
第 04 天 jquery

【教学内容】

Jquery 简述

- ◆ Jquery 介绍
- ◆ Jquery 快速入门
- ◆ Jquery 核心函数
- ◆ Jquery 函数别名
- ◆ Jquery 的对象

Jquery 选择器

- ◆ 基本选择器
- ◆ 层次选择器
- ◆ 过滤器

Jquery Dom 操作

- ◆ 筛选节点 API
- ◆ 文档对象属性操作
- ◆ 文档对象 css 样式操作
- ◆ 设置或获取 html 文本和值
- ◆ 文档 DOM 对象 CRUD

Jquery 事件操作

【教学总结】

【第一阶段】

【学习目标】

了解什么是 jquery,掌握其基本使用

【内容:jquery 简述】

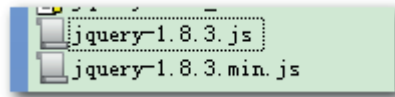
Jquery 介绍

jQuery 是一个兼容多浏览器的 javascript 库，核心理念是 write less,do more(写得更少,做得更多)。jQuery 在 2006 年 1 月由美国人 John Resig 在纽约的 barcamp 发布。jQuery 是免费、开源的，使用 MIT 许可协议。jQuery 的语法设计可以使开发者更加便捷，例如操作文档对象、选择 DOM 元素、制作动画效果、事件处理、使用 Ajax 以及其他功能。jQuery 还有一个比较大的优势是，它的文档说明很全，而且各种应用也讲得很详细，同时还有许多成熟的插件可供选择。jQuery 能够使用户的 html 页面保持代码和 html 内容分离，也就是说，不用再在 html 里面插入一堆 js 来调用命令了，只需要定义 id 即可。

官网下载 jquery-版本.js

最新版 jquery 2.x ， 企业主流 jquery1.x (1.4、 1.6)

课程以 jquery1.8 学习 （主要学习 通用 API ）



jquery-1.8.3.js 原版源码

jquery-1.8.3.min.js 压缩版的 js 互联网项目减少 流量

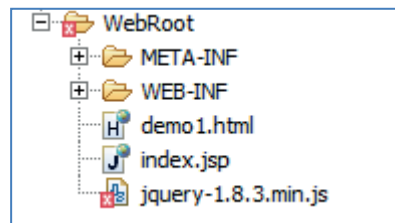
学习 jquery 主要参考 官方 API 和课外参考书

API 分类

- HTML 元素选取
- HTML 元素操作
- CSS 操作
- HTML 事件函数
- JavaScript 特效和动画
- HTML DOM 遍历和修改
- AJAX
- Utilities

Jquery 快速入门

第一步： 新建 web 项目，引入 jquery.js



第二步： 编写 jsp 页面

引入 jquery

```
<script type="text/javascript" src="jquery-1.8.3.min.js"></script>
```

使用传统 js 编写 onload 函数

```
//1. 传统用法  
window.onload=function(){  
    alert("传统用法1");  
};  
window.onload=function(){  
    alert("传统用法2");  
};*/
```

Jquery onload

```
//使用jquery完成操作，可以加载多次
jQuery(document).ready(function(){
    alert("jquery load1");
});
jQuery(document).ready(function(){
    alert("jquery load2");
});

//简化操作1
jQuery(function(){
    alert("jquery load3");
});

//简化操作2
$(function){
    alert("jquery load4");
});
```

jQuery(document) 这里 jQuery 是一个函数，是 jquery 框架核心函数，将 document 对象转换为 jquery 对象，转换 jquery 对象后，才能使用 jquery 对象的方法 ready

```
$(function){

    //1. 抓取页面元素
    //alert($("#div").html());

    //2. 转换成jquery对象
    var arr=[1,2,3];
    $(arr).each(function){
        //alert(this);
    });

    alert($("#<p>hello jquery</p>").html());

});
```

Jquery 核心函数

- `jQuery(callback)` // 页面onload 函数
- `jQuery(expression, [context])` // 查找指定对象
- `jQuery(elements)` // 将dom对象转换为jQuery对象
- `jQuery(html, [ownerDocument])` // 将html转换jQuery对象

用法一： 传入 callback 函数，页面加载后执行

```
// jQuery(callback) 在页面加载后执行
jQuery(function() {
    alert("页面加载后执行...");
});
```

用法二： 传入 expression 表达式

通过 jquery 提供九种选择器，编写不同表达式，用来选中页面中元素

```
<div>传智播客</div>
```

```
// 用法二： jQuery(expression) 选中页面的元素
alert(jQuery("div").html());
```

用法三： 传入 elements ， 将元素转换为 jquery 对象

```
// 用法三： jQuery(elements) 转换为jquery对象
var array = [10,20,30];
jQuery(array).each(function() {
    alert(this);
});
```

用法四： 传入 HTML 片段 ， 转换 jquery 对象

```
// 用法四： jQuery(html) 将HTML片段转换为 jquery对象
var p = jQuery("<p>今天下雨</p>");
alert(p.html());
```

Jquery 函数别名

简化 jQuery 函数编写，`$=jQuery`，使用`$` 取代`jQuery`
onload 函数最简单写法
`$(function){...};`

Jquery 的对象

jQuery 对象是 jQuery 独有的，其可以使用 jQuery 里的方法

```
// 修改div中内容，传统DOM写法
document.getElementById("mydiv").innerHTML = "传统DOM操作";

// 使用 jquery操作
$("#mydiv").html("jquery 操作");
```

`document.getElementById(id)` ---- 返回 DOM 对象

`$("#id")` ----- 返回 jQuery 对象

- **jQuery对象无法使用DOM对象属性方法**
- **DOM对象也无法使用jQuery对象属性方法**

JQuery 对象 可以和 DOM 对象相互转换 ！

可以将 jquery 对象，理解为是一个 DOM 对象数组，这个数组的第一个元素就是 DOM 对象 ！

- DOM 对象 ---> jQuery 对象 `$(dom 对象)`
- JQuery 对象 ---> DOM 对象 `jquery 对象[0]` 或者 `jquery 对象.get(0)`

案例：

```
// DOM对象转换为jquery对象
$(document.getElementById("mydiv")).html("DOM转换jQuery");

// jquery对象转换为DOM对象，对div操作
$("#mydiv")[0].innerHTML = "jQuery转换DOM";
```

【笔试面试题】

描述 jquery 是什么，它的作用

Jquery 中的\$有什么作用

Jquery 中怎样在 dom 与 jquery 对象之间转换

【重点总结】

掌握 jquery 的基本使用，掌握 dom 对象与 jquery 对象之间的转换。

【第二阶段】

【学习目标】

掌握 jquery 的选择器。

【内容:jquery 选择器】

jQuery，顾名思义，也就是 JavaScript 和查询（Query），一切操作都从选择器开始，操作之前必须选中。

选择器语法：\$(expression 表达式)

- jQuery内部提供了9种选择器
 - 基本选择器
 - 层级选择器
 - 基本过滤选择器
 - 内容过滤选择器
 - 可见性过滤选择器
 - 属性过滤选择器
 - 子元素过滤选择器
 - 表单过滤选择器
 - 表单对象属性过滤选择器

如果 CSS3 选择器有基础，jQuery 选择器 非常好学!!

基本选择器

- **#id** 根据id属性查找一个元素 例如: \$("mydiv")
- **.class** 根据class属性查找元素 例如: \$(".myClass");
- **element** 根据元素标签名称查找所有元素 例如: \$("div");
- ***** 匹配所有元素 例如 \$("*");
- **selector1,selector2,selectorN** 将每一个选择器匹配到的元素合并后一起返回 例如: \$("div, p);

\$("div") 选中页面所有 div 元素

\$("#itcast") 选中 id 为 itcast 的元素

\$(".dellink") 选中所有 class 属性为.dellink 元素

\$("p,div") 选中页面中所有 p 元素和 div 元素

代码:

```
<div>DIVAAAA</div>
<div class="itcast">DIVBBBB</div>
<div>DIVCCCC</div>
<div>DIVDDDD</div>
<div class="itcast">DIVEEEE</div>
<div id="foo">DIVFFFF</div>
<p>PAAAA</p>
<p class="itcast">PBBBB</p>
<p>PCCCC</p>
```

- 练习:
- 通过each() 在每个div元素前 加入 "传智播客"
- 通过size() / length 打印页面中 class属性为 itcast 的元素数量
- 通过index() 打印 id属性为foo 的div标签 是页面内的第几个div标签

each 函数和 html 函数的使用

```
$("#div").each(function(){
    // 每个div都会执行这个函数
    // this 函数中的this关键字都指向一个不同的DOM元素
    $(this).html("传智播客" + $(this).html());
});
```

size 函数 和 length 属性

```
alert($(".itcast").length);
alert($(".itcast").size());
```

index 函数

```
// index() 搜索匹配的元素，并返回相应元素的索引值，从0开始计数。
alert($("#div").index($("#div#foo")));
```

层次选择器

层次选择器是根据 DOM 元素的层级关系，通过后代元素、子元素、兄弟元素进行选择

两个元素位置关系

- 祖先
- 后代
- 父元素

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/477161056123006144>