# 2018.02.04

## 课堂笔记

### 什么是jQuery---框架（库）

JQuery is fast, small, feature-rich JavaScript liberay.

理念：write less , do more.

框架（库）：封装了很多预定义的对象和函数（别人写好的一个功能库，只需要拿过来用。）

开源：免费，代码可见（reactor Vue....）

### JavaScript框架有哪些

[www.ijavascript.cn/prototype/index.html](http://www.ijavascript.cn/prototype/index.html)

1. prototype侧重功能扩展，被jQuery干掉了
2. Yui: Yahoo user interface, 实现页面布局，提供许多组件
3. Dojo: 功能多，企业级，得到IBM，SUN 和 BEA等大公司的支持。学习曲线陡，文档不齐全，API不稳定，每次升级都可能导致已有的程序失效
4. ExtJs: 07年出来的高级，有许多绚丽效果的框架（例：多个div可以拖拽实现位置切换）。缺点：臃肿庞大，性能一般，兼容性不及jQuery
5. MooTools（面向对象）
6. jQuery （写的更少，做的更多，兼容性好）

....

### jQuery的特点：

1. 开源产品，语义易懂，丰富文档（注意：学习API）
2. write less, do more
3. jQuery是一个轻量级的脚本，其代码非常小，开发版278K，压缩版94k
4. jQuery支持CSS1~CSS3定义的属性和选择器
5. jQuery 是跨浏览器的，它支持的浏览器包括IE 6.0+ FF1.5+ Safari2.0+ Opera9.0+ Chrome，兼容性好。
6. 插件丰富

### jQuery被用在哪儿

1. 处理HTML文档（动态增删HTML元素）
2. Event 各种鼠标，键盘，浏览器事件
3. 实现动画效果 展开和折叠，隐藏和显示，淡入和淡出...
4. 提供Ajax交互 异步加载，局部刷新

### 版本的选择

1. 兼容性好的时候[兼容IE6,7,8]，选择版本1.xxx 政府，银行...
2. 网页加载速度快，选择版本2.xxx（缺点兼容性稍差，不兼容IE678）
3. 需要体验新特性（新功能）选择版本3.xxx

### 学习方法

1. 查看离线文档【手册】，不推荐看课件，查看老师的code，同学之间相互讨论
2. 百度，同事沟通（相互学习），实践，看淘宝京东的做法

### jQuery中的$符号

写得少，做得多【时间就是金钱】

$运用场景：

1. $(“字符串”) 获取页面元素，如：$(“#box1”) $(“.box1”)
2. $.ajax() 将$作为函数对象，使用该对象上面的方法
3. $(function(){}) 注册页面加载完毕事件。等效于window.onload
4. $(domObject); dom对象转为jquery对象。$（window）
5. $(“html代码”) 创建html代码的元素对象。

### jQuery使用的大前提

#### 引入jQuery文件

<src 找提示，回车。

#### 保证要查找的元素确实存在

#### 保证执行的代码是在元素在加载之后

<script></script>要在body脚下.................或者通过jQuery页面加载事件。

### jQuery中的选择器

css的选择器：

类:.class

id: #id

标签: element

jquery的选择器：

$(“.class”)

$(“#id”)

$(“element”)

jQuery选择器分类：基本选择器，层次选择器，过滤选择器，表单选择器

jQuery中的选择器跟css中的选择器非常类似，唯一的区别是$(“”)

### css()方法

1. 获取样式值

语法：$(“选择器”).css(“属性名”);

1. 设置样式值

语法：$(“选择器”).css(“属性名”, “属性值”);

1. 设置多个属性值

语法：$(“选择器”).css({“属性名1”: “属性值1”, “属性名2”: “属性值2”...});

### 基本选择器:

#### id选择器

---选择有id=”value”的元素

#### class选择器

---选中的是class=”value”的集合

#### element选择器

---选中的是对应的所有元素集合

#### 联合选择器

$(“#id1, #id2...”)---选中的是具有id值为id1，id2的选择器

#### 通用选择器

$(“\*”)---选中所有（在css中可能用到通用选择器来重置样式，jQuery用不到）

### 层级选择器

#### 后代选择器（子子孙孙）

$(“#id p”)---选中所有id=”value”下面的所有的p标签

#### parent>child父子选择器（所有的儿子）

$(“#id>p”)---选中所有id=”value”下面的所有的”儿子”p标签

#### prev + next（下一个邻居元素）

$(“#id+p”)---选中所有id=”value”下一个p标签

#### prev ~ siblings（兄弟节点【集合】 等同于nextAll()）

$(“#id~p”)---选中所有id=”value”以后的所有的兄弟p标签

### 过滤选择器

#### 基本过滤选择器

1. :first----- $(“tr:first”)第一个
2. :last------ $(“tr:last”)最后一个
3. :not------ $(“#div p:not(‘#div1’)”)去除特定值以外的所有的兄弟元素
4. **:even------ 偶数的元素 从0开始**
5. **:odd------ 奇数的元素 从0开始**
6. :eq(index)------ 选中给定的索引值的元素 从0开始
7. :gt(index)------ 选中大于给定的索引值的元素 从0开始
8. :lt(index)------ 选中小于给定的索引值的元素 从0开始
9. :header------ 匹配h（1,2,3....）元素

用途：主要运用于---后台管理（表格、列表）

语法：$(“object:selector”)

#### 内容过滤选择器

1. :contains(text)-----$(“div:contains(‘John’)”)选中的是“给定文本”标签
2. :has(selector)-----$(“div:has(‘p’)”).addClass(“test”)中的是“给定元素”标签
3. :empty------$(“div:empty)”)选中内容为空（不包含子元素或者文本）的标签
4. :parent------$(“div:parent)”)...

#### 可见性过滤选择器

1. :visible-----$(“input:visible)”)匹配所有可见元素
2. :hidden-----$(“input:hidden)”)匹配所有不可见元素

主要针对 input--》type=”hidden”, style--》 display:none

#### 属性过滤选择器

1. [attribute]-----$(“input[id])”).addClass(“highlight”)匹配特定属性的元素
2. [attribute=value]-----$(“input[name=’basketball’])”).attr(“checked”,true)匹配特定属性的特定值的元素
3. [attribute!=value]-----匹配除去特定属性的元素
4. [attribute^=value]-----匹配特定内容的属性开头的元素
5. [attribute$=value]-----匹配特定内容的属性结束的元素
6. [attribute\*=value]-----匹配包含特定内容的属性的元素

#### 子元素过滤选择器

:first-child----匹配第一个子元素

:last-child----匹配最后一个子元素

:nth-child(para)----匹配指定参数的子元素

:nth-child(even/odd)----匹配偶奇数子元素（计数从1开始）

:nth-child(n)----匹配指定位置子元素（计数从1开始）

:nth-child(3n+1)----匹配3n+1位置子元素（计数从1开始），其中n的取值从0开始

应用场景：

手机端：插入广告

# 2018.02.05

## 课堂笔记

### DOM操作

document object model：文档对象模型

#### DOM Core

JavaScript中获得Dom节点：

....getElementById() getElementsByTagName()------》jQuery------ $(“element”)

#### HTML-DOM

#### CSS-DOM

### jQuery中DOM操作

围绕DOM树，进行增删改查（昨天的选择器就是查的部分）

#### 创建节点

var li\_1 = $(“<li></li>”); //利用工厂函数创建了一个li的节点

$(“ul”).append(li\_1); //添加节点到指定位置

var li\_2 = $(“<li class=’author’>添加一个有内容的节点</li>”);

$(“ul”).append(li\_2);

#### 插入节点

##### 父子前后关系：

append() 父子--------后----关系

语法：$(“选择器（父）”).append(“节点”/变量)

appendTo() 父子--------后----关系

语法：$(“节点”/变量).appendTo(“选择器（父）”)

prepend() 父子-------前---关系

语法：$(“选择器（父）”).prepend(“节点”/变量)

prependTo() 父子-------前---关系

语法：$(“节点”/变量).prepend(“选择器（父）”)

##### 兄弟前后关系

after() 兄弟--------后----关系[插入的节点跟匹配的节点同级（之后）]

语法：$(“选择器（兄）”).after(“节点”/变量)

insertAfter() 兄弟--------后----关系[插入的节点跟匹配的节点同级（之后）]

语法：$(“节点”/变量).insertAfter(“选择器（兄）”)

before() 兄弟--------前----关系[插入的节点跟匹配的节点同级（之前）]

语法：$(“选择器（弟）”).before(“节点”/变量)

insertBefore() 兄弟--------前----关系[插入的节点跟匹配的节点同级（之前）]

语法：$(“节点”/变量).insertAfter(“选择器（弟）”)

#### 删除节点

##### remove-----移除

语法：$(“选择器”).remove();

保留自身（用变量接收起来，可以用于append），但是如果被移除的对象绑定了事件就没有了。

$(“#h2”).click(function(){

alert(“remove”);

})

var a = $(“#h2”).remove(); //这个变量可以继续使用

$(“ul”).append(a); //原来的变量被作为新元素被添加到ul中，但是事件消失

##### empty-----清空

语法：$(“选择器”).empty(); 清空匹配标签，但是原来的还是保留。

##### detach

语法：var a = $(“选择器”).detach();

保留自身，但是被移除的对象绑定了事件也还在。

$(“#li”).click(function(){

alert(“detach”);

})

var b = $(“#li”).detach(); //这个变量可以继续使用

$(“ul”).append(b); //原来的变量被作为新元素被添加到ul中，事件未消失

#### 复制节点

clone----克隆

语法：$(“选择器”).clone(参数);

参数：一个boolean(true, false),指代的是*事件处理函数*是否被复制

#### 替换节点

replacewith()

$(old).replaceWith(new); $(“p”).replaceWith(“<b>替换的内容</b>”);

replaceAll()

$(new).replaceAll(old); $(“<b>替换的内容</b>”).replaceAll(“p”);

替换节点方法能够替换某个节点

#### 包裹节点

$(ele).wrap(); //包裹在每一个ele外面 <b><p></p></b> <b><p></p></b>

$(ele).wrapAll(); //一个包裹全部的ele <b><p></p> <p></p> <p></p> </b>

$(ele).wrapInner(); //包裹在每一个ele里面 <p><b></b></p> <p><b></b></p>

#### 属性操作

##### attr ----------attribute主要应用于自定义属性操作

获取属性值

语法：$(“ele”).attr(“name”);

设置属性值

语法：$(“ele”).attr(“name”, “value”);

##### prop----------property主要应用于w3c属性操作

获取属性值

语法：$(“ele”).prop(“name”);

设置属性值

语法：$(“ele”).prop(“name”, “value”);

应用场景：购物车-----实现全选，反选

##### removeAttr()------直接删除

##### removeProp()-----只是清空属性值变为undefined，属性名还在

#### 样式操作

##### addClass()----添加class值

语法：$(“ele”).addClass(“value”);

##### removeClass()----删除class值

语法：$(“ele”).removeClass(“value”);----删除指定类 的值

$(“ele”).removeClass();----删除所有类 的值

##### hasClass()----有无某值的class

语法：$(“ele”).hasClass(“value”);----判断是否指定值的类 返回boolean值

##### toggleClass()----切换样式

语法：$(“ele”).toggleClass(“value”);----点击添加样式，再点击取消样式

应用场景：tab滑动菜单，切换效果，京东导航菜单

#### 设置和获取HTML文本值

##### html()

获取和设置文本值的时候，可以获取和设置html标签

获取值：$(“ele”).html()

设置值：$(“ele”).html(“值”)

##### text()

获取和设置文本值的时候，只能获取和设置“纯文本”

获取值：$(“ele”).text()

设置值：$(“ele”).text(“值”)

##### val()---针对表单操作

获取值：$(“ele”).val()

设置值：$(“ele”).val(“值”)

#### 遍历节点

##### next() 下一个兄弟节点

语法：$(“ele”).next();

例子：$(“ele”).next().css();

##### prev() 上一个兄弟节点

语法：$(“ele”).prev();

##### siblings() 除了自己以外，其他所有的兄弟节点

语法：$(“ele”).siblings();

应用: 滑动菜单，其他菜单背景色改变，并且内容响应的改变

补充： index()-----获取索引值

eq(index)----选择对应的索引值

find()

$(“ul li”).find(“span”).css(“color”,”red”);

##### each()

带参数：（取出索引和纯文本）

$(“ul li”).each(function(index,item){

//index:索引

//item:项目

console.log(item);---------打印所有的标签，包含属性

console.log(“索引”+index+”文本内容”+$(item).text());

})

不带参数：（取出纯文本）

$(“ul li”).each(function(){

console.log($(this).text());

})

##### closest() 最近的元素

语法：$(“ele”).closest();

例子：$(li).closest(“div”)

<div>

<div>

<ul>

<li></li>

</ul>

</div>

</div>

方法一：$(this).closest(“tr”).remove();

方法二：$(this).parent().parent().remove();

##### children()

#### CSS DOM操作

##### css()

##### offset() 偏移量

top垂直方向的偏移量

left水平方向的偏移量

语法：$(“ele”).offset().top;

应用场景：楼层滚动效果

##### position()

针对top left获取他们的值

语法：$(“ele”).position().top/left;

##### scrollTop()

滚动条距离顶部的值

$(window).scroll(function(){

$(“body,html”).scrollTop();

})

应用场景：

特效---------返回顶部，拖拽效果

# 2018.02.07

## 页面加载事件

原生JavaScript：window.onload = function(){};

jQuery：$(function(){}) 或者 $().ready(function(){});

一个页面可以由多个$(function(){

//DOM加载完毕要执行的代码

})

## 事件三要素

事件源+事件类型+事件处理函数

### 事件绑定事件

bind()

语法：$(“element”).bind(‘type’,function(){});

简写：

$(“element”).type(function(){})

#### 多个事件绑定一个处理函数

$(element).bind(“mouseover mouseout”,function(){})

#### 一个事件绑定多个处理函数

$(element).bind(“click”,function(){}).bind(“click”,function(){})

**简化：$(element).click(function(){})**

#### 多个事件绑定不同处理函数

$(element).bind({

“mouseover”,function(){},

“mouseout”,function(){}

})

### 合成事件

### 阻止浏览器默认动作

event.preventDefault()

### 事件冒泡

event.stopPropagation();

### 事件委派 delegate

给所匹配的函数附加一个事件处理函数，*即使这个元素是后加进来的也有效*。

语法：$(“外部已经存在的元素”).delegate(“新增元素”,”type”,function(){})

应用场景：新加入的HTML，进行操作，就需要delegate(‘’,’’,function(){})

例：添加删除银行卡

### 事件对象的属性

pageX

pageY

获取光标相对于页面的坐标....在document中获得

应用场景：移动类型的特效上。

### 移除事件（取消绑定）

unbind()

语法：$(element).unbind();-----移除所有（慎用）

$(element).unbind(‘click’);-----移除指定

类似：removeClass()

### 模拟操作

$(element).triggerHandler(“submit”);

应用于测试。同时也可以使用preventDefault(==价与return false)；

### 其他的用法

on 和 off 同bind 和unbind一样

## 基本动画函数 show() hide()

show() 显示匹配的隐藏的元素 $(element).show(time/effect,[callback]);

time可以是纯数字，单位是毫秒。

hide() 隐藏匹配的显示的元素 $(element).hide(time/effect,[callback]);

effect可以是normal fast slow

show() hide()的本质是转为display:none display:block

应用场景：导航菜单，tab滑动菜单等，需要显示隐藏效果的地方

加载时默认隐藏，$(element).hide()--->display:none

toggle()切换

## 淡入，淡出效果 fadeIn fadeOut fadeTo

fadeIn() 淡入 $(“element”).fadeIn(time/effect,[callbacks])

time可以是纯数字，单位是毫秒。

fadeOut() 淡出 $(“element”).fadeIn(time/effect,[callbacks])

effect可以是normal fast slow

fadeIn() fadeOut()的本质是转为display:none display:block

fadeTO()设置透明度 区间值：0-1 本质是设置opacity

## 滑动动画函数slideUp slideDown

slideUp() 隐藏--通过改变高度

slideDown() 显示--通过改变高度

应用场景：主用应用于后台管理系统，菜单栏

## 自定义动画 animate

$(“element”).animate({attr:value, attr:value},time,[callback])

## 停止动画和判断是否处于动画状态

stop(clearQueue,gotoEnd);

clearQueue:清空队列，立即结束动画

gotoEnd:让当前正在执行的动画立即完成

$(“element”).stop(true,true);

应用场景：

分享菜单，处理动画累积

is(“:animated”)判断是否处于动画状态，可用于防止动画累积

延迟执行动画delay（time）

$(“element”).delay().动画函数;

# 2018.02.08

## 作业问题

### .eq() 和 :eq()的区别

:eq() 匹配一个给定索引值的元素----本质还是选择器

.eq() 获取第N个元素------本质是一个方法（在一个集合中单独拿出指定索引值的元素）

# 2018.02.09

## 插件

plugin，又称为extension。关键点，掌握插件的使用方法和步骤，参数配置一般都在官网上参看。

网上搜插件：

plugins.jquery.com

17sucai.com

jq22.com

js-css.cn

应用于：动画特效

### jQuery插件使用通法：

1. 引入jQuery库
2. 引入插件样式（有些插件不需要样式）
3. 引入插件js库（有的插件不止一个js库）
4. 在页面按照插件编写HTML结构和样式，调用插件（留出插件的位置）
5. 调试

### 插件使用常见的问题

1. 样式兼容性问题
2. 插件依赖的jq版本问题
3. 目录改变导致背景图片或者按钮找不到的情况--------------【修改图片路径】
4. 没有参数说明-------【打开插件源码，里面有注释】
5. 有的插件不止一个样式或者不止一个js文件！！

补充：jQuery的引用顺序.....强调jQuery直接引用在最前面

1. 没有参数说明文档的时候，可以百度插件名称，有可能找到相关参数说明
2. 插件demo里面可能引用有外部链接，如果网络慢会加载很久，调用插件办法

### 轮播插件的使用步骤

1. 项目的预留位置

2. 引用jQuery库 \*\*\*\*\*\*\*注意，引用顺序

3. 引用插件js文件【可能不止一个js文件】 \*\*\*\*\*\*\*注意，引用顺序

4. 引用css文件

5. 复制，粘贴--调用插件----内容：html代码

6. 修改样式----修改成我们想要的款式【图片，文件路径】

7. 调试

# 春节假后2018.02.26

## 复习：

delegate

$("外部选择器[父级]").delegate("新增元素选择器","事件类型",function(){})

语法：$(“外部已经存在的元素”).delegate(“新增元素”,”type”,function(){})

应用场景：新加入的HTML，进行操作，就需要delegate(‘’,’’,function(){})

例：添加删除银行卡

显示: show() fadeIn() slideDown()

隐藏: hide() fadeOut() slideUp()

遍历节点: prev() next() siblings()

补充知识点: parent() closest()

prepend() 向前添加子元素

append() 向后添加子元素

remove() 删除后，保留自身（内容），绑定事件没有

detach() 删除后，保留自身，绑定事件还在

empty() 删除后，不保留自身，绑定事件还在

addClass()

removeClass()

offset()

scrollTop()

hover() ----相当于mouseover mouseout

preventDefault() 阻止浏览器默认行为---> return false

stopPropagation() 阻止冒泡

自定义动画（本质：改变属性值；在动画里的属性是驼峰命名法；不支持颜色渐变改变）

$(“selection”).anminate({属性:属性值,属性:属性值....},时间,[callback])

stop(true,ture).delay(200).animate({"left":0},3000);

注意：顺序

stop(clearQueue,gotoEnd);

clearQueue:清空队列，立即结束动画

gotoEnd:让当前正在执行的动画立即完成

$(“element”).stop(true,true);

应用场景：

分享菜单，处理动画累积

# 2018.02.27

## 项目：

project pakage下的内容：

css文件夹

img文件夹>images文件夹(单独放后台传输图片：banner和products之类的大图) 其他的放 页面小图标

js文件夹

plugin文件夹

项目日志TXT

命名尽量 用\_下划线，因为复制的时候可以选中整体

切图---视频

F:\itsource\06JQUERY\_Project\20180227\teacher\video\02PS切图-查看字体样式

步骤：**先写出结构**

项目人员：

一般的小商城项目，一个UI（甚至可以省，直接用网上的设计），两个前端，两个后台

和前端协同开发的人有:

后台程序员,还有UI,前端,测试人员

小组长(主管)--大公司

项目经理 产品经理 老总

做出版心的单独类

查看关于通用类的样式：

F:\itsource\06JQUERY\_Project\20180227\teacher\code\css\common.css

<input></input>

<input></input>两个换行的input有间隙

<input></input><input></input>两个并排的input没有间隙

使用ps获取字体大小

使用FSCapture710获取宽度和颜色