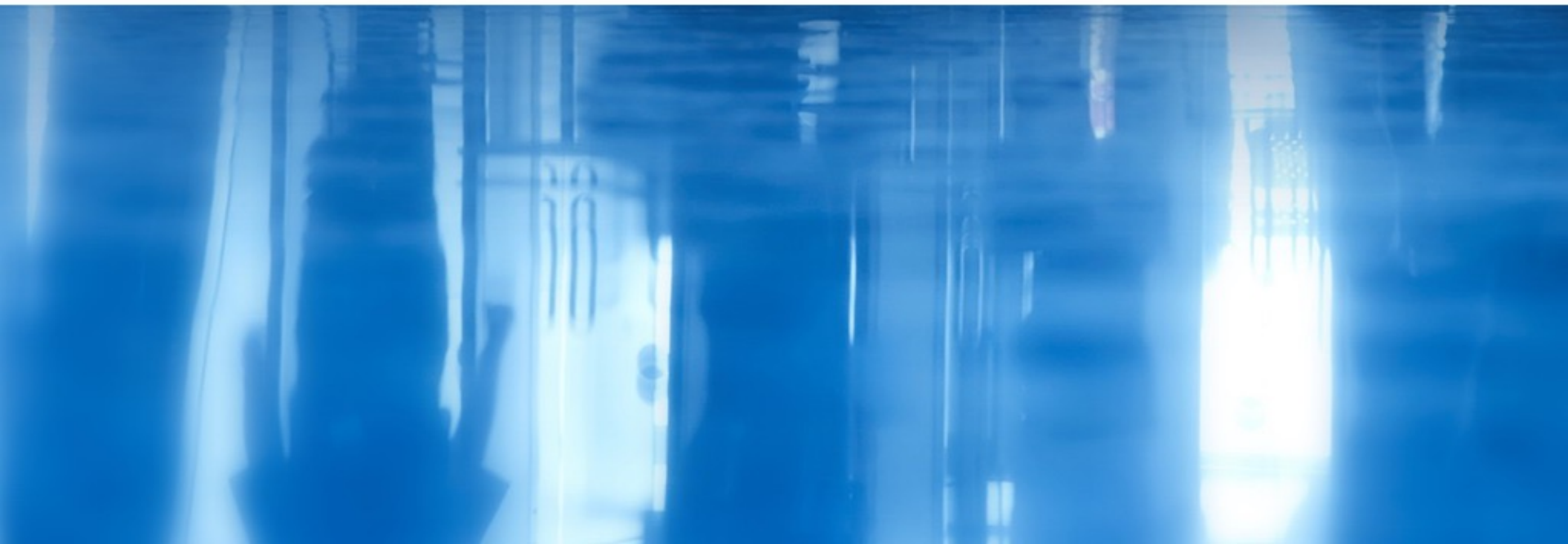


第10章

AJAX技术基础



本章内容

- 10.1 Web 2.0与AJAX
- 10.2 XMLHttpRequest
- 10.3 DOM和JavaScript
- 10.4 AJAX常用应用

AJAX技术概述

- AJAX技术是Web 2.0阶段系列技术和相关产品服务中非常重要的一种技术。
- 作为一种客户端技术，AJAX应用实现客户浏览器与服务器的异步交互。
- 在异步交互中，客户使用XMLHttpRequest对象发送请求并获得服务器的响应，同时AJAX可以在不刷新整个页面的情况下用JavaScript操作DOM以实现页面动态更新。

AJAX技术概述

- 本章主要介绍
 - 什么是AJAX及相关技术；
 - AJAX异步通信的工作原理；
 - AJAX的主要应用等。
- 通过本章的学习，读者应该学会使用AJAX开发具有更大交互性和更丰富用户体验的Web应用。

10.1 Web 2.0与AJAX技术概述

- 2005年Web 2.0成为了人们关注的焦点。Web 2.0代表的的是一个新的网络阶段，它本身并没有明确的标准来进行描述，一般将促成这个阶段的各种技术和相关产品服务统称为Web 2.0。
- 伴随者Web 2.0的诞生，互联网进入了一个更加开放、交互性更强、由用户决定内容并参与共同建设的可读写网络阶段。

10.1.1 什么是AJAX

- AJAX是英文Asynchronous JavaScript And XML的缩写，意为异步JavaScript与XML。
- 该术语最早是由Jesse James Garrett创造的。

10.1.1 什么是AJAX

- Web应用使用HTTP协议，它通过请求/响应机制为客户提供服务。
- 当客户发出一个请求时，服务器把整个页面发送给客户。如果页面中只有一小部分内容需要修改，服务器仍然发送整个页面，也就是需要完全页面刷新。这样就需要占用更多的网络资源，响应时间也较长。
- AJAX技术就是为了解决这个问题而产生的，使用该技术如果只需更新页面中一小部分内容时，只需部分刷新页面即可。。

10.1.2 AJAX相关技术简介

- AJAX不仅仅只包含异步JavaScript和XML，它实际上是包含多种技术的一个综合技术，其中包括JavaScript脚本、XHTML、CSS、DOM、XML、XSTL以及最重要的XMLHttpRequest对象。
- 实际上，AJAX是包含允许浏览器与服务器异步通信而无需完全刷新当前页面的所有技术。

10.1.2 AJAX相关技术简介

- 开发人员可以用XHTML和CSS实现数据信息的标准化显示；使用DOM实现浏览器丰富的动态显示效果；使用XML和XSTL进行浏览器和服务器的数据交换和处理；使用XMLHttpRequest实现客户与服务器之间的异步请求和响应；使用JavaScript脚本语言对所有数据进行处理。
- 下面概括介绍一下AJAX技术中几个最核心的技术。

1. XHTML与CSS

- XHTML (eXtensible HyperText Markup Language) 指的是可扩展的标记语言，它是一种为适应XML而重新改造的HTML语言。
- 从本质上说，XHTML是一种过渡技术，它结合了XML中的部分强大功能和HTML中的大部分的简单特性。
- CSS (Cascading Style Sheet) 指的是级联样式单，它是万维网联盟开发的一种为了实现HTML页面中数据与格式相分离的一种技术。

2. JavaScript脚本语言

- JavaScript是一种可以与HTML标记语句混合使用的脚本语言，其编写的程序可以直接在浏览器中解释执行。
- 在JavaScript脚本中可以调用浏览器中提供的对象，利用这些对象提供的属性和方法可以实现页面效果的动态控制。

3. DOM

- DOM (Document Object Model) 指的是文档对象模型，它是W3C的一个规范，可以用一种独立于平台和语言的方式访问和修改文档的内容和结构。
- DOM是面向HTML和XML文档的API，为文档提供了结构化表示，并定义了如何通过脚本来访问文档结构。文档中的每个元素都是DOM的一部分。在HTML页面中通常使用JavaScript脚本语言来访问DOM。

4. XML与XSLT

- XML (eXtensible Markup Language) 称为可扩展的标记语言。
- 它是在SGML和HTML的基础上发展起来的。XML吸取了两者的优点，克服了SGML过于复杂和HTML局限性等缺点。
- 目前XML已成为网上数据交换的标准。

4. XML与XSLT

- XML具有如下的一些特点：
 - **可扩展性。** XML不是标记语言，它本身并不包含任何标记。它允许用户自己定义标记和属性，可以有各种定制的数据格式。
 - **更多的结构和语义。** XML侧重于对文档内容的描述，而不是文档的显示。用户定义的标记描述了数据的语义，便于数据的理解和机器处理。HTML只能表示文档的格式，而用XML可以描述文档的结构和内涵。

4. XML与XSLT

- **自描述性**。对数据的描述和数据本身都包含在文档中，使数据具有很大的灵活性。
- **数据与显示分离**。XML所关心的是数据本身的语义，而不是数据的显示，所以可以在XML数据上定义多种显示形式。

4. XML与XSLT

- XSL (eXtensible Stylesheet Language) 称为可扩展的样式单语言，它是一种用来转换XML文档结构的语言。使用XSL可以从一个XML文档中提取信息，并使用该信息创建另一个XML文档。
- 下面是一个简单的XML文档，它描述了一个图书馆中图书和杂志的信息，其中包括两本图书和一本杂志。
- [程序10.1 library.xml](#)

5. XMLHttpRequest

- XMLHttpRequest是浏览器中定义的对象，它是AJAX技术中的核心对象。
- 通过JavaScript脚本可以创建XMLHttpRequest对象。
- XMLHttpRequest对象定义了若干属性和方法，通过这些属性和方法就可以向服务器发出异步请求和处理响应结果。再结合上面的技术就可以实现在不刷新整个页面的情况下更新页面数据，从而实现更丰富的用户体验。

本章内容

- 10.1 [Web 2.0与AJAX](#)
- 10.2 [XMLHttpRequest](#)
- 10.3 [DOM和JavaScript](#)
- 10.4 [AJAX常用应用](#)

10.2 XMLHttpRequest对象

- 正是XMLHttpRequest对象提供了Web应用程序与服务器之间的异步通信。
- 使用XMLHttpRequest对象，客户可以直接从Web服务器检索数据，或向Web服务器提交XML数据而不需要刷新整个页面。XML数据在客户端使用DOM与XSLT转换成HTML。

10.2.1 创建XMLHttpRequest对象

- 用JavaScript创建一个XMLHttpRequest对象。
- 创建XMLHttpRequest对象时应先检查浏览器是否支持ActiveX对象。如果浏览器支持ActiveX对象，就可以使用ActiveX对象来创建XMLHttpRequest对象，否则，就要使用本地JavaScript对象技术创建XMLHttpRequest对象。

10.2.1 创建XMLHttpRequest对象

```
var xmlHttp;  
function createXMLHttpRequest(){  
    if(window.ActiveXObject){  
        xmlHttp =  
            new ActiveXObject("Microsoft.XMLHTTP");  
    }else if(window.XMLHttpRequest){  
        xmlHttp = new XMLHttpRequest();  
    }  
}
```

10.2.2 XMLHttpRequest的属性

属性名	说 明
onreadystatechange	为异步请求设置事件处理程序。每当XMLHttpRequest对象的状态改变时都会触发这个事件处理器，通常会调用一个JavaScript函数
readyState	该属性表示请求的状态。它可有下面几个不同的值：0：未初始化；1：正在加载；2：已加载；3：交互中；4：完成
responseText	检索服务器响应，并表示为文本
responseXML	检索服务器响应，并表示为XML DOM对象
status	检索服务器的HTTP状态码。如404表示Not Found，200表示OK
statusText	检索服务器的HTTP状态码的文本
responseBody	检索响应体。该属性是IE 7及以后版本的window对象，但不是W3C的规范

10.2.3 XMLHttpRequest的方法

方法名	说 明
<code>abort()</code>	取消当前HTTP请求
<code>getAllResponseHeaders()</code>	返回所有的请求头。如果 <code>readyState</code> 属性的值不是3或4, 将返回 <code>null</code>
<code>getResponseHeader(string header)</code>	返回指定的响应头。如果 <code>readyState</code> 属性的值不是3或4, 将返回 <code>null</code>

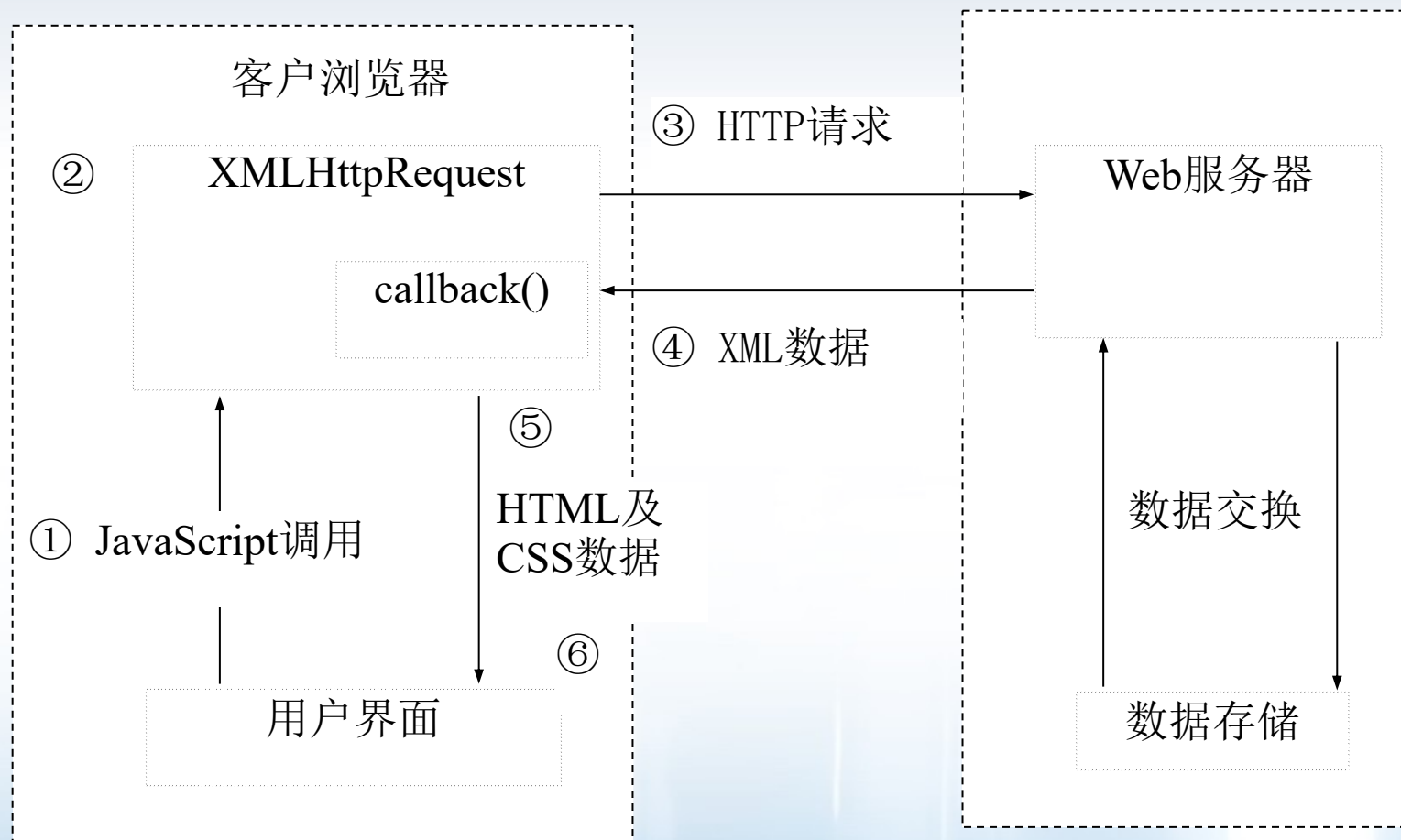
10.2.3 XMLHttpRequest的方法

<code>open(string method, string url, boolean asynch, string username, string password)</code>	打开一个HTTP请求，但并没有发送请求。调用open()方法将readyState属性设置为1，responseText、responseXML、status和statusText属性设置为初始值。在open()方法中需要指定使用的HTTP方法（GET、POST或PUT）和服务器的URL（相对或绝对）。另外，还可以传递一个Boolean值，指示这个调用是异步的还是同步的。默认值为true，表示请求是异步的。如果这个参数是false，浏览器就会等待，直到从服务器返回响应为止。username和password指定服务器端验证的用户名和密码。后三个参数是可选的
<code>send(data)</code>	向服务器发送HTTP请求并检索响应。数据可以是字符串、无符号字节数组或XML DOM对象等。发送的数据是可选的，其值可以为null。根据open()方法中asynch参数的值不同，send()方法可以是同步的，也可以是异步的。如果是同步的，send()方法只有在接收到全部响应才返回，如果是异步的，该方法立即返回。在调用send()方法后，readyState属性被设置为2，当请求完成时，readyState属性被设置为4
<code>setRequestHeaders(string header, string value)</code>	设置请求的HTTP头。header为请求头名，value为请求头的值

10.2.4 一个简单的示例

- 下面是一个简单的HTML页面。其中有一个按钮，当单击该按钮时将向服务器发送一个异步请求。服务器将发回一个简单的静态文本文件作为响应。在处理这个响应时，会在警告窗口中显示该文本文件的内容。
- [程序10.2 simpleRequest.html](#)

10.2.5 AJAX的交互模式



1. 客户触发事件

- 一个客户事件触发一个AJAX事件。客户事件从简单的onchange事件到某个特定的用户动作，很多这样的事件都能触发AJAX事件。例如：

```
<input type="button" value="开始异步请求"
```

```
onclick="startRequest();" />
```

- 这里，当用户单击按钮时，将触发onclick事件，程序调用startRequest()函数。

2. 创建XMLHttpRequest对象

- 在startRequest()函数中通过调用createXMLHttpRequest()函数创建了XMLHttpRequest对象，代码如下：

```
function createXMLHttpRequest() {  
    if (window.ActiveXObject) {  
        xmlHttp = new  
ActiveXObject("Microsoft.XMLHTTP");  
    }  
    else if (window.XMLHttpRequest) {  
        xmlHttp = new XMLHttpRequest();  
    }  
}
```

3. 向服务器发出请求

- 在向服务器发出请求之前，应该通过XMLHttpRequest对象的onreadystatechange属性设置回调函数。当XMLHttpRequest对象的内部状态改变时就会调用回调函数，因此回调函数是处理响应的地方。

```
function startRequest() {  
    createXMLHttpRequest();  
    xmlHttp.onreadystatechange =  
        handleStateChange;  
    xmlHttp.open("GET",  
        "simpleResponse.xml", true);  
    xmlHttp.send(null);}
```

4. 服务器处理请求并返回响应

- 如果请求的是静态资源，服务器将返回该资源。如果请求的是动态资源，服务器将执行动态资源，这可能需要访问数据库甚至是另一个系统，然后向用户返回响应。
- XMLHttpRequest对象提供了两个访问服务器响应的属性。一个是`responseText`，它将响应提供为一个字符串。另一个属性是`responseXML`，它将响应提供为一个XML对象。

4. 服务器处理请求并返回响应

- 由于XMLHttpRequest对象只能处理text/html类型的结果，所以如果请求的是动态资源（如Servlet），需要将Content-Type响应头设置为text/xml，另外，为了避免浏览器在本地缓存结果，需要将Cache-Control响应头设置为no-cache，如下所示：

```
response.setHeader("Cache-  
Control", "no-cache");
```

5. 通过回调函数处理结果

- 通过回调函数可以对响应结果进行处理。在回调函数中首先应该检查XMLHttpRequest对象的`readyState`属性和`status`属性的值。当`readyState`属性值为4、`status`属性的值为200时表示响应完成，这时才能使用XMLHttpRequest对象的`responseText`或`responseXML`检索请求结果。例如，下面是程序10.2中的回调函数：

5. 通过回调函数处理结果

```
function handleStateChange() {  
    if (xmlHttp.readyState == 4) {  
        if (xmlHttp.status == 200) {  
            alert("The server  
replied with: " +  
xmlHttp.responseText);  
        }  
    }  
}
```

6. 更新HTML DOM对象

- 客户使用新的数据更新HTML DOM页面表示元素。JavaScript脚本可以使用DOM API获得HTML的每个元素的引用。
- 一般方法是使用 `document.getElementById(“userIdMessage”)`，这里，userIdMessage是HTML文档一个元素的id属性值。
- 有了元素的引用，JavaScript就可以修改元素的属性、修改元素的style属性或者添加、删除、修改子元素。修改元素内容的一个常用方法是设置元素的innerHTML属性值。

10.2.6 使用innerHTML属性创建动态内容

- 结合HTML元素的innerHTML属性，XMLHttpRequest对象的responseText属性就会变得更有用。innerHTML属性是一个非标准的属性，最早在IE中实现，后来也为其他许多流行的浏览器所采用。innerHTML属性是一个简单的字符串，表示一组开始标记和结束标记之间的内容。
- 下面的例子使用XMLHttpRequest对象的responseText属性和HTML元素的innerHTML属性实现生成HTML内容。
- [程序10.3 innerHTML.html](#)

10.2.6 使用innerHTML属性创建动态内容

- 程序中通过document对象的getElementById()返回id为results块的对象，然后将该元素的innerHTML属性的值设置为XMLHttpRequest对象的responseText的值。
- 下面的XML文件innerHTML.xml是客户要请求和返回的文件：
- [程序10.4 innerHTML.xml](#)

10.3 DOM和JavaScript

- 10.3.1 什么是DOM
- 10.3.2 DOM与JavaScript
- 10.3.3 使用DOM动态编辑页面
- 10.3.4 发送请求参数

10.3.1 什么是DOM

- DOM的含义是**文档对象模型**。在W3C主页上对DOM的定义为：文档对象模型是与平台和语言无关的接口，允许程序和脚本动态地访问和更新文档的内容、结构和样式。文档可以进一步处理，处理的结果可以放回到所提供的页面中。

10.3.2 DOM与JavaScript

- DOM与JavaScript很容易混淆。DOM是面向HTML和XML文档的API，为文档提供了结构化的表示，并定义了如何通过脚本来访问文档结构。
- DOM独立于具体的编程语言，通常通过JavaScript访问DOM，不过并不严格要求这样，可以使用任何脚本语言来访问DOM，这要归功于DOM一致的API。

10.3.2 DOM与JavaScript

- 表10-3列出了DOM元素的一些常用属性，表10-4列出了一些常用方法。

属性名	说明
childNodes	返回当前元素所有子元素的数组
firstChild	返回当前元素的第一个下级子元素
lastChild	返回当前元素的最后一个子元素
nextSibling	返回紧跟在当前元素后面的元素
previousSibling	返回紧邻当前元素之前的元素
nodeValue	返回结点值
parentNode	返回元素的父结点

10.3.2 DOM与JavaScript

表10-4 遍历XML文档的DOM元素方法

方法名	说明
getElementById(id)	返回文档中由id指定的元素
getElementsByTagName(name)	返回当前元素中指定标记名的子元素的数组
hasChildNodes()	返回一个布尔值，指示元素是否有子元素
getAttribute(name)	返回指定名称的元素的属性值

10.3.2 DOM与JavaScript

- 有了DOM，就能编写简单的跨浏览器的脚本，从而充分利用XML的强大功能和灵活性，将XML作为浏览器和服务端之间数据交换的媒介。
- 程序10.5是一个HTML文档，它通过异步请求访问服务器上的XML文档，然后对其解析，通过警告框显示返回的数据。页面中包括两个按钮，一个查看图书信息，一个查看杂志信息。
- [程序10.5 parseXML.html](#)

10.3.3 使用DOM动态编辑页面类

- 表10-5列出了用于动态创建内容的DOM属性和方法。

属性/方法	说明
<code>document.createElement(tagName)</code>	创建由tagName指定的元素
<code>document.createTextNode(text)</code>	创建一个包含静态文本的节点
<code><element>.appendChild(childNode)</code>	将指定的节点添加到当前元素的子节点列表，作为一个新的子节点
<code><element>.getAttribute(name)</code>	获得元素中name属性的值
<code><element>.setAttribute(name, value)</code>	设置元素中name属性的值

10.3.3 使用DOM动态编辑页面类

- 下面的例子展示了如何使用DOM和JavaScript来动态创建内容。本例仍然使用程序10.1的library.xml文件，运行结果如图10-4所示。单击不同按钮，从XML文件中查询图书和杂志信息，并动态构建表格显示数据。
- HTML文件的代码如程序10.6所示。
- [程序10.6 dynamicContent.html](#)

10.3.4 发送请求参数

- 客户向服务器发送的请求可以带请求参数。对于不同的HTTP方法，请求参数的传递有些不同。
- 采用GET方法发送的请求，请求参数是作为“名/值”对放在URL中传递。资源URL后面有一个问号（?），问号后面就是“名/值”对。“名/值”对采用name=value的形式，多个“名/值”对之间用与号（&）分隔。下面是一个例子：

<http://localhost:8080/helloweb/login?name=Adam&birthday=1988-09-08>

10.3.4 发送请求参数

- 采用POST方法发送的请求，参数的格式与GET请求相同，只不过参数串是在请求体中发送的。一般情况下，使用POST方法发送参数是通过表单实现的，即在<form>元素中将method属性值指定为“post”。

10.3.4 发送请求参数

- 使用XMLHttpRequest对象也可以发送参数，并且可以使用GET方法或POST方法发送请求参数。但需要开发人员利用JavaScript创建查询串，其中包含的数据作为请求的一部分发送给服务器。
- 不论使用的GET方法还是POST方法，创建查询串的技术是一样的。唯一的区别是，当使用GET发送请求时，查询串会追加到请求URL中，而使用POST发送请求时，则在调用XMLHttpRequest对象的send()时发送查询串。

10.4 AJAX的常用应用

- AJAX技术在Web应用开发中有很多应用，本节通过一些例子介绍几种常见的应用，其中包括数据验证、动态加载列表框、创建工具提示、动态更新Web页面等。
- 在这些例子中，使用Java Servlet作为服务器端组件。

10.4 AJAX的常用应用

- 10.4.1 表单数据验证
- 10.4.2 动态加载列表框
- 10.4.3 创建工具提示
- 10.4.4 动态更新Web页面

10.4.1 表单数据验证

- 数据验证包括客户端验证和服务器端验证。客户端验证通常使用JavaScript编写，它只能在客户端验证用户在表单中输入的数据。这种验证不需要与服务器交换数据。服务器端验证需要与服务器交换数据。

10.4.1 表单数据验证

- 在AJAX技术出现之前，实现服务器端验证必须提交整个页面才能验证数据。使用AJAX实现验证就不受这个限制，并且可以为用户提供更好的交互性的体验。下面介绍一个常见的注册用户名的验证。
- 程序10.7是一个HTML文件，它包含一个标准文本框，接收用户输入的注册用户名，单击“检测”按钮向服务器发送请求。
- [程序10.7 register.html](#)

10.4.1 表单数据验证

- 程序中通过onclick事件触发验证方法validate()。在该方法中首先得到用户名，然后构造一个URL并向服务器的ValidationServlet发送请求。在回调方法中通过XMLHttpRequest对象的responseXML返回验证的结果信息。
- 服务器端的代码也很简单，它是通过Servlet实现验证功能的，代码如下。
- [程序10.8 ValidationServlet.java](#)

10.4.2 动态加载列表框

- 在Web应用开发中经常需要动态构建列表框的内容。如果要求在动态构建列表框时不刷新整个页面，早期的办法是使用隐藏数据。这种方法在数据量比较大时不适用。使用AJAX技术就可以很容易实现动态加载列表框。
- 程序10.9 dynamicList.html
- 程序10.10 RefreshNameServlet.java

10.4.3 创建工具提示

- 很多应用程序中都带有工具提示功能，如在Word环境中将鼠标指向一个工具按钮



10.4.3 创建工具提示

- 下面的例子实现了简单的工具提示功能。
- [程序10.11 toolTip.html](#)
- [程序10.12 ToolTipServlet.java](#)

10.4.4 动态更新Web页面

- 下面例子通过Web页面输入客户信息，单击“添加”按钮，将这些数据提交到服务器，在这里可将它们存储到数据库中。服务器发送一个状态码向浏览器作出响应，指示数据库操作是否成功。假设数据库成功插入，浏览器会使用DOM操作客户信息动态更新页面内容。这个例子中还创建了“删除”按钮，可从数据库中删除客户信息。
- [程序10.13 customerList.html](#)

10.4.4 动态更新Web页面

- 单击“添加”按钮将调用addCustomer()

The screenshot shows a web browser window with the title 'Customer List'. The address bar displays 'http://localhost:8080/helloweb/customerList.html'. The page content includes a heading '请输入客户信息' (Please enter customer information), followed by three input fields for '客户名:' (Customer Name), 'Email地址:' (Email Address), and '电话:' (Phone Number). Below these fields is a '添加' (Add) button. Underneath, the text '客户信息如下:' (Customer information as follows:) is displayed above a table. The table contains two rows of customer data, each with a '删除' (Delete) button.

客户信息如下:			
王小明	wang@163.com	8899123	删除
Smith	smith@yahoo.com	123456999	删除

10.4.4 动态更新Web页面

- 当服务器接收到请求时调用Servlet的doGet(), 在该方法中根据action请求参数的值确定调用哪个方法, 若是增加信息, 将调用addCustomer(), 若是删除信息则调用deleteCustomer()。
- 程序10.14 CustomerListServlet.java

10.5 小 结

- AJAX表示异步JavaScript与XML，它实现一种允许浏览器与服务器通信而无需完全刷新页面的技术。AJAX技术涉及到多种技术的结合，其中包括XHTML、CSS、DOM、XML、JavaScript、XSTL以及最重要的XMLHttpRequest对象。
- XMLHttpRequest对象是Ajax技术的核心，它提供了Web应用程序与服务器之间的异步通信。使用XMLHttpRequest对象，客户可以直接从Web服务器检索数据，或向Web服务器提交XML数据而不需要刷新整个页面。

10.5 小 结

- 开发人员可以使用XHTML和CSS实现数据信息的统一化、标准化显示；使用DOM实现浏览器丰富的动态显示效果；使用XML和XSTL进行浏览器和服务器的数据交换和处理；使用JavaScript脚本语言对所有数据进行处理；使用XMLHttpRequest实现客户与服务器之间的异步请求和响应。
- AJAX技术在Web应用开发中有着很多应用。本章介绍了几种常见的应用，其中包括表单数据验证、动态加载列表框、创建工具提示、动态更新Web页面等。另外使用AJAX技术还可以实现读取响应首部、显示进度条、创建自动刷新页面、自动完成、访问Web服务等功能。