第 1 章

JavaScript 概述

学习目标

(1) 了解 JavaScript 的发展。
(2) 了解 JavaScript 的作用及特点。
(3) 了解 JavaScript 的组成。
(4) 掌握脚本的基本结构。
(5) 掌握网页引入脚本的方式。
(6) 了解浏览器兼容性问题。

JavaScript 是目前 Web 应用程序开发者使用广泛的客户端脚本编程语言,它不仅可以用来开发交互式的 Web 页面,更重要的是它将 HTML、XML、Ajax 和 Java Applet 等功能强大的 Web 对象有机结合起来,使开发人员能快速生成 Internet 或 Intranet 上使用的分布式应用程序。

任务 1.1 认识 JavaScript

1.1.1 JavaScript 的起源

20 世纪 90 年代中期,大部分互联网用户使用 28.8Kbps 的调制解调器连接到网络进行网上冲浪,为解决网页功能简单的问题,HTML 文档已经变得越来越复杂和庞大,更让用户痛苦的是,为了能验证一个表单的有效性,客户端必须与服务器端进行多次的数据交互,甚至等待几十秒之后,服务器端返回的不是“提交成功”,而是错误提示。当时业界已开始考虑开发一种客户端脚本语言,用于解决诸如表单合法性验证等简单而实用的问题。

1995 年,Netscape 公司和 Sun 公司联合开发出 JavaScript 脚本语言,并在 Netscape Navigator 2 中实现了 JavaScript 脚本规范的第一个版本,即 JavaScript 1.0 版,该脚本语言不久就显示了其强大的生机和发展潜力。当时 Netscape Navigator 主宰着 Web 浏览器市场,为了跟上 Netscape 的步伐,Microsoft 在其 Internet Explorer 3 中以 Jscript 为名,发布了一个 JavaScript 1.0 的克隆版本 Jscript 1.0。

1997 年,为了避免无序竞争,同时解决 JavaScript 几个版本中语法、特性等方面的混乱问题,JavaScript 1.1 作为一个草案被提交给欧洲计算机制造商协会(ECMA),经协商后推出了 ECMA-262 标准,其中定义了 ECMAScript 这种全新的脚本语言。然而至今各个浏览

器对 JavaScript 的支持未完全遵循该标准。

到了 21 世纪,网上各种广告和滚动提示条越来越多,JavaScript 被很多网页制作者乱用,直到 2005 年年初,Google 公司的网上产品使 Ajax 快速兴起,并受到广泛好评,作为 Ajax 最重要元素之一的 JavaScript 才重新找到了自己的定位。

如今 JavaScript 正朝着提高用户体验、增强网页友好性的方向发展,并越来越受到开发人员的关注,各种 JavaScript 的功能插件层出不穷,网页的功能在它的基础上更加丰富多彩。工程师要用发展的眼光看待事物,只要新事物代表了事物的发展方向,就具有强大的生命力。

1.1.2 JavaScript 的作用

JavaScript 的诞生无疑给网页注入了新的活力,它除了实现普通的表单验证外,还可以制作各种漂亮的页面特效,越来越多的网页使用了这一脚本语言。随着 Web 技术的发展和成熟,JavaScript 还被用在服务器的通信上,也就是近年来应用越来越广泛的 Ajax 技术上。

1. 客户端表单验证

JavaScript 最开始出现的目的就是解决验证方面的问题,这也是 JavaScript 最基本和最重要的作用。表单验证的应用场合比较常见,例如,网站中常见的会员注册,填写注册信息时,如果某项信息格式输入错误(例如,密码长度位数不够、出生日期输入字符等),表单页面将及时给出错误提示。这些错误在没有提交到服务器端前,在客户端提前进行验证,称为客户端表单验证。这样,用户得到了及时的交互,同时也减轻了网站服务器端的压力。

2. 页面动态效果

在 JavaScript 中,用户可以编写响应鼠标、键盘等事件的代码,创建动态页面特效,从而高效地控制页面内容。例如,幻灯片切换特效(见图 1-1)和层的隐藏/显示特效(见图 1-2 和图 1-3),它们的应用增强了客户端的体验,使网站更加有动感、有魅力,吸引了更多的浏览者。

图 1-1 幻灯片切换特效

图 1-2　层的隐藏

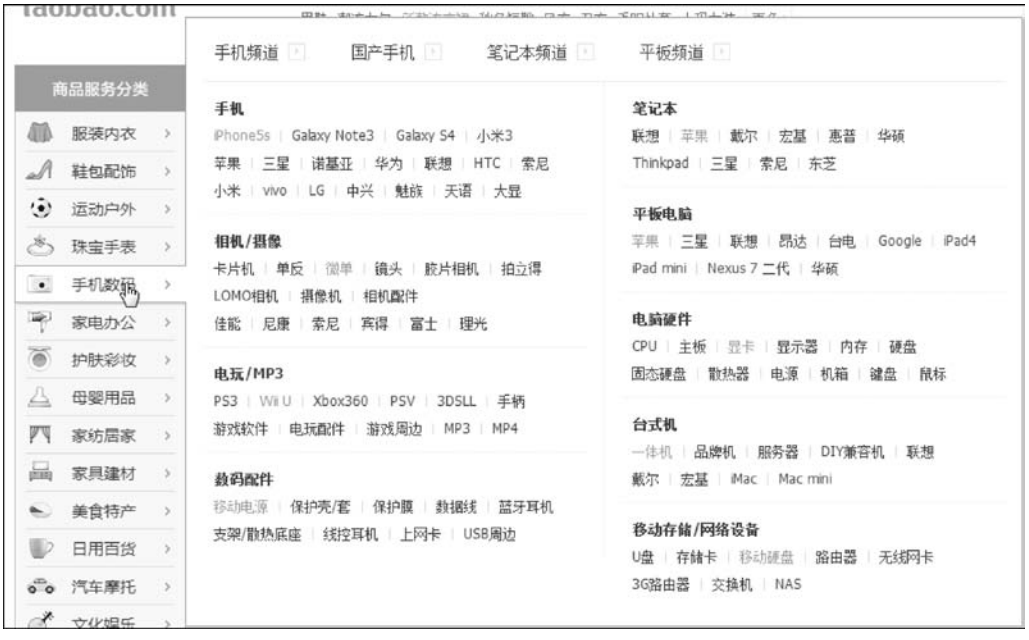

图 1-3　层的显示

3. 动态改变页面内容

在实际应用中，通常需要动态地在页面中为表格添加一行、删除一行或者改变表格中的内容，这些功能经常在购物网站中使用，用户可以方便地修改购物的数量等，如图 1-4 所示。

商品名称	数量	价格	操作
防滑真皮休闲鞋	12	¥568.50	删除 修改
抗疲劳神奇钛项圈	2	¥49.00	删除 修改
抗疲劳神奇钛项圈	3	¥49.00	删除 修改
抗疲劳神奇钛项圈	6	¥49.00	删除 确定
抗疲劳神奇钛项圈	5	¥49.00	删除 修改
抗疲劳神奇钛项圈	6	¥49.00	删除 修改
抗疲劳神奇钛项圈	7	¥49.00	删除 修改
增加订单			

图 1-4　动态改变表格内容

1.1.3　JavaScript 的基本特点

JavaScript 是一种基于对象和事件驱动的客户端脚本语言,并具有相对的安全性,主要用于创建交互性较强的动态页面。其主要特点如下。

(1) 基于对象: JavaScript 是基于对象的脚本编程语言,能通过 DOM(文档对象模型)及自身提供的对象和操作方法来实现所需的功能。

(2) 事件驱动: JavaScript 采用事件驱动方式,能响应键盘、鼠标及浏览器窗口事件等,并执行指定的操作。

(3) 解释性语言: JavaScript 是一种解释性脚本语言,无须专门的编译器进行编译,在嵌入 JavaScript 脚本的 HTML 文档被浏览器载入时逐行地解释,大量节省了客户端与服务器端进行数据交互的时间。

(4) 实时性: JavaScript 事件处理是实时性的,无须经服务器端即可对客户端的事件做出响应,并用处理结果实时更新目标页面。

(5) 动态性: JavaScript 提供简单高效的语言流程,灵活处理对象的各种方法和属性,同时及时响应文档页面事件,实现页面的交互性和动态性。

(6) 跨平台: JavaScript 脚本的正确运行依赖于浏览器,而与具体的操作系统无关。只要客户端装有支持 JavaScript 脚本的浏览器,JavaScript 脚本运行结果就能正确反映在客户端浏览器平台上。

(7) 开发使用简单: JavaScript 的基本结构类似于 C 语言,采用小程序段的方式编程,并提供了简易的开发平台和便捷的开发流程,能嵌入 HTML 文档中供浏览器解释执行,同时 JavaScript 的变量类型是弱类型,使用不严格。

(8) 安全性: JavaScript 是客户端脚本,通过浏览器解释执行。它不允许直接访问本地计算机,并且不能将数据存到服务器上,它也不允许对网络文档进行修改和删除,只能通过浏览器实现信息浏览或动态交互,从而有效地防止数据的丢失。

综上所述,JavaScript 是一种有较强生命力和发展潜力的脚本描述语言,可被直接嵌入 HTML 文档中,供浏览器解释执行; 它可直接响应客户端事件,如验证数据表单合法性等,并调用相应的处理方法,迅速返回处理结果并更新页面,实现 Web 交互性和动态性

的要求。同时将大部分的工作交给客户端处理，使 Web 服务器资源服务器的消耗降到最低。

1.1.4 JavaScript 的组成

尽管 ECMAScript 是一个重要的标准，但它并不是 JavaScript 的唯一部分，也不是唯一被标准化的部分。实际上，一个完整的 JavaScript 是由以下 3 个不同部分组成的，如图 1-5 所示。

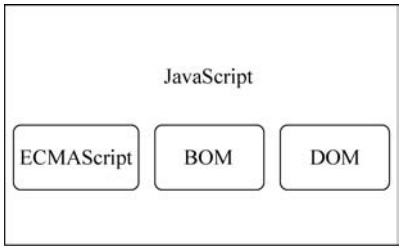

图 1-5 JavaScript 的组成部分

1. ECMAScript 标准

ECMAScript 是一种开放的、被国际上广为接受的、标准的脚本语言规范，它不与任何的浏览器绑定，实际上，它也没有提到用于任何用户输入/输出的方法，那么 ECMAScript 标准主要对哪些内容实行了规范呢？简单地说，ECMAScript 主要描述了以下内容。

(1) 语法。

(2) 变量和数据类型。

(3) 关键字、保留字。

(4) 运算符。

(5) 逻辑控制语句。

(6) 对象。

ECMAScript 是 JavaScript 语言的标准，它在不断升级更新，2009 年 ECMAScript 5 问世，经过持续几年的磨砺，2015 年发布了 ECMAScript 第 6 个版本(ES6)，它已成为 JavaScript 有史以来最实质的升级，特性涵盖范围甚广，以后每一年都会推出一个版本，目前已更新到 ECMAScript12。本书中将 2015 年以后的版本统称为 ES6，将 2015 年之前的版本统称为 ES5，不会详细去区分到底是哪个版本。

2. 浏览器对象模型(BOM)

从 Internet Explorer 3 和 Netscape Navigator 3 开始，浏览器都提供一种被称为 BOM (browser object model)的特性，它可以对浏览器窗口进行访问和操作。利用 BOM 的相关技术，Web 开发者可以移动窗口、改变状态栏以及执行一些与页面内容不相关的操作。

3. 文档对象模型(DOM)

DOM 是 document object model(文档对象模型)的简称，是 HTML 文档对象模型(HTML DOM)定义的一套标准方法，用来访问和操纵 HTML 文档。DOM 由万维网联盟(world wide web consortium，W3C)定义，最新的浏览器都支持第一级和第二级 DOM(第一级和第二级 DOM 是一种标准)，通过 JavaScript，可以重构整个 HTML 文档。

任务 1.1 认识 JavaScript 微课

任务 1.2　在页面显示个人信息

任务描述

使用浏览器打开页面时,在页面使用 JavaScript 语言输出个人信息,包括所在班级、学号、姓名、性别和爱好。页面效果如图 1-6 所示。

图 1-6　个人信息页面

任务分析

本任务要在页面使用 JavaScript 语言输出个人信息,完成该任务可以使用下列步骤。

(1) 选择 JavaScript 脚本编辑器编辑脚本代码。

(2) 嵌入脚本代码到 HTML 文档中。

(3) 使用 document. write 语句在页面输出信息。

(4) 使用浏览器浏览该 HTML 文档。

1.2.1　选择 JavaScript 脚本编辑器

编写 JavaScript 脚本代码可以选择普通的文本编辑器,如记事本、UltraEdit、EditPlus 等,只要选择的编辑器能将所编辑的代码最终保存为 HTML 文档(扩展名为. html、. htm 等)即可。也可以选择专业的前端开发工具,如 HBuilder、Sublime Text、Dreamweaver、WebStorm 等,这些专业的开发工具内部集成了 JavaScript 脚本的开发环境,使用起来简单、方便。本书案例采用 HBuilder 工具实现。

1.2.2　脚本的基本结构

通常 JavaScript 代码通过＜script＞标签嵌入 HTML 文档中,可以将多个脚本嵌入一个文档中,只需将每个脚本都封装在＜script＞标签中。浏览器遇到＜script＞标签时,将逐行读取内容,直到遇到＜/script＞结束标签为止。浏览器将检查 JavaScript 语句的语法,如果有错误,将提示错误; 如果没有错误,浏览器将解释执行语句。

1. 脚本的基本结构

脚本的基本结构如下:

```
<script language="javascript 1.2" type="text/javascript">
    //JavaScript 语句;
</script>
```

其中，<script>...</script>标签对将 JavaScript 脚本代码进行封装，同时告诉浏览器其间代码为 JavaScript。language 属性指定封装代码的脚本语言及版本；type 属性指定插入的脚本代码类型。

2. 脚本的执行原理

在脚本的执行过程中，浏览器客户端与服务器端采用请求/响应模式进行交互，如图 1-7 所示。

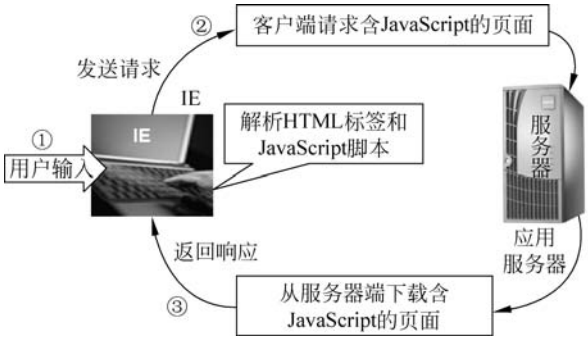

图 1-7　脚本的执行原理

下面解析一下这个过程。

(1) 浏览器接收用户请求，即用户在浏览器的地址栏中输入要访问的页面(页面包含 JavaScript 脚本代码)。

(2) 浏览器把请求消息发送到 Web 服务器端，等待服务器端响应。

(3) Web 服务器端响应请求，Web 服务器端找到请求的页面，将整个页面包含 JavaScript 脚本代码作为响应内容，发送回客户端；客户端浏览器打开回应的网页文件内容，从上往下逐行读取并显示其中的 HTML 标签和 JavaScript 脚本代码。

使用客户端脚本有以下好处。

(1) 含脚本的页面只要下载一次即可，减少了不必要的网络通信。

(2) 脚本是从服务器端下载到客户端，然后在客户端进行的，不占用服务器端的资源，客户端分担了服务器端的任务，大大地减轻了服务器的压力，从而间接地提升了服务器的性能。

1.2.3　在网页中引用 JavaScript 的方式

JavaScript 脚本代码可以通过多种方式嵌入 HTML 文档中，主要有以下几种：①使用<script>...</script>标签对；②链接外部的 JavaScript 文件；③JavaScript 伪 URL 引入。

1. 使用<script>...</script>标签对

使用<script>...</script>标签对将 JavaScript 脚本代码嵌入 HTML 文档中，是最常用的方法。

【实例 1-1】

```
<!DOCTYPE html>
```

```
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>script 标签方式</title>
        <script type="text/javascript">
          alert("知识是伟大的!");
        </script>
    </head>
    <body>
    </body>
</html>
```

浏览器载入嵌有 JavaScript 脚本的 HTML 文档时,能自动识别 JavaScript 脚本代码起始标签<script>和结束标签</script>,并将其间的代码解释执行,然后将解释结果返回 HTML 并在浏览器窗口显示。

<script>...</script>的位置并不是固定的,可以包含在<head>...</head>和<body>...</body>中的任何位置。这种方式通常适用于 JavaScript 脚本代码比较少,并且网站中的每个页面使用的 JavaScript 脚本代码均不同的情况。在实际项目开发中,开发人员有时希望在多个页面运行 JavaScript 实现相同的页面效果,这时通常使用外部 JavaScript 文件。

2. 链接外部的 JavaScript 文件

外部 JavaScript 文件是将 JavaScript 写入一个外部文件中,以.js 为扩展名保存这个文件,然后通过<script>标签的 src 属性将脚本代码嵌入 HTML 文档中。

【实例 1-2】

```
<script type="text/javascript" src="hello.js"></script>
```

hello.js 就是外部 JavaScript 文件,src 属性表示指定外部 JavaScript 文件的路径,一般采用相对路径。

外部文件中不能包含<script>标签,把.js 文件放到通常存放脚本的目录中,这样容易管理和维护。

3. JavaScript 伪 URL 引入

通过伪 URL 方式调用语句来引入 JavaScript 脚本代码的格式如下:

```
JavaScript: JavaScript 脚本代码
```

【实例 1-3】

```
<h4>通过伪 URL 方式引入 JavaScript 脚本代码: </h4>
<form>
  <input type="button" value="伪 URL 方式" onclick="javascript:alert('这是另一种方式调用脚本代码')"/>
</form>
```

1.2.4 常用的输入/输出语句

在 JavaScript 中常用的输入/输出语句有：警告对话框 alert()、提示对话框 prompt() 和输出语句 document.write()。

1. 警告对话框 alert()

alert()方法会创建一个警告对话框，用于将浏览器或文档的警告信息传递给客户。参数可以是变量、字符串或表达式，警告对话框无返回值。

alert()方法的基本语法格式如下：

```
alert("提示信息");
```

(1) 参数是字符串

```
alert("欢迎来到 JavaScript 世界!");
```

效果如图 1-8 所示。

(2) 参数是变量

```
var myName = "Jacky";
alert("大家好,\n 我是" + myName);
```

效果如图 1-9 所示。

说明：

JavaScript 中提供了一些特殊字符，允许在字符串中包括一些无法直接输入的字符，每个字符以反斜杠\开始，其中\n 就是转义字符，表示换行，其他常用的转义字符还有\r 回车符，\t 制表符，\"双引号，\\反双斜杠。

(3) 参数是表达式

```
alert("7 + 8 = " + (7 + 8));
```

效果如图 1-10 所示。

图 1-8 参数是字符串

图 1-9 参数是变量

图 1-10 参数是表达式

2. 提示对话框 prompt()

prompt()方法会弹出一个提示对话框，用于收集客户关于特定问题的反馈信息，提示对话框具有返回值。

prompt()方法的基本语法格式如下：

```
prompt("提示信息","输入框的默认信息");
```

该方法的返回值可以被引用或存储到变量中。

【实例 1-4】

```
<script type="text/javascript">
    var sports = prompt("输入你最喜欢的运动:","打羽毛球");
    alert("你最喜欢的运动是:" + sports);
</script>
```

运行效果如图 1-11 所示,单击“确定”按钮后,对话框如图 1-12 所示。

图 1-11 提示对话框

图 1-12 单击“确定”按钮的效果

说明:

如果用户单击“取消”按钮,prompt()方法将返回 null;如果用户单击“确定”按钮,prompt()方法将返回用户输入的字符串或默认字符串;如果用户没有设置 prompt()方法的第二个参数,默认将得到 undefined 的值。

3. 输出语句 document.write()

document.write()方法可以向文档写文本、HTML 表达式或 JavaScript 脚本代码。该方法需要一个字符串参数,它是写到文档 HTML 的内容,这些字符串参数可以是变量或值为字符串的表达式,写入的内容常常包括 HTML 标签。

document.write()方法的基本语法格式如下:

```
document.write("输出内容");
```

(1) 参数是字符串

```
document.write ("JavaScript 的学习非常有趣");
```

(2) 参数是变量

```
var myName = "Jacky";
document.write ("大家好,我是" + myName);
```

(3) 参数是 HTML 标签

```
document.write ("<h1>网页制作常用工具</h1><ul><li>图像处理工具 Photoshop</li><li>网页编辑工具 Dreamweaver</li></ul>");
```

运行效果如图 1-13 所示。