

JavaScript+jQuery前端开发基础教程

微课版

第7章 jQuery简介

本章主要内容:

- 了解jQuery
- jQuery资源
- 使用jQuery

7.1 了解jQuery

- jQuery是一个免费、开源的轻量级JavaScript库，其设计宗旨是“Write Less, Do More”，即提倡用更少的代码，做更多的事。
- jQuery最初由John Resig开发，于2006年1月在纽约的BarCamp会议上发布。现在，jQuery已发展为一个开源项目。
- jQuery UI以jQuery为基础，提供用于构建Web图形界面的UI组件。
- jQuery Mobile则以jQuery为基础，用移动平台专用组件对其进行了扩展，用于移动应用开发。

7.1.1 jQuery主要功能

- jQuery主要提供下列功能。
 - 选取和操作HTML元素：jQuery提供了丰富、高效的选择器，可准确选取HTML文档中的一个或多个元素，并可操作HTML元素的外观和行为。
 - 操作CSS：直接使用JavaScript操作页面中的CSS样式表会受限于浏览器的兼容性，jQuery很好地解决了浏览器兼容问题。
 - 标准化HTML事件处理：jQuery提供了丰富的页面事件处理方法，不仅解决了浏览器兼容问题，而且使事件处理更加简单。
 - 支持网页特效和动画：jQuery提供了丰富的页面特效，通过简单地调用内置的动画函数，即可实现高级动画特效。
 - 简化了HTML DOM操作：jQuery降低了JavaScript操作DOM的复杂性，只需极少的代码即可完成复杂的DOM操作。
 - 简单的AJAX操作：开发人员只需简单地调用函数即可完成AJAX请求，获得服务器端的响应，而无须考虑客户端和服务器之间的通信细节。

7.1.2 jQuery主要特点

- 1. 简洁

- jQuery库非常小，未压缩的库（3.7.1版）只有287KB。

- 2. 功能强大

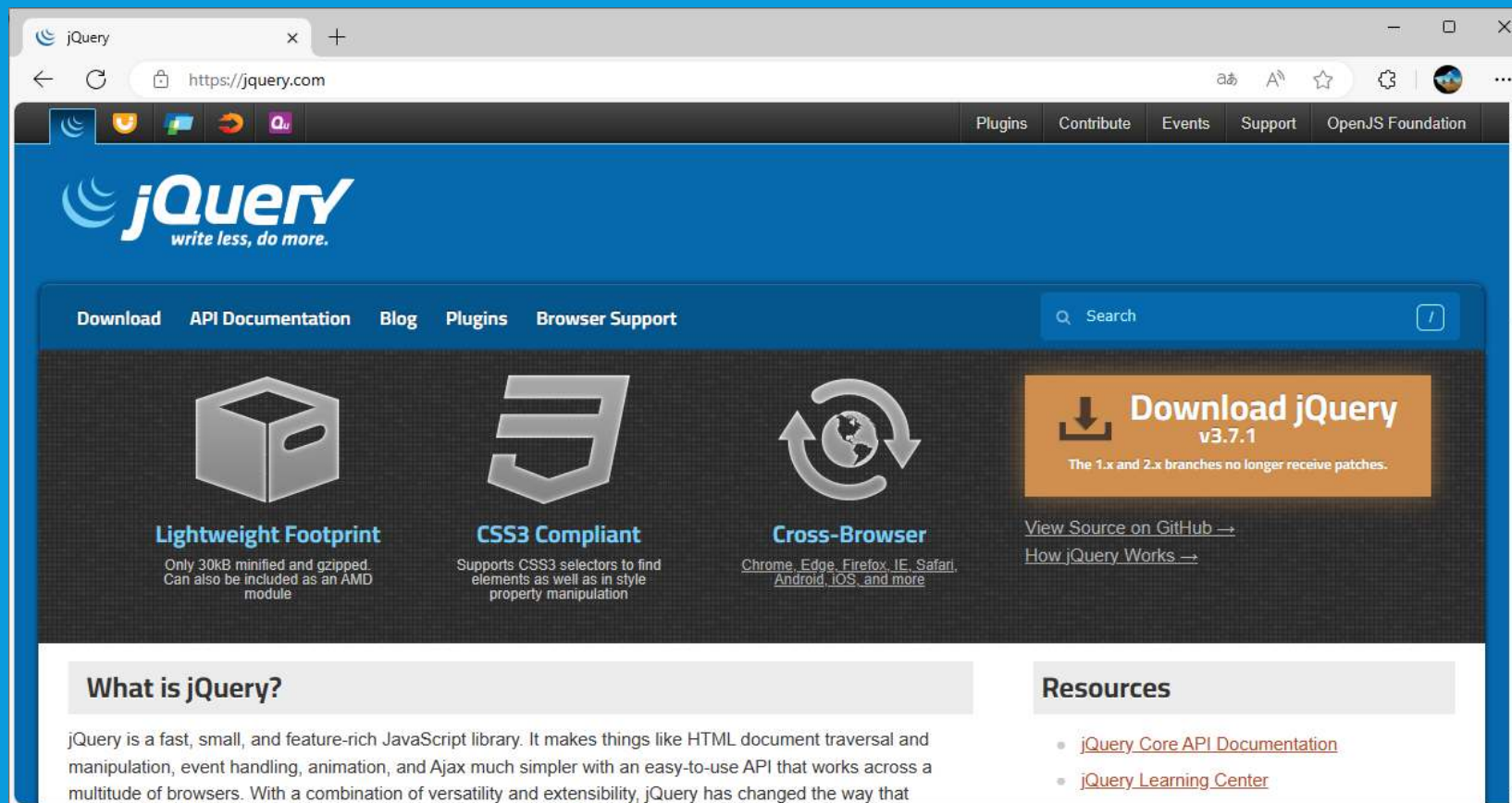
- 传统的JavaScript脚本编程方式需要开发人员具备良好的程序设计功底，并熟练掌握HTML、CSS和DOM等各种Web开发技术。
- JavaScript脚本在客户端浏览器中解释执行，在大型Web应用中调试和维护JavaScript脚本往往会耗费开发人员的大量精力。
- jQuery改变了传统的JavaScript编程方式。使用jQuery提供的函数，即可快速实现各种功能，代码更加简洁、结构清晰。

▪ 3. 兼容各种浏览器

- 本书前面的章节回避了JavaScript的浏览器兼容问题，原因是目前的各种浏览器对JavaScript的支持越来越全面。
- 而使用jQuery不需要考虑浏览器兼容问题。
- jQuery具有良好的浏览器兼容能力，支持各种主流浏览器：Chrome、Edge、Firefox、Safari和Opera等。

7.2 jQuery资源

- jQuery主页提供各种相关资源

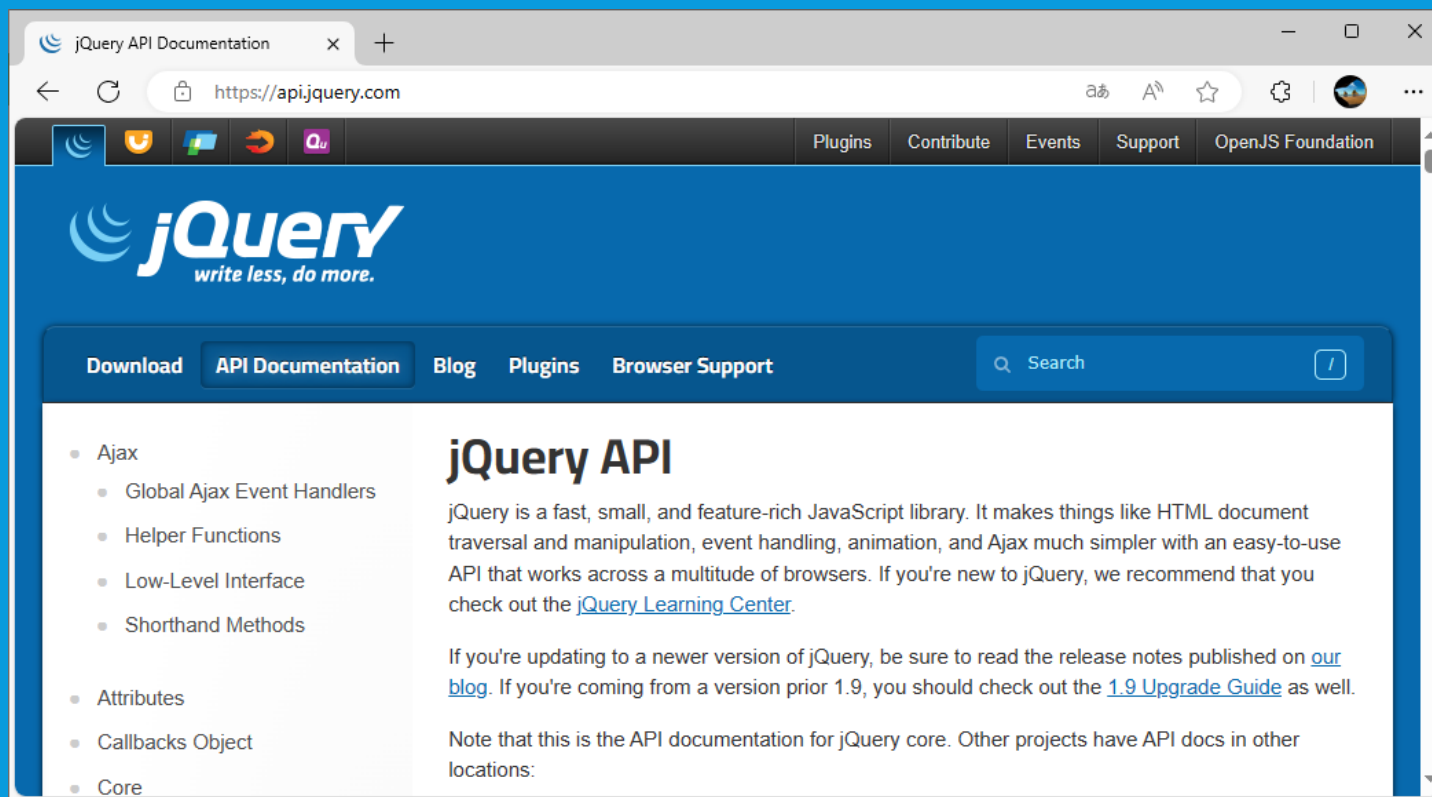


7.2.1 下载jQuery

- jQuery库有3种类型。
 - uncompressed: 未压缩版, 包含各种注释、空白和换行符等, 适用于开发阶段。
 - compressed: 压缩版, 删除了各种注释、空白和换行符等, 适用于Web应用发布。
 - slim: 瘦身版, 不包含ajax和effects模块。
- jQuery库是一个单独的js代码文件。

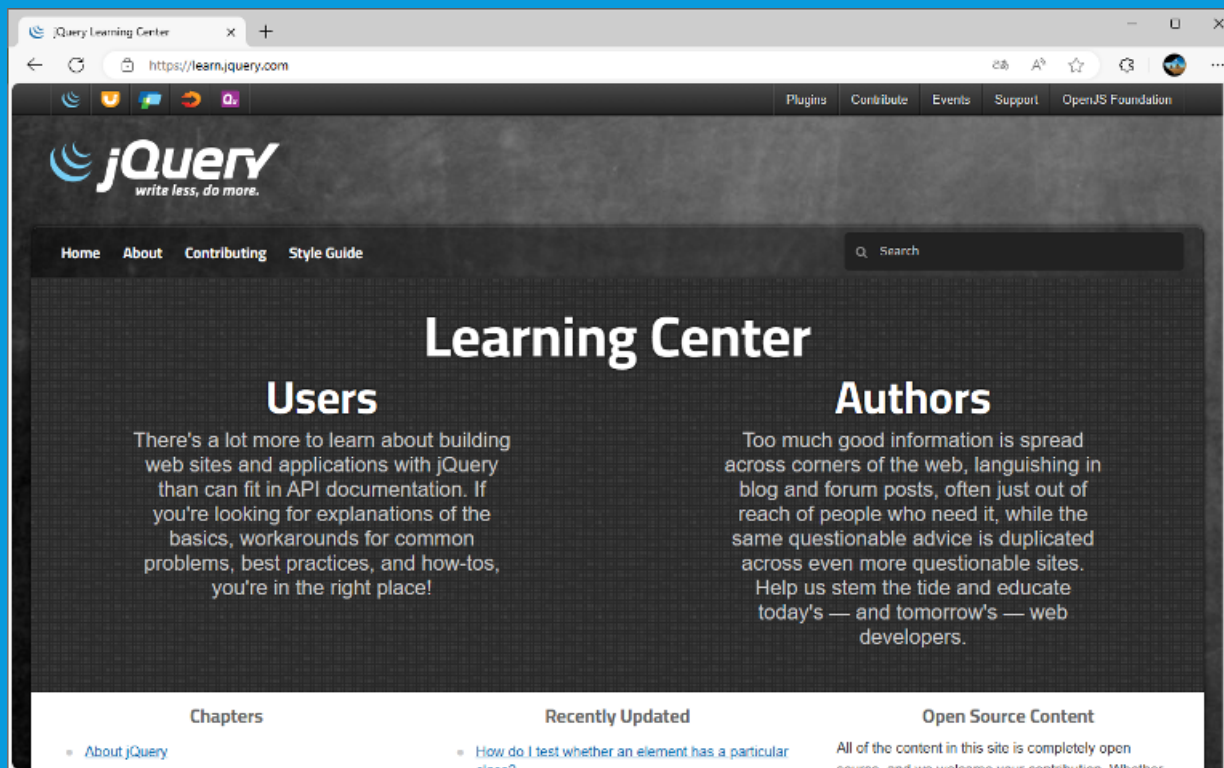
7.2.2 查看 jQuery 文档

- 在 jQuery 主页中单击“API Documentation”按钮，可进入 jQuery 文档中心



7.2.3 jQuery学习中心

- 在jQuery主页右侧的资源列表中单击“jQuery Learning Center”链接，可进入jQuery学习中心页面



7.2.4 中文学习资源

- 菜鸟教程提供了在线的中文jQuery学习资源



7.3 使用jQuery

- 要使用jQuery库，需要在HTML文档中将其引入。
- 引入后，即可在脚本中调用jQuery库提供的各种函数。

7.3.1 引入jQuery

- 在HTML文档中使用<script>标记来引入jQuery。
- 有两种引入jQuery库的方法：
 - 引用本地jQuery库
 - 引用CDN上的jQuery库。

1. 引入本地jQuery库

- 可将jQuery库下载到本地，放在和HTML文档相同的文件夹或子文件夹中，也可放在本地Web服务器中。对初学者而言，建议将jQuery库下载到本地，放在和HTML文档相同的文件夹中。然后，在HTML文档中使用下面的语句将其引入。
 - `<script src="jquery-3.7.1.min.js"></script>`
- jquery-3.7.1.min.js是下载到本地的jQuery库文件名，该名称包含了版本号，“min”表示是压缩版的库文件。

2. 引入CDN上的jQuery库

- CDN (Content Delivery Network, 内容分发网络) 是互联网中提供文本、图片、脚本、应用程序或其他资源的网络服务器。通常, CDN只提供各类资源的稳定版本。
- 两个常用CDN中的jQuery查询地址和引用地址如下。
 - jQuery CDN: <https://releases.jquery.com>。jQuery 3.7.1引用地址为<https://code.jquery.com/jquery-3.7.1.js>。
 - 微软CDN: https://docs.microsoft.com/en-us/aspnet/ajax/cdn/overview#jQuery_Releases_on_the_CDN_0。jQuery 3.7.1引用地址为<https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.7.1.js>。
- 例如, 下面的语句从jQuery CDN引入压缩版的jQuery库。
 - `<script src="https://code.jquery.com/jquery-3.7.1.js"></script>`

7.3.2 jQuery语法

- jQuery() 函数是jQuery库中最重要的一个函数，大多数jQuery脚本都是从jQuery() 函数开始的。
- \$是jQuery的别名，绝大多数开发人员喜欢使用\$() 而不是jQuery()。
- jQuery基础语法格式如下。
 - `$(selector).action()`
- 其中，selector为选择器，用于选取HTML标记，action() 为对选中HTML标记执行的操作。

- `<html>`
- `<head>`
- `<!--引入本地jQuery库-->`
- `<script src="jquery-3.7.1.min.js"></script>`
- `</head>`
- `<body>`
- `<div id="show"></div>`
- `<script>`
- `$(function () {`
- `$("#show").text("你好，jQuery! ")//修改id为show的标记`
的文本内容
- `});`
- `</script>`

jQuery代码的基本结构

- `<script>`
- `$(document).ready(function () {`
- `...//jQuery脚本`
- `})`
- `</script>`
- jQuery脚本放在`$(document).ready()`的回调函数中

ready () 函数的调用方式

- jQuery提供了多种调用ready () 函数的方式。
 - `$(回调函数)`。
 - `$(document).ready(回调函数)`。
 - `$("#document").ready(回调函数)`。
 - `$("#img").ready(回调函数)`。
 - `$().ready(回调函数)`。
- jQuery 3.x推荐使用第1种方法，其他方法仍可使用但已过时。参数“回调函数”可以是函数名称，也可以是一个匿名函数。

7.3.3 选取HTML标记

- `$()` 函数的第1个参数有多种形式：字符串形式的CSS选择器、字符串形式的HTML标记、一个或多个DOM元素或者一个函数。
- `$()` 函数返回一个jQuery对象，该对象封装了参数匹配的HTML标记或者新建的HTML标记。
- 如果有多个匹配的HTML标记，则返回对象是一个jQuery对象数组。对jQuery对象执行的操作将作用于其包含的所有标记。

【例 7-2】 使用\$()函数操作多个 HTML 标记。源文件：[07\test7-2.html](#)。

```
<html>
<head>
  <script src="jquery-3.7.1.min.js"></script>
</head>
<body>$( )函数匹配多个标记
  <p>第一个段落</p><p>第二个段落</p><p>第三个段落</p>
  <script>
    $(function () {
      $('p').css({ "text-decoration": "underline" }) //统一设置<p>标记样式
    })
  </script>
</body>
</html>
```

7.3.4 上下文

- `$()` 函数的第2个参数指定上下文——HTML标记的选择范围。
- 如果没有指定上下文，则在整个HTML文档中选择标记。

```
<body>$( )函数匹配多个标记
  <p>第一个段落</p>
  <div><p>第二个段落</p></div>
  <p>第三个段落</p>
  <script>
    $(function () {
      $('p', 'div').css({"text-decoration": "underline"}) //修改<div>中的<p>标记样
    })
  </script>
```

7.3.5 将HTML标记转换为jQuery对象

`$()` 函数可将HTML标记转换为jQuery对象。

```
<html>
<head>
  <script src="jquery-3.7.1.min.js"></script>
</head>
<body>
  <table id="t1" border="1"></table>
  <script>
    var t1 = document.getElementById('t1')
    $(t1).html('<tr><td>数据 1</td><td>数据 2</td></tr>')
  </script>
</body>
</html>
```

7.3.6 使用链接方法调用

- jQuery中的大部分方法都会返回其操作的jQuery对象，所以可使用句点符号来实现链接方法调用，使代码更简洁。

```
<div>原始数据</div>
<script>
  $(function () {
    $('div').append('<br>第 2 行').css({ color: "red" }).append('<br>').append('第 3 行')
  })
</script>
```


7.3.7 jQuery命名空间

- jQuery引入了命名空间的概念。
- jQuery脚本中的所有全局变量均属于jQuery命名空间，jQuery和\$均表示jQuery命名空间。
- 在与其他JavaScript库一起使用时，可能会出现\$标识符冲突的情况。
- jQuery提供了noConflict()方法用于避免冲突。
- noConflict()方法返回全局jQuery对象，可将其赋给一个变量，然后用该变量来代替\$标识符。

【例 7-6】 使用变量代替\$标识符。源文件：07\test7-6.html。

```
<html>
<head>
  <script src="jquery-3.7.1.min.js"></script>
</head>
<body>
  <div>原始数据</div>
  <script>
    var $j = jQuery.noConflict();
    $j(function () { alert('页面中的<script>标记个数: ' + $j('script').length) })
  </script>
</body>
</html>
```

- 另一种可行的避免冲突的方式是将所有脚本代码放在ready()方法中，并将\$作为ready()方法参数。这样，ready()方法内部的\$标识符代表jQuery，方法外的\$标识符代表其他库。

【例 7-7】 在 ready() 方法中封装\$标识符。源文件：[07\test7-7.html](#)。

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <script src="jquery-3.7.1.min.js"></script>
  <script>
    jQuery.noConflict()
    jQuery(document).ready(function ($) {
      alert('页面中的<script>标记个数: ' + $('script').length)
    })
  </script>
```

编程实践：在页面加载视频

- 本节综合应用本章所学知识，设计一个HTML文档，使用脚本在页面中加载国产大飞机C919的新闻视频，如图7-13所示。



JavaScript+jQuery前端开发基础教程

微课版

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/926144135151010231>