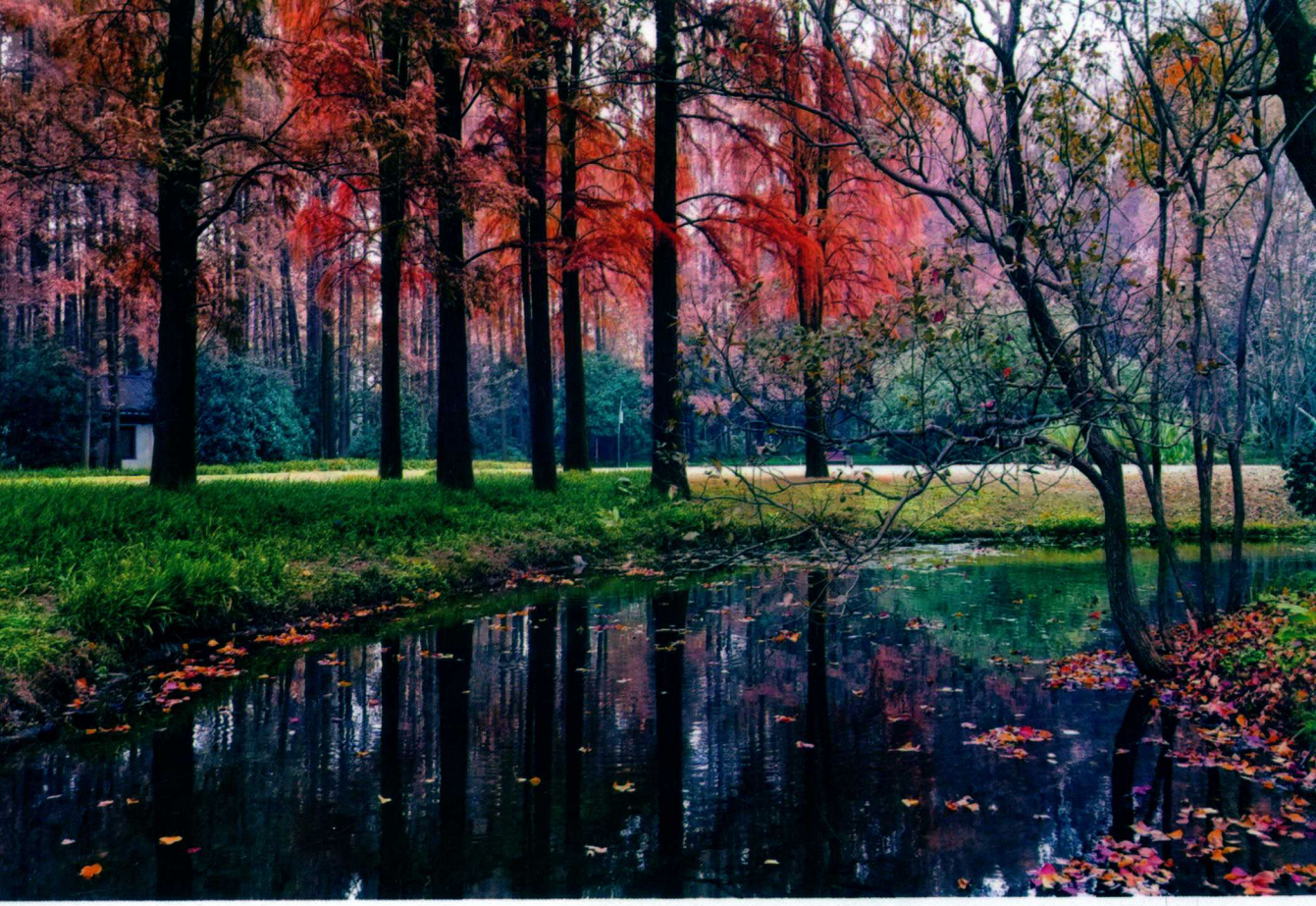

JavaScript+Vue+React 全程实例

Web 前端技术丛书

通过JavaScript实例掌握Web前端开发技术

- 涵盖目前流行的特效、流行的JS技术、流行的Vue与React框架
- 包括众多JavaScript应用场景，直接感受实际开发出来的页面效果
- 围绕实例进行讲解，每一节都可以让读者掌握一种实用技术

郑均辉　薛燚　编著

范例程序源代码

清華大學出版社

JavaScript+Vue +React 全程实例

Web 前端技术丛书

郑均辉　薛燚　编著

清华大学出版社
北京

内 容 简 介

本书基于理论知识与开发实践相结合的思想，精选当前简单、实用和流行的百余个 JavaScript 代码实例，帮助读者学习掌握 JavaScript 脚本语言。全书内容翔实、重点突出、通俗易懂，涵盖了 JavaScript 前端开发的方方面面。

全书共分为 13 章，包括 JavaScript 前端设计、调试和开发的一些必备知识，表单处理、DOM 控制、控件特效、日期时间、网页特效、DIV+CSS、Ajax 应用等方面的应用实例，还特别增加了对当下非常流行的 React 和 Vue.js 框架的介绍。本书的全部代码实例均是对 JavaScript 技术最具代表性的实践应用，可以帮助读者深入学习 JavaScript 的开发技巧。

本书是学习掌握 JavaScript 技术非常好的图书，既适合 JavaScript、Vue、React 前端初学者阅读，也适合从事前端网页设计以及需要学习前端技术的后端开发工程师阅读，同时还可作为高等院校和培训学校相关专业的教材。相信本书丰富的内容和大量的实例能够帮助初学者快速步入 Web 前端开发的捷径，并衷心地希望每一名前端爱好者都可以成为有代码实践和技术深度的 JavaScript 高手。

图书在版编目（CIP）数据

JavaScript+Vue+React 全程实例/郑均辉，薛燚编著. —北京：清华大学出版社，2019
（Web 前端技术丛书）
ISBN 978-7-302-53164-7

Ⅰ. ①J… Ⅱ. ①郑… ②薛… Ⅲ. ①JAVA 语言－程序设计 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2019）第 116023 号

责任编辑： 夏毓彦
封面设计： 王　翔
责任校对： 闫秀华
责任印制： 杨　艳

出版发行： 清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者： 清华大学印刷厂
经　　销： 全国新华书店
开　　本： 190mm×260mm　　**印　　张：** 20.5　　**字　　数：** 525 千字
版　　次： 2019 年 8 月第 1 版　　**印　　次：** 2019 年 8 月第 1 次印刷
定　　价： 69.00 元

产品编号：082162-01

前　言

读懂本书

JavaScript 无处不在

二十多年前，布兰登•艾奇（Brendan Eich）为 Netscape 浏览器草草地设计出网页脚本语言（最早的 JavaScript 原型）的时候，可能根本不会预料到如今 JavaScript 会成为 Web 开发领域的第一编程语言。

在权威的编程语言排行榜（TIOBE）中，JavaScript 多年来一直稳居在前几名之列。尽管自 JavaScript 诞生之日起，就伴随着诸如“语法不够严谨”“逻辑不够清晰”“代码管理混乱”这类的批评之声，但这并没有阻止广大程序员对其的喜爱。

JavaScript 之所以能够得到广泛的欢迎和普及，与其简单易学、使用灵活，跨平台兼容的这些特性密不可分。新手可以很快地掌握一些基本技巧并实践在网页开发中，高手可以凭借扎实的基本功、构建出逻辑复杂且功能强大的 Web 应用。

如今，JavaScript 支持在绝大多数的平台上进行开发：PC 客户端的应用程序，Web 服务器端的业务逻辑，嵌入式芯片设计，物联网设备研发等，均是 JavaScript 可以发挥魔力的舞台。毫不夸张地讲，今天的 JavaScript 几乎是无处不在。

JavaScript 技术特点

JavaScript 是一种基于对象（Object）和事件驱动（Event Driven）并具有相对安全性的脚本语言。JavaScript 广泛应用于互联网的 Web 开发，通过为 HTML 网页添加动态响应功能来提高用户交互体验。

JavaScript 的设计目标，就是一种基于原型对象、弱类型、事件驱动、跨平台兼容的解释性动态脚本语言。同时，由于 JavaScript 具有很强的扩展性，因此可以基于 JavaScript 原生语法开发出功能更为强大的类库或框架。

当然，也正是由于 JavaScript 的灵活性太强，各大浏览器厂商设计的版本兼容性不好。不过，这种情况随着 EMCA TC39 委员会推出的标准化 ECMAScript 脚本语言得到了很好的改善，相信未来 JavaScript 脚本语言的兼容性会越来越好。

JavaScript 扩展类库和框架

JavaScript 之所以无处不在、广受欢迎，相信与其庞大的扩展类库和框架群不无关系。例如，ProtoType、jQuery、jQuery Mobile、AngularJS、React、Vue.js 等，这些耳熟能详的名字都是 JavaScript 扩展类库和框架的优秀代表。

以上这些优秀的 JavaScript 扩展类库和框架不单单是将核心功能进行抽象、集成和扩展，而是在设计模式、功能架构、性能优化等方面做足了功夫，带给了设计人员无与伦比的编程体验以及代码性能和运行效率的显著提升。

本书真的适合你吗

本书大量的基础代码实例可以帮助读者快速掌握 JavaScript 的编程技巧，并应用到实践开发之中。尤其是关于 JavaScript 框架的提高内容中，通过对目前流行的 React 和 Vue.js 框架的介绍，帮助读者去了解前端 Web 技术的前沿方向。无论是基础内容或提高内容，相信读者都可以从中获益。

本书涉及的主要软件工具、技术或框架

Apache HTTP	Mozilla Firefox	Dreamweaver CS6	UltraEdit
WebStorm	Sublime Text	Notepad	EditPlus
HTML	MIME	HTTP	React
HTML5	JavaScript	ECMAScript	Vue.js
CSS3	AJAX	RegExp	JSON

本书特点

（1）本书完全是从简单、通用的 JavaScript 代码实例出发，抛开枯燥的纯理论知识介绍，通过实例讲解的方式帮助读者学习 JavaScript 脚本语言设计。

（2）本书内容涵盖 JavaScript 所涉及的、绝大部分的前端开发知识，将这些内容整合到一起可以系统地了解掌握这门语言的全貌，为介入大型 Web 项目的开发做了很好的铺垫。

（3）本书对于实例中的知识难点做出了详细的分析，能够帮助读者有针对性地提高 JavaScript 编程开发技巧。

（4）本书在知识点上按照类别进行合理的划分，全部的代码实例都是独立的，读者可以从头开始阅读，也可以从中间开始阅读，不会影响学习进度。

（5）本书代码遵循重构原理，避免代码污染，真心希望读者能写出优秀、简洁、可维护的代码。

代码下载

本书代码下载地址（注意数字与字母大小写）：https://share.weiyun.com/5K9SHun。如果下载有问题，请联系 booksaga@163.com，邮件主题为“JavaScript+Vue+React”。

本书读者

- JavaScript、Vue、React 前端开发初学者
- 从事前端网页设计的开发工程师
- 需要学习前端技术的后端开发工程师
- 高等院校和培训学校相关专业的师生

本书第 1~11 章由平顶山学院的郑均辉编写，第 12~13 章由薛燚编写。

关于封面照片

封面照片由蜂鸟网摄影家 ptwkzj 先生友情提供，在此表示衷心感谢。

编　者

2019 年 6 月

目　录

第 1 章　JavaScript 环境搭建

本书将用新颖的视角去认识 JavaScript，通过简单流行的代码实例深度阐述 JavaScript 的特性，尽量利用 IT 世界里有意思的东西来激发读者的学习兴趣。本章将概括性地介绍 JavaScript 的书写方式、调试方式和开发工具。

1.1　HTML 中书写 JavaScript 的几种方式

编写 JavaScript 代码其实无须特殊软件，一个普通的文本编辑器（如记事本）和一个 Web 浏览器就足够了。用 JavaScript 编写的代码需要放在 HTML 文档中才能被浏览器执行，有以下两种方式可以做到这一点。

第一种方式是将 JavaScript 代码放到文档<head>标签的<script>标签中：

```
<!DOCTYPE html>
<html>
 <head>
  <title>hello world</title>
  <script>
     alert('hello world!');
  </script>
 </head>
 <body>
 </body>
</html>
```

将上面的代码保存到 HTML 文件中（在记事本中编写，然后另存为扩展名为 html 的文件），用任意浏览器打开，就可以看到一个弹出对话框。

第二种方式是把 JavaScript 代码存为一个扩展名为 js 的独立文件。以前的做法是在文档<head>里用<script>标签的 src 属性来指向该文件：

```
<!DOCTYPE html>
<html>
 <head>
  <title>hello world</title>
  <script src="helloworld.js"></script>
 </head>
 <body>
```

```
08    </body>
09  </html>
```

目前业界推荐的做法是把<script>放到 HTML 文档最后，</body>标签之前（第 08 行和第 09 行之间）。这样做的目的是，使浏览器更快地加载页面并展示给用户，从而增强用户体验。

1.2 选择开发工具

近几年 JavaScript 的开发工具也得到了蓬勃发展，大小工具琳琅满目，难易程度也是不尽相同。比如重量级的集成开发平台，包括 Adobe Dreamweaver、Visual Studio 系列和 jetBrains WebStorm 系列，具有简单易学、适应性强、功能完善等特点，深受广大开发者的喜爱。一些轻量级的代码编辑器工具（EditPlus 和 Sublime Text）也非常受专业人士欢迎，这些编辑器看似功能简单实则扩展性很强大，初学者虽不易掌握，但熟练运用后一定会让设计人员爱不释手。这里简单介绍三种 JavaScript 开发工具。

1. Adobe Dreamweaver

这就是曾经被誉为“网页三剑客”之一的 Dreamweaver，备受广大网页设计和开发人员的喜爱，历史非常悠久。它的运行效果如图 1.1 所示。

图 1.1 Adobe Dreamweaver CS6

Dreamweaver CS6 版本支持 CSS 3、HTML 5，并集成了 jQuery 代码提示功能，是网页开发人员开发大型项目或长期使用的必备工具。

2. jetBrains WebStorm

jetBrains WebStorm 是 JavaScript 集成开发平台中针对性最强、功能最完善且非常简单易学的一款开发工具。jetBrains 公司是近年来最成功的一家专业软件平台开发商，推出了多系列专业性非

常强的开发工具，在用户体验方面十分下功夫。WebStorm 开发平台就是专门针对前端开发（HTML、JavaScript、CSS）而设计的。图 1.2 就是开发平台界面的演示效果。

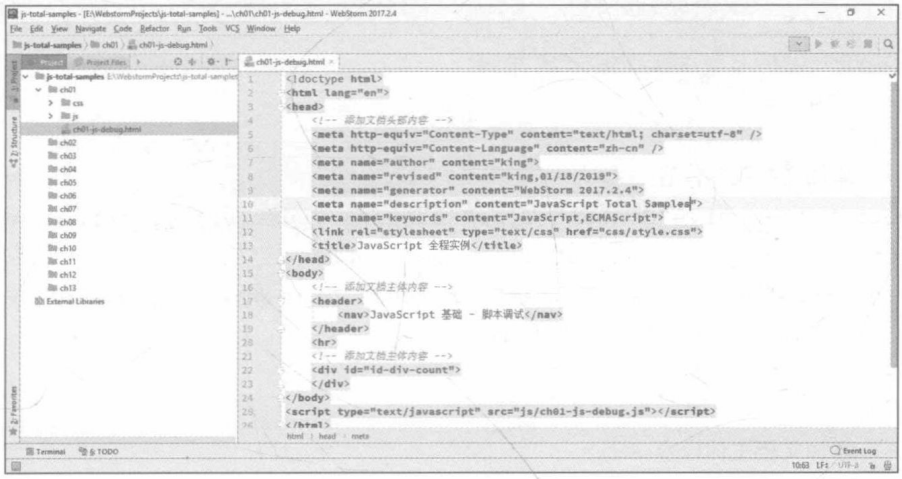

图 1.2　jetBrains WebStorm

3. Sublime Text

Sublime Text 是 JavaScript 开发环境 IDE 中比较漂亮（见图 1.3）的且对开发支持非常良好的一款文本编辑器，简洁、强大、高效。Sublime 做了很多用户体验方面的改进和支持，对审美有要求的读者可果断入手。

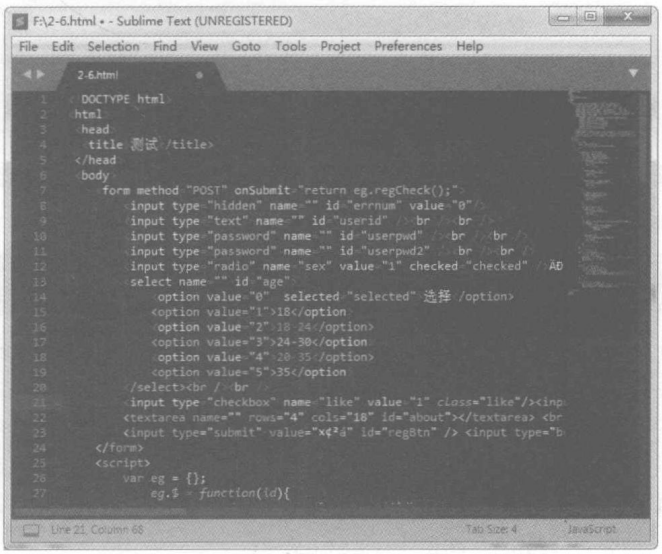

图 1.3　Sublime Text 3

1.3　JavaScript 的调试

目前的主流浏览器均支持直接调试 JavaScript 代码，本节以常见的 Chrome 和 Firefox 两款浏览器为例进行介绍。

这里先给出一段测试代码（详见源代码目录 ch01-js-debug.html 文件）：

【代码 1-1】

```
<!doctype html>
<html lang="en">
<head>
  <!-- 添加文档头部内容 -->
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Language" content="zh-cn" />
  <link rel="stylesheet" type="text/css" href="css/style.css">
  <title>JavaScript 全程实例</title>
</head>
<body>
  <!-- 添加文档主体内容 -->
  <header>
      <nav>JavaScript 基础 - 脚本调试</nav>
  </header>
  <hr>
  <!-- 添加文档主体内容 -->
  <div id="id-div-count">
  </div>
</body>
<script type="text/javascript" src="js/ch01-js-debug.js"></script>
</html>
```

其中，【代码 1-1】中第 20 行代码引入了一个外链式 js 脚本文件，具体代码如下（详见源代码目录 ch01-js-debug.js 文件）：

【代码 1-2】

```
var v_id_div_count = document.getElementById("id-div-count");
var strLine;
for(var i=1; i<10; i++) {
    strLine = "i=" + i.toString() + "<br>";
    console.log(strLine);
    v_id_div_count.innerHTML += strLine;
}
```

下面分别使用 Chrome 和 FireFox 浏览器调试上面定义的 js 代码。

1. 使用 Chrome 调试

下面使用 Chrome 浏览器运行测试代码所在的 HTML 网页，如图 1.4 所示。打开 Chrome 浏览器的开发工具面板（或按 F12 键），如图 1.5 所示。

图 1.4　使用 Chrome 浏览器打开

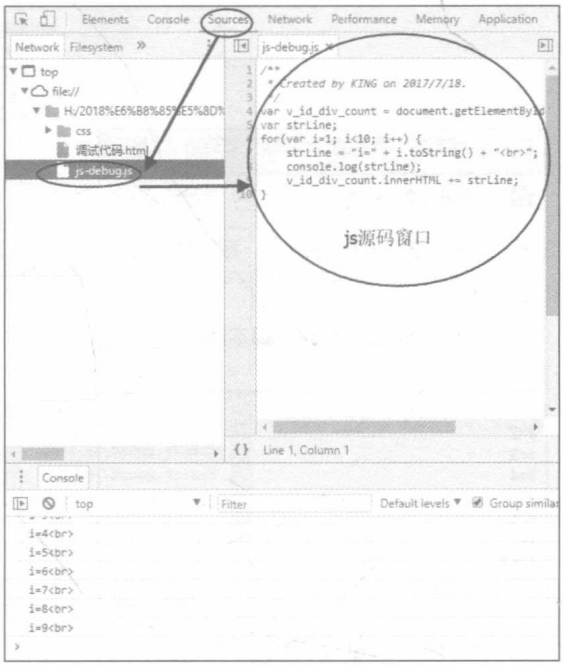

图 1.5　打开 Chrome 浏览器 js 调试功能

可以在 JavaScript 代码的第 08 行添加断点，单击行号就可以，如图 1.6 所示。这样每次程序执行到该行时中断，如图 1.7 所示。将鼠标放在某个变量上，就会显示当前变量的值，然后通过右侧的调试控制条可以控制是否继续执行下一行。

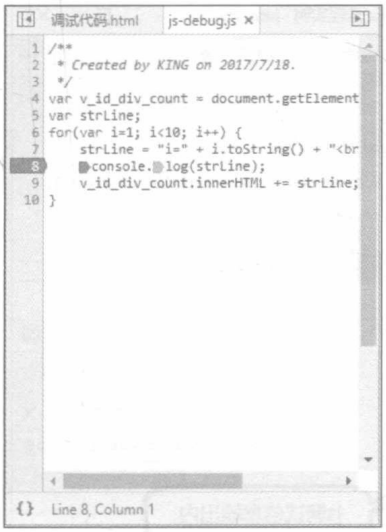

图 1.6　为 js 脚本代码设置断点

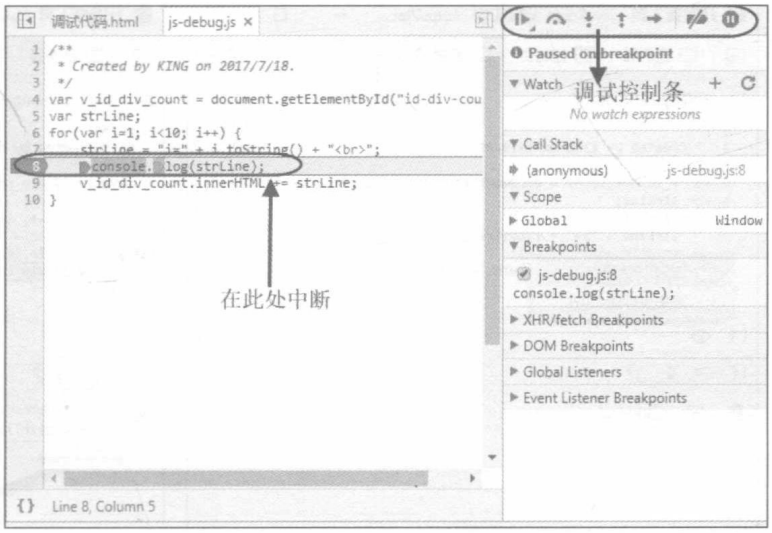

图 1.7　调试脚本代码

2. 使用 Firefox 调试

下面使用 Firefox 浏览器运行测试代码所在的 HTML 网页，如图 1.8 所示。打开 Firefox 浏览器的调试功能面板，如图 1.9 所示。

下面在图 1.9 的 js 源码窗口中为【代码 1-2】中的第 05 行脚本语句设置断点，如图 1.10 所示。

图 1.8 使用 Firefox 浏览器调试 js 脚本

图 1.9 打开 Firefox 浏览器 js 调试功能面板

然后，按“F5”功能键重新刷新页面，再按步进“F11”功能键来调试执行 js 代码，页面效果如图 1.11 和图 1.12 所示。在图 1.11 和 1.12 中可以看到，每次执行到【代码 1-2】中第 05 行脚本语句设置断点处时，js 代码均会被中断，然后在日志窗口中输出调试信息（变量“i”计数器的数值）。以上就是 JavaScript 脚本语言开发与调试的基本过程方法。

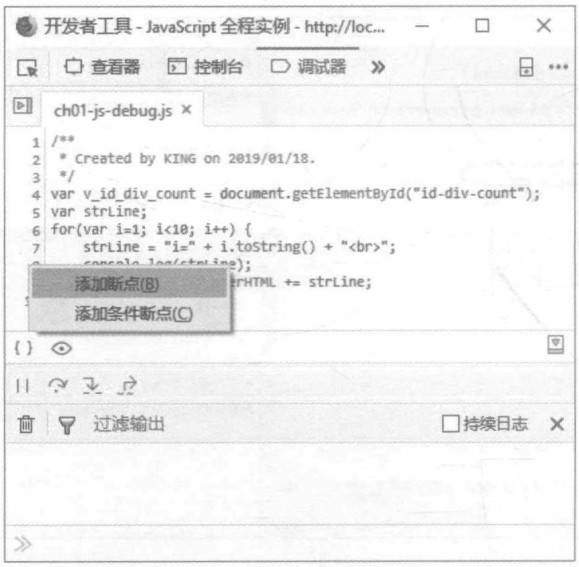

图 1.10 使用 Firefox 浏览器为 js 脚本代码设置断点

图 1.11 使用步进（F11）方式调试脚本代码

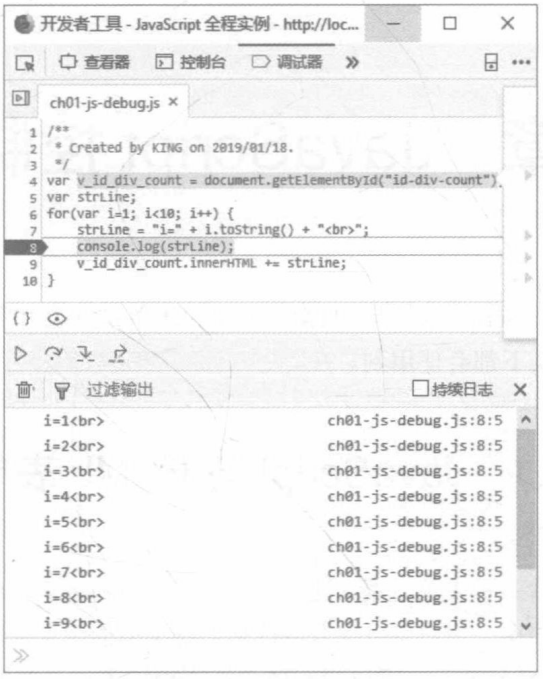

图 1.12　使用跨越（F10）方式调试脚本代码

第 2 章　JavaScript 控制表单

本章介绍如何通过 JavaScript 来控制 HTML 表单。表单是网页开发中比较常见且十分重要的一种元素，在很多应用场景下都会使用到。

2.1　JavaScript 与 HTML 表单

HTML 表单是一个包含多种表单元素（比如：文本域、下拉列表、单选框、复选框、提交按钮等）的区域。在 HTML 表单中，用户可以输入或选择多种类型的信息，然后提交到服务器来处理。

在 Web 开发中，通过 JavaScript 可以很有效地操作控制表单。例如，验证表单域中用户输入数据的合法性，格式化表单域中的用户输入数据，遍历表单域中的全部项，动态修改表单域中某项数据，控制表单的提交与重置，等等。

另外，在 HTML 5 规范中，为表单（<form>）增加了多个新的输入类型，提供了更好的输入控制和验证，与 JavaScript 结合得更加完美。

HTML 表单是通过标签（<form>）来定义的，具体语法格式如下：

【代码 2-1】

```
<form>
…form elements…
</form>
```

2.2　JavaScript 遍历表单

HTML 表单对象（Form）中定义有一个 elements 集合属性，可以返回表单中所有元素的数组集合。那么，JavaScript 通过 for 循环语句就可以实现遍历表单中全部表单项的操作了。

【代码 2-2】（详见源代码目录 ch02-js-traverse-form.html 文件）

```
<!doctype html>
<html lang="en">
<head>
  <!-- 添加文档头部内容 -->
  <title>JavaScript 全程实例</title>
```

```
06 </head>
07 <body>
08   <!-- 添加文档主体内容 -->
09   <header>
10       <nav>JavaScript 控制表单 - 遍历表单</nav>
11   </header>
12   <hr>
13   <!-- 添加文档主体内容 -->
14   <form name="formTraverse" method="get">
15       <p>First name: <input type="text" name="fname" /></p>
16       <p>Last name: <input type="text" name="lname" /></p>
17       <input type="submit" value="Submit" />
18   </form>
19 </body>
20 <script type="text/javascript">
21    var i;
22    var els = formTraverse.elements;
23    for(i in els) {
24       var el = els[i];
25       if(el.type)
26          console.log("element type : " + el.type +
                        ",element name : " + el.name);
27    }
28 </script>
29 </html>
```

关于【代码 2-2】的说明：

- 第 14 ~ 18 代码通过标签<form>定义了一个表单，内部通过标签<input>定义一组文本框和一个提交按钮。
- 第 20 ~ 28 行通过<script>标签定义了 JavaScript 脚本代码，用于实现对表单的遍历操作。
  - 第 22 行代码通过表单 form 的 elements 属性，获取了表单域中全部表单项元素的数组集合（els）。
  - 第 23 ~ 27 行代码通过 for...in...语句遍历数组集合（els），并在控制台中输出每项表单元素的 type 属性和 name 属性。

下面使用 Firefox 浏览器运行测试该 HTML 网页，具体效果如图 2.1 所示。