讲师：佟刚

新浪微博：尚硅谷-佟刚

JavaScript 库作用及对比

- 为了简化 JavaScript 的开发, 一些 JavsScript 库诞生了. JavaScript 库封装了很多预定义的对象和实用函数。能帮助使用者建立有高难度交互的 Web2.0 特性的富客户端页面, 并且兼容各大浏览器
- 当前流行的 JavaScript 库有:

jQuery 简介

- jQuery 是继 Prototype 之后又一个优秀的 JavaScript 库
- jQuery 理念: 写得少, 做得多. 优势如下:
  - 轻量级
  - 强大的选择器
  - 出色的 DOM 操作的封装
  - 可靠的事件处理机制
  - 完善的 Ajax
  - 出色的浏览器兼容性
  - 链式操作方式
  - ……

jQuery: HelloWorld

```
<!-- 引入 jQuery -->
<script type="text/javascript" src="scripts/jquery-1.3.1.js"></script>
<script type="text/javascript">
    $(document).ready(function(){ //等待 dom 元素加载完毕, 类似 window.onload
        alert("HelloWorld"); //弹出一个对话框
    });
</script>
```

jQuery 对象

- jQuery 对象就是**通过 jQuery($()) 包装 DOM 对象后产生的对象**
- **jQuery 对象是 jQuery 独有的**. 如果一个对象是 jQuery 对象, 那么它就可以使用 jQuery 里的方法: $("#persontab").html();
- **jQuery 对象无法使用 DOM 对象的任何方法, 同样 DOM 对象也不能使用 jQuery 里的任何方法**
- **约定**：如果获取的是 jQuery 对象, 那么要在变量前面加上 $.
  - var **$**variable = **jQuery 对象**
  - var variable = DOM 对象

jQuery 对象转成 DOM 对象

- jQuery 对象不能使用 DOM 中的方法, 但如果 jQuery 没有封装想要的方法, 不得不使用 DOM 对象的时候, 有如下两种处理方法:
- (1) **jQuery 对象是一个数组对象**, 可以通过 [index] 的方法得到对应的 DOM对象.

```
var $cr = $("#cr");
var cr = $cr[0];
```

- (2) 使用 jQuery 中的 get(index) 方法得到相应的 DOM 对象

```
var $cr = $("#cr");
var cr = $cr.get(0);
```

DOM 对象转成 jQuery 对象

- 对于一个 DOM 对象, 只需要用 **$()** 把 **DOM 对象包装起来(jQuery 对象就是通过 jQuery 包装 DOM 对象后产生的对象)**, 就可以获得一个 jQuery 对象.

```
var cr = document.getElementById("cr");
var $cr = $(cr);
```

转换后就可以使用 jQuery 中的方法了

jQuery 选择器

- 选择器是 jQuery 的根基, 在 jQuery 中, 对事件处理, 遍历 DOM 和 Ajax 操作都依赖于选择器
- jQuery 选择器的优点:
  - 简洁的写法

```
$("#id")          //document.getElementById("id");
$("tagName")      //document.getElementsByTagName("tagName");
```

  - 完善的事件处理机制

```
//若在网页中没有 id 为 "id" 的元素, 浏览器会报错
//document.getElementById("id").style.color = "red";

//需要先判断 document.getElementById("id") 是否存在
if(document.getElementById("id"))
    document.getElementById("id").style.color = "red";

//使用 jQuery 获取网页中的元素即使不存在页不会报错
$("#id").css("color", "red");
```

基本选择器

- 基本选择器是 jQuery 中最常用的选择器, 也是最简单的选择器, 它通过元素 id, class 和标签名来查找 DOM 元素(在网页中 id 只能使用一次, class 允许重复使用).

选择器	描述	返回
#id	根据给定的 id 匹配一个元素	单个元素组成的集合
.class	根据给定的类名匹配元素	集合元素
element	根据给定的元素名匹配元素	集合元素
*	匹配所有元素	集合元素
selector1, selector2, ... , selectorN	将每一个选择器匹配到的元素合并后一起返回	集合元素

基本选择器示例

- 改变 id 为 one 的元素的背景色为 # bbffaa
- 改变 class 为 mini 的所有元素的背景色为 # bbffaa
- 改变元素名为 的所有元素的背景色为 # bbffaa
- 改变所有元素的背景色为 # bbffaa
- 改变所有的元素和 id 为 two 的元素的背景色为 # bbffaa

层次选择器

- 如果想通过 DOM 元素之间的层次关系来获取特定元素, 例如后代元素, 子元素, 相邻元素, 兄弟元素等, 则需要使用层次选择器.

选择器	描述	返回
$("ancestor descendant")	选取 ancestor 的所有 descendant(后代)元素	集合元素
$("parent > child")	选取 parent 元素下的 child(子) 元素, 与 $("ancestor descendant") 有区别, $("ancestor descendant")选择的是后代元素	集合元素
$("prev + next")	选取紧接在 prev 元素后的下一个 next 元素	集合元素
$("prev ~ siblings")	选取 prev 元素后的所有 siblings 元素	集合元素

- 注意: (“prev ~ div”) 选择器只能选择 “# prev” 元素**后面的同辈元素**; 而 jQuery 中的方法 **siblings()** 与前后位置无关, 只要是**同辈节点**就可以选取

层次选择器示例

- 改变 <body> 内所有 的背景色为 # bbffaa
- 改变 <body> 内子 的背景色为 # bbffaa
- 改变 id 为 one 的下一个 的背景色为 # bbffaa
- 改变 id 为 two 的元素后面的所有兄弟 的元素的背景色为 # bbffaa
- 改变 id 为 two 的元素所有 兄弟元素的背景色为 # bbffaa

过滤选择器

- **过滤选择器**主要是通过特定的过滤规则来筛选出所需的 DOM 元素, 该选择器**都以 “:” 开头**
- 按照不同的过滤规则, 过滤选择器可以分为基本过滤, 内容过滤, 可见性过滤, 属性过滤, 子元素过滤和表单对象属性过滤选择器.

基本过滤选择器

选择器	描述	返回
:first	选取第一个元素	单个元素组成的集合
:last	选取最后一个元素	集合元素
:not(selector)	去除所有与给定选择器匹配的元素	集合元素
:even	选取索引时偶数的所有元素，索引从0 开始	集合元素
:odd	选取索引时奇数的所有元素，索引从0 开始	集合元素
:eq(index)	选取索引等于 index 的元素，索引从0 开始	集合元素
:gt(index)	选取索引大于 index 的元素，索引从0 开始	集合元素
:lt(index)	选取索引小于 index 的元素，索引从0 开始	集合元素
:header	选取所有的标题元素，如：h1，h2 等	集合元素
:animated	选取当前正在执行动画的所有元素	集合元素

基本过滤选择器示例

- 改变第一个 div 元素的背景色为 # bbffaa
- 改变最后一个 div 元素的背景色为 # bbffaa
- 改变class不为 one 的所有 div 元素的背景色为 # bbffaa
- 改变索引值为偶数的 div 元素的背景色为 # bbffaa
- 改变索引值为奇数的 div 元素的背景色为 # bbffaa
- 改变索引值为大于 3 的 div 元素的背景色为 # bbffaa
- 改变索引值为等于 3 的 div 元素的背景色为 # bbffaa
- 改变索引值为小于 3 的 div 元素的背景色为 # bbffaa
- 改变所有的标题元素的背景色为 # bbffaa
- 改变当前正在执行动画的所有元素的背景色为 # bbffaa

内容过滤选择器

- 内容过滤选择器的过滤规则主要体现在它所包含的子元素和文本内容上

选择器	描述	返回
:contains(text)	选取含有文本内容为 text 的元素	集合元素
:empty	选取不包含子元素或者文本的空元素	集合元素
:has(selector)	选取含有选择器所匹配的的元素的元素	集合元素
:parent	选取含有子元素或者文本的元素	集合元素

内容过滤选择器示例

- 改变含有文本 ‘di’ 的 div 元素的背景色为 # bbffaa
- 改变不包含子元素(或者文本元素) 的 div 空元素的背景色为 # bbffaa
- 改变含有 class 为 mini 元素的 div 元素的背景色为 # bbffaa
- 改变含有子元素(或者文本元素)的div元素的背景色为 # bbffaa

可见性过滤选择器

- 可见性过滤选择器是**根据元素的可见和不可见状态**来选择相应的元素

选择器	描述	返回
:hidden	选取所有不可见的元素	集合元素
:visible	选取所有可见的元素	集合元素

- 可见选择器 :hidden 不仅包含样式属性 display 为 none 的元素, 也包含文本隐藏域 (<input type="hidden">)和 visible:hidden 之类的元素

可见性过滤选择器示例

- 改变所有可见的div元素的背景色为 # bbffaa
- 选取所有不可见的元素, 利用 jQuery 中的 show() 方法将它们显示出来, 并设置其背景色为 # bbffaa
- 选取所有的文本隐藏域, 并打印它们的值

属性过滤选择器

- 属性过滤选择器的过滤规则是**通过元素的属性来获取相应的元素**

选择器	描述	返回
[attribute]	选取拥有此属性的元素	集合元素
[attribute=value]	选取指定属性的值为value 的元素	集合元素
[attribute!=value]	选取指定属性的值不等于 value 的元素	集合元素
[attribute^=value]	选取指定属性的值以 value 开始的元素	集合元素
[attribute$=value]	选取指定属性的值以 value 结束的元素	集合元素
[attribute*=value]	选取指定属性的值含有 value 的元素	集合元素
[selector1][selector2]...[selectorN]	用属性选择器合并成一个复合属性选择器，满足多个条件．每选择一次，缩小一次范围	集合元素

属性过滤选择器示例

- **选取下列元素,改变其背景色为 # bbffaa**
- 含有属性title 的div元素.
- 属性title值等于"test"的div元素.
- 属性title值不等于"test"的div元素(**没有属性title的也将被选中**).
- 属性title值 以"te"开始 的div元素.
- 属性title值 以"est"结束 的div元素.
- 属性title值 含有"es"的div元素.
- 选取有属性id的div元素，然后在结果中选取属性title值含有“es”的 div 元素.

子元素过滤选择器

选择器	描述	返回
:nth-child(index/even/odd/equation)	选取每个父元素下的第 index 个子元素或者奇偶元素(index 从 1 算起)	集合元素
:first-child	选取每个父元素的第一个子元素	集合元素
:last-child	选取每个父元素的最后一个子元素	集合元素
:only-child	如果某个元素是它父元素中唯一的子元素，那么将被匹配.	集合元素

- nth-child() 选择器详解如下：
  - (1) :nth-child(even/odd): 能选取每个父元素下的索引值为偶(奇)数的元素
  - (2):nth-child(2): 能选取每个父元素下的索引值为 2 的元素
  - (3):nth-child(3n): 能选取每个父元素下的索引值是 3 的倍数 的元素
  - (3):nth-child(3n + 1): 能选取每个父元素下的索引值是 3n + 1的元素

子元素过滤选择器示例

- 选取下列元素,改变其背景色为 # bbffaa
- 每个class为one的div父元素下的第2个子元素.
- 每个class为one的div父元素下的第一个子元素
- 每个class为one的div父元素下的最后一个子元素
- 如果class为one的div父元素下的仅仅只有一个子元素，那么选中这个子元素

表单对象属性过滤选择器

- 此选择器主要对所选择的表单元素进行过滤

选择器	描述	返回
:enabled	选取所有可用元素	集合元素
:disabled	选取所有不可用元素	集合元素
:checked	选取所有被选中的元素(单选框，复选框)	集合元素
:selected	选取所有被选中选项元素(下拉列表)	集合元素

表单对象属性过滤选择器示例

- 利用 jQuery 对象的 val() 方法改变表单内可用 <input> 元素的值
- 利用 jQuery 对象的 val() 方法改变表单内不可用 <input> 元素的值
- 利用 jQuery 对象的 length 属性获取多选框选中的个数
- 利用 jQuery 对象的 text() 方法获取下拉框选中的内容

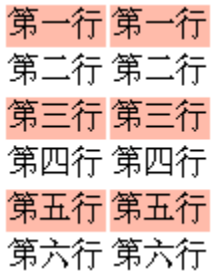
表单选择器

选择器	描述	返回
:input	选取所的 <input>, <textarea>, <select>,和<button>元素	集合元素
:text	选取所有的单行文本框	集合元素
:password	选取所有的密码框 元素	集合元素
:radio	选取所有的单选框	集合元素
:checkbox	选取所有的多选框	集合元素
:submit	选取所有的提交按钮	集合元素
:image	选取所有的图像按钮	集合元素
:reset	选取所有的重置按钮	集合元素
:button	选取所有的按钮	集合元素
:file	选取所有的上传域	集合元素
:hidden	选取所有的不可见元素	集合元素

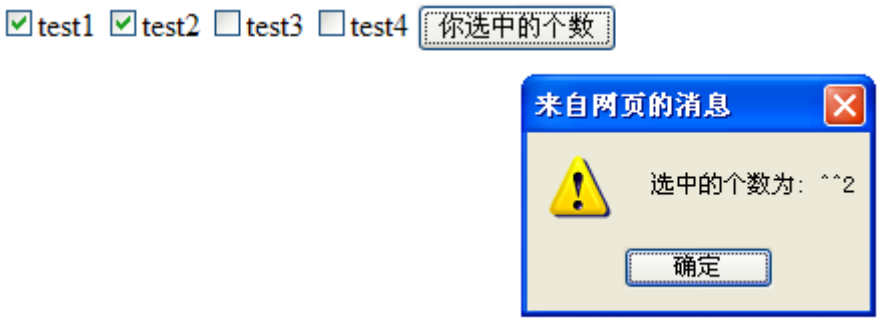
练习

- 1. 给网页中所有的 <p> 元素添加 onclick 事件

- 2. 是一个特定的表格隔行变色



- 3. 对多选框进行操作, 输出选中的多选框的个数



jQuery 中的 DOM 操作

- DOM(Document Object Model—文档对象模型)：一种与浏览器, 平台, 语言无关的接口, 使用该接口可以轻松地访问页面中所有的标准组件
- DOM 操作的分类:
  - DOM Core: DOM Core 并不专属于 JavaScript, 任何一种支持 DOM 的程序设计语言都可以使用它. 它的用途并非仅限于处理网页, 也可以用来处理任何一种是用标记语言编写出来的文档, 例如: XML
  - HTML DOM: 使用 JavaScript 和 DOM 为 HTML 文件编写脚本时, 有许多专属于 HTML-DOM 的属性
  - CSS-DOM:针对于 CSS 操作, 在 JavaScript 中, CSS-DOM 主要用于获取和设置 style 对象的各种属性

查找节点

- 查找节点:
  - 查找属性节点: 通过 jQuery 选择器完成.
  - 操作属性节点: 查找到所需要的元素之后, 可以调用 **jQuery 对象**的 attr() 方法来获取它的各种属性值
  - 操作文本节点：通过 text() 方法

创建节点

- 创建节点: 使用 jQuery 的工厂函数 $(): **$(html)**; 会根据传入的 html 标记字符串创建一个 DOM 对象, 并把这个 DOM 对象包装成一个 **jQuery 对象**返回.
- 注意:
  - **动态创建的新元素节点不会被自动添加到文档**中, 而是需要使用其他方法将其插入到文档中;
  - 当创建单个元素时, 需**注意闭合标签和使用标准的 XHTML 格式**. 例如创建一个 元素, 可以使用 $(" ") 或 $("** **"), 但不能使用 $(" ") 或 $("<P>")
- 创建文本节点就是在创建元素节点时直接把文本内容写出来; 创建属性节点也是在创建元素节点时一起创建

插入节点(1)

- 动态创建 HTML 元素并没有实际用处, 还需要将新创建的节点插入到文档中, 即成为文档中某个节点的子节点

方法	描述
append()	向每个匹配的元素的**内部的结尾处**追加内容
appendTo()	将每个匹配的元素追加到指定的元素中的**内部的结尾处**
prepend()	向每个匹配元素的**内部的开始处**插入内容
prependTo()	将每个匹配的元素插入到指定的元素**内部的开始处**

插入节点(2)

方法	描述
after()	向每个匹配的元素的之后插入内容
insertAfter()	向每个匹配的元素的插入到指定的元素之后
before()	向每个匹配的元素的之前插入内容
insertBefore()	向每个匹配的元素的插入到指定的元素之前

- 以上方法不但能将新创建的 DOM 元素插入到文档中, 也**能对原有的 DOM 元素进行移动**.

删除节点

- remove(): 从 DOM 中删除所有匹配的元素, 传入的参数用于根据 jQuery 表达式来筛选元素. 当某个节点用 remove() 方法删除后, 该节点所包含的所有后代节点将被同时删除. 这个方法的返回值是一个指向已被删除的节点的引用.
- empty(): 清空节点 – 清空元素中的所有后代节点(不包含属性节点).

复制节点

- clone(): 克隆匹配的 DOM 元素, 返回值为克隆后的副本. 但此时复制的新节点不具有任何行为.
- clone(true): 复制元素的同时也复制元素中的的事件

替换节点

- replaceWith(): 将所有匹配的元素都替换为指定的 HTML 或 DOM 元素
- replaceAll(): 颠倒了的 replaceWith() 方法.
- 注意: 若在替换之前, 已经在元素上绑定了事件, 替换后原先绑定的事件会与原先的元素一起消失

包裹节点

- wrap(): 将指定节点用其他标记包裹起来. 该方法对于需要在文档中插入额外的结构化标记非常有用, 而且不会破坏原始文档的语义.
- wrapAll(): 将所有匹配的元素用一个元素来包裹. 而 wrap() 方法是将所有的元素进行单独包裹.
- wrapInner(): 将每一个匹配的元素的子内容(包括文本节点)用其他结构化标记包裹起来.

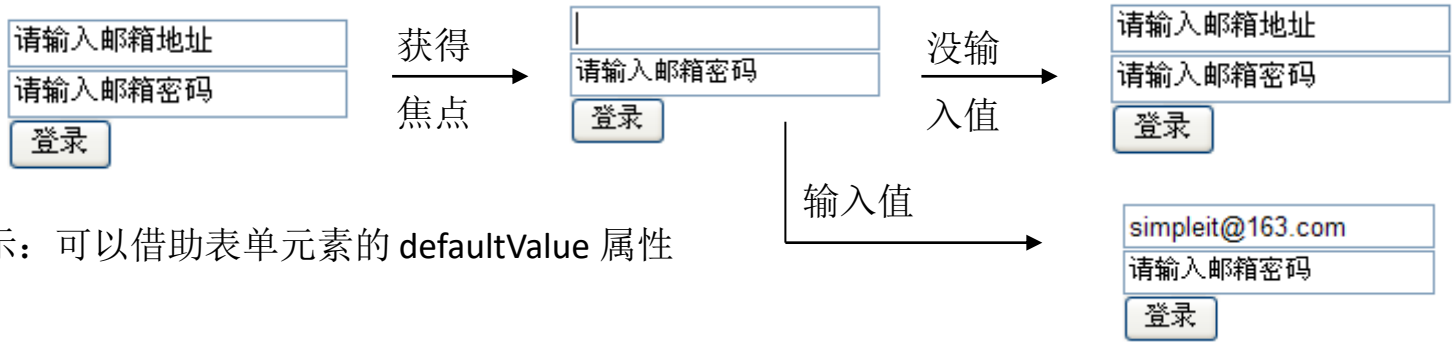
属性操作

- attr(): 获取属性和设置属性
  - 当为该方法传递一个参数时, 即为某元素的获取指定属性
  - 当为该方法传递两个参数时, 即为某元素设置指定属性的值
- jQuery 中有很多方法都是一个函数实现获取和设置. 如: attr(), html(), text(), val(), height(), width(), css() 等.
- removeAttr(): 删除指定元素的指定属性

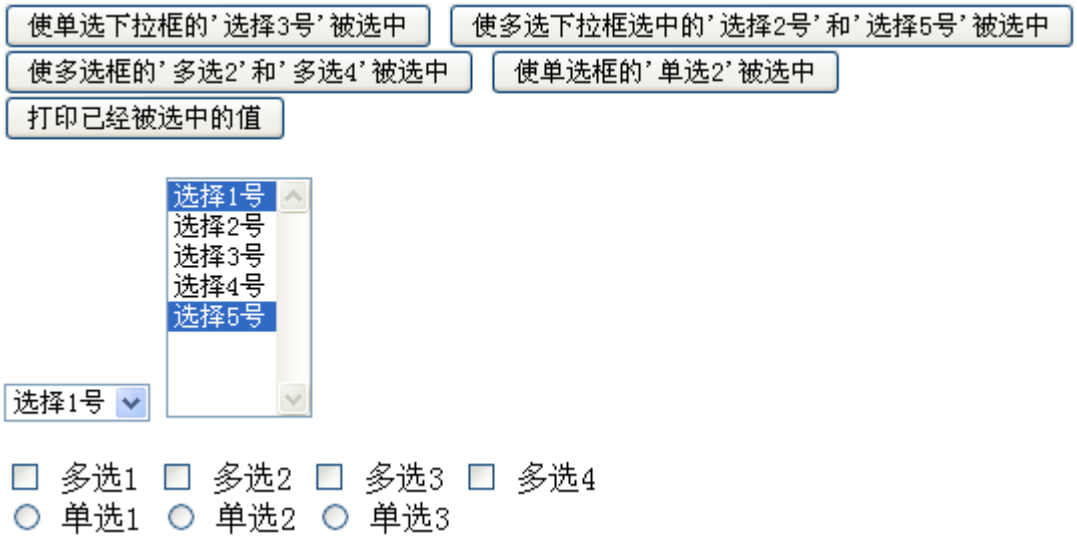
设置和获取 HTML，文本和值

- 读取和设置某个元素中的 **HTML 内容**: html() . 该方法可以用于 XHTML, 但不能用于 XML 文档
- 读取和设置某个元素中的**文本内容**: text(). 该方法既可以用于 XHTML 也可以用于 XML 文档.
- 读取和设置某个元素中的值: val() --- 该方法类似 JavaScript 中的 value 属性. 对于**文本框, 下拉列表框, 单选框**该方法可返回元素的值(**多选框只能返回第一个值**). 如果为多选下拉列表框, 则返回一个包含所有选择值的数组

val() 方法的两个练习



提示：可以借助表单元素的 defaultValue 属性



提示：js 中数组的表示方法

["1", "2"]

常用的遍历节点方法

- 取得匹配元素的所有子元素组成的集合: children(). 该方法只考虑子元素而不考虑任何后代元素.
- 取得匹配元素后面紧邻的同辈元素的集合(但集合中只有一个元素): next()
- 取得匹配元素前面紧邻的同辈元素的集合(但集合中只有一个元素): prev()
- 取得匹配元素前后所有的同辈元素: **siblings()**

样式操作

- 获取 class 和设置 class : class 是元素的一个属性, 所以获取 class 和设置 class 都可以使用 attr() 方法来完成.
- 追加样式: addClass()
- 移除样式: removeClass() --- 从匹配的元素中删除全部或指定的 class
- 切换样式: toggleClass() --- 控制样式上的重复切换.如果类名存在则删除它, 如果类名不存在则添加它.
- 判断是否含有某个样式: hasClass() --- 判断元素中是否含有某个 class, 如果有, 则返回 true; 否则返回 false

CSS-DOM 操作

- 获取和设置元素的样式属性: css()
- 获取和设置元素透明度: opacity 属性
- 获取和设置元素高度, 宽度: height(), width(). 在设置值时, 若只传递数字, 则默认单位是 px. 如需要使用其他单位则需传递一个字符串, 例如 $(“p:first”).height(“2em”);
- 获取元素在当前视窗中的相对位移: offset(). 其返回对象包含了两个属性: top, left. 该方法只对可见元素有效

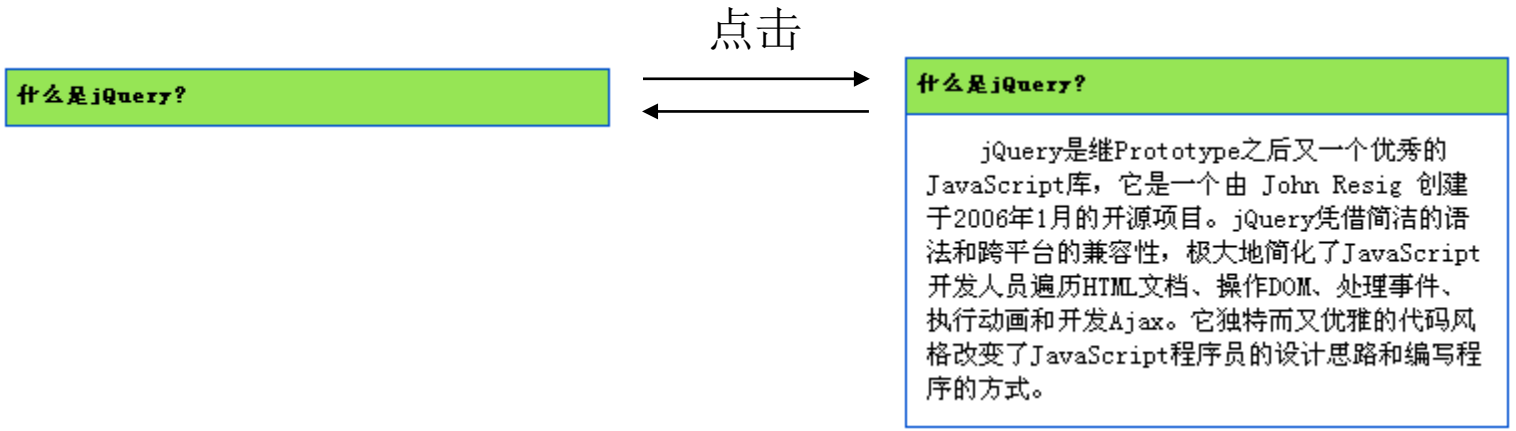
jQuery 中的事件-- 加载 DOM

- 在页面加载完毕后, 浏览器会通过 JavaScript 为 DOM 元素添加事件. 在常规的 JavaScript 代码中, 通常使用 window.onload 方法, 在 jQuery 中使用 $(document).ready() 方法.

方法	window.onload	$(document).ready()
执行时机	必须等待网页中的所有内容加载完毕后(包括图片)才能执行	网页中的所有 DOM 结构绘制完毕后就执行, 可能 DOM 元素关联的东西并没有加载完
编写个数	不能同时编写多个	能同时编写多个
简化写法	无	$()

事件绑定

- 对匹配的元素进行特定的事件绑定: bind()



提示： 使用 jQuery 的 **is()** 方法判断元素是否可见

合成事件

- hover(): 模拟光标悬停事件. 当光标移动到元素上时, 会触发指定的第一个函数, 当光标移出这个元素时, 会触发指定的第二个函数.
- toggle(): 用于模拟鼠标连续单击事件. 第一次单击元素, 触发指定的第一个函数, 当再一次单击同一个元素时, 则触发指定的第二个函数, 如果有更多个函数, 则依次触发, 直到最后一个.
- toggle() 的另一个作用: 切换元素的可见状态.

事件冒泡

- 事件会按照 DOM 层次结构像水泡一样不断向上只止顶端
- 解决: 在事件处理函数中返回 false, 会对事件停止冒泡. 还可以停止元素的默认行为.

事件对象的属性

- 事件对象: 当触发事件时, 事件对象就被创建了. 在程序中使用事件只需要为函数添加一个参数. 该事件对象只有事件处理函数才能访问到. 事件处理函数执行完毕后, 事件对象就被销毁了.
- event.pageX, event.pageY: 获取到光标相对于页面的 x, y 坐标.

移除事件

- 移除某按钮上的所有 click 事件: $(“btn”).unbind(“click”)
- 移除某按钮上的所有事件: $(“btn”).unbind();
- one(): 该方法可以为元素绑定处理函数. 当处理函数触发一次后, 立即被删除. 即在每个对象上, 事件处理函数只会被执行一次.

```
$("a").one("click", function(){
    alert("click me just once!");
    return false;
});
```

jQuery 中的动画：隐藏和显示

- hide(): 在 HTML 文档中, 为一个元素调用 hide() 方法会将该元素的 display 样式改为 none. 代码功能同 css(“display”, “none”);
- show(): 将元素的 display 样式改为先前的显示状态.
- 以上两个方法在不带任何参数的情况下, 作用是立即隐藏或显示匹配的元素, 不会有任何动画. 可以通过制定速度参数使元素动起来.
- 以上两个方法会同时减少(增大)内容的高度, 宽度和不透明度.

jQuery 中的动画(2)

- fadeIn(), fadeOut(): 只改变元素的透明度. fadeOut() 会在指定的一段时间内降低元素的不透明度, 直到元素完全消失. fadeIn() 则相反.
- slideDown(), slideUp(): 只会改变元素的高度. 如果一个元素的 display 属性为 none, 当调用 slideDown() 方法时, 这个元素将由上至下延伸显示. slideUp() 方法正好相反, 元素由下至上缩短隐藏.

jQuery 中的动画(3)

- toggle(): 切换元素的可见状态: 如果元素时可见的, 则切换为隐藏; 如果元素时隐藏的, 则切换为可见的.
- slideToggle(): 通过高度变化来切换匹配元素的可见性.
- fadeTo(): 把不透明度以渐近的方式调整到指定的值 (0 – 1 之间).

练习3：品牌列表

注意: 两个过滤函数 is 和 filter 的使用

练习4：超链接和图片提示效果

提示1

提示2 这是我的超链接提示1

自带提示1

自带提示2

提示1

提示2

自带提示1

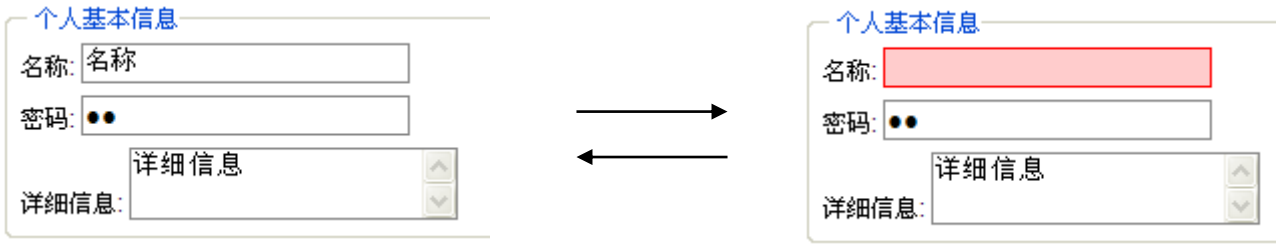
这是自带提示1

有效果：

无效果：

```
<div id='tooltip'><img src='images/apple_4_bigger.jpg' /><br>苹果 Mac</div>
```

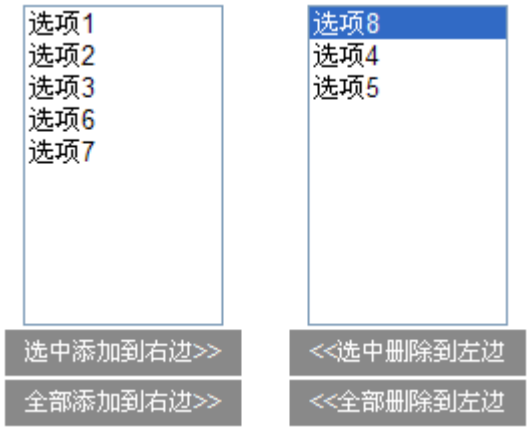
练习5：单行文本框的用户体验



练习6：多选框应用

你爱好的运动是？ ☐全选/全不选
☐足球 ☐篮球 ☐羽毛球 ☐乒乓球
全　选 | 全不选 | 反　选 | 提　交

练习7：下拉框应用


选项1
选项2
选项3
选项6
选项7
选项8
选项4
选项5
选中添加到右边>>
全部添加到右边>>
<<选中册除到左边
<<全部册除到左边

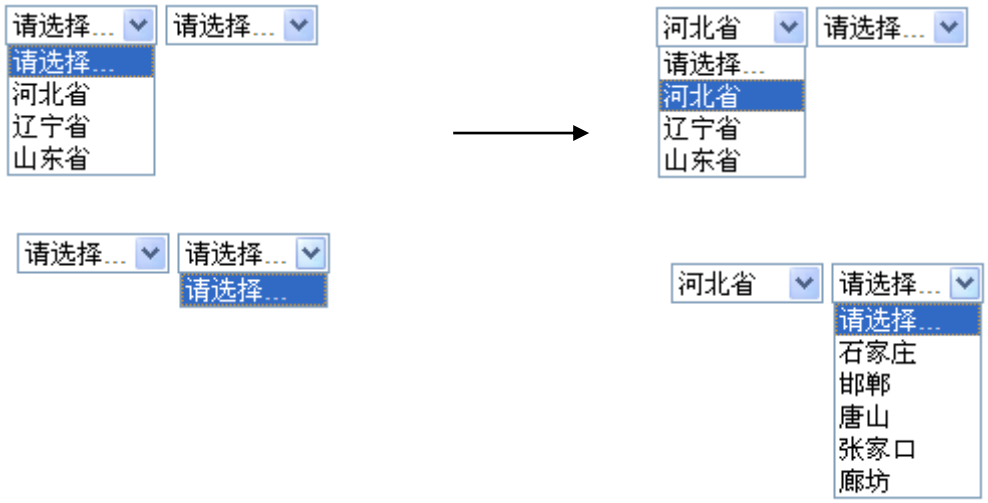
JQuery 加载并解析 XML

- JQuery 可以通过 $.get() 或 $.post() 方法来加载 xml.

```
$(function(){
    $.get("cities.xml", function(xml){
        alert(xml);
    });
});
```

- JQuery 解析 XML 与解析 DOM 一样, 可以使用 find(), children() 等函数来解析和用 each() 方法来进行遍历

练习8：使用 JQuery 实现



练习9：使用 JQuery 实现

添加新员工

name: email: salary:

Submit

Name	Email	Salary	
Tom	tom@tom.com	5000	Delete
Jerry	jerry@sohu.com	8000	Delete
Bob	bob@tom.com	10000	Delete


尚硅谷
www.atguigu.com