

HTML5/CSS3时代，我们要在web里做动画选择其实已经很多了：

你可以用CSS3的animation+keyframes;

你也可以用css3的transition;

你还可以用通过在canvas上作图来实现动画，也可以借助jQuery动画相关的API方便地实现;

当然最原始的你还可以使用window.setTimeout()或者window.setInterval()通过不断更新元素的状态位置等来实现动画，前提是画面的更新频率要达到每秒60次才能让肉眼看到流畅的动画效果。

现在又多了一种实现动画的方案，那就是还在草案当中的window.requestAnimationFrame()方法。

初识requestAnimationFrame

来看[MDN](#)上对其给出的诠释：

The window.requestAnimationFrame() method tells the browser that you wish to perform an animation and requests that the browser call a specified function to update an animation before the next repaint. The method takes as an argument a callback to be invoked before the repaint.

window.requestAnimationFrame() 将告知浏览器你马上要开始动画效果了，后者需要在下次动画前调用相应方法来更新画面。这个方法就是传递给window.requestAnimationFrame()的回调函数。

也可这个方法原理其实也就跟setTimeout/setInterval差不多，通过递归调用同一方法来不断更新画面以达到动起来的效果，但它优于setTimeout/setInterval的地方在于它是由浏览器专门为动画提供的API，在运行时浏览器会自动优化方法的调用，并且如果页面不是激活状态下的话，动画会自动暂停，有效节省了CPU开销。

基本语法

可以直接调用，也可以通过window来调用，接收一个函数作为回调，返回一个ID值，通过把这个ID值传给window.cancelAnimationFrame()可以取消该次动画。

```
requestAnimationFrame(callback) //callback为回调函数
```

一个简单的例子

模拟一个进度条动画，初始div宽度为1px,在step函数中将进度加1然后再更新到div宽度上，在进度达到100之前，一直重复这一过程。

为了演示方便加了一个运行按钮（看不到例子请刷新）。

0%

Run

```
<div id="test" style="width:1px;height:17px;background:#0f0;">0%</div>
<input type="button" value="Run" id="run"/>
```



```
window.requestAnimationFrame = window.requestAnimationFrame || window.mozRequestAnimationFrame ||
window.webkitRequestAnimationFrame || window.msRequestAnimationFrame;
var start = null;
var ele = document.getElementById("test");
var progress = 0;

function step(timestamp) {
    progress += 1;
    ele.style.width = progress + "%";
    ele.innerHTML=progress + "%";
    if (progress < 100) {
        requestAnimationFrame(step);
    }
}
requestAnimationFrame(step);
document.getElementById("run").addEventListener("click", function() {
    ele.style.width = "1px";
    progress = 0;
    requestAnimationFrame(step);
}, false);
```

