

## 第1章 jQuery 入门

学习要点：

- 1.什么是 jQuery
- 2.学习 jQuery 的条件
- 3.jQuery 的版本
- 4.jQuery 的功能和优势
- 5.其他 JavaScript 库
- 6.是否兼容低版本 IE
- 7.下载及运行 jQuery

主讲教师：李炎恢

官方网站：<http://www.ycku.com>

合作网站：<http://www.ibeifeng.com>

### 一. 什么是 jQuery

jQuery是一个JavaScript库，它通过封装原生的JavaScript函数得到一整套定义好的方法。它的作者是John Resig，于2006年创建的一个开源项目，随着越来越多开发者的加入，jQuery已经集成了JavaScript、CSS、DOM和Ajax于一体的强大功能。它可以用最少的代码，完成更多复杂而困难的功能，从而得到了开发者的青睐。



主旨：以更少的代码、实现更多的功能

### 二. 学习jQuery的条件

jQuery是JavaScript库，所以jQuery在使用上要比原生的JavaScript要简单，但是对于网页编程来说，有些通用的基础知识是必备的：

1.XHTML或HTML5（含CSS，网页必备的基础技术，XHTML已完结90课）；

2.JavaScript（虽然jQuery使用比JavaScript简单，但各种语法来自JavaScript，只要掌握已出课程的第一季149课即可，并不需要完全精通，只要理解语法和项目简单的用法即可。）；

3.服务器语言如：PHP（jQuery属于前端技术，那和后端技术是相辅相成、互相呼应的，而PHP课程目前出到第四季，而jQuery只需要PHP第一季136课的基础即可。）；

### 三. jQuery 的版本

从2005年8月开始，进入公共开发阶段，随之而来的新框架于2006年1月14日正式以jQuery的名称发布。

2006 年 8 月发布了 jQuery1.0，第一个稳定版本，具有对 CSS 选择符、事件处理和 Ajax 交互的支持。

2007 年 1 月发布了 jQuery1.1，极大的简化 API。合并了许多较少使用的方法。

2007 年 7 月发布了 jQuery1.1.3，优化了 jQuery 选择符引擎执行的速度。

2007 年 9 月发布了 jQuery1.2，去掉了 XPath 选择器，新增了命名空间事件。

2008 年 5 月发布了 jQuery1.2.6，引入了 Dimensions 插件到核心库中。

2009 年 1 月发布了 jQuery1.3，使用了全新的选择符引擎 Sizzle，性能进一步提升。

2010 年 1 月发布了 jQuery1.4，进行了一次大规模更新，提供了 DOM 操作，增加了很多新的方法或是增强了原有的方法。

2010 年 2 月发布了 jQuery1.4.2，添加了 .delegate() 和 .undelegate() 两个新方法，提升了灵活性和浏览器一致性，对事件系统进行了升级。

2011 年 1 月发布了 jQuery1.5，重写了 AJAX 组件，增强了扩展性和性能。

2011 年 5 月发布了 jQuery1.6，重写了 Attribute 组件，引入了新对象和方法。

2011 年 11 月发布了 jQuery1.7，引入了 .on() 和 .off() 简介的 API 解决事件绑定及委托容易混淆的问题。

2012 年 3 月发布了 jQuery1.7.2，进行一些优化和升级。

2012 年 7 月发布了 jQuery1.8，8 月发布了 1.8.1，9 月发布了 1.8.2，重写了选择符引擎，修复了一些问题。

2013 年 1 月发布了 jQuery1.9，CSS 的多属性设置，增强了 CSS3。

2013 年 5 月发布了 jQuery1.10，增加了一些功能。

**2013 年 4 月发布了 jQuery2.0，5 月发布了 jQuery2.0.2，一个重大更新版本，不在支持 IE6/7/8，体积更小，速度更快。**

在制作这套课程的时候，我们最新的中文版的 API 文档是 1.8 版本的，有在线和离线两种手段查阅：

- 1.在线的可以访问：<http://t.mb5u.com/jquery/>
- 2.离线的 AP 文档将打包提供给大家
- 3.如果要看最新的文档，英文版的：<http://api.jquery.com/> 看不懂可以打开谷歌翻译，虽然不是特别准。

关于版本学习的问题：

版本的版本号升级主要有三种：第一种是大版本升级，比如 1.x.x 升级到 2.x.x，这种升级规模是最大的，改动的地方是最多的，周期也是最长的，jQuery 从 1.x.x 到 2.x.x 用了 7 年。第二种是小版本更新，比如 1.7 升级到 1.8，改动适中，增加或减少了一些功能，一般周期半年到一年左右。第三种是微版本更新，比如 1.8.1 升级到 1.8.2，修复一些 bug 或错误之类。

版本的内容升级主要也有三种：第一种是核心库的升级，比如优化选择符、优化 DOM 或者 AJAX 等；这种升级不影响开发者的使用。第二种是功能性的升级，比如剔除一些过时的方法、新增或增强一些方法等等；这种升级需要了解和学习。第三种就是 BUG 修复之类的升级，对开发者使用没有影响。

学习者有一种担忧，比如学了 1.3 版本的 jQuery，那么以后升级新版本是不是还需要重学？没必要，因为并不是每次升级一个版本都会增加或剔除功能的，一半左右都是内部优化，升级到新版本并不需要任何学习成本。就算在新的版本增加了一些功能，只需要几分钟了解

一下即可使用，无需清零之前的知识，只需后续累加。当然，在早期的 jQuery 版本都创建了最常用的功能，而新版本中增加的功能，也不是最常用的，无需立即学习，立马用起。

#### 四. jQuery 的功能和优势

jQuery 作为 JavaScript 封装的库，他的目的就是简化开发者使用 JavaScript。主要功能有以下几点：

1. 像 CSS 那样访问和操作 DOM
2. 修改 CSS 控制页面外观
3. 简化 JavaScript 代码操作
4. 事件处理更加容易
5. 各种动画效果使用方便
6. 让 Ajax 技术更加完美
7. 基于 jQuery 大量插件
8. 自行扩展功能插件

jQuery 最大的优势，就是特别的方便。比如模仿 CSS 获取 DOM，比原生的 JavaScript 要方便太多。并且在多个 CSS 设置上的集中处理非常舒服，而最常用的 CSS 功能又封装到单独的方法，感觉非常有心。最重要的是 jQuery 的代码兼容性非常好，你不需要总是头疼着考虑不同浏览器的兼容问题。

#### 五. 其他 JavaScript 库

目前除了 jQuery，还有 5 个库较为流行，他们分别是 YUI、Prototype、Mootools、Dojo 和 ExtJS。

YUI，是雅虎公司开发的一套完备的、扩展性良好的富交互网页工具集。

Prototype，是最早成型的 JavaScript 库之一，对 JavaScript 内置对象做了大量的扩展。

Dojo，Dojo 强大之处在于提供了其他库没有的功能。离线存储、图标组件等等。

Mootools，轻量、简洁、模块化和面向对象的 JavaScript 框架。

ExtJS，简称 Ext，原本是对 YUI 的一个扩展，主要创建前端用户界面。(付费的)

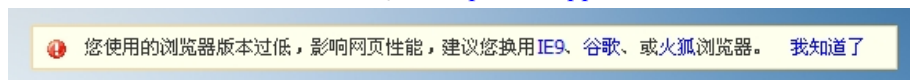
#### 六. 是否兼容 IE 低版本

这次 jQuery 发布了大版本 2.x.x，完全放弃兼容 IE6/7/8。不单单如此，很多国际上的大型站点也开始逐步不再支持 IE6/7/8。但对于国内而言，比较大型的网站最多只是抛弃 IE6，或者部分功能不支持 IE6 的警示框，还没可能一下子把 IE6/7/8 全面抛弃。这里我们就谈一谈你的项目是否有必要兼容 IE6/7/8。

完全不支持 IE6 的示例：网易云课堂 -- <http://study.163.com>



部分功能不支持 IE6 的示例：Q+ -- <http://web.qq.com>



完全不支持的做法，就是检测到是否为 IE6 或者 IE6/7/8，然后直接跳转到一个信息错误界面，让你更换或升级浏览器，否则无法访问使用。

部分功能不支持的做法，就是判断你是 IE6 或 IE6/7/8，然后给一个警示条或弹窗，告诉你使用此款浏览器性能降低或部分功能使用不正常或不能使用的提示，但还可以访问使用。

虽然大部分国内网站用 IE6 去运行都能基本兼容，但很多细节上还是有些问题，导致不能流畅的去使用。

疑问：我们新做的项目应不应该再兼容 IE6 或 IE6/7/8？

这个问题争论很久，支持兼容的人会拿国情和使用率来证明。不支持兼容的人会用技术落后导致整个落后别国来证明。其实这两种都有商榷的地方。

首先，传统行业失败率为 97%，而新新的 IT 行业更高达 99% 以上（数据可能不精确，但可以说明失败率很高）。那么站在更高的角度去看你的项目，你不管是付出 3 倍成本去完成一个用户体验一般，但兼容性很好的项目；还是付出正常成本去完成用户体验很好，但不兼容低版本浏览器；这两种情况不管是哪一种，最终可能都会失败。那么你愿意选择哪种？

是否兼容 IE6 或 IE6/7/8 并不单纯是用户基数和国情的问题，而很多项目发起人只一味的用这种理由去判定需求，那么失败也在所难免。除了这个方面，我们还应该考虑更多的方面：

#### 1. 成本控制

很多项目往往在 6、12、18、32... 个月就会发生财务问题，比如资金紧缩甚至断裂。所以，成本控制尤为重要。项目如果不是老站升级，也不是大门户的新闻站，成本控制和尽快上线测试才是最重要的。而如果新站一味要求全面兼容，会导致成本加剧（随着功能多少，成本倍率增加）。为了锁紧时间，就不停的加班再加班，又导致员工抵触，工作效率降低，人员流动开始频繁，新员工又要接手开发一半的项目。这样成本不停的再累加。最终不少项目，根本没上线就失败了。

#### 2. 用户选择

用户一般可以分为两种用户：高质量用户和低质量用户。所谓高质量用户，就是为了一款最新的 3D 游戏去升级一块发烧级的显卡，或直接换一台整机。所谓低质量用户，发现不能玩最新的 3D 游戏，就放弃了，去玩“植物大战僵尸”解解馋算了。在用户选择上有一个很好的案例，就是移动互联网。网易和腾讯在他们的新闻应用上，他们兼容了几乎所有的手机平台，比如 IOS、安卓、黑莓、塞班等等，因为新闻应用的核心在新闻，而新闻的用户基数巨大，需要兼顾高质量和低质量用户。而腾讯在 IOS 上的几十个应用，除了新闻、QQ、浏览器，其他的基本都只有 IOS 和安卓，在塞班和黑莓及其他上就没有了。

所以，你的应用核心是哪方面？兼容的成本有多大？会不会导致成本控制问题？用户选择尤为重要，放弃低质量用户也是一种成本控制。在用户基数庞大的项目上，放弃低质量用户就有点愚笨，比如某个新闻站有 1 亿用户，2000 万为使用低版本浏览器的低质量的用户，而面对 2000 万用户，你兼容它或单独为 2000 万做个低版本服务，成本虽然可能还是 3 倍，但对于庞大的用户基数来看，这种成本又非常低廉。而你的用户基数只有 1000 人，而低质量用户有 50 人，那么为了这 50 人去做兼容，那么 3 倍的成本就变得非常的昂贵。

#### 3. 项目侧重点

你的项目重点在哪里？是为了看新闻？是为了宣传线下产品？那么你其实有必要兼容低版本浏览器。首先这种类型的站不需要太好的用户体验，不需要太多的交互操作，只是看，而兼容的成本比较低，并且核心在新闻或产品！但如果你的项目有大量的交互、大量的操作，



比如全球最大的社交网已经不兼容 IE6/7，最大的微博也不再兼容 IE6/7，就是这个原因。所以，项目并不是一味的全面兼容，或者全面不兼容，主要看你的项目侧重点在哪里！

#### 4. 用户体验

如果你的项目在兼容低版本浏览器成本巨大，比如社交网，有大量的 JS 和 AJAX 操作。那么兼容 IE6/7 的成本确实很高，如果兼容，用户体验就会很差。兼容有两种，一种是高版本浏览器用性能好，体验好的模式；低版本的自动切换到兼容模式。第二种就是，不管高版本或低版本都用统一的兼容模式。这两种成本都很高。用户体验好的模式，能增加用户粘度，增加付费潜在用户，而用户体验差的总是被用户归纳为心目中的备胎（所谓备胎就是实在没有了才去访问，如果有，很容易被抛弃）。

#### 5. 数据支持

如果对某一种类型的网站项目有一定的研究，那么手头必须有支持的数据分析。有数据分析可以更好的进行成本控制，更有魄力的解决高低质量用户的取舍。

浏览器	浏览次数   百分比	独立访客	IP	人均浏览次数	平均停留时间(秒)	历史
总计	13839   100%	3380	2740	4.09	58.53	-
Chrome	3414   24.67%	822	656	4.15	55.97	历史
360安全浏览器	2906   21.00%	738	634	3.94	58.66	历史
Firefox	2584   18.67%	604	442	4.28	62.12	历史
MSIE 8.0	1680   12.14%	376	284	4.47	56.28	历史
sogou	794   5.74%	223	208	3.56	59.29	历史
MSIE 7.0	504   3.64%	97	88	5.20	67.93	历史
MSIE 9.0	483   3.49%	162	124	2.98	51.76	历史
MSIE 6.0	392   2.83%	98	77	4.00	52.73	历史
Maxthon	297   2.15%	81	72	3.67	55.15	历史
Safari	291   2.10%	81	67	3.59	51.73	历史
TheWorld	240   1.73%	38	31	6.32	71.22	历史
QQ浏览器	175   1.26%	36	32	4.86	62.22	历史
TencentTraveler	40   0.29%	13	13	3.08	104.08	历史
Opera	34   0.25%	10	9	3.40	68.41	历史
SaaYaa	5   0.04%	1	3	5.00	109.20	历史

http://t.qq.com/yc60.com

瓢城 Web 俱乐部浏览器比例图

#### 6. 教育用户

很多项目可能是有固定客户群，或者使用该项目人员质量普遍较高。那么，面对零星一点的低质量用户，我们不能再去迎合他。因为迎合他，就无法用高质量的用户体验去粘住忠实用户，又不能获取到低质量用户的芳心。所以，我们应有的策略是：牢牢把握住高质量的忠诚用户，做到他们心目中的第一；教育那部分低质量用户（比如企业级开发项目，可以直接做企业培训，安装高版本浏览器等等。互联网项目，就给出提示安装高版本浏览器即可）。那么一部分低质量用户被拉拢过来，还有一小撮死性不改的就只有放弃。切不可捡了芝麻丢了西瓜，不要贪大求全。

PS：我们用了这六点讨论了一下是否需要兼容 IE6 或 IE6/7/8，结论就是必须根据实际情况，你项目的成本情况、人员情况、用户情况和项目本身类型情况而制定，没有一刀切的兼容或不兼容。

### 七. 下载及运行 jQuery

目前最新的版本，是 1.10.1 和 2.0.2，我们下载开发版，可以顺便读一读源代码。如果你需要引用到你线上的项目，就必须使用压缩版，去掉了注释和空白，是容量最小。

本课程使用的软件是：Nodepad++

使用测试的浏览器为：Firefox3.6.8、Firefox21+、Chrome、IE6/7/8/9、Opera 和 Safari

使用的版本为：1.10.1、2.02 作为抛弃 IE6/7/8 版本会在单独章节中讲解

使用的 html 版本为：xhtml1.0，在必要的时候将会使用 html5

使用的调试工具：Firefox 下的 firebug

//单击按钮弹窗

```
$(function () {  
    $('input').click(function () {  
        alert('第一个 jQuery 程序! ');  
    });  
});
```

# 感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城 Web 俱乐部(ycku.com)联合提供：

本次主讲老师：李炎恢

谢谢大家，再见！