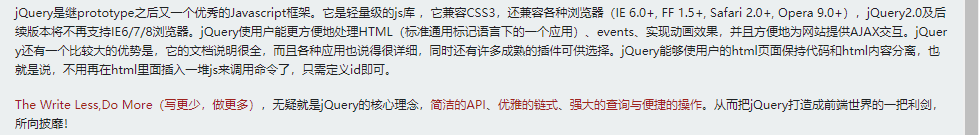
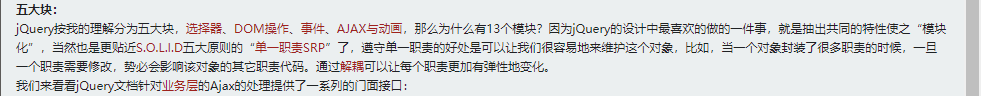
# jQuery源码解析（架构与依赖模块）

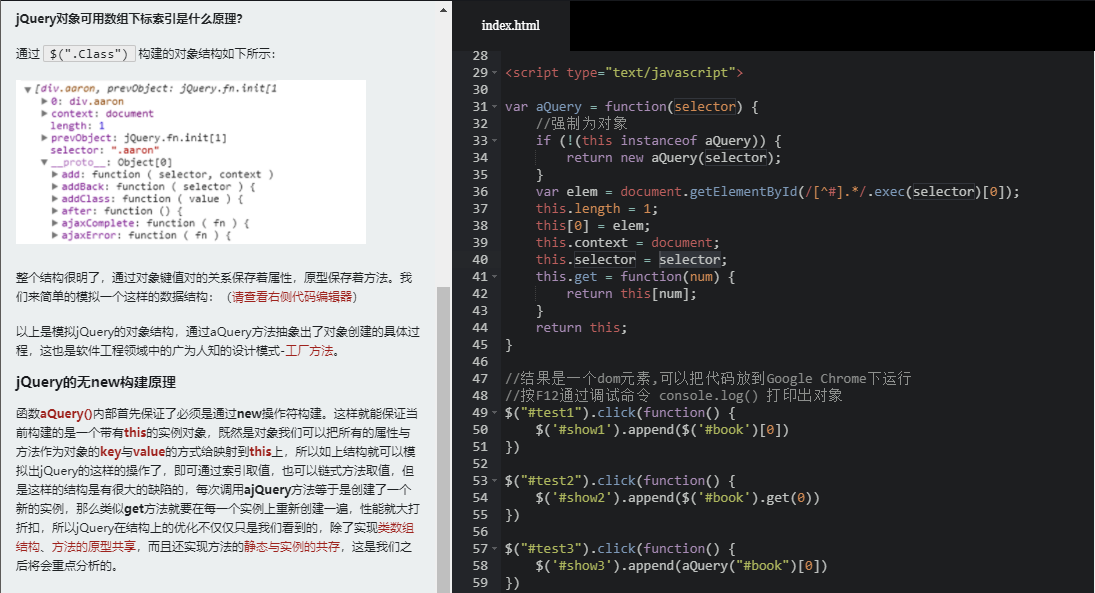
1. jQuery源码解析（架构与依赖模块）：<https://www.imooc.com/learn/172>；
2. 设计理念-写更少做更多



1. 整体架构



1. 设计模式 - 工厂方法 & jQuery的类数组对象结构



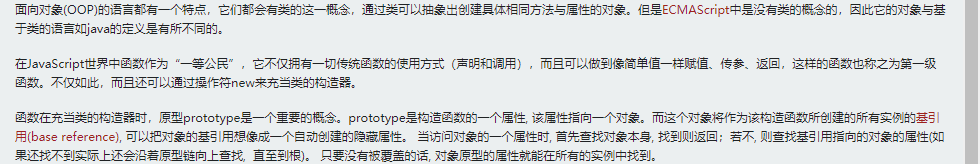
1. 匿名函数 & 自执行



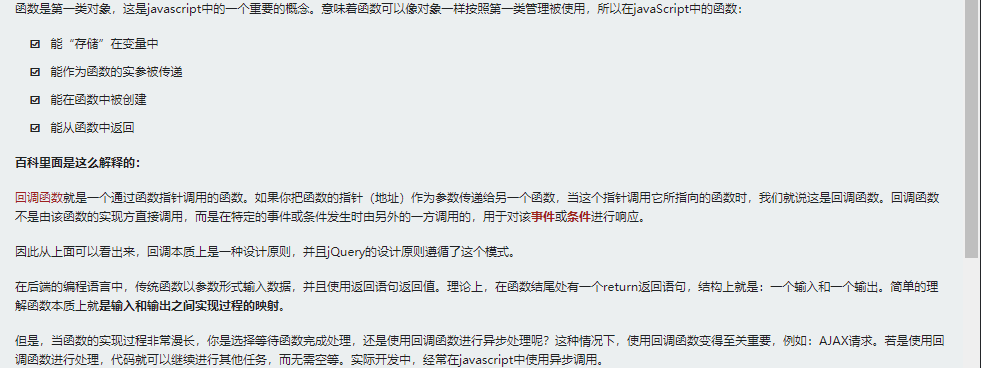
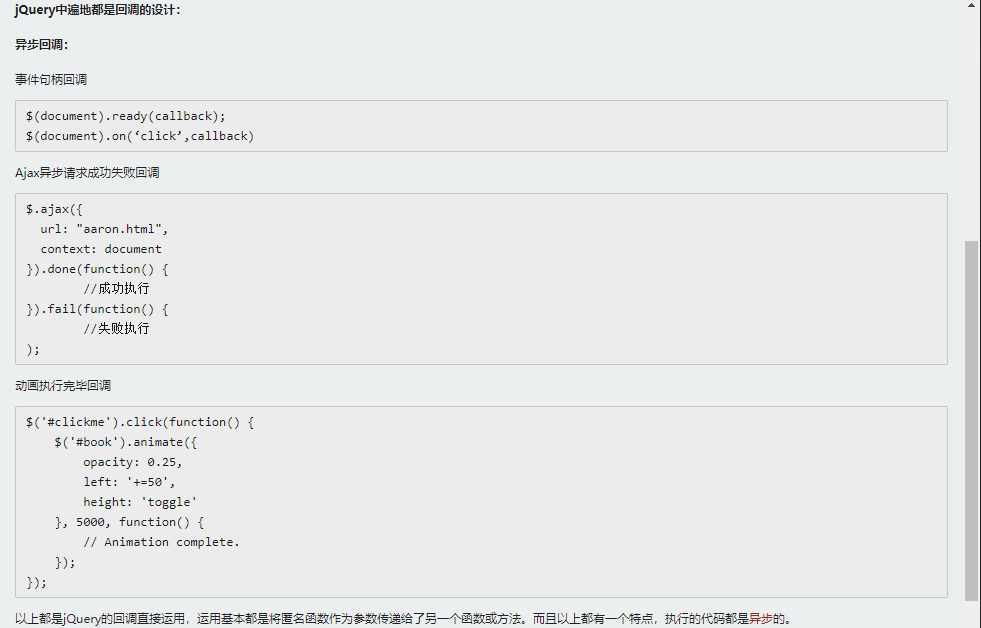
1. jQuery中ready与load事件



1. 对象构建

1. 回调函数

1. 设计模式 - 观察者模式



· 这种模式感觉就是说我准备了一百种菜给你，但不知道你想吃哪几种，我给个盘子给你看着夹吧，而不是全部菜都塞给你吃；

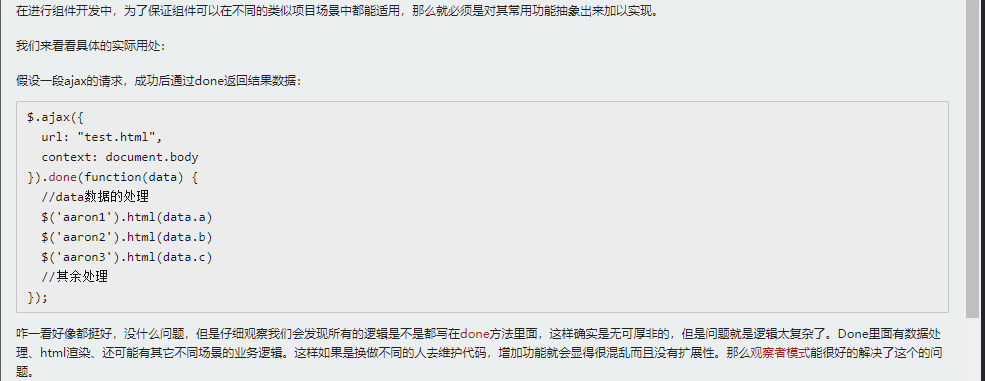
· Observable是被观察者，里面的callbacks数组中的每一个都是一个观察者，每一个元素都是函数形式；

· 将观察者都搞在一个集合中，当被观察者发生变化，就直接找到这个集合，并运行其中的函数；

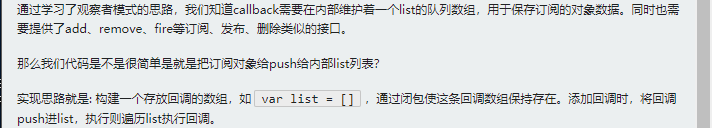
· 跟多个函数放到一个函数中一次执行有什么不同，不明白？到时是对的，但是在调用的时候，只用放一次，而且放到一个函数中这个过程是通过方法来添加的，而不是直接在代码中添加。所谓的“组件”的思想。而且，最重要的是，注意这句话：当一个对象的改变需要同时改变其它对象，并且它不知道具体有多少对象需要改变的时候，就应该考虑使用观察者模式。 如果一次把多个函数放到一个函数中，不仅仅可能造成这个函数过于庞大（比如要同时执行几十上百个函数），而且每次十分不灵巧，函数变得不够“单纯”，换句话说，太low了。。。；

· 跟多个函数放到一个函数中一次执行有什么不同，不明白？如果把多个函数直接放到一个函数中，那如果你需要新增一个函数呢，改动原先的函数？如果那段代码不是你写的，你是不是还要先去搞懂对方写的是什么意思，然后再把函数添加到相应的地方，如果使用观察者模式通过暴露一个接口让你来添加新的函数，这样不是更好；

1. 观察者模式实际应用

1. jQuery回调对象 - jQuery.Callbacks – 观察者模式在jQuery中的应用

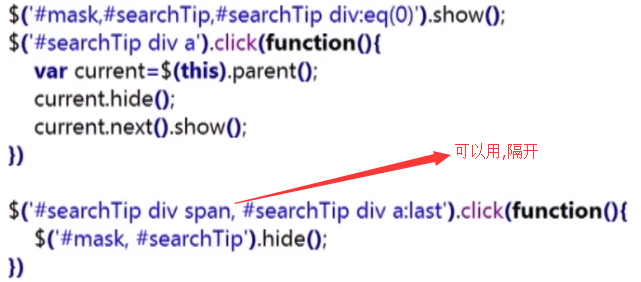


# 如何实现“新手引导”效果

1. 隐藏按钮文字的小技巧：text-indent:-999px;overflow：hidden；
2. 时间加上30天快捷方法



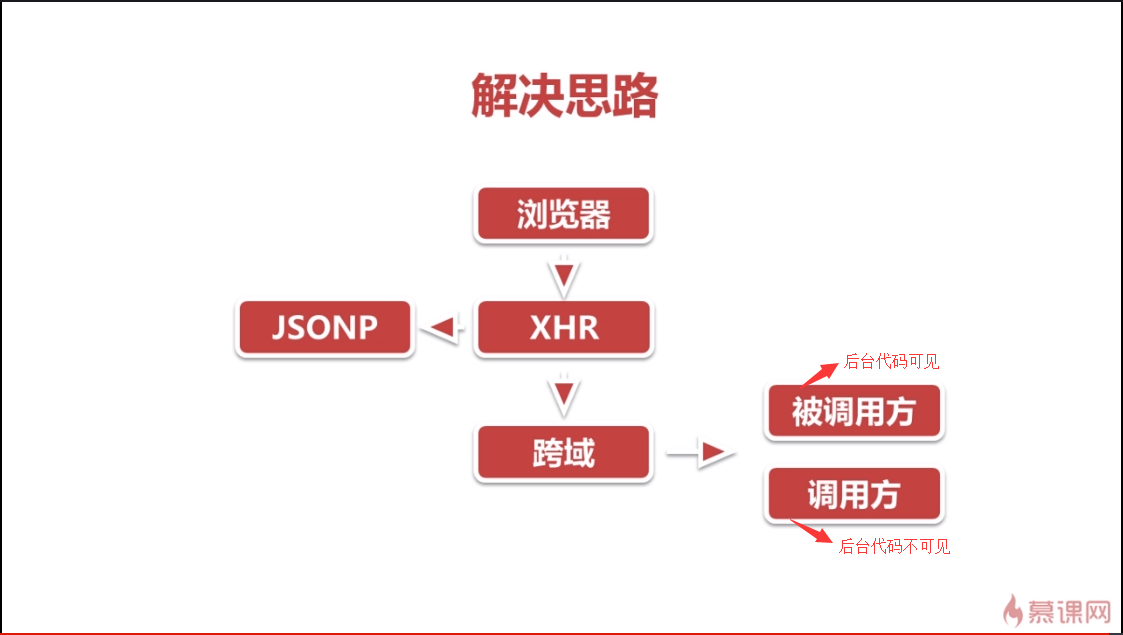
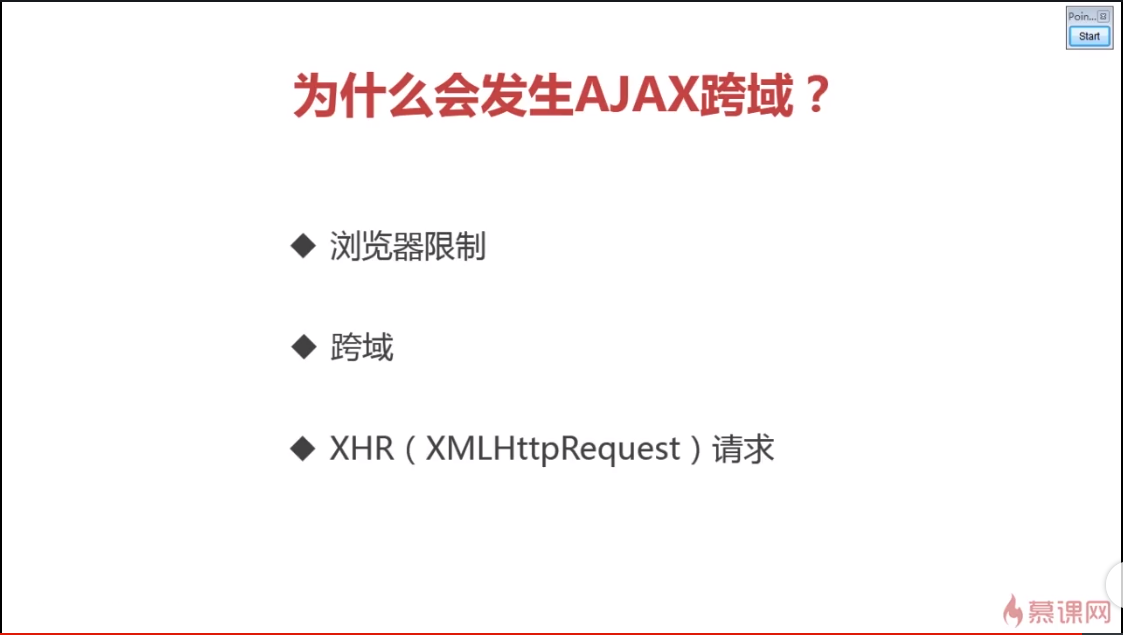
1. jQuery选择器



1. 先将新手引导的图片制作完成，可使用绝对定位的方式将图片显示在对应区域；

# 跨域

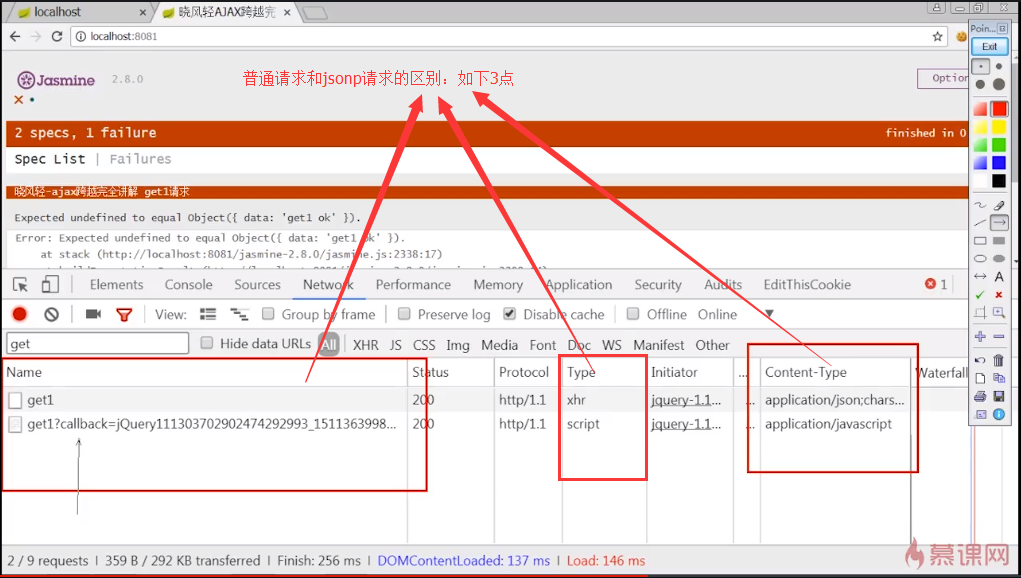
1. 跨域小结(为什么form表单提交没有跨域问题，但ajax提交有跨域问题)：http://blog.csdn.net/wabiaozia/article/details/78771709
2. 产生跨域问题的原因



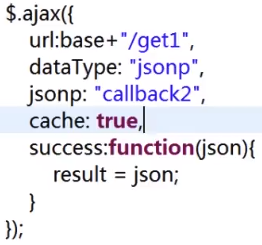
1. 浏览器限制，并不是后台限制；→ 指定参数，让浏览器不做校验，但这需要每个用户改变客户端；以下参数就是禁止浏览器校验



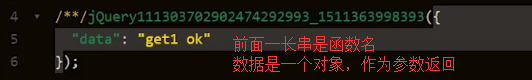
1. 端口、域名、协议等不一致；→ 被调用方修改代码（支持跨域），支持基于HTTP协议关于跨域方面的要求而作修改，当a域名调用b域名时，在b域名返回的信息里加入字段告诉浏览器我允许a域名调用，浏览器通过校验，就不会报跨域问题；调用方修改（隐藏跨域），通过代理，从浏览器发出的都是a域名的请求，在代理里面将指定的url转到b域名里面，在浏览器看上去就是同一个域名；
2. 发送的是xhr请求就存在跨域，若请求图片是json请求就不存在跨域；→ jsonp，动态创建了script，在script中发送了请求，弊端多；jsonp是json的补充使用方式，不是官方协议；jsonp是动态创建了script标签，script标签返回的是js代码，若服务器后台不作任何改动，返回的还是json对象，所以浏览器将json对象当作js代码来解析就会报错；因此使用jsonp是需要修改后台代码的；

 jsonp实现原理：请求发出是增加了一个callback参数，（默认参数名称为callback，可设置jsonp参数），后台发现有callback参数就知道是一个jsonp请求，然后就会把返回的数据由json转化为javascript，而javascript的内容就是一个函数调用；该JavaScript代码的函数名是callback参数名，他的函数参数是原来要返回的json对象；

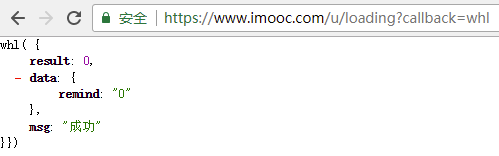
\_参数是一个随机数字，防止请求被缓存，cache设置为true，表示请求可以被缓存，则请求链接上不会有这个\_参数。

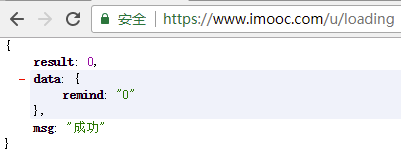






慕课网也支持jsonp请求



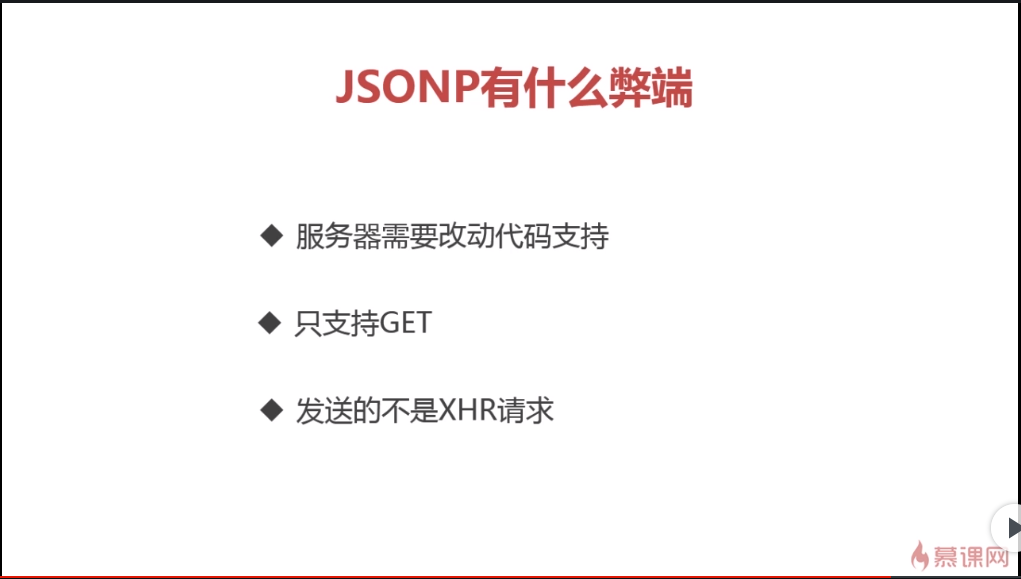


1. jsonp弊端

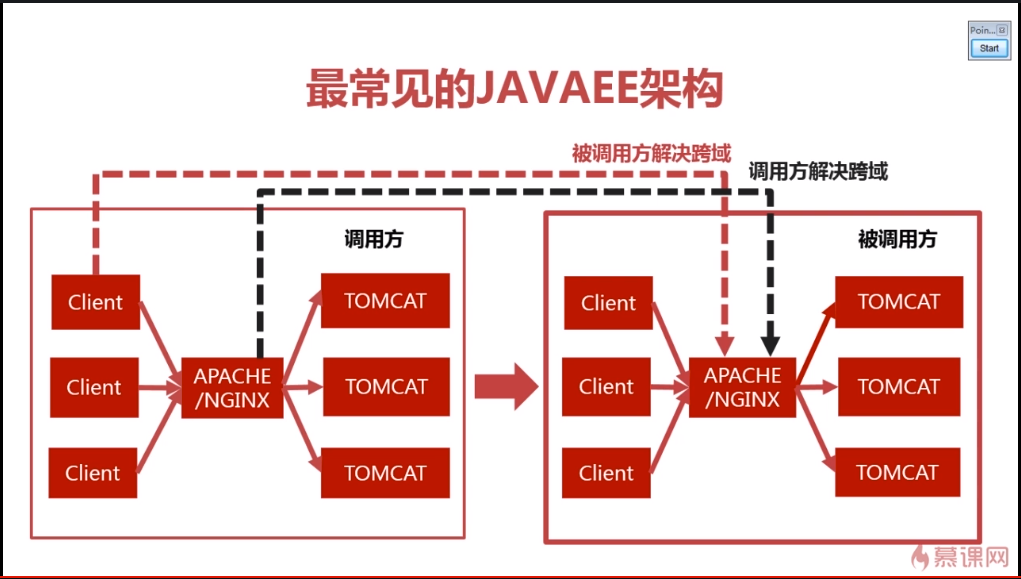
若调用的接口不是自己的代码，jsonp就无能为力了

即使在ajax中指定了post请求，jsonp实际还是get请求，script请求只能是get请求

xhr有很多新的特性，如异步，但jsonp都没有

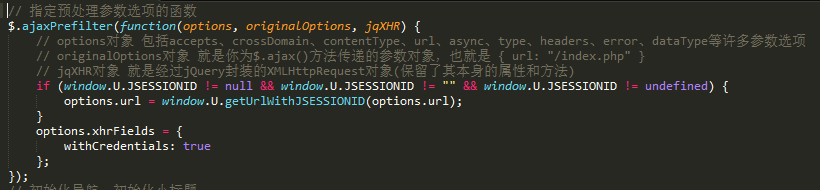


1. 被调用方解决跨域 （修改被调用方的HTTP服务器）&& 调用方解决跨域（修改调用方的HTTP服务器）



1. 被调用方解决跨域-非简单请求

前台请求传参（看完课程之后发现可以不用增加withCredentials参数）



后台处理设置

预检命令缓存：”Access-Control-Max-Age”告诉浏览器在这段时间内可以缓存这段信息，不需要再发送预检信息:

非简单请求时，后台先加上这句话，刷新页面会发送OPTIONS预检命令，请求成功；此时后台去掉这句话，刷新页面不会发送预检请求，会直接请求成功，这就是”Access-Control-Max-Age”的缓存。

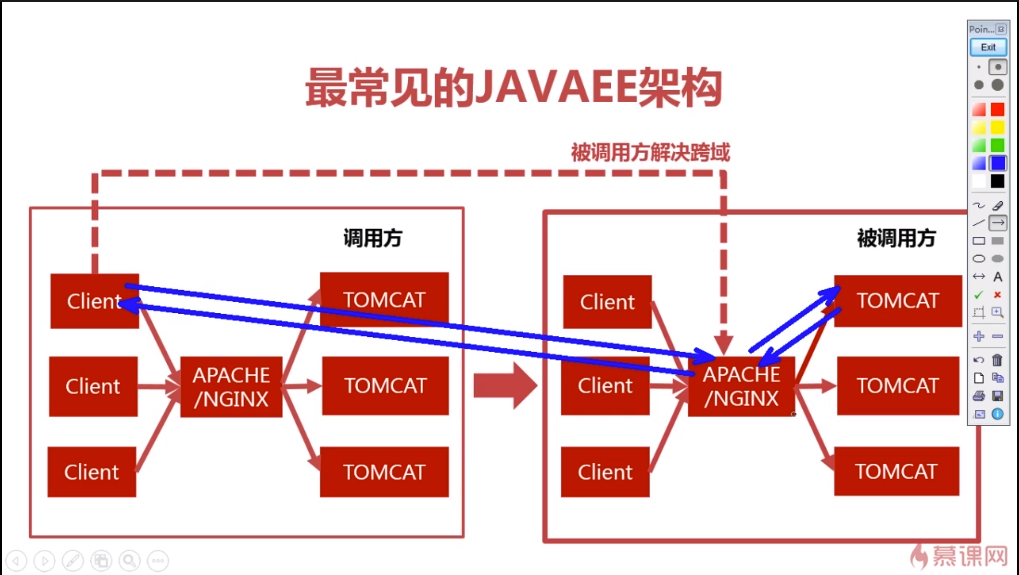
将disabled cache勾上再刷新（或其他清除缓存方式），则请求不成功

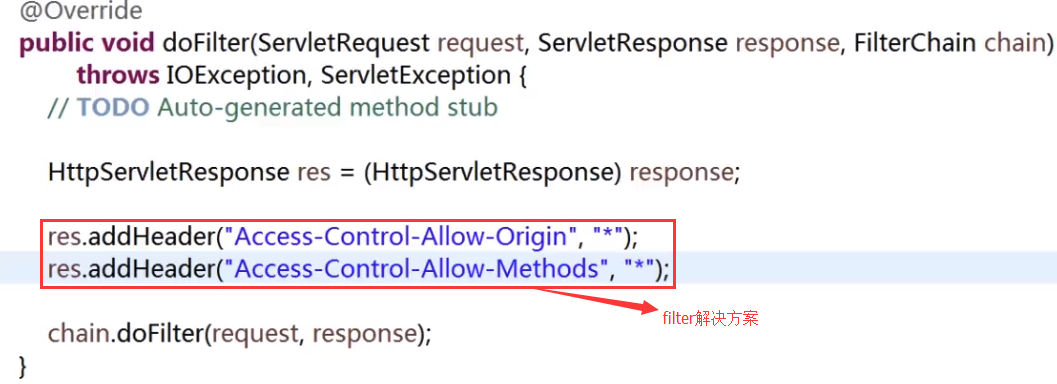
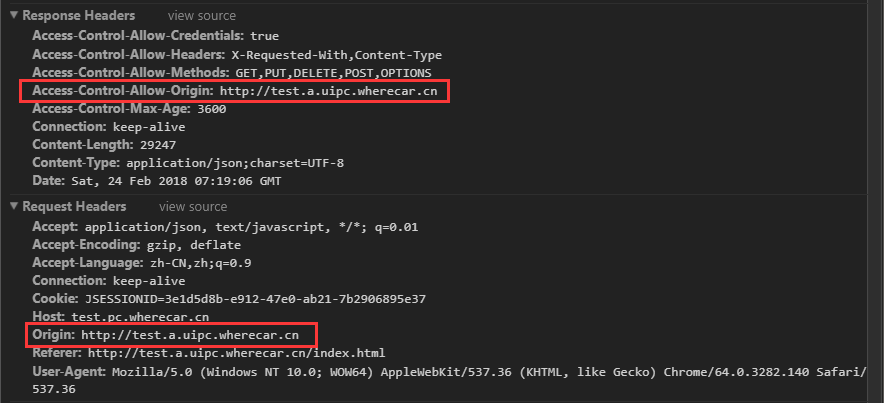


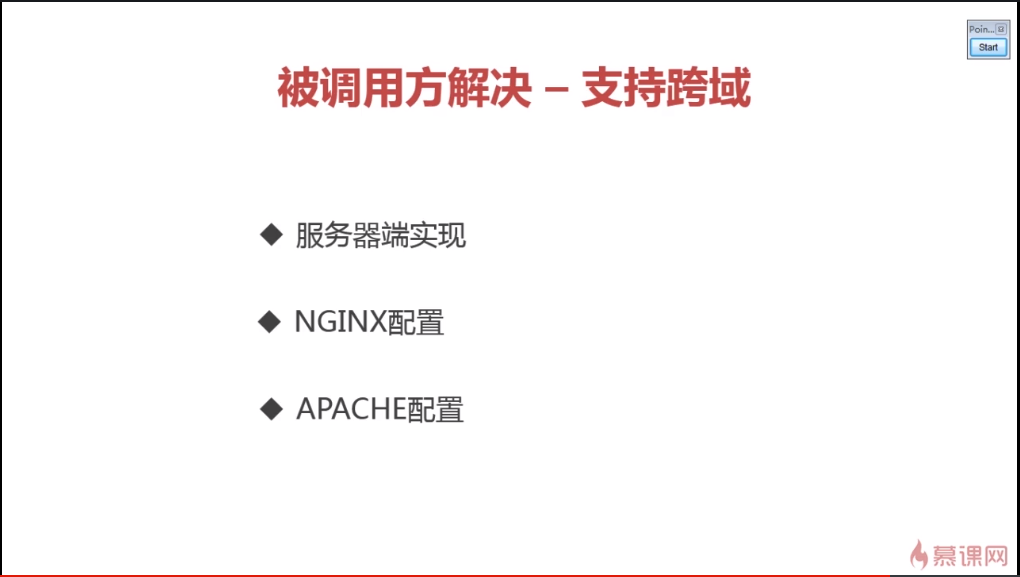
请求带cookie的时候，origin必须是全匹配，不能使用\*，同时后台加上”Access-Control-Allow-Credentials”,”true”，cookie是被调用方的cookie

请求发出时浏览器会判断是否是简单请求；若为简单请求，则先执行后判断；若为非简单请求，则先发送预检命令，再发送跨域请求

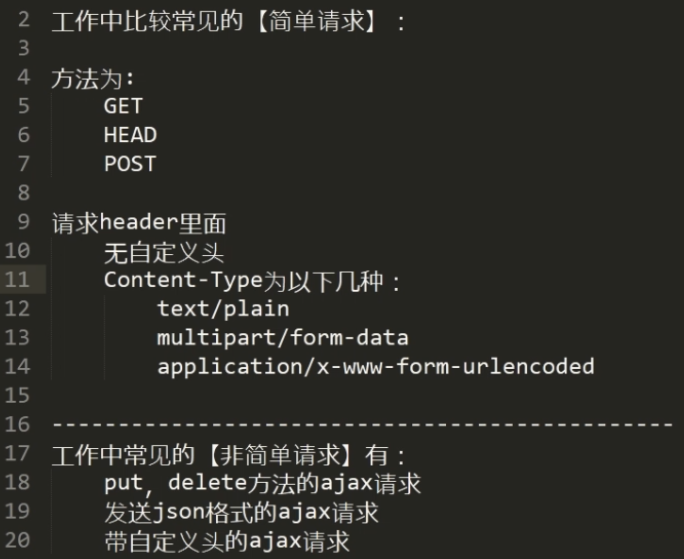
1. 被调用方解决跨域-简单请求



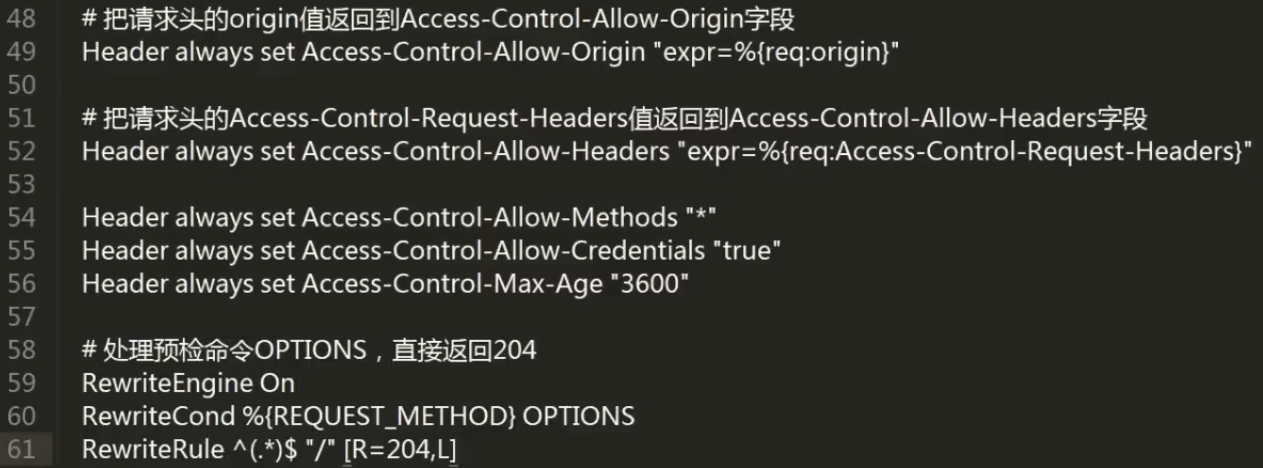
当请求为跨域请求时，在请求头RequestHeaders中有origin字段，（当前域名信息），若响应头ResponseHeaders中存在该域名，则允许跨域； 



1. 简单请求和非简单请求



1. 请求带cookie的时候，origin必须是全匹配，不能使用\*，同时后台加上”Access-Control-Allow-Credentials”,”true”
2. 调用方解决跨域（反向代理）访问同一个域名的两个不同url，最后去到两个不同的服务器？？？。
3. 被调用方解决跨域 - apache解决方案 – nginx解决方案

先配置虚拟主机再写代码

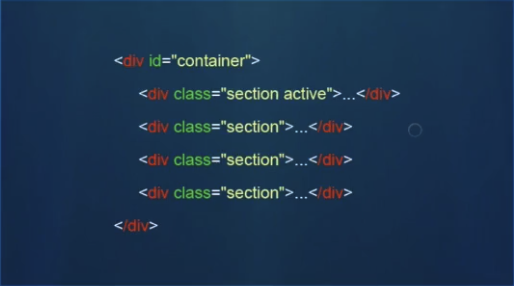


1. 总结

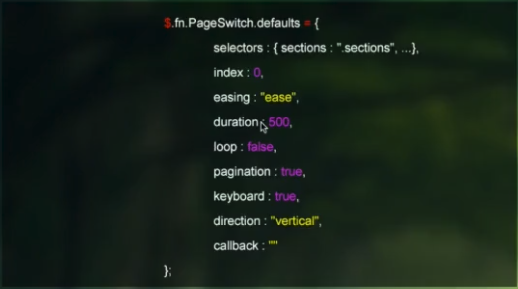


# 全屏切换效果

1. Dom结构

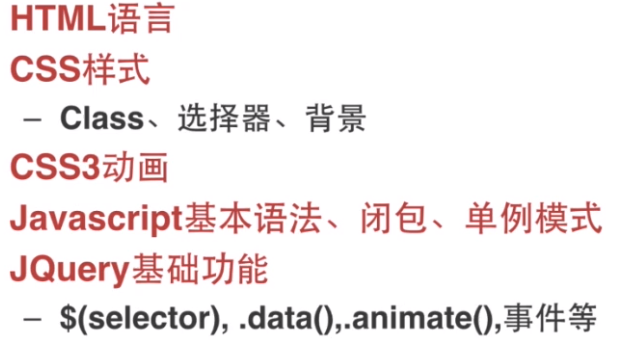


1. 插件引用



1. 预备知识：

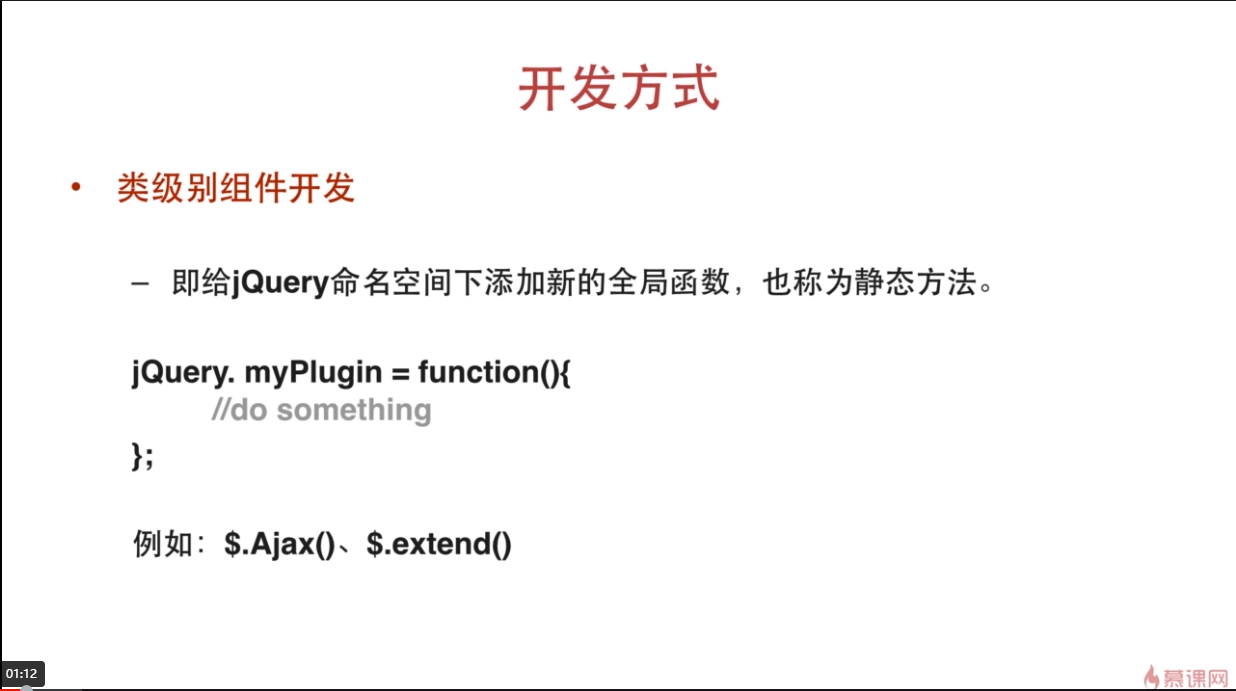
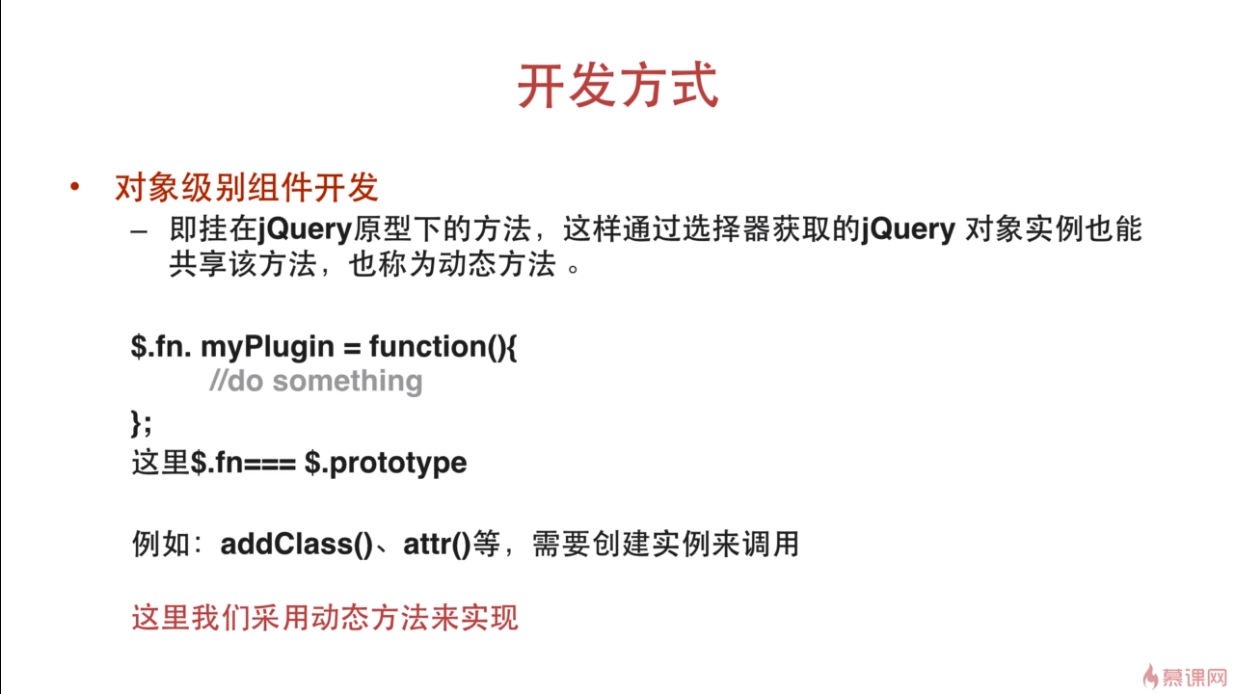
通过CSS使div实现全屏；jQuery插件框架；鼠标滚轮事件；键盘事件；实现滑动动画；

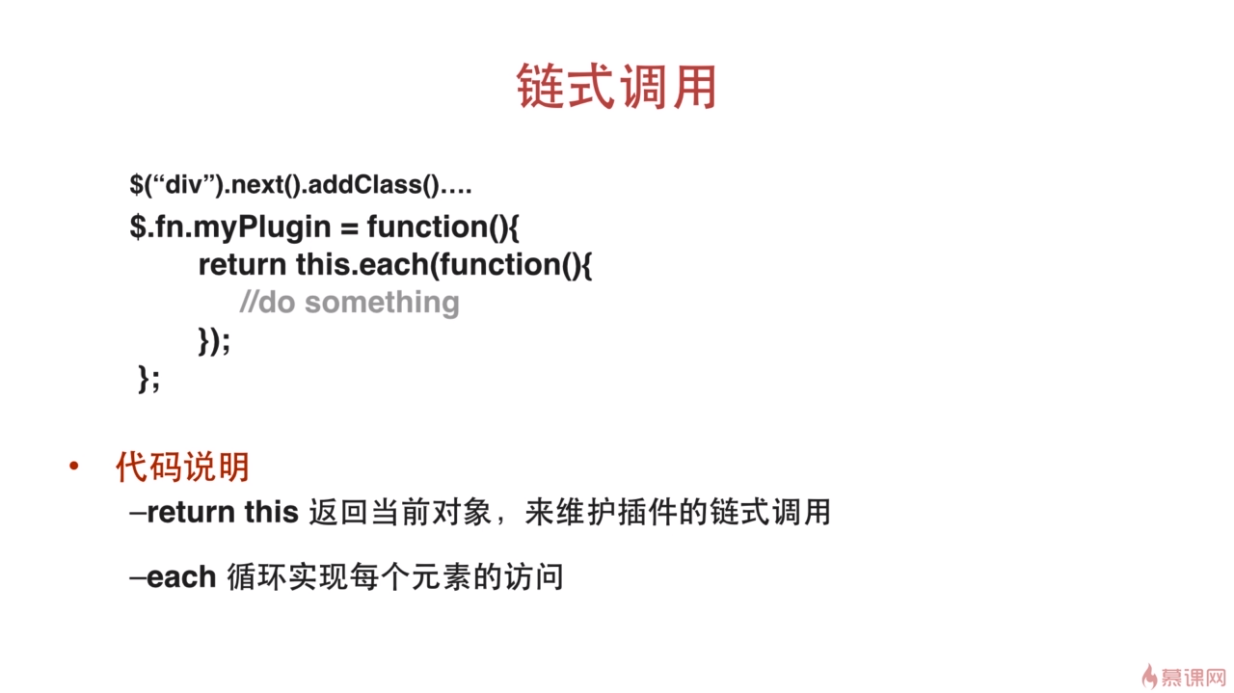


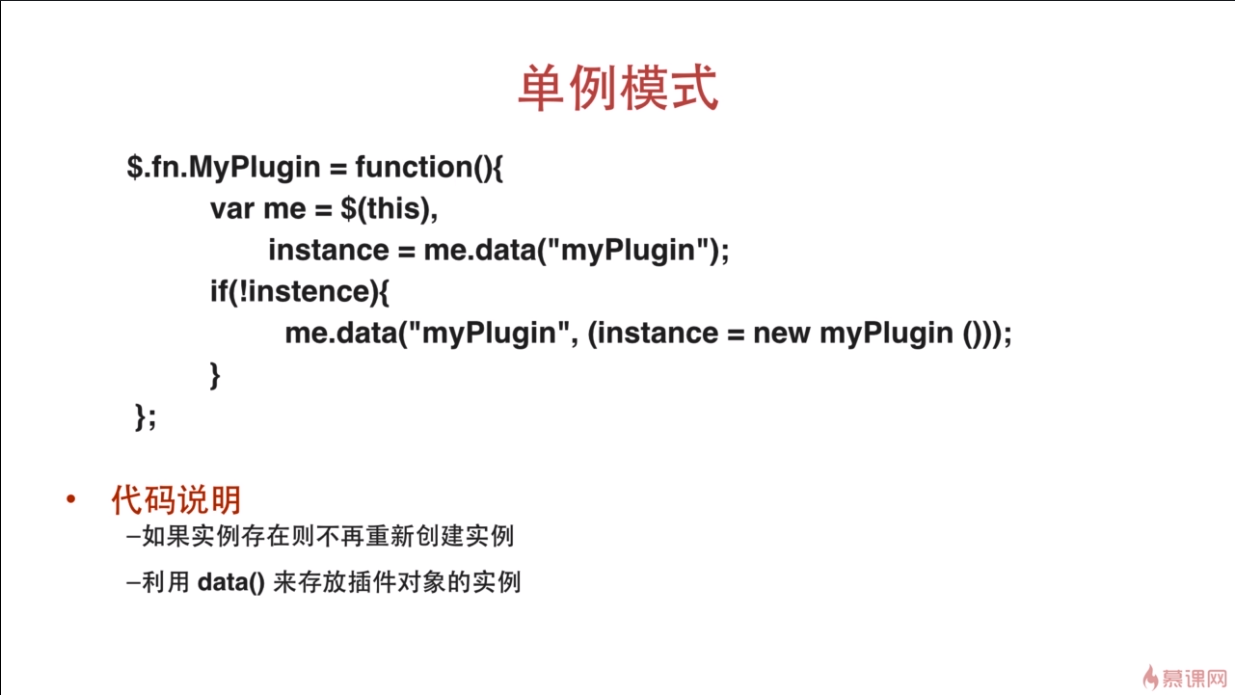
1. 闭包的作用



1. jQuery插件开发方式

1. jQuery链式调用
2. 单例模式



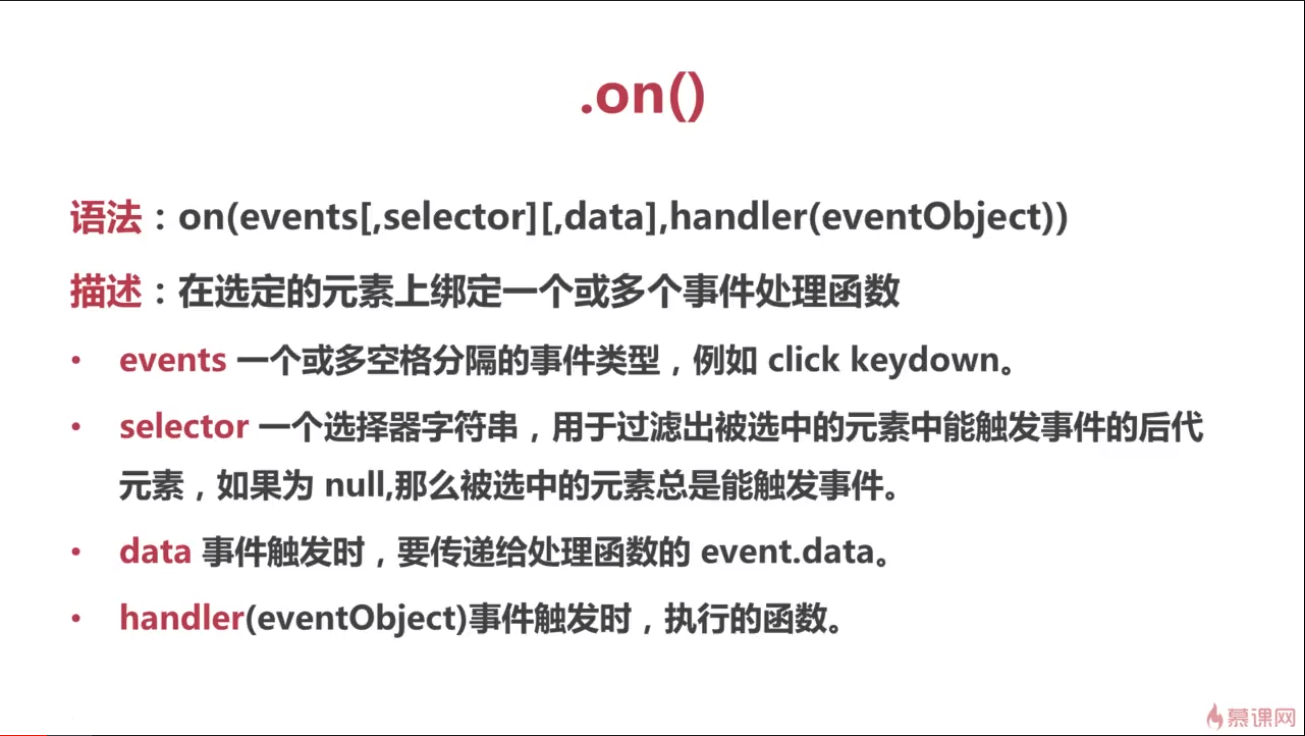
1. jQuery插件基础开发教程：<https://www.cnblogs.com/ajianbeyourself/p/5815689.html>；



1. js基础知识复习

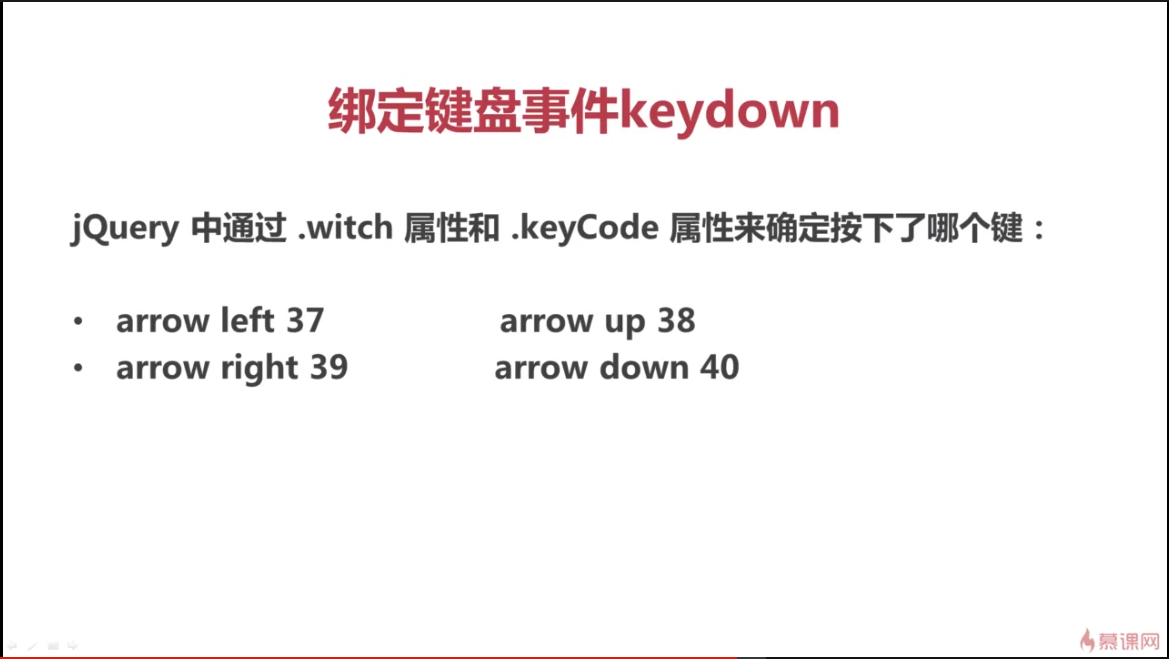
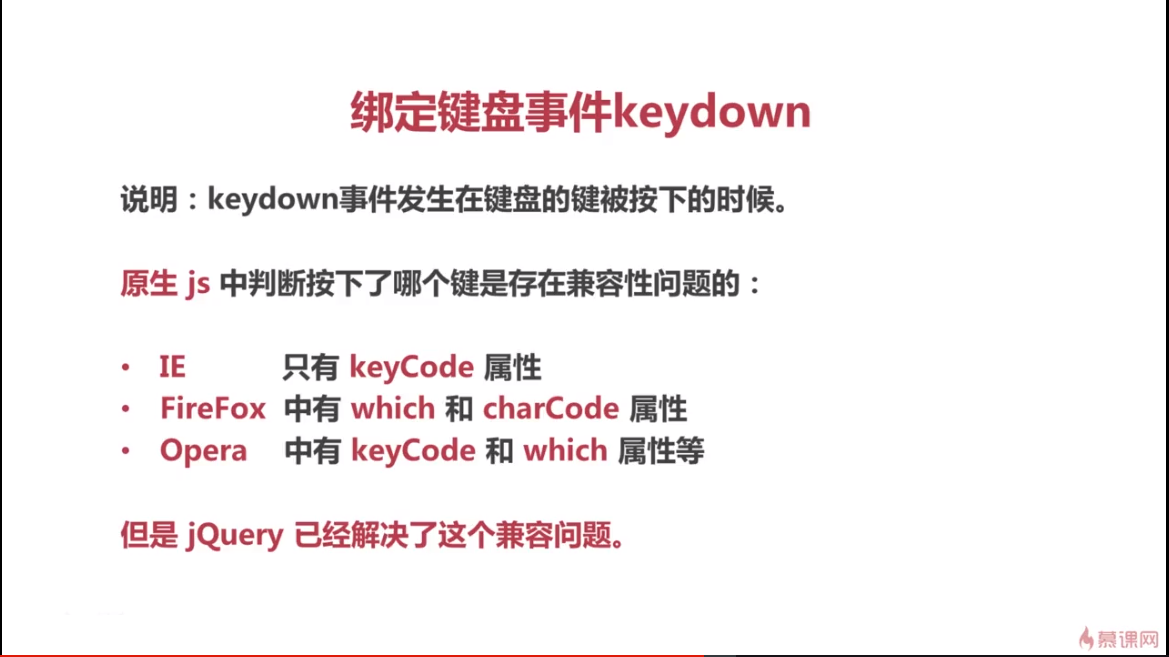


1. 事件委托：委托事件不仅可以给未创建的后代元素绑定事件外，当需要监听很多元素的时候，委托事件的开销更小；

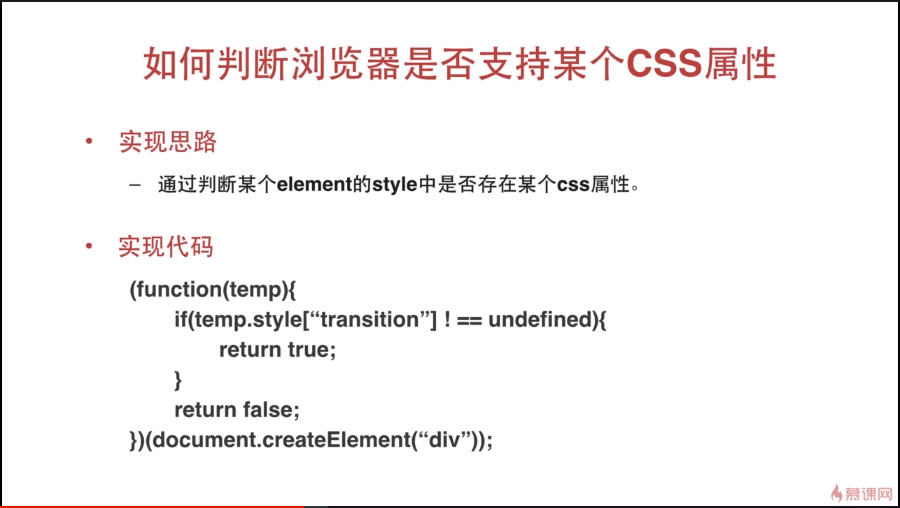


1. 鼠标滚轮事件、键盘事件





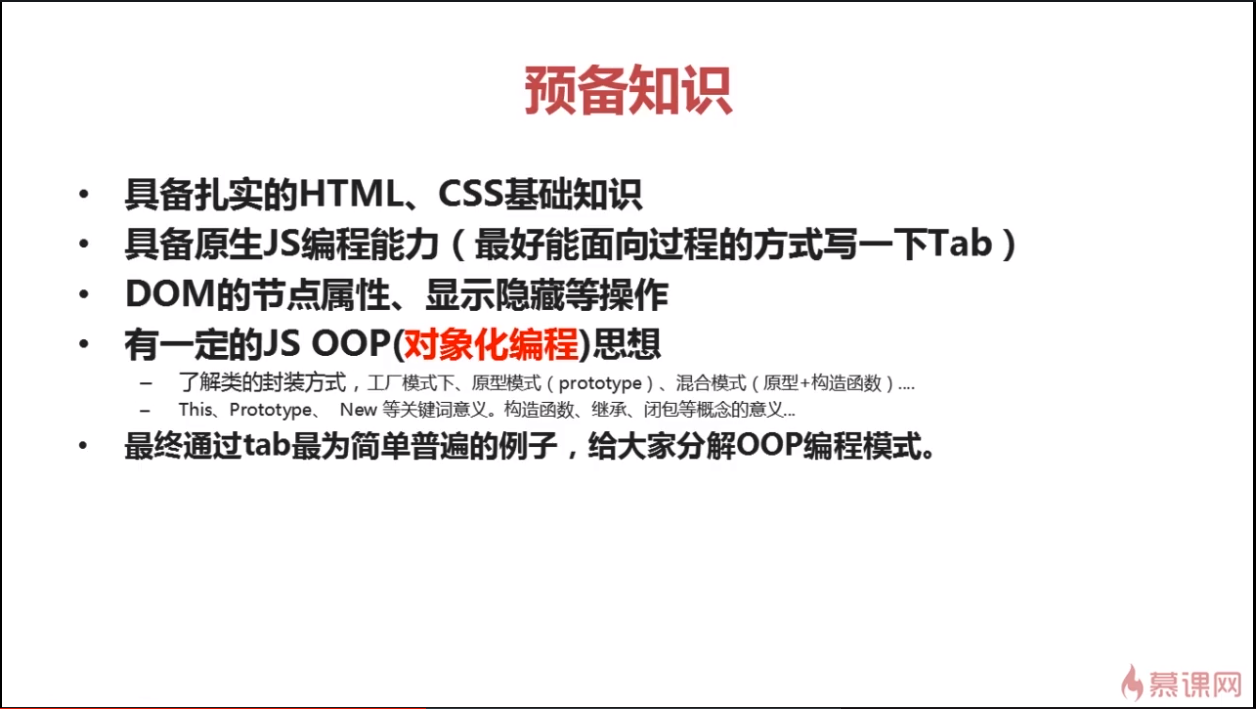
1. 如何判断浏览器是否支持某个CSS属性



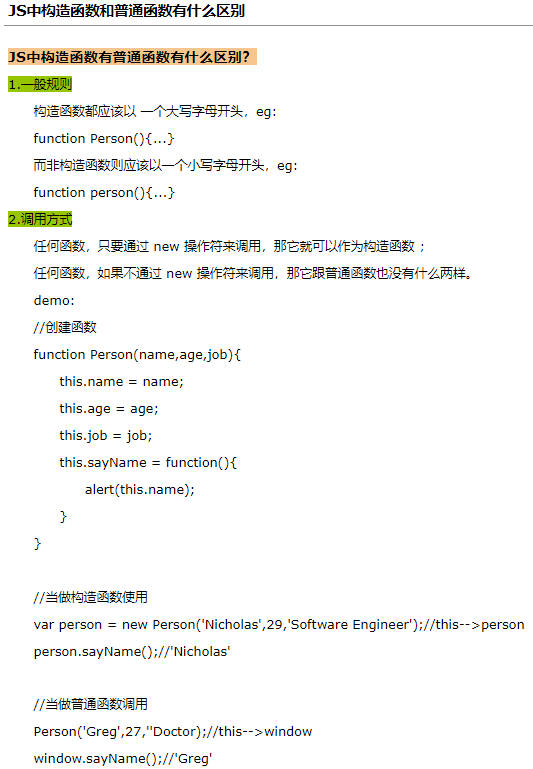
1. 对于不支持transition属性的低版本浏览器，可以通过jQuery的animate方法实现动画效果；

# JS插件开发之-Tab选项卡

1. 预备知识



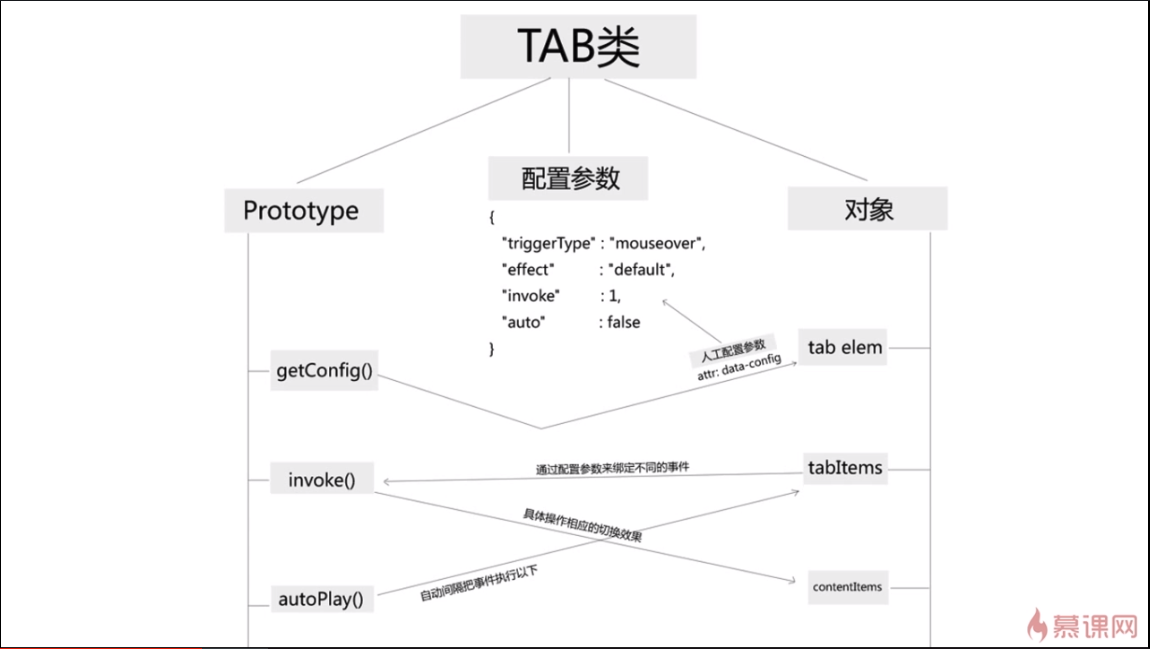
1. 深入理解js构造函数：<https://www.cnblogs.com/wangyingblog/p/5583825.html>；



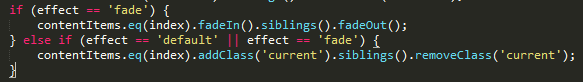
1. OOP（对象化编程）（Object Oriented Programming）思想；
2. OOP三大基本要素：对象获取、事件绑定、交互功能；核心思想：封装，继承，多态；
3. JavaScript构造函数及原型对象：<http://blog.csdn.net/a153375250/article/details/51083245>；



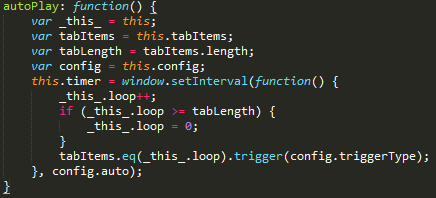
1. 构造Tab类



1. 给当前元素添加某个样式同时移除当前元素所有同胞元素的某个样式： currentTab.addClass('actived').siblings().removeClass('actived');
2. 容错处理，如果传参错误，则取一个默认值



1. trigger() 方法触发被选元素的指定事件类型。



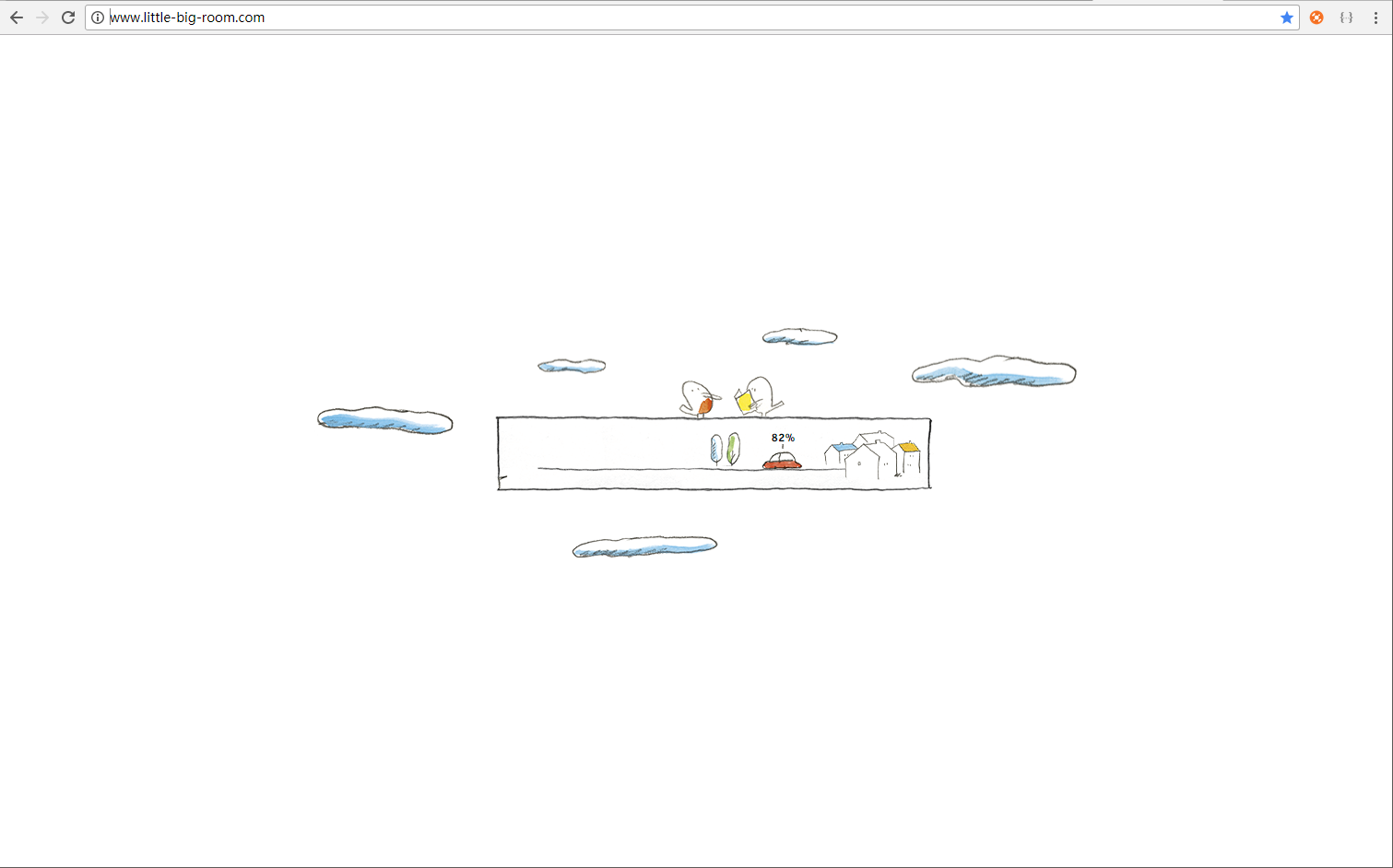
1. jQuery hover()用法

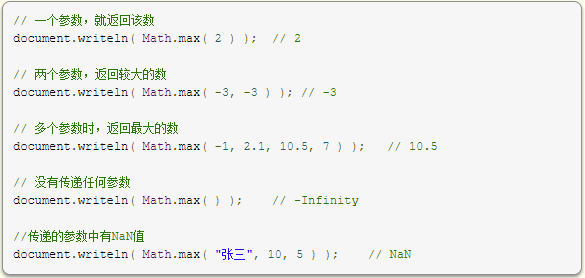


1. 1
2. 1

# Img图片预加载 · 插件封装

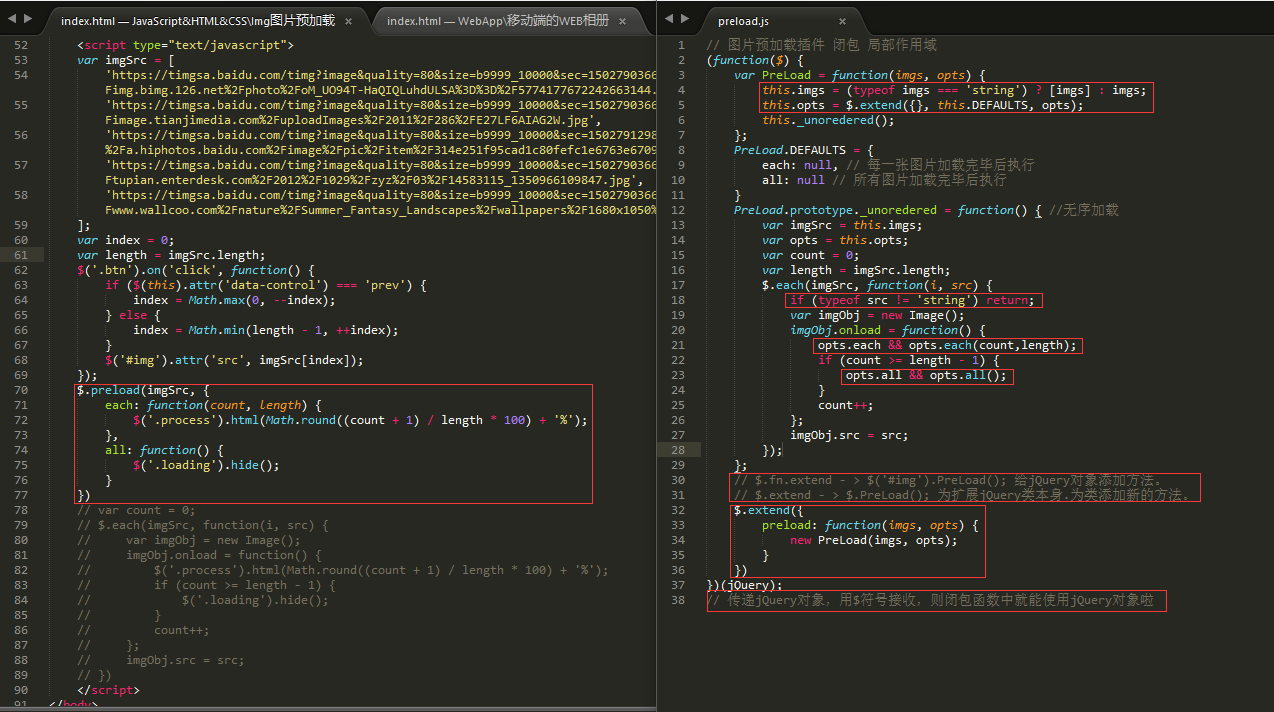
1. www.little-big-room.com



1. 预加载图片的特点：1. 提前加载所需图片；2. 更好的用户体验；
2. 有序预加载（如：漫画）和无序预加载（如：表情插件）；
3. max,min小技巧



1. 《理解jquery的$.extend()、$.fn和$.fn.extend()》http://caibaojian.com/jquery-extend-and-jquery-fn-extend.html



1. 懒加载就是把图片的url放置元素标签的自定义属性里，由滚动来判断当前图片是否在视窗里，然后setattribute，getattribute等属性来设置

# DatePicker时间选择控件

1. js日期对象的用法





1. new Date()\*1 获得当前时间戳；

# jasmine测试框架

1. 1
2. 1
3. 1
4. 1
5. 1
6. 1

# Jquery插件transform旋转焦点图

1. 1
2. 1
3. 1
4. 1
5. 1

# 瀑布流布局

1. 1
2. 1
3. 1
4. 1
5. 1
6. 1