Write less, do more, I like jQuery.



jQuery是最常用的js库，整体来说非常轻量并易于扩展，对于移动应用可以使用其更轻量的孪生兄弟Zepto代替。其是由John Resig于2006创建的开源项目，极大的简化了javascript开发人员遍历HTML文档、操作DOM、处理事件、开发Ajax等操作，最有特色的形式是$(document).ready(function() {});其省略形式为：$(function() {});。

在使用jQuery时，需要注意jQuery对象和DOM对象的区别，通常对于jQuery对象，会在变量前加$，如var $variable=jQuery对象;。可以通过如下方式对两者进行转化。

jQuery对象转化为DOM对象: var $cr = $('#cr');var cr = $cr[0];

DOM对象转化为jQuery对象: var cr = document.getElementById('cr');var $cr = $(cr);

此外，在使用多个不同的javascript库时，有可能会出现库之间的冲突，可以通过如下方式解决。

|  |
| --- |
| jQuery.noConflict();  jQuery(function() {  jQuery('p').click(function() {  alert(jQuery(this).text());  });  });  //推荐下面的方式  var $j = jQuery.noConflict();  $j(function() {}); |

在介绍jQuery选择器之前，首先引入**CSS选择器**，如下表所示。

|  |  |
| --- | --- |
| 选择器类型 | 示例 |
| 标签选择器 | td{} |
| ID选择器 | #note{} |
| 类选择器 | div.classname{} |
| 群组选择器 | 其实就是选择器的组合 td,p,div,a{} |
| 后代选择器 | #links a{} |
| 通配选择器 | \*{} |
| 其他选择器 | 伪类选择器、子选择器、临近选择器和属性选择器 |

那么相应的jQuery也会提供对应的选择器，并增加部分自定义的选择器用于方便操作，主要包括基本选择器、层次选择器、过滤选择器、表单选择器等4类选择器。

|  |  |
| --- | --- |
| jQuery选择器 | 示例 |
| 基本选择器 | |
| #id, .class, element | 最基本的id,类,元素选择器 |
| \* | 匹配所有元素, $("\*") |
| selector1,selector2,… | 将每一个选择器匹配到元素合并后一起返回 |
| 层次选择器 | |
| $("ancestor descendant") | $('div span')选取<div>里所有的<span>元素 |
| $("parent>child") | $('div>span')选取<div>元素下的<span>子元素 |
| $("prev+next") | $('.one+div')选取class为one的下一个<div>元素 |
| $("prev~siblings") | $('#two~div')选取id为two的元素后面所有<div>兄弟元素 |
| 过滤选择器 | |
| :first, :last | 选择第一个/最后一个元素 |
| :not(selector) | 去除所有与给定选择器匹配的元素 |
| :even, :odd | 选取索引为偶数/奇数所有的元素，初始索引为0开始 |
| :eq(index),gt(index)  ,:lt(index) | 选取索引等于/大于/小于index的元素 |
| :header | 选取所有标题元素 |
| :animated | 选取当前正在执行动画的元素 |
| :contains(text) | 选取含有文本内容的为text的元素 |
| :empty | 选取不包含子元素或者文本的空元素 |
| :has(selector) | 选取含有选择器所匹配元素的元素 |
| :parent | 选取含有子元素或者文本的元素 |
| 表单选择器 | |
| :input | 获取所有<input>,<textarea>,<select>和<button>元素 |
| :text,:password,:hidden | 获取所有单行文本框/密码框/不可见元素 |
| :radio, :checkbox | 获取所有单选框/多选框 |
| :submit,:reset,:button | 选取单选按钮、多选按钮、任意按钮 |
| :image, :file | 选取所有图像按钮，所有上传域 |

Tip：html元素的id包含#，（，]等特殊字符时，需要通过//进行转义，例如：<div id='id[1]'></div>， $('#id**\\**[1\\]') //转义特殊字符

**DOM**(Document Object Modal)文档对象模型是一种与浏览器、平台、语言无关的接口，通过该接口可以轻松的访问浏览器中所有标准组件。一般来说，DOM操作可以分为3类：DOM Core，包含getElementById(),getAttribute()等方法；HTML\_DOM，例如document.forms,element.src;CSS\_DOM,例如element.style.color，对于jQuery来说，将会用更简洁的方式实现这部分功能，如下表所示。

|  |  |
| --- | --- |
| DOM操作 | 示例 |
| 查找结点 | 获取元素结点：var $li = $('ul li:eq(1)');  获取属性结点：var p\_txt = $li.attr('name'); |
| 创建结点 | jQuery工厂方法:$(html语句);  var $li\_1 = $('<li></li>');$('ul').append('$li\_1');  文本结点和属性结点于此类似 |
| 插入结点 | 插入结点的方式很多：  append(), appendTo(), prepend(),prependTo(),添加到**内部**  after(), insertAfter(),before(),insertBefore() |
| 删除结点 | 删除节点：$('ul li:eq(1)').remove();  清空结点，即清空结点中的内容(子节点)  $('ul li:eq(1)').empty(); |
| 复制结点 | $('#xionger').click(function(){  $(this).clone().appendTo('ul');  }); |
| 替换结点 | $('p').replaceWith('<strong>fruit</strong>');  .replaceALl与其操作方向相反 |
| 包裹结点 | 包裹结点:$('strong').wrap('<b></b>');  包裹结点内容：$('strong').wrapInner('<b></b>'); |
| 属性操作 | 获取属性值：var p\_txt = $('p').attr('title');  设置属性值：$('p').attr('title', 'title xionger');  删除属性：$('p').removeAttr('title'); |
| 样式操作 | 获取样式：var p\_style = $('p').attr('class');  设置样式：$('p').attr('class', 'handsome');  追加样式：$('p').addClass('another'); |
| 设置和获取HTML、文本和值 | .html(), .text(), .val()  $('#address').focus(function() {  var txt\_value = $(this).val();  if (txt\_value == '请输入邮箱地址') {  $(this).val('');  }  }); |
| 遍历结点 | .children(),获取匹配元素的子元素集合  .next(), prev(), siblings()获取前一个/后一个/所有兄弟  .closest(),获取最近的匹配元素  $(document).bind('click', function(e) {  $(e.target).closest('li').css('color', 'red');  }); |
| CSS-DOM操作 | var color\_txt = $('p').css('color');  $('p').css({'fontSize':'30px', 'backgroundColor':'#888888'}); |

在jQuery中，对Ajax方法进行了封装，如下表所示。

|  |  |
| --- | --- |
| Ajax方法 | 示例 |
| .load() | 参数：url[,data][,callback]  $('#resText').load('test.html'); |
| $.get()/$.post() | 参数包括url,data,callback，以及返回内容的格式的type,包括xml,html,script,\_default  $.post('/tasks', $(#form01).serialize(), handler, 'json'); |
| $.getScript()  $.getJSON() | 动态加载js， $.getScript('/public/js/test.js');  $.getJSON('/public/resources/test.json');  JSONP形式加载其他网站JSON数据示例：  $('#send').click(function() {  $.getJSON('http://api.flickr.com/services/feeds/photos\_public.gne?tags=car&tagmode=any&format=json&jsoncallback=?', function(data) {  $.each(data.items, function(i, item) {  $("<img class='para'/>").attr('src', item.media.m).appendTo('#resText');  if (i == 3) {  return false;  }  });  });  }); |
| $.ajax(); | 参数包括url,type, timeout（ms）,data,  datatype(xml,html,sscript,json.jsonp,text)  beforeSend, complete, success, error, global |
| 序列化元素 | $(#form01).serialize()序列化为字符串,其等价于:  'username' + encodeURIComponent($('#content').val())+'&'+ 'xxx';  .serializeArray()序列化为JSON数据  $.param(obj)将普通对象序列化 |
| 全局事件 | $('#loading').ajaxStart(function(){});  .ajaxStop(), .ajaxComplete(),.ajaxSend(),.ajaxSuccess() |

之前就介绍的$(document).ready()相信大家都不会陌生，其与window.onload实际上有一些区别的，前者是当DOM加载完成后触发，而后者则是页面中的所有元素(包括所有关联元素)加载完成后执行。在jQuery中，通常事件绑定是通过bind(type[,data],fn)方法实现的，其第一个参数为事件类型，非常丰富，包括blur,focus,load,resize,unload,click,dblclick,mousedown/up/move/over/out/enter/leave,change,select,submit,keypress/down/up等。

还可以通过.unbind解绑事件，.trigger(eventType)来模拟触发事件（包含默认行为），.triggerHandler()触发事件且不执行默认行为。此外，jQuery提供了合成事件，比如.hover(enter,leave),toggle(fn1,fn2)。

对于jQuery中的事件来说，其均使用事件冒泡机制，而不支持时间捕获机制(可以用原生js)。事件冒泡就是当页面上有个元素时，其中一个嵌套在另一个中，如果均绑定了click事件，那么触发内层元素的click事件时，会同时触发外部的click事件。这样的机制在有时是需要的，但有时却不是期望的，那么就绪要通过event.stopPropagation()方法来禁止事件的冒泡了，

此外，在jQuery中，可以通过event.preventDefault()方法来阻止元素的默认行为，常用与a链接元素和submit表单提交行为。需要注意的，jQuery实际上对event进行了封装，以屏蔽不同浏览器的差异，event.target用于获取触发事件的元素，.relatedTarget获取相关元素(mouseover,mouseout)，.pageX/pageY获取页面坐标，.which获取鼠标左中右键(值为1,2,3),.metaKey获取Ctrl键，.originalEvent获取原始事件对象。

接下来，将介绍jQuery库非常吸引人的动画操作，如下表所示。

|  |  |
| --- | --- |
| 方法 | 诠释 |
| .show(),.hide(),.toggle() | 显示和隐藏元素，$('element').show(1000); |
| .fadeIn(),fadeOut()  .fadeTo() | 改变元素的不透明度 |
| .slideUp(), .slideDown()  .slideToggle() | 通过修改元素高度达到元素上下移动的效果 |
| .animate() | 自定义动画方法,  $('element').animate({left:'500px'},3000);  .stop(param),停止动画,param=true，可以清空动画队列 |

jQuery插件非常多，但最常用的要数验证插件了，最后通过一个示例来了解validation插件。

|  |
| --- |
| $('#commentForm').validate({  rules: {  username: { required: true, minlength: 2 },  email: { require: true, email: true },  url: 'url',  comment: 'required'  };  }); |

参考资料：

1. 单东林. 锋利的jQuery（第2版）[M]. 北京:人民邮电出版社, 2012.