

JavaScript及web前端开发

第8讲 JQuery

唐大仕

课程进度

14

综合应用

15 Node服务端

10

网站开发

11

手机开发

12

文本及绘图

9

Bootstrap

8

Jquery

5

HTML5

6

CSS3

7

Javascript进阶

2

HTML

3

CSS

4

Javascript基础

1

序论

相关网站

■ 学习网站

- 教程 <https://www.w3school.com.cn/jquery/index.asp>
- 手册 https://www.w3school.com.cn/jquery/jquery_reference.asp

■ 官网

- <https://jquery.com> 核心
- <https://api.jquery.com> 文档
- <https://openjsf.org/> 更多

本讲内容

- jQuery概述
- 选择
- 事件
- 效果
- 操作
- 遍历
- 插件

概述

jQuery

■ jQuery是一套方便操作文档对象的JavaScript库

- DOM

- jQuery 是一个 JavaScript 库

- jQuery 极大地简化了 JavaScript 编程

- jQuery 很容易学习和使用

■ 应用相当广泛

- 有统计表明80%网站用了jQuery

- jQuery使用统计 <http://trends.builtwith.com/javascript/jQuery>

jQuery 基本理念

■ 通过 jQuery

- 可以选取（查询，query）HTML 元素
- 并对它们执行“操作”（actions）

■ jQuery 口号

- write less, do more
- “写得更少，但做得更多”



■ 发明者 John Resig



jQuery快速入门

■选择→操作

- 内容 `$("#mydiv").html("hello");` `//.innerHTML=xxx` //简便书写
- 样式 `$(".book").css("color","red");` 如果选择出多个对象，它会一起操作
- 隐藏 `$(".detail").hide();` `//.style.display="none"` //不同浏览器都适用
- 事件 `$("button").on("click", function(){...});` `//.onclick=xxxx`//多个事件
- 获取
 - `alert($("#mydiv").html());`
 - `var h = $("#mydiv").height();` //一件复杂的事情

■注：\$ 也可写为 jQuery

案例分析

■ 使用jquery的示例

□ <http://www.cmdpe.com/> 其中的common.js

```
rotateForce = 45
moveForce = 30
$(window).mousemove(function(e) {
    var docX = $(window).width();
    var docY = $(window).height();
    var moveX = (e.pageX - docX/2) / (docX/2) * -moveForce;
    var moveY = (e.pageY - docY/2) / (docY/2) * -moveForce;
    var rotateY = (e.pageX / docX * rotateForce*2) - rotateForce;
    var rotateX = -((e.pageY / docY * rotateForce*2) - rotateForce);
    $('.popup')
        .css('left', moveX+'px')
        .css('top', moveY+'px')
        .css('transform', 'rotateX('+rotateX+'deg) rotateY('+rotateY+'deg)');
});
}
```

jQuery链式操作

- 多个操作可以一直写下去

```
$(".mydiv")  
    .html("hello")  
    .css("color","red")  
    .css("font-size", "18px")  
    .on("click", function(){alert("ok"); } )  
    .on("mouseover", function(){ $(this).css("background", "yellow"); } )  
    .on("mouseout", function(){ $(this).css ("background", "white"); } )  
    ;
```

- this是一特殊的存在

引用jQuery

■ 直接引用网络上的jQuery

□ 引用CDN(内容分发网) 上的库

- 如 **BootCdn**.cn <https://cdn.bootcss.com/jquery/3.4.1/jquery.js>
- 如 `<script src=" http://libs.baidu.com/jquery/1.11.1/jquery.js`
`"></script>`

□ 官网

- `<script src=" http://code.jquery.com/jquery-1.11.3.js "></script>`

■ 也可以下载到本机后进行引用

□ `<script src="jquery.js"></script>`

jQuery 写到哪里

■ 一般写到body的尾部

□ 并将jQuery 操作写到document ready 函数中

□ 以防止文档在完全加载（就绪）之前运行 jQuery 代码：

`$(document).ready(function () {。。。。});`

```
$(document).ready(function() {  
  
    --- jQuery functions go here ---  
  
});
```

□ 简单地写为

`$(function(){ });`

小技巧：浏览器支持类似jQuery

■ 在elements中

- 右键，复制为css path或selector
- 这就是选择器

■ 在console中

- `$(".book")` 要求引用了jQuery
- `$$` 类似于`$`，但有不同，不要求引用jquery
- `$0, $1, $2` 表示当前选择的元素（最后一个，倒数第2个，倒数第3个等）

The background is a solid green color. It is decorated with several white circles of varying sizes, some of which are partially cut off by the edges of the frame. The word "选择" is written in a bold, white, sans-serif font on the left side of the image.

选择

jQuery 选择器

■ jQuery 选择器

- 通过标签名、属性名或内容对 HTML 元素进行选择。
- 选择器允许对 HTML 元素（或者叫DOM节点）或一组元素进行操作
 - `$("div").css("color", "red");` 对所有的div都进行了操作
- 如果没有选取到元素，它并不出错，只是不起作用

选择器的写法

■ 与css选择器类似

- `$("div")` 元素
- `$("#id1")` id
- `$(".class1")` 类
- `$("ul li")` 子孙元素
- `$("ul>li")` 直接子元素

jQuery 选择器

■ jQuery 元素选择器

- `$("p")` 选取 `<p>` 元素。
- `$("p.intro")` 选取所有 `class="intro"` 的 `<p>` 元素。
- `$("p#demo")` 选取 `id="demo"` 的第一个 `<p>` 元素。

jQuery 选择器

■ jQuery 属性选择器

- jQuery 使用 XPath 表达式来选择带有给定属性的元素。
- `$("[href]")` 选取所有带有 href 属性的元素。
- `$("[href='#']")` 选取所有带有 href 值等于 "#" 的元素。
- `$("[href!='#']")` 选取所有带有 href 值不等于 "#" 的元素。
- `$("[href$='.jpg']")` 选取所有 href 值以 ".jpg" 结尾的元素。
- `$("a[href^='http']")` 选取所有 href 值以 "http" 开始的a元素。

jQuery 选择器

■ 更多的选择器实例

语法	描述
<code>\$(this)</code>	当前 HTML 元素
<code>\$("p")</code>	所有 <code><p></code> 元素
<code>\$("p.intro")</code>	所有 <code>class="intro"</code> 的 <code><p></code> 元素
<code>\$(".intro")</code>	所有 <code>class="intro"</code> 的元素
<code>\$("#intro")</code>	<code>id="intro"</code> 的第一个元素
<code>\$("ul li:first")</code>	每个 <code></code> 的第一个 <code></code> 元素
<code>\$("[href\$='.jpg']")</code>	所有带有以 <code>".jpg"</code> 结尾的属性值的 <code>href</code> 属性
<code>\$("div#intro .head")</code>	<code>id="intro"</code> 的 <code><div></code> 元素中的所有 <code>class="head"</code> 的元素

jQuery工作原理

■ 内部是如何选择的

□ 使用document的以下方法

- getElementById
- getElementsByTagName
- getElementByName
- getElementsByName
- querySelectorAll (简称QSA) (这就是console中的\$\$)
- querySelector

□ 并进行了优化与包装

jQuery选择出的对象

■ jQuery对象与html对象的关系

□ jQuery对象是html对象的**包装**，它们是不一样的

- 但是可以这样：`$("#hoo").get(0)` 或者 `$("#hoo")[0]`

- 就是与 `document.getElementById("hoo")` 一样的了

□ `$("#hoo").attr("href", ".....")` 等价于：

- `$("#hoo")[0].href = "....."`

- `document.getElementById("hoo").href = "...."`

■ 为什么可以进行链式操作

□ 这是因为`$()`函数返回的结果是自己

The background is a solid light green color. It is decorated with several thin, white-outlined circles of various sizes scattered across the page. The Chinese characters '事件' (Shìjiàn) are written in a bold, white, sans-serif font on the left side.

事件

jQuery 事件函数

- 当按钮的点击事件被触发时会调用一个函数：
- 该方法隐藏所有 `<p>` 元素：

```
$ ("button").click(function() {..some code
```

```
$ ("p").hide();
```

更一般地

■ on: 注册事件

- `$(xxx).on("click", function(){ });` 推荐使用.on 原生native, .onclick= 或 .addEventListener()
 - 这与 `$(xxx).click(function(){ });` 效果是一样的 .click是早期的写法
- 又如`$(xxx).on("mouseover", function(){}). on("mouseout", function(){});`
 - 相当于`$(xxx).hover(function(){} , function(){})`

■ off: 取消事件

- `$("#p").off("click");`

■ trigger: 激发事件

- `$("#p").trigger("click");` 模拟点击，它用在setInterval()中
- `$("#input").trigger("select");` 由程序进行模拟选择

- 相当于xxxx.addEventListner("click",function(){})

■ 事件函数中，可以用 `$(this)`来表示当前的对象

- 相当于 event.targment

效果

隐藏和显示

■ **hide()** 和 **show()** 来隐藏和显示：

□ `xxx.style.display="none"` 或 `""`

```
$("#hide").click(function() {  
    $("#p").hide();  
});  
$("#show").click(function() {  
    $("#p").show();  
});
```

切换可见性

■ 切换 toggle()

- jQuery toggle() 函数使用 show() 或 hide() 函数来切换 HTML 元素的可见状态。
- 隐藏显示的元素，显示隐藏的元素。

```
$(selector).toggle(speed, callback)
```

滑动效果

■ slideDown, slideUp, slideToggle

□ jQuery 拥有以下滑动函数：

```
$(selector).slideDown(speed, callback)
```

```
$(selector).slideUp(speed, callback)
```

```
$(selector).slideToggle(speed, callback)
```

渐显、渐隐

■ fadeIn(), fadeOut(), fadeTo()

□jQuery 拥有以下 fade 函数：

```
$(selector).fadeIn(speed, callback)
$(selector).fadeOut(speed, callback)
$(selector).fadeTo(speed, opacity, callback)
```

- speed 参数可以设置这些值："slow", "fast", "normal" 或 毫秒 如500
- fadeTo() 函数中的 opacity 参数规定减弱到给定的不透明度
- callback 参数是在该函数完成之后被执行的函数名称

jQuery 效果的参数

■ jQuery效果的参数

□ hide() 和 show() 都可以设置两个可选参数：speed 和 callback。

```
$(selector).hide(speed,callback)  
$(selector).show(speed,callback)
```

□ speed 参数规定显示或隐藏的速度。可以为："slow", "fast", "normal" 或毫秒。

□ callback 参数是在 hide 或 show 函数完成之后被执行的函数名称

Callback 函数 -- 当动画完成后执行

■ jQuery 动画的问题

□ 一般动画是需要时间的，如果我们顺序书写，但实际是按语句执行的。

- `$(".a").hide("slow");`

- `$(".b").show();`

- 则后者可能先执行

□ 为了避免这个情况，可以以参数的形式添加 Callback 函数

- callback 又称回调函数

Callback 函数—让事件先后执行

- 当动画 100% 完成后，即调用 Callback 函数（回调）。

```
$(selector).hide(speed,callback)
```

- callback 参数是一个在 hide 操作完成后被执行的函数。

- 错误（没有 callback）

```
$("#p").hide(1000);  
alert("The paragraph is now hidden");
```

- 正确（有 callback）

```
$("#p").hide(1000,function(){  
    alert("The paragraph is now hidden");  
});
```

- 结论：如果希望在一个涉及动画的函数之后来执行语句，请使用 callback 函数。

自定义动画

■ animate

- jQuery 函数创建自定义动画的语法：

```
$(selector).animate({params},[duration],[easing],[callback])
```

- 关键的参数是 params。它定义产生动画的 CSS 属性。可以同时设置多个此类属性：

```
animate({width:"70%",opacity:0.4,marginLeft:"0.6in",fontSize:"3em"});
```

- 第二个参数是 duration。它定义用来应用到动画的时间。它设置的值是："slow", "fast", "normal" 或毫秒。

应用示例

■ 北大its主页上 <https://its.pku.edu.cn>

□ 右侧的小工具 `$("#gotoTop").fadeIn(100) fadeOut`

□ 还有 `slideUp slideDown Toggle`

The background is a solid light green color. It is decorated with several thin, white-outlined circles of various sizes. Some circles are partially cut off by the edges of the frame. The Chinese characters '操作' (Operation) are written in a bold, white, sans-serif font on the left side of the image.

操作

jQuery HTML 操作

■ 改变 HTML 内容

```
$(selector).html(content)
```

□ html() 函数改变所匹配的 HTML 元素的内容（innerHTML）。

jQuery HTML 操作

■ 添加 HTML 内容

□ append() 函数向所匹配的 HTML 元素内部追加内容。

```
$(selector).append(content)
```

□ prepend() 函数向所匹配的 HTML 元素内部预置 (Prepend) 内容。

```
$(selector).prepend(content)
```

□ after() 函数在所有匹配的元素之后插入 HTML 内容。

```
$(selector).after(content)
```

□ before() 函数在所有匹配的元素之前插入 HTML 内容。

```
$(selector).before(content)
```

属性操作

■.html() 内部html

- 设置 `$(selector).html("aaaaa");`
- 获取 `var t = $(selector).html();`

■.text() 内部文本

■.val() 值（主要针对input）xxxx.value

■.attr() 属性，用来访问各种attribute

- 设置 `$("a").attr("href", "http://a.com");`
- 获取 `var flg = $("input[type=radio]").attr("checked")`

jQuery CSS 函数

■ jQuery CSS 操作

□ jQuery 拥有三种用于 CSS 操作的重要函数：

□ • `$(selector).css(name,value)`

□ • `$(selector).css({properties})`

□ • `$(selector).css(name)`

对css类的操作

■.addClass

□相当于 `xxxxx.classList.add()`

■.removeClass

■.toggleClass

jQuery CSS 函数

■ jQuery Size 操作

□ jQuery 拥有两种用于尺寸操作的重要函数：

□ • `$(selector).height(value)`

□ • `$(selector).width(value)`

应用示例

■ 北大主页上 <https://www.pku.edu.cn>

□ addClass

□ CSS

遍历

遍历

- 遍历是在已选择的基础再查找**相关**的元素
- .parent() 父亲
- .parents("ul") 祖先（所有）
- .closest("ul") 祖先（最近，含自己）
- .children("p.1") 子女
- .find("p") 子孙
- .siblings() 同胞
- .next(), .pre() 前/后一个同胞

过滤

- 过滤是在已选择的集合缩减一部分
- `.first()` 首元素
- `.last()` 尾元素
- `.eq(n)` 第n个元素
- `.filter(".intro")` 按条件选择
- `.not(".intro")` 与filter相反

对每个元素单独操作

■ 使用each

■ .each(function(index,element){

- 其中，可以使用index(在集合中的下标), element(当前HTML DOM元素)

- 也可以使用 `$(this)` 表示当前对象（jQuery对象）

■ });

- `$('p').css(...)`

```
$('p').each( function(i,e){
```

```
    $(this).css()      $('..').eq(i).css()    $(e).css()
```

```
    console.log("%d %s", i, e === $(this)[0]);
```

```
});
```

集合的一些特性

- .length属性，表示元素的个数
- 在each函数中，
 - \$(this) 当前对象 (jQuery对象)
 - this 代表当前的html元素
 - \$(this).index(), 当前对象的下标
 - 或 \$(父亲).index(this)

应用示例：下拉式nav



应用示例：选项卡



应用示例：手风琴菜单



应用示例：图片轮播

- 自己写的简单的图片轮播放

图片轮播

■ 第三方组件 swiper

- <https://www.swiper.com.cn/>

- demo

- <https://www.swiper.com.cn/demo/index.html>

- 其中 自动切换 <https://www.swiper.com.cn/demo/280-autoplay.html>

The background is a solid light green color. It is decorated with several thin, white-outlined circles of various sizes scattered across the page. The Chinese characters '插件' (Add-on) are written in a large, bold, white font on the left side.

插件

jQuery插件

■ 有大量的基于jQuery插件

■ 官网

□ <https://plugins.jquery.com/>

■ 其他

□ <https://www.jq22.com> （国内）

Popular Tags

- [ui](#) (542)
- [jquery](#) (482)
- [form](#) (285)
- [animation](#) (273)
- [input](#) (252)
- [image](#) (210)
- [responsive](#) (184)
- [slider](#) (172)
- [ajax](#) (154)
- [scroll](#) (140)

插件

- 各种改进的控件
- 导航、选项卡、菜单
- UI、布局等
- 一些特殊的
 - 照片墙
 - 弧形文字 <https://circletype.labwire.ca/>

更多（了解）

■ jQuery的替代方案

- 著名的是 zepto.js

- <https://zeptojs.com/>

- 还有

- min.js <https://github.com/remy/min.js>

- dolla.js <https://github.com/lelandrichardson/dolla>

■ 一些工具类

- underscore <https://underscorejs.net/>

- 学习其源码 <https://underscorejs.net/docs/underscore.html>

jQuery太好了，它革了自己的命

■现在浏览器自己支持了

□\$ → \$\$, document.querySelectorAll(".news")

□.on → .addEventListener

□.addClass → .classList.add

□.insert → .parentNode.insertBefore

■参考文章：

- 如何做到 jQuery-free ? <https://kb.cnblogs.com/page/178891>

本讲小结

- 引用jquery, 书写 `$(function(){ });`
- 选择 `$("....")`
- 事件 `.on("click", function(){.....});`
- 效果 `.hide() .show() .toggle() .slideToggle()`
- 操作 `.html() .attr() .css() .addClass()`
- 遍历 `.each(function(){ }); .eq(5) $(this).siblings()`
- 插件 `swiper`

