**动画的常规优化技巧**

1、边界判断：一定加上步长进行判断

2、setTimeout每一次执行动画前，先把之前的没用的清除掉

3、对于定时器返回值timer，把它写在元素的自定义属性上，来避免全局变量的过多使用或避免不同的元素对象之间产生冲突

**什么是jQuery：**

它是一个JS文件，它里面包含了我们项目中常用的JS方法和属性(都是用原生的JS实现的)，换句话说就是一个类库(和我们自己写的DOM库，包含了很多常用的方法，并且方法把兼容问题都处理了，以后引进来直接用即可)

**jQuery实现的原理：**这个类库是用构造函数模式进行封装的，而jQuery是一个类，选择器$(),相当于new jQuery创建了jQuery这个类的一个实例

$()可以传递两个参数：selector(选择器内容)、context(上下文，不传默认是document)

属性、css、动画、文档处理、事件等提供的属性和方法都是定义在jQuery原型上的-->jQuery.prototype，属于实例的公有方法，并且每一个jQuery实例都可以使用这些方法

**jQuery选择器原理**

ID选择器：通过ID获取一个jQuery对象(jQuery的一个实例)(不是一个元素对象)

var $div1 = $("#div1");

我想获取一个#div1元素对象--->将jQuery对象转换为原生的js对象，但是真正的意思是在jQuery的一个实例中，把属性名为0的属性值拿出来用

var oDiv1 = $div1[0];

var oDiv1 = $div1.get(0);//在jQuery的原型上提供了一个get方法，通过索引获取指定的元素对象

**this和$(this)的区别：**this是原生的对象，$(this)是jQuery对象，this不能直接使用jQuery的方法

$("#div1").addClass("dd").html("珠峰培训"); //jQuery中的链式写法 每一次执行完方法，都把当前的jQuery实例返回(return this;)

jQuery有一部分常用的方法是定义在jQuery.prototype上的(属性、css、文档处理、事件、动画、筛选...)，这些方法只有通过选择器创建一个实例才能够使用

$("#div1").addClass("dd");

还有一部分是把jQuery当做一个普通的对象，增加的属性名和属性值(ajax、工具)

$.ajax();

$.each();

jQuery中提供了扩展原型和私有对象属性上的方法

当做对象扩展的：通常是在完善类库

$.extend({

aryDistinct: function (ary) {

var obj = {};

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

var cur = ary[i];

if (obj[cur] == cur) {

ary[i] = ary[ary.length - 1];

ary.length -= 1;

i--;

continue;

}

obj[cur] = cur;

}

obj = null;

return ary;

}

});

var ary = [1, 2, 3, 2, 1, 2, 3, 4, 2, 3, 4, 2, 1];

var newAry = $.aryDistinct(ary);

console.log(newAry);

扩展到jQuery原型上的：我们通常用来写插件

$.fn.extend({

a: function () {

alert(1);

}

});

$().a();

当一个页面出现多个类库，都是用到了$，那么我们把jQuery的使用权转让给别的

**var j = $.noConflict();//转让使用权，并且起了个别名j，以后用j代替$**

$("#div1").addClass("dd");//Error:$ is not a function

jQuery("#div1").addClass("dd");

j("#div1").addClass("dd");