

简介

\$

入口函数

DOM对象与jQuery对象

选择器

事件

事件类别

事件机制

动画

样式操作

属性操作

节点操作

jQuery特殊属性操作

val方法

html与text方法:

width与height方法

scrollTop与scrollLeft方法

offset与position方法

补充知识点

链式编程

each方法

插件

简介

jq是一个类库 (library) 。write less, do more

环境

开发环境：写代码

测试环境：测试服务器

生产环境：对外开放服务器

git、svn：代码管理

优点

查找元素的方法多种多样，非常灵活

隐式迭代特性，jQuery会偷偷地自动遍历，不需要我们遍历。因此不需要再手写for循环了

链式编程

完全没有兼容问题，例如：低版本火狐浏览器不支持innerText，支持textContent

实现动画非常简单，而且功能更加的强大

代码简单、粗暴

版本

1.X -- 兼容ie 678

2.X -- 放弃兼容ie 678

3.X

压缩与否

1.压缩版本的 (min.js)，上线的时候用，节省体积约200K，用在生产环境

2.正常版本，没有压缩，常用于学习或开发环境

引入的两种方式：

1.直接下载到本地,使用script标签进行导入:<script type="text/javascript" src="js/jquery2.js"></script>

2.CDN:就是远程服务器上的资源

\$

一个函数，以后用\$的时候，记得跟小括号 \$()；

参数不同，功能就不同

三种参数的不同用法

i. 参数是一个function, 入口函数

ii. 把dom对象转换成jquery对象 \$(document).ready(function () {});\$ (btn),\$("<a>");

iii. 参数是一个字符串,用来找对象

入口函数

js的入口函数只能写一个，后面的会覆盖前面的；jq的入口函数可以写多个，且不会被覆盖。

js的入口函数执行要比jQuery的入口函数执行得晚一些。

jq的入口函数会等待页面的加载完成才执行，但是不会等待图片的加载。

js的入口函数会等待页面加载完成，并且等待图片或外部引入资源加载完成才开始执行。

推荐些jq入口函数，更加灵活方便。

两种方式：

1. `$(document).ready(function(){})`

2. `$(function(){})`

DOM对象与jQuery对象

DOM对象(js对象)：用js的方式获取到的对象是DOM对象

jq对象：用jq的方式获取到的对象是jq对象

区别：DOM对象和jq对象是两个不同的对象。DOM对象不能调用jq对象的方法，jq对象也不能调用DOM对象的方法

联系：jq对象其实就是js对象的一个集合，伪数组，里面存放了一大堆的js对象

转换：DOM --> jQuery `$(document)`

jQuery --> DOM `$li[0]` `$li.get(0)`

选择器

jq选择器基本兼容了css所有选择器，并且还添加了很多其他复杂的选择器。

基本选择器

元素选择器： `$("div")`

类选择器： `$(".box")`

id选择器： `$("#box")`

交集选择器： `$("s1s2")`

并集选择器： `$("s1,s2")`

层级选择器

后代选择器： `$("s1 s2")`

子代选择器： `$("s1>s2")`

过滤选择器：这类选择器都带冒号：

下标为偶数过滤： `$("li:even")`

下标为奇数过滤： `$("li:odd")`

下标小于过滤： `$("li:lt(1)")`

下标等于过滤： `$("li:eq(2)")`

下标大于过滤： `$("li:gt(2)")`

第一个过滤： `first`

最后一个过滤：last

筛选选择器：筛选选择器的功能与过滤选择器有点类似，但是用法不一样，筛选选择器主要是方法。

children():找儿子

find():找后代

parent():找爹

parents():找祖先

siblings():找兄弟，不包括自己

next():下一个兄弟

prev():上一次兄弟

eq():指定下标

\$(this)

事件

事件类别

mouseover: 鼠标经过事件，不论鼠标指针穿过被选元素还是其子元素，都会触发

mouseover事件，触发次数多

mouseout:鼠标离开事件

mouseenter:鼠标进入事件，只有鼠标指针穿过被选元素时才会触发

mouseleave: 鼠标离开事件

hover: 两个function

scroll: 滚动事件 scrollTop

change:

resize:

blur/focus

事件机制

发展历程:

简单注册事件只能注册单个事件，不能注册多个事件>bind()可以事件注册多个事件功能，但不支持动态事件绑定 >delegate()解决了动态事件绑定问题，支持注册委托事件，但因此所需要的注册事件太多了>统一on()

事件执行的顺序:

先执行委托事件，再执行自己的事件。

`$(selector).bind(event,[data],function,[map])`

事件映射：只传一个对象{};

添加多个事件;

unbind()

`$(selector).delegate(childSelector,event,[data],function)`

事件委托，利用事件冒泡原理,好处：1.提高性能； 2.动态添加的元素事件也生效；

undelegate()

`$(selector).on(event,[childSelector],[data],function)`

在被选元素及子元素上添加一个或多个事件处理程序。

// 第一个参数：events，绑定事件的名称可以是由空格分隔的多个事件（标准事件或者自定义事件）

// 第二个参数：selector，执行事件的后代元素（可选），如果没有后代元素，那么事件将只有自己执行。

// 第三个参数：data，传递给处理函数的数据，事件触发的时候通过event.data来使用（不常使用）

// 第四个参数：handler，事件处理函数

对应移除事件处理方法off()

// 解绑匹配元素的所有事件：\$(selector).off();

// 解绑匹配元素的所有click事件：\$(selector).off("click");

`$(selector).one(event,[data],function)`

为被选元素添加一个或多个事件处理程序，并规定当事件发生时运行的函数。

当使用 one() 方法时，每个元素只能运行**一次**事件处理程序函数。

触发事件

```
$("#btn").on("click",function () {  
    $("p").click();//方式一  
    $("p").trigger("click");//方式二  
    $(".father").triggerHandler("change")  
});
```

事件对象

//screenX和screenY 对应屏幕最左上角的值

//clientX和clientY 距离页面左上角的位置（忽视滚动条）

//pageX和pageY 距离页面最顶部的左上角的位置（会计算滚动条的距离）

//event.keyCode 按下的键盘代码

//event.data 存储绑定事件时传递的附加数据

//event.stopPropagation() 阻止事件冒泡行为

//event.preventDefault() 阻止浏览器默认行为

//return false:既能阻止事件冒泡，又能阻止浏览器默认行为。

动画

fast:200ms normal:400ms slow:600ms

显示: `$(selector).show(speed,callback);`

隐藏: `$(selector).hide(speed,callback);`

显示隐藏切换: `$(selector).toggle(speed,callback);`

淡入: `$(selector).fadeIn(speed,callback);`

淡出: `$(selector).fadeOut(speed,callback);`

淡入淡出切换: `$(selector).fadeToggle(speed,callback);`

渐变不透明度: `$(selector).fadeTo(speed,opacity,callback);` 渐变为给定的不透明度 (值介于 0 与 1 之间)

下滑: `$(selector).slideDown(speed,callback);`

上滑: `$(selector).slideUp(speed,callback);`

上下滑切换: `$(selector).slideToggle(speed,callback);`

自定义动画: `$(selector).animate({params},speed,callback);`

第一个参数: 对象, 里面可以传需要动画的样式

第二个参数: speed 动画的执行时间

第三个参数: 动画的执行效果, swing: 秋千 摇摆, linear: 线性 匀速

第四个参数: 回调函数

几乎可以用来操作CSS所有属性! 不过, 需要记住一件重要的事情: 当使用 animate() 时, 必须使用 Camel 标记法书写所有的属性名, 比如, 必须使用 paddingLeft 而不是 padding-left, 使用 marginRight 而不是 margin-right, 等等。

同时, 色彩动画并不包含在核心 jQuery 库中。

如果需要生成颜色动画, 您需要从 jquery.com 下载 [颜色动画](#) 插件。

您甚至可以把属性的动画值设置为 "show"、"hide" 或 "toggle":

jQuery 提供针对动画的队列功能

动画队列: 把这些动画存储到一个动画队列里面, 一个一个按顺序执行

停止动画: `$(selector).stop(stopAll,goToEnd);`

停止正在执行的动画。位置放在动画方法之前

参数1: clearQueue: 是否清除动画队列 true false

参数2: jumpToEnd: 是否跳转到当前动画的最终效果 true false

delay的用法:

`$("#div").fadeIn(1000).delay(2000).fadeOut(1000);`

样式操作

js原生代码中有两种方式改变样式，一种是style设置，另一种是添加className设置。同样的jq也有两种操作样式的方法，css方法和class方法

.css(), 返回或设置匹配的元素的一个或多个样式属性。

`$(selector).css(name)`, 获取想要的样式, name想要获取的样式

`$(selector).css(name,value)`, 设置单个样式

`$(selector).css({property:value, property:value, ...})`, 设置多个样式

`$(selector).css(name,function(index,value){})`, 设置所有匹配的元素中样式属性的值。此函数返回要设置的属性值。接受两个参数, index 为元素在对象集中的索引位置, value 是原先的属性值。

class()

添加类: `$("li").addClass("basic")`

删除类: `$("li").removeClass("bigger")`

切换类: `$("li").toggleClass("bigger");`

判断类: `$("li").hasClass("bigger")`。hasClass() 方法检查被选元素是否包含指定的类名称。如果被选元素包含指定的类, 该方法返回 "true"。

属性操作

样式与属性的区别:

样式: 在style里面写的, 用css来操作。

属性: 在标签里面写的, 用attr方法操作。

.attr(), 用法同css方法一样, 参考css方法。获取属性值, 修改单个属性, 修改或添加多个属性.....

删除属性.removeAttr()

.prop, 对于布尔类型的属性, 不要attr方法, 应该用prop方法, prop用法跟attr方法一样

`$("#ck").prop("checked", true);`

节点操作

创建节点: 直接使用单引号或双引号引起来的标签即创建完成

1.`$("#box").append('传智大前端');`

2.`$("#box").append($("#我是创建的li"));`

3.`$("#box").append(``

``

`再从创建多个`

`我是第${num}个 //此处, 变量必须使用${}括起来`

``

)

添加节点:

`$("#div").html($("#option"))`;对比记忆。html添加元素时会清空之前的元素内容,但append是在原来的基础上继续追加。

```
$("#div").append($("#p"));
$("#p").appendTo($("#div"));
$("#div").prepend($("#p"));
$("#p").prependTo($("#div"));
$('div').after($("#p"));
$('div').before($("#p"));
```

清空节点:

`.html("")`,可以清空一个元素的内容,不建议使用,会发生内存泄漏(不能清空相关元素的事件,事件占用空间)

`.empty()`,不仅可以清空元素内容,同时清空元素匹配的事件

删除节点: `.remove()`

克隆节点: `.clone()`--深度复制

不传参数,或者参数为false,都是深度复制,但不会复制元素事件;

参数为true,深度复制的同时也复制保留了元素事件;

jQuery特殊属性操作

val方法

实际上可以通过属性操作`attr("val")`,但jQuery单独给出一个方法方便使用。

不传参数指获取值,传参数指修改值。

html与text方法:

`.text()`,获取或设置文本

`.html()`,获取或创建元素,创建元素

width与height方法

通过css方法获取到的是字符串: 200px;

设置或获取元素的宽高,获取到的是数值型;

获取可视区的宽高: `$(window).width()`,获取的值随着可视区的改变而改变。对应client家族,呼应resize事件

扩展:

```
$("#div").width();//width
$("#div").innerWidth();//padding+width
$("#div").outerWidth();//padding+width+border
$("#div").outerWidth(true);//padding+width+border+margin
```


scrollTop与scrollLeft方法

设置或者获取垂直滚动条的位置

获取的值是数值型

offset与position方法

offset方法获取元素距离document-可视区的位置, .offset() ---> {top:xx,left:xx}

position方法获取的是元素距离有定位的父元素的位置, .position()---->

{top:yy,left:yy}

获取到的都是数值型

三大家族: offset client scroll

补充知识点

链式编程

通常情况下, 只有设置操作才能把链式编程延续下去。因为获取操作的时候, 会返回获取到的相应的值, 无法返回 jQuery对象

end(); // 筛选选择器会改变jQuery对象的DOM对象, 想要回复到上一次的状态, 并且返回匹配元素之前的状态。

each方法

```
$(selector).each(function(index,element){});
```

// 参数一表示当前元素在所有匹配元素中的索引号

// 参数二表示当前元素 (DOM对象)

多库共存

jQuery使用\$作为标示符, 但是如果与其他框架中的\$冲突时, jQuery可以释放\$符的控制权.

```
var c = $.noConflict();//释放$的控制权,并且把$的能力给了c
```

插件

钢琴案例需要再深度学习!!!

数组对象

对象数组

- 1.选择器
- 2.DOM操作
- 3.样式操作
- 4.动画

5.事件处理

6.ajax(没有说到)

7.丰富的插件

音频标签：audio

视频标签：vidio

节流阀：钢琴案例