfade内部列表自动淡入淡出插件
- 纯CSS实现帅气的SVG路径描边动画效果
- 超级强大的SVG SMIL animation动画详解
- 再说CSS3 animation实现点点loading动画
- 如何使用Tween.js各类原生动画运动缓动算法
- 使用CSS offset-path让元素沿着不规则路径运动
- Canvas中颜色过渡动画效果的实现

- CSS3 Transitions, Transforms和Animation使用简介与应用展示
- web上渐进使用jQuery Mobile中animate相关CSS
- CSS3 box-shadow兼容loading效果兼IE10+ CSS Hack介绍
- 小tips: 点击页面出现富强、民主这类文字动画效果

分享到:        { 0 }

标签: animation, fade, Web Animations, 动画

## 发表评论（目前3条评论）

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

1. **yyI说道:**  
2018年05月24日 13:02  
母猪羞愧的低下了头  
[回复](#)



2. **敖爽说道:**  
2018年03月23日 10:13  
厉害厉害  
[回复](#)



3. **wktrf说道:**  
2018年03月23日 08:31  
高产如那啥  
[回复](#)



### 最新文章

- » 常见的CSS图形绘制合集
- » 粉丝群第1期CSS小测点评与答疑
- » 分享三个纯CSS实现26个英文字母的案例
- » 小tips: 纯CSS实现打字动画效果
- » CSS/CSS3 box-decoration-break属性简介
- » CSS :placeholder-shown伪类实现Material Design占位符交互效果

- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的？](#)
- » [周知：CSS -webkit-伪元素选择器不再导致整行无效](#)

#### 今日热门

- » [常见的CSS图形绘制合集](#) <sup>(178)</sup>
- » [粉丝群第1期CSS小测点评与答疑](#) <sup>(113)</sup>
- » [未来必热：SVG Sprite技术介绍](#) <sup>(111)</sup>
- » [HTML5终极备忘大全（图片版+文字版）](#) <sup>(85)</sup>
- » [让所有浏览器支持HTML5 video视频标签](#) <sup>(83)</sup>
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) <sup>(80)</sup>
- » [CSS3下的147个颜色名称及对应颜色值](#) <sup>(78)</sup>
- » [小tips: 纯CSS实现打字动画效果](#) <sup>(72)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(69)</sup>
- » [分享三个纯CSS实现26个英文字母的案例](#) <sup>(69)</sup>

#### 今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) <sup>(76)</sup>
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) <sup>(64)</sup>
- » [看，for..in和for..of在那里吵架！](#) <sup>(60)</sup>
- » [是时候好好安利下LuLu UI框架了！](#) <sup>(47)</sup>
- » [原来浏览器原生支持JS Base64编码解码](#) <sup>(35)</sup>
- » [妙法攻略：渐变虚框及边框滚动动画的纯CSS实现](#) <sup>(33)</sup>
- » [炫酷H5中序列图片视频化播放的高性能实现](#) <sup>(31)</sup>
- » [CSS scroll-behavior和JS scrollIntoView让页面滚动平滑](#) <sup>(30)</sup>
- » [windows系统下批量删除OS X系统.DS\\_Store文件](#) <sup>(26)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(26)</sup>

#### 猜你喜欢

- [jQuery-innerfade内部列表自动淡入淡出插件](#)
- [纯CSS实现帅气的SVG路径描边动画效果](#)
- [超级强大的SVG SMIL animation动画详解](#)
- [再说CSS3 animation实现点点点loading动画](#)
- [如何使用Tween.js各类原生动画运动缓动算法](#)
- [使用CSS offset-path让元素沿着不规则路径运动](#)
- [Canvas中颜色过渡动画效果的实现](#)
- [CSS3 Transitions, Transforms和Animation使用简介与应用展示](#)
- [web上渐进使用jQuery Mobile中animate相关CSS](#)
- [CSS3 box-shadow兼容loading效果兼IE10+ CSS Hack介绍](#)
- [小tips: 点击页面出现富强、民主这类文字动画效果](#)