动画初探[中]

学习要点：

1.问题所在

2.设置代码

本节课，我们要讲一下JavaScript在动画中的实现，让大家了解动画是怎样形成的。

一．问题所在

最简单的动画已经可以运动，但还包含着一些问题，

二．设置代码

问题1：如果目标长度并不等于移动到目标的长度，比如按照每50毫秒7像素，那么

可能就达到不一个整数可能会多出一个或几个像素，所以我们判断的时候，用大于等于比较妥当，否则会一直运动下去。

if (getStyle(element, attr) >= target) {}

问题2：怎么才能让移动到目标值到达指定的目标值停止，而不是多出一个或几个像素。

if (getStyle(element, attr) >= target) {

element.style[attr] = target + 'px';

clearInterval(timer);

} else {

element.style[attr] = getStyle(element, attr) + step + 'px';

}

问题3：虽然可以剪掉多余的像素，但剪掉的时候，会后退一下，很突兀。

element.style[attr] = getStyle(element, attr) + step + 'px';

if (getStyle(element, attr) >= target) {

element.style[attr] = target + 'px';

clearInterval(timer);

}

问题4：如果通过事件，比如点击等可能会导致创建多个定时器，速度就会翻倍变快。

clearInterval(window.timer);

timer = setInterval(function () {

element.style[attr] = getStyle(element, attr) + step + 'px';

if (getStyle(element, attr) >= target) {

element.style[attr] = target + 'px';

clearInterval(timer);

}

}, t);

PS：对于每多少毫秒执行一次定时器，这个参数我们可以内置，因为绝大部分情况下，只要一开始设定好，一般来说不需要改变。并且，如果修改了，整体加速或者整体减速。

问题5：可以设置向右或向下移动，无法向左或向上移动。并且之前用负数有点变扭。

if (getStyle(element, attr) > target) step = -step;

clearInterval(window.timer);

timer = setInterval(function () {

element.style[attr] = getStyle(element, attr) + step + 'px';

if (step > 0 && getStyle(element, attr) >= target) {

element.style[attr] = target + 'px';

clearInterval(timer);

} else if (step < 0 && getStyle(element, attr) <= target) {

element.style[attr] = target + 'px';

clearInterval(timer);

}

}, t);

问题6：当点击一次按钮时，运动一次，第二次点击时，就不运动了。主要原因是已经到目标点了。所以，我们每次点击的时候可以手工重置一下。

element.style[attr] = start + step + 'px';

start += step;

PS：但这种方法需要对应CSS的位置，如果不一致，一开始会闪烁一下。

问题7：参数太多，搞不清位置，我们通过封装传参来解决这个问题。

Base.prototype.animate = function (obj) {

for (var i = 0; i < this.elements.length; i ++) {

var element = this.elements[i];

var attr = obj['attr'] == 'x' ? 'left' : obj['attr'] == 'y' ? 'top' : 'left';

var start = obj['start'] != undefined ? obj['start'] : getStyle(element, attr);

var t = obj['t'] != undefined ? obj['t'] : 50;

var step = obj['step'] != undefined ? obj['step'] : 10;

var target = obj['alter'] + start;

if (start > target) step = -step;

element.style[attr] = start + 'px';

　　 clearInterval(window.timer);

timer = setInterval(function () {

element.style[attr] = getStyle(element, attr) + step + 'px';

if (step > 0 && getStyle(element, attr) >= target) {

element.style[attr] = target + 'px';

clearInterval(timer);

} else if (step < 0 && getStyle(element, attr) <= target) {

element.style[attr] = target + 'px';

clearInterval(timer);

}

}, t);

}

return this;

};

PS：我们把目标值，改成了增量值，这样在调用的时候会更加清晰。attr属性值采用x表示横轴，y表示纵轴，这样更加符合语义，更加清晰。当然，对于极少部分人群会不知道x轴和y轴的，你也可以用hengzhou和zongzhou来代替，原理一样。