# JQuery

1. 简化JavaScript开发，使用JS的封装库，建立高难度的Web2.0界面，兼容各大浏览器。

封装DOM、事件处理机制、完善的Ajax、链式操作。**必须导入JQuery包：**

**<!--导入JQuery包-->  
 <script type="text/javascript" src="jquery-1.7.2.js"></script>  
 <script type="text/javascript">**

1. **JQuery对象，通过Jquery($())封装DOM对象得到的。**

* **JQuery对象、DOM对象，这两者都不能访问对方的方法**
* **JQuery对象的表示方法：在变量面前加上$**

**eg:**var $param = Jquey对象

var param = DOM对象

* **JQuery对象是一个数组：是一个DOM对象数组，用$()包装DOM对象**
* **JQuery和DOM对象互转：**
* **JQuery转为DOM：转换之后可以使用DOM的方法**

（1）使用[index]方法获取

var $cr = $(“#cr”);

var cr = $cr[0];

（2）使用get(index)获取

var $cr = $(“#cr”);

var cr = $cr.get(0);

* **DOM转为JQuery**：转换之后可以使用JQuery的方法

var cr = document.getElementById(“cr”);

var $cr = $(cr);

* **示例代码：**

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
 <title>Title</title>  
 **<!--导入JQuery包 ： jquery-1.7.2.js 下载，到官网复制连接，用迅雷下载-->**  
 <script type="text/javascript" src="jquery-1.7.2.js"></script>  
 <script type="text/javascript">  
 **//$(function(){})等同于window.onload = function(){}**  
 $(function () {

**//根据名称获取对象，并且编写响应函数**  
 **$**("button").**click**(function () {  
 alert("hello world");  
 **//获取JQuery方法**  
 var $btn = $("button");  
 **//获取JQuery方法对象是一个数组：这里有两个按键，长度=2**  
 alert($btn.length);  
 **//获取JQuery转为DOM：打印第2个按键的值**  
 alert($btn[1].firstChild.nodeValue);  
 })  
 })  
 </script>  
</head>  
<body>  
 <button>Click</button>  
 <br><br>  
 <button>ClickMe2</button>  
</body>  
</html>

1. **选择器(最重要的部分)：基本选择器、层次选择器、过滤选择器结合使用**

* **基本选择器：**

通过元素的id、class、标签名来查找DOM元素。网页中：id只能使用一次，class可以多次使用。**5个基本选择器如下：**

* **$(“#id”)：**根据id匹配，返回集合
* **$(“.class”)：**根据类名匹配元素，返回集合元素
* **$(“element”)：**根据给定元素名匹配
* **$(“\*”)：**匹配所有元素
* **$(“selector1、selector2..”)：**合并多个选择器，再一起使用

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
 <title>Title</title>  
 **<!--导入JQuery包-->**  
 <script type="text/javascript" src="jquery-1.7.2.js"></script>  
 <script type="text/javascript">  
 $(function () {  
 $(**"#btn1"**).click(function () {  **//按键的点击操作函数**  
 **//根据id选择**  
 $(**"#one**").css("background", "#ffbba");  
 });  
 $("#btn2").click(function () {

**//根据class名选择**  
 $(**".min"**).css("background", "#ffbbaa");  
 });  
 $("#btn3").click(function () {  
  **//根据元素名选择：实际上时标签名**  
 $(**"div"**).css("background", "#ffbbaa");  
 });  
 $("#btn4").click(function () {  
  **//选择所有**  
 $(**"\*"**).css("background", "#ffbbaa");  
 });  
 $("#btn5").click(function () {

**//合并几个选择器**  
 $(**"sapn, #two"**).css("background", "#ffbbaa");  
 });  
 })  
 </script>  
</head>

<body>  
 <input type="button" value="选择 id 为 one 的元素" id="btn1" />  
 <input type="button" value="选择 class 为 mini 的所有元素" id="btn2" />  
 <input type="button" value="选择 元素名是 div 的所有元素" id="btn3" />  
 <input type="button" value="选择 所有的元素" id="btn4" />  
 <input type="button" value="选择 所有的 span 元素和id为two的元素" id="btn5" />  
 <br><br>  
 <**div** class="**one**" id="**two**" title="test">  
 id为two,class为one,title为test的div  
 <div class="**mini**" title="other">class为mini,title为other</div>  
 <div class="mini" title="test">class为mini,title为test</div>  
 </div>  
 <div style="display:none;" class="**none**">style的display为"none"的div</div>  
 <**span** id="span">^^span元素^^</span>  
</body>  
</html>

* **层次选择器：**

通过DOM元素之间的层次关系获取特定元素，eg：后代元素、子元素、相邻元素、兄弟元素。

**几个常用的层次选择器如下：**

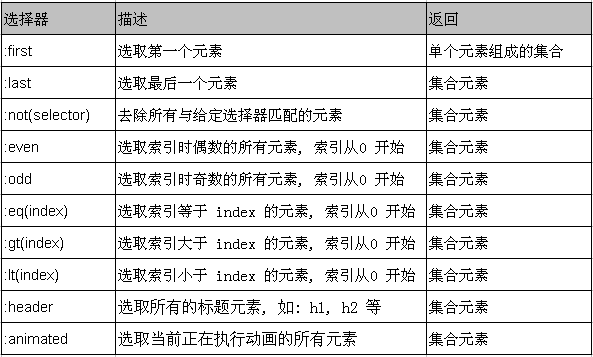
* $(“ancestor descendant”)：选择ancestor元素的所有descendant元素
* $(“parent > child”) :选择parent元素下的子child元素
* $(“prev + next”) :选择紧邻prev元素的下一个next元素
* $(“prev ～ siblings”) :选择prev元素之后的所有siblings元素
* $(“#id”).siblings(“”) ：等同于$(“prev ～ siblings”)
* $(“#id”).nextAll(“”)
* $(“#id”).prevAll(“”)

* **过滤选择器：**

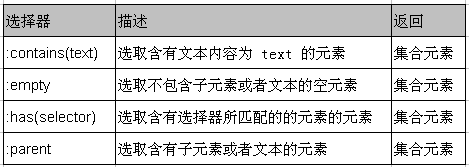
过滤选择器使用规则来过滤DOM元素，**该选择器用“：”开头。**

过滤选择器分类：内容、可见性、属性、子元素、表但对象属性过滤。

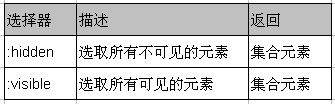
* **基本过滤选择器：**



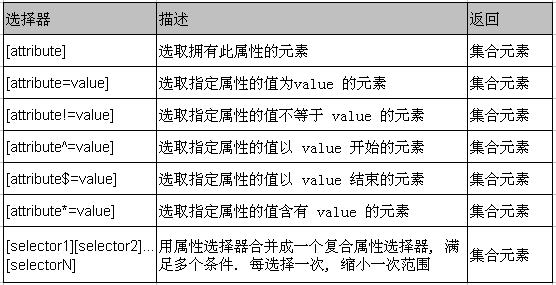
* **内容过滤选择器：过滤子元素、文本内容**



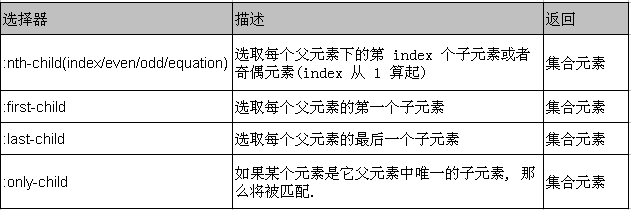
* **可见性过滤选择器：过滤可见、不可见状态的元素**



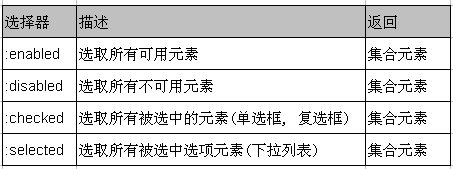
* **属性过滤选择器：**



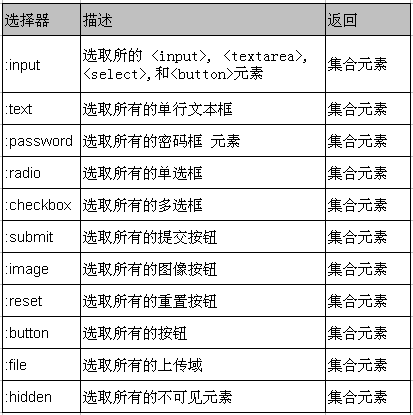
* **子元素过滤选择器：**



* **表单对象属性过滤选择器：过滤表单元素**



* **表单选择器：**



* **示例代码：**

**//子节点选择器：需要在选择器前添加一个空格.**

$("#btn1").click(function(){

$("div.one **:nth-child(2)**").css("background", "#ffbbaa");

});

**//表单选择器：使所有的可用的单行文本框的 value 值变为‘尚硅谷’**

$("#btn2").click(function(){

alert($("**:text**:enabled").val());

$("**:text**:enabled").val("尚硅谷");

});

1. **JQuery操作DOM**

* **查找节点： 结合选择器操作DOM元素**

**//1. 操作文本节点: 通过 jQuery 对象的 text() 方法，修改id=bj的text值=北京**

$("#bj")**.text**("尚硅谷");

**//2. 操作属性节点: 通过 jQuery 对象的 attr() 方法.**

//注: 直接操作 value 属性值可以使用更便捷的 val() 方法.

$(":text[name=**'username'**]")**.attr**("**value**", "尚硅谷");

<body>

<ul id="**city**"><li id="bj" name="BeiJing">**北京**</li>

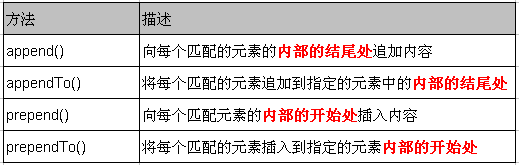
**<br><br> //换两行**

**//默认值：atguigu**

name: <input type="text" name="**username**" **value**="atguigu"/>

</body>

* **插入节点：**



**//下面两者等价**

$("<li id='atguigu'>尚硅谷</li>")**.prependTo**($("#city"));

$("#city")**.prepend**("<li id='atguigu'>[尚硅谷]</li>");

* **删除节点：**

**//DOM 节点直接删除.**

$("#bj")**.remove()**;

**//清空子节点**

$("#game")**.empty()**;

<body>

<ul id="game">

<li id="rl">红警</li>

<li>实况</li>

<li>极品飞车</li>

<li>魔兽</li>

</ul>

</body>

* **复制节点：**

1、clone 节点时，**若原节点有 id属性, 则克隆后, 会出现在一个文档中有两个 id 相同的节点的情况.**

2、.**clone(true):** 在克隆节点的同时, 克隆节点包含的事件.

3、**.attr**("id", "bj2") ：克隆节点后，修改id属性

4、**.insertAfter**($("#rl")：克隆后，插入到id=rl节点之后

**示例代码：**

$("#bj")**.clone**(true)

.attr("id", "bj2")

.insertAfter($("#rl"));

* **替换节点：replaceAll、repalceWith**

**//1. 创建一个 <li>尚硅谷</li> 节点, 替换 #city 的最后一个 li 子节点**

$("<li>尚硅谷</li>")**.replaceAll**($("#city li:last"));

**//2. 创建一个 <li>[尚硅谷]</li> 节点, 替换 #city 的第二个 li 子节点**

$("#city li:eq(1)")**.replaceWith**($("<li>[尚硅谷]</li>"));

**//3、互换节点：需要先克隆节点**

var $bj2 = $("#bj").clone(true); **//克隆id=bj的节点**

var $rl = $("#rl").replaceWith($bj2); **//用$bj2克隆节点，去替换id=rl的节点**

$("#bj").replaceWith($rl);  **//将原本id=rl节点，赋值给id=bj的节点**

* **包裹节点：**

**用于css修改标记，插入额外的结构化标记，不会修改原来的代码意思，只是添加代码。**

* **.wrap()：**分开包裹，每个对象上都被添加包裹
* **.wrapAll()：**将整体进行包裹，对象的上一级，被包裹
* **.wrapInner()：**仅包裹文字
* **示例代码：**

**//包装 li 本身**

$("#game li").wrap("<font color='red'></font>");

**//包装所有的 li**

$("#city li").wrapAll("<font color='red'></font>");

**//包装 li 里边的文字.**

$("#language li").wrapInner("<font color='red'></font>");

* **属性操作：**
* **基本方法：**
* **attr()：获取属性**
* **html(),：获取所选元素的内容（包括标签）**
* **text()：获取所选元素的文本内容**
* **val()：获取表单字段的值**
* **height()：**
* **width()：**
* **css()：css操作**
* **示例代码：**

**//1、显示id=city，的所有子节点**

alert($("#city").html());

**//2、将id=city的所有子节点替换为。。。。**

$("#city").html("<li id='atguigu'>尚硅谷</li>");

**//3、获取id=city，对应的value属性值**

alert($("#city").val);

**//4、获取id=city，的id属性值**

alert($("#city").attr("id"));

**//5、遍历每个li元素的文本**

$("button").click(function(){

$("li")**.each**(function(index){ //index：用于计数的参数

var n = index + 1;

alert(n); //输出index数值：1、2、3……

alert($(this).text()); //输出文本数据

});

});

* **样式操作：CSS对DOM的操作**
* 先获取class；所有的样式操作，都是基于class属性的
* 样式操作的方法：
* addClass():追加样式
* removeClass():移除样式，
* toggleClass()：切换样式，控制样式的重复切换
* hasClass():判断是否含有某个样式
* **示例代码：**

**//1. hasClass(): 某元素是否有指定的样式**

alert($("div:first").**hasClass**("**SubCategoryBox**")); //true

**//2. 移除样式：选取div标签的第一个元素**

$("div**:first**").**removeClass**("**SubCategoryBox**");

**//3. 切换样式: 存在, 则去除; 没有, 则添加.**

$(".**showmore**").click(function(){

$("div:first").**toggleClass**("**SubCategoryBox**");

return false;

});

<div class="**SubCategoryBox**">

<ul>

<li ><a href="#">佳能</a><i>(30440) </i></li>

<li ><a href="#">索尼</a><i>(27220) </i></li>

<li ><a href="#">其它品牌相机</a><i>(7275) </i></li>

</ul>

<div class="**showmore**">

<a href="more.html"><span>显示全部品牌</span></a>

</div>

1. **事件处理：结合选择器使用**

* 加载DOM的两种方法：JQuery、window.onload
* JQuery的方法：可以编写多个，

**$(function(){**

**});**

* window.onload的方法：只能编写一个，必须在所有内容加载完，才能够使用

**$(document).ready(function(){**

**});**

* **方法：**
* **事件绑定**：**.bind()**
* **移除事件**：**.unbind()**
* **合成事件：**
* hover():鼠标悬停位置检测，触发函数
* toggle():循环切换，可用于鼠标点击时，切换执行函数
* mouseover():鼠标移上去时，触发函数
* mouseout():鼠标移出去时，触发函数
* mousemove():追踪鼠标的移动，可用pageX、pageY获得鼠标当前时刻的位置
* one()：仅添加一次事件——即事件只触发一次。
* **事件冒泡：**

1. **事件冒泡：一般情况下，在触发子节点事件之后，会自动向上触发父节点的事件；**
2. **阻止事件冒泡：在对应的子节点末尾添加return false；。事件会按照DOM的层次来执行，从内到外，只有遇到return false；才会停止。**

**示例代码：若点击内层的span，则会先触发span函数，再触发content函数。**

<script>

$("#content").click(function(){ **//外层的响应函数：content**

alert("div click");

});

$("span").click(function(){ **//内层的响应函数：span**

alert("span click");

//return false; **//通过在响应函数的结尾返回 false, 可以阻止冒泡.**

});

</script>

<body>

<div id="content">

外层div元素

<span>内层span元素</span>

外层div元素

</div>

</body>

* **获取鼠标位置：**

$(function(){

**//事件的 pageX, pageY 属性**

$("**body**").**mousemove**(function(**obj**){

$("#msg").text("x: " + **obj.pageX**

+ ", y: " + **obj.pageY**);

});

})

<body>

……

</body>

* **示例代码：**

**//1、bind: 为某 jQuery 对象绑定事件：click事件**

$(".head").**bind**("click", function(){

**//使用 is() 方法, 来判断某个给定的 jQuery 对象是否符合指定的选择器.**

var flag = $("**.content**")**.is(":hidden")**;

if(flag){

$(".content").**show**();//show() 方法: 使隐藏的变为显示

}else{

$(".content").**hide**();

}

});

**//2、mouseover: 鼠标移上去，mouseout: 鼠标移出.**

$(".head").**mouseover**(function(){

$(".content").show();

}).**mouseout**(function(){

$(".content").hide();

});

**//3、合成事件: hover: 鼠标移上去执行第一个函数, 移出执行第二个函数.**

$(".head").**hover**(function(){

$(".content").show();

}, function(){

$(".content").hide();

});

**//4、合成事件: toggle: 第一次点击执行第一个函数, 第二次点击执行第二个**

//函数 ... 循环执行.

$(".head").**toggle**(function(){

$(".content").show();

}, function(){

$(".content").hide();

});

<body>

<div id="panel">

<h5 class="**head**">什么是jQuery?</h5>

<div class="**content**">

jQuery是继Prototype。。。

</div>

</div>

</body>