# jQuery框架

第1天课堂笔记（本课程共3天）

# 目录

[jQuery框架](#_Toc10557)

[目录](#_Toc19180)

[一、传统轮播和呼吸轮播复习](#_Toc30525)

[1.1 传统轮播](#_Toc29920)

[1.2 三位置传统轮播图](#_Toc4131)

[1.3 呼吸轮播（交叉淡入淡出轮播）](#_Toc28164)

[二、异形滚动](#_Toc16599)

[三、jQuery框架](#_Toc20647)

[3.1 JavaScript编程比较恶心的地方](#_Toc24397)

[3.2 jQuery就是轮子轮子轮子轮子轮子的集合](#_Toc13387)

[3.3 简介](#_Toc16624)

[四、jQuery整体感知](#_Toc3289)

[4.1 选择问题轻松解决](#_Toc21893)

[4.2 样式问题轻松解决](#_Toc12406)

[4.3 动画的问题轻松解决](#_Toc4658)

[4.4 批量添加监听、节点关系](#_Toc8449)

[五、$()函数](#_Toc30076)

[5.1 jQuery对象不是原生JS对象](#_Toc24194)

[5.2 引号问题](#_Toc29823)

[5.3 支持的选择器](#_Toc29200)

[5.4 筛选器](#_Toc25266)

[六、CSS函数](#_Toc7423)

[七、animate函数](#_Toc12525)

[八、事件监听](#_Toc13903)

先不直接介绍jQuery，先把之前遗留的问题讲了

# 一、传统轮播和呼吸轮播复习

## 1.1 传统轮播

是一个a标签，但是我们不希望点击之后有任何默认的事情，默认事情通常是刷新当前页：

|  |
| --- |
| 1. <a href="javascript:;" class="leftBtn" id="leftBtn"></a> |

最最关键的就是右按钮、左按钮的策略：

右按钮，处理最后一张的时候，先拉后瞬移：

|  |
| --- |
| 1. rightBtn.onclick = function(){ 2. idx ++; 3. animate(m\_unit, {.......},500,function(){ 4. if(idx > 4){ 5. idx = 0; 6. m\_unit.style.left = “0px”; 7. } 8. }) 9. } |

左按钮，处理第一张的时候，先瞬移后拉动：

|  |
| --- |
| 1. leftBtn.onclick = function(){ 2. idx --; 3. if(idx < 0 ){ 4. idx = 4; 5. m\_unit.style.left = -560 \* 5 + "px"; 6. } 7. animate(m\_unit, {.....},500) 8. } |

## 1.2 三位置传统轮播图

页面效果，随着人们的认识深入，就产生更多的原理。

传统轮播的“火车法”的问题是，小圆点点击的时候，拉动的图片，跳跃性大。

比如现在是0号图，点击3号小圆点，就会看见1号、2号，停留在3号图。不是平滑过渡。

三位置传统轮播图，特点是，所有的li不是火车，而是都绝对定位。他们的left值只有三种可能：

-560 0 560



## 1.3 呼吸轮播（交叉淡入淡出轮播）

摞在一起，不管点击哪个按钮，老图淡出，新图淡入。点击小圆点，老图淡出，新图淡入。

我们讲一个bug，就是元素opacity:0了，也能够被点击。

解决办法就是，页面上有且又有一个li的display:block；

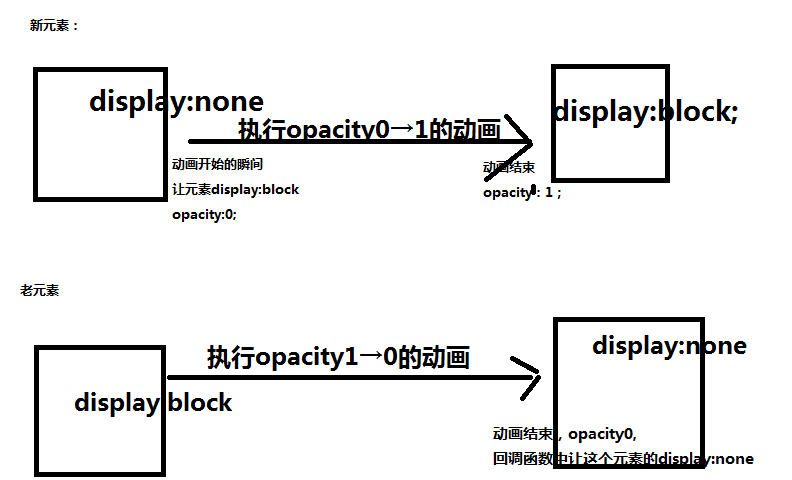
当点击右按钮的时候：

当前这个图片执行opacity1→0的动画，动画结束之后，让这个元素display:none;

新图片执行opacity0→1的动画，动画开始之前，瞬间让新图片display:block;然后再动画

一定要牢记：呼吸轮播动画的起点是display:block； 动画的终点是display:none;

但是JS是没法过渡display属性的，所以我们就往中间加上了opacity



# 二、异形滚动

每次老师吹哨的时候：

小明擦玻璃

小红拖地

小刚扫地

让小明改名叫小红

让小红改名叫小刚

让小刚改名叫小明

所以老师每次吹哨，这三个学生依次做不同的劳动。

先运动，然后改变身份。

lisArr = [lis.stateA,lis.stateB,lis.stateC,lis.stateD……]

点击一下按钮，我们永远让数组的第0项瞬移到6号位置，第1项动画变为0，第2项动画变为1……

lisArr.push(lisArr.shifi());

[lis.stateB,lis.stateC,lis.stateD……lis.stateA]

# 三、jQuery框架

## 3.1 JavaScript编程比较恶心的地方

恶心1：选择元素麻烦，全线兼容的方法只有getElementById()和getElementsByTagName()两个。其他的方法是不都兼容的。getElementsByClassName()通过类名选择元素，IE9开始兼容。

恶心2：样式操作麻烦，得到原生样式，需要我们自己造轮子fetchComputedStyle()

恶心3：动画麻烦，需要我们自己造轮子animate();

恶心4：批量控制麻烦，大量出现的for循环语句；排他操作麻烦

恶心5：HTML节点操作麻烦

恶心的东西，能用“轮子”来解决，事实上我们已经造了两个轮子：fetchComputedStyle、animate。

能不能把所有轮子组合起来，成为一个框架呢？jQuery就是这样的一个东西。

## 3.2 jQuery就是轮子轮子轮子轮子轮子的集合

jQuery是DOM编程领域的霸主，极大的简化了原生JavaScript的DOM编程。

jQuery中含有丰富的轮子，完美解决了选择元素难、样式难、动画难、批量操作难等各种兼容问题，**让工程师只需要思考业务，而不必思考细枝末节的兼容问题**。

John Resig jQuery的第一创始人



## 3.3 简介

**官网：**www.jquery.com



**口号：**写更少的代码，做更多的事情

**官方的自我介绍：**

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

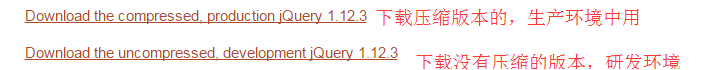
简单翻译：jQuery是一个快速、小型的、特性很多的JS库，它把很多事儿都变得简单。jQuery是免费的、开源的。

**jQuery有两条版本线**：1.X、2.X。

jQuery 2.x has the same API as jQuery 1.x, but does not support Internet Explorer 6, 7, or 8.

jQuery2.X和1.X的功能完全一致，API完全一致，但是2.X不兼容IE6、7、8。

下载的时候有两个选择：



# 四、jQuery整体感知

## 4.1 选择问题轻松解决

$()函数就是jQuery的核心函数，query就是选择的意思，也就是说jQuery的核心招牌功能就是选择元素：

|  |
| --- |
| 1. **$**("#box ul li.haha span").css("background-color","red"); |

语法：

|  |
| --- |
| 1. $(“选择器”) |

选择的就是所有符合条件的元素，而不是一个。

$可以用jQuery来代替，$和jQuery是同一个函数：

|  |
| --- |
| 1. **jQuery**("#box ul li.haha span").css("background-color","red"); |

一辈子就和document.getElmentById()无缘了，拜拜了你内！！

jQuery选择的元素，所有浏览器兼容！

## 4.2 样式问题轻松解决

得到样式（得到计算后的样式）：

|  |
| --- |
| 1. $(".box").css("width"); |

通过$()函数选择出来的东西，都是jQuery对象，所有的jQuery对象，都可以继续打点调用css函数，css函数已经封装了计算后的样式。

设置样式：

|  |
| --- |
| 1. $(".box").css({ 2. "width":700, 3. "height" : 300, 4. "opacity" : 0.4 5. }); |

语法

|  |
| --- |
| 1. $(“选择器”).css(JSON); |

## 4.3 动画的问题轻松解决

jQuery内部含有一个运动框架，比我们上课写的牛逼十万倍！

|  |
| --- |
| 1. $(".box").animate({"left":900},4000,function(){ 2. alert("运动完成"); 3. }); |

## 4.4 批量添加监听、节点关系

所有的li都有监听了：

|  |
| --- |
| 1. $(".circles ol li").mouseenter(function(){ 2. //自己变红，自己的兄弟恢复为橙色   $(this).css("background-color","red").siblings().css("background-color","orange");   1. }); |

siblings()表示兄弟节点。

# 五、$()函数

## 5.1 jQuery对象不是原生JS对象

$()函数，是招牌功能，能够根据CSS选择元素。

比如：

|  |
| --- |
| 1. $("#box") |

选择页面上id为box的盒子。

**注意，选择出来的东西，是一个类数组对象，是jQuery自己的对象，这个jQuery对象后面不能跟着原生JS的语法：**

|  |
| --- |
| 1. ~~$("#box").style.backgroundColor = "red";~~ |

因为.style.backgroundColor是原生JS语法，$()原则的对象是jQuery对象，不能跟着原生。

所以，如果想把jQuery对象，转为原生JS对象，要加[0]就行了：

|  |
| --- |
| 1. $("#box")[0].style.backgroundColor = "red"; |

$等价于jQuery

|  |
| --- |
| 1. jQuery("#box") 2. 等价于 3. $("#box") |

## 5.2 引号问题

|  |
| --- |
| 1. $("选择器") |

注意引号不能丢！！在jQuery世界中，只有三个东西不能加引号，其他必须加引号：

|  |
| --- |
| 1. $(this) 2. $(document) 3. $(window) |

上述的三个东西，不能有引号，干吗用的，后面说。

## 5.3 支持的选择器

jQuery支持所有CSS2.1的选择器：

|  |
| --- |
| 1. $("p") 2. $(".box") 3. $("#box") 4. $("#box ul li") 5. $("li.special") 6. $("ol , ul") 7. $("\*") |

也支持部分CSS3的选择器，我们CSS3的选择器在CSS3课程介绍，所以也不讲了。

## 5.4 筛选器

这些都是关于序号的：

|  |
| --- |
| 1. $("p") 所有的p 2. $("p:first") 第一个p 3. $("p:last") 最后一个p 4. $("p:eq(3)") 下标为3的p 5. $("p:lt(3)") 下标小于3的p 6. $("p:gt(3)") 下标大于3的p 7. $("p:odd") 下标是奇数的p 8. $("p:even") 下标是偶数的p |

特别的，eq可以单独提炼为方法，可以连续打点：

|  |
| --- |
| 1. $("p").eq(3).animate({"width":400},1000); |

等价于

|  |
| --- |
| 1. $("p:eq(3)").animate({"width":400},1000); |

提炼出来的好处是，可以用变量

|  |
| --- |
| 1. var a = 3; 2. $("p").eq(a).animate({"width":400},1000); |

# 六、CSS函数

css函数可以读样式，可以设样式。

读样式，可以读取计算后样式，写一个参数，是不是驼峰，无所谓，但是必须加引号：

|  |
| --- |
| 1. $("p:first").css("background-color"); 2. $("p:first").css("backgroundColor"); |

设置样式，设置样式，有两种语法，**如果你只想设置一个样式，逗号隔开k和v**：

|  |
| --- |
| 1. $("p:odd").css("backgroundColor"**,**"blue"); |

**如果想设置很多样式，就写JSON:**

|  |
| --- |
| 1. $("p:odd").css(**JSON**); |

所有的数值，不需要单位：

|  |
| --- |
| 1. $("p:lt(4)").css({ 2. "width" : 20, 3. "height" : 20, 4. "backgroundColor" : "red" 5. }) |

当然，你也可以这样，不厌其烦的用逗号：

|  |
| --- |
| 1. $("p:lt(4)").css("width",20); 2. $("p:lt(4)").css("height",20); 3. $("p:lt(4)").css("backgroundColor","red"); |

特别的，还支持+=写法：

|  |
| --- |
| 1. $("p:eq(5)").css("width","+=20px"); |

# 七、animate函数

动画

|  |
| --- |
| 1. $("p").animate({"left":1000},2000,function(){ 2. $(this).css("background-color","red"); 3. }); |

我们骄傲的告诉大家，我们封装的animate语法基本和jQuery一样，只不过jQuery是对象打点：

|  |
| --- |
| 1. $("选择器").animate(终点JSON,动画时间,回调函数); |

有没有缓冲呢，有，jQuery需要插件来完成，我们日后说。

jQuery默认不是匀速，是easeInOut

和我们封装的框架不一样，jQuery默认有一个处理机制，叫做动画排队。当一个元素接收到了两个animate命令之后，后面的animate会排队：

|  |
| --- |
| 1. $("p").animate({"left":1000},2000); 2. $("p").animate({"top":400},2000); |

先2000毫秒横着跑，然后2000毫秒竖着跑。动画总时长4000。

如果想让元素斜着跑，就是同时变化left和top，就写在同一个JSON里面：

|  |
| --- |
| 1. $("p").animate({"left":1000,"top":400},2000); |

不同的元素，不排队，是同时的。

# 八、事件监听

jQuery颠覆了我们的行文习惯：

|  |
| --- |
| 1. $(".box1").click(function(){ 2. //点击box1之后做的事情 3. }); |

事件名一律不写on。特别的，鼠标进入改成了mouseenter，鼠标离开改为了mouseleave。

|  |
| --- |
| 1. oDiv.onclick = function(){ 3. } |

可以看做等价于：

|  |
| --- |
| 1. $("#box").click(function(){ 3. }) |