

14-第十四章 运动

—requestAnimationFrame()

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 `window.requestAnimationFrame()` 方法。

1、基本方法使用

```
1. var num = 0;
2. function fn(){
3.     num++;
4.     document.title = num;
5.     requestAnimationFrame(fn); //在内部根据用户浏览器(电脑性能)情况,重复调用 fn,
6.
7. }
8. //requestAnimationFrame(fn); //调用一次fn;
9. //也可以如下
10. fn();
```

2、`cancelRequestAnimationFrame(timer)`

添加`manage`定时管理器

```

1. var num = 0;
2. var timer;
3. function fn(){
4.     num++;
5.     document.title = num;
6.     timer = requestAnimationFrame(fn); //在内部根据用户浏览器情况,重复调用
    fn,
7.     if(num==250){
8.         cancelAnimationFrame( timer );
9.     }
10.
11. }
12. //requestAnimationFrame(fn); //调用一次fn;
13. //也可以如下
14. fn();
15.

```

把 `requestAnimationFrame(fn)` 赋值给 `timer` , `timer` 就是 定时管理器

3、`RequestAnimationFrame()` 兼容

```

1. //requestAnimationFrame 兼容
2. window.requestAnimationFrame = window.requestAnimationFrame || wi
    ndow.mozRequestAnimationFrame || window.webkitRequestAnimationFra
    me || window.msRequestAnimationFrame || function (fn){
3.     setTimeout(fn,1000/60);
4.
5. };
6.
7. //cancelAnimationFrame 兼容
8. window.cancelAnimationFrame = window.cancelAnimationFrame || wind
    ow.mozCancelAnimationFrame || window.webkitCancelAnimationFrame
    || window.msCancelAnimationFrame || clearTimeout;
9.

```

二、速度版运动框架

```

1.     var Obox = document.getElementById('box');
2.     move(Obox,'left',20,-6);//调用move函数,指定传递实参
3.     function move(obj,attr,target,speed){//指定接受形参
4.         target = parseFloat(target);//转化为number
5.         var init = parseFloat( getStyle(obj,attr));//获取初始样
        式值
6.         var time;//动画管理器
7.         (function rQAF(){
8.             init+= speed;
9.             time = window.requestAnimationFrame(rQAF);//调用r
                equestAnimationFrame函数
10.            if( Math.abs(target-init)<=Math.abs(speed) ){//用
                绝对值判断是否到达目标值
11.                init = target;//
12.                window.cancelAnimationFrame(time);//删除reques
                tAnimationFrame动画
13.            }
14.            obj.style[attr] = init+'px';//设置样式
15.        })()
16.
17.
18.    }
19.    function getStyle(obj,attr){//定义获取样式函数
20.        return window.getComputedStyle? window.getComputedSty
                le(obj)[attr]:obj.currentStyle[attr];
21.    }

```

三、时间版运动框架

```

1.      move(Obox, 'left', '800px', 1000); //调用move函数,指定传递实参
2.      function move(obj, attr, target, tar_t){ //指定接受形参
3.          target = parseFloat(target); //转化为number
4.          var init = parseFloat( getStyle(obj, attr)); //获取初始样
      式值
5.          var init_t = new Date(); //获取开始时间
6.          var style_val;
7.          (function rQAF(){
8.              var cur_t = new Date() - init_t; //获取动画时长
9.              var prop = cur_t / tar_t; //获取动画执行时长与动画预设总时
      间比值
10.             if( prop >= 1 ) { //动画执行时长与动画预设总时间比值大于等于
      1时,
11.                 prop = 1;
12.             } else {
13.                 window.requestAnimationFrame(rQAF);
14.             }
15.             style_val = (target - init) * prop; //根据时间比例获取运动
      路程比例
16.             obj.style[attr] = init + style_val + 'px'; //设置样式
17.         })()
18.
19.
20.     }
21.     function getStyle(obj, attr){ //定义获取样式函数
22.         return window.getComputedStyle? window.getComputedStyle
      le(obj)[attr]: obj.currentStyle[attr];
23.     }

```

四、时间加速版运动框架

加速度

例：假如两辆汽车开始静止，均匀地加速后，达到10m/s的速度，A车花了10s，而B车只用了5s。它们的速度都从0变为10m/s，速度改变了10m/s。所以它们的速度变化量是一样的。但是很明显，B车变化得更快一些。我们用加速度来描述这个现象：B车的加速度（ $a = \Delta v / \Delta t$ ，其中的 Δv 是速度变化量）>A车的加速度。显然，当速度变化量一样的时候，花时间较少的B车，加速度更大。也就是说B车的启动性能相对A车好一些。因此，加速度是表示物体速度变化快慢的物理量。

s 是距离，**v_f** 终速度，**t** 是加速的时间。**v_f=at**，**a** 是加速度，替换可得这是匀加速运动距离的通常表达式

$$s = \frac{1}{2} at^2$$

```
1.      move(Obox,'left','800px',1500);//调用move函数,指定传递实参
2.      function move(obj,attr,target,tar_t){//指定接受形参
3.          target = parseFloat(target);//转化为number
4.          var init = parseFloat( getStyle(obj,attr));//获取初始样
   式值
5.          var init_time = new Date();//获取开始时间
6.          var sty_v;
7.          var a = 2*(target-init)/Math.pow(tar_t,8);//获取加速度
8.          (function rQAF(){
9.              var cur_t = new Date()- init_time;//获取动画时长
10.             if( cur_t>=tar_t ){//动画执行时长与动画预设总时间比值大
   于等于1时,
11.                 //style_val = target;//
12.                 cur_t=tar_t;
13.                 // window.cancelAnimationFrame(time);//删除requ
   estAnimationFrame动画
14.             }else{
15.                 window.requestAnimationFrame(rQAF);
16.             }
17.             sty_v = a*Math.pow(cur_t,8)/2;//根据时间比例获取运动
   路程比例
18.             obj.style[attr] = init+ sty_v+'px';//设置样式
19.         })()
20.
21.
22.     }
23.     function getStyle(obj,attr){//定义获取样式函数
24.         return window.getComputedStyle? window.getComputedSty
   le(obj)[attr]:obj.currentStyle[attr];
25.     }
```

五、多值时间版运动框架

```

1.     move(Obox,{ width : '200px',
2.                 height : "200px",
3.                 left : "800px",
4.                 opacity : 1},2000,function(){ console.log('o
k')});//调用move函数,指定传递实参
5.
6.     function move(obj,json,targ_t,callback){//指定接受形参
7.         var target = {}/*目标值*/,init ={}/*初始值*/,styleV/*样
式*/;
8.         for(var attr in json){
9.             target[attr] = parseFloat( json[attr] );//目标值转
化为number类型
10.            init[attr] = parseFloat( getStyle(obj,attr));//获
取初始样式值并转化为number类型
11.        }
12.        var init_t = new Date();//获取开始时间
13.        (function rQAF(){
14.            var cur_t = new Date()-init_t;//获取当前时间与开始时
间的差值--动画执行时长
15.            if( cur_t>=targ_t){//判断动画执行时长是否大于预设目标
16.                cur_t=targ_t;//让动画执行时长等于预设目标
17.            }else{
18.                window.requestAnimationFrame(rQAF);//调用rQAF
函数一次
19.            }
20.
21.            for(var attr in json){
22.                var a = 2*(target[attr]-init[attr])/Math.po
w(targ_t,2);//获取对象属性的加速度
23.                styleV = a*Math.pow(cur_t,2)/2;//根据动画时长设
置样式
24.
25.
26.                if(attr=='opacity'){//
27.                    obj.style[attr] = init[attr]+styleV;//设置
样式
28.                    obj.style.filter = 'alpha(opacity='+style
V*100+')';//opacity兼容
29.
30.                }else{
31.                    obj.style[attr] = init[attr]+styleV+'p
x';//设置样式
32.                }
33.
34.            }
35.            cur_t==targ_t?callback&&callback.call(obj):'';//根
据动画时长是否等于了预设目标, true执行回调函数, 并绑定this

```

```
36.         })()
37.
38.
39.
40.     }
41.     function getStyle(obj,attr){//定义获取样式函数
42.         return window.getComputedStyle? window.getComputedStyle
43.         le(obj)[attr]:obj.currentStyle[attr];
    }
```