zjybushiren88888 的专栏

- 空间
- 博客
- 好友
- 相册
- 留 言

用户操作

[留言] [发消息] [加为好友]

订阅我的博客

zjybushiren88888 的公告

文章分类

- ASP.NET
- C#
- CSS
- Html
- MS SQLSERVER
- MSSQL 索引优化
- T-SQL QUERYING

存档

- 2009 年 10 月(2)
- 2009 年 09 月(9)
- 2009 年 08 月(4)
- 2009 年 07 月(24)

什么是 DOM 收藏

文档对象模型（Document Object Model，DOM）是一种用于HTML和XML文档的编程接口。它给文档提供了一种结构化的表示方法，可以改变文档的内容和呈现方式。我们最为关心的 是，DOM把网页和脚本以及其他的编程语言联系了起来。

脚本开发人员可以通过文档对象的属性、方法和事件来掌控、操纵和创建动态的网页 元素。每一个网页元素（一个HTML标签）都对应着一个对象（object，所谓“对象”，用白话说就是“东西”。object这个词在台湾通常翻译成 “物件”）。网页上的标签是一层层嵌套的，最外面的一层是<HTML>，文档对象模型也这样一层层嵌套着，但是通常被理解成一棵树的形状。树 根是window或document对象，相当于最外层的标签的外围，也就是整个文档。树根之下（这棵树的图通常是倒着画，就好像遗传谱系或者家谱那样。 树根就是唯一的共同祖先）是子一级的对象，子对象也有它自己的子对象，除了根对象以外，所有的对象都有自己的父对象，同一对象的子对象之间就是兄弟的关 系。

在这种由“父子兄弟”组成的“单性繁殖家族图谱树”框架结构中，每个网页元素都 可以被确切地定位。文档对象模型把整张网页组织成这样的一个树状的结构，树结构中的每一个元素都被视为一个节点（node）。包括JavaScript在内的各种编程语言都可以通过文档对象模型来访问和改变网页的各种细节。

万维网协会（World Wide Web Consortium，W3C）已经给文档对象模型制定了一系列标准，并且正在制定更多的相关标准。当代的浏览器除支持其中的一部分标准之外，还支持某些 早在W3C标准制定以前就流行了的历史既成的编程接口。也就是说现在浏览器使用的技术历史由来纷繁复杂，有些人们普遍使用的DOM技术并无标准可依。

我们将深入所有通用DOM的细节（包括IE浏览器中“与众不同”的某些技术），以全面掌握面向实践的技术。

DOM和JavaScript

我经常在QQ、MSN和email中被大家问到的“有关JavaScript” 的问题，95%其实是DOM的问题。人们在习惯上不爱说DOM，要么就说JavaScript，要么就扯到“Ajax”（一度火爆的“概念”，最近刚刚有 所降温，一如上世纪末的“DHTML”那样。对于这些热点词汇的产生，我个人感到非常欣慰，因为每一次都带来人们对JavaScript技术的热捧。下一 个热点词汇是什么？也许我们可以炮制一个也说不定……Pseudo-Mashup，如何？）。

我们用JavaScript对网页进行的所有操作都是通过DOM进行的。DOM 属于浏览器，而不是JavaScript语言规范里的规定的核心内容，所以如果你下载一个JavaScript语言的参考帮助文档来查的话，就连妇孺皆知 的document.write方法也找不到。

下面这段代码的作用是用一个提示框逐个显示网页中所有链接的网址，代码中被标为 红色的部分就是DOM。

```
var anchorTags = document.getElementsByTagName("a");
for (var i = 0; i < anchorTags.length ; i++)
{
   alert("Href of this a element is : " + anchorTags[i].href + "\n");
}
```

这样一来哪些是核心JavaScript，哪些是DOM，各自起什么作用，就可 以一目了然了。

```
var anchorTags =
```

创建了一个名为 anchorTags 的 JavaScript 变量。

```
document.getElementsByTagName("a")
```

Document 接口是 DOM1核心（DOM1 Core）规范 中定义的第一个接口，而 document 是实现了Document接口的一个宿主对象。document掌控着网页里的所有东西。

DOM1核心 为Document 接口定义了 getElementsByTagName() 方法。这个方法返回一个节点列表（NodeList） ，也就是一种DOM特有的包含节点的数组，包含了所有符合匹配参数条件的标签，按照在文档中出现的顺序排列。于是anchorTags变量现在就成了一个 节点列表。

```
;
```

分号是JavaScript里的语句结束符号。

```
for (var i = 0; i <
```

这是编程语言里典型的 “for循环”。声明了循环变量i，逐个处理anchorTags节点列表里的每一个节点。这也是JavaScript的东西。

```
anchorTags.length
```

DOM1 核心 定义了NodeList接口的 length 属性。这个属性返回一个整数，就是节点列表里包含的节点数目。说起来JavaScript 的数组也有一个 length属性。

```
; i++) {
```

典型的JavaScript变量增1运算。

```
alert(
```

alert() 是一个DOM方法，弹出一个提示框，显示传递给该方法的参数（字符串）。话说这个东西是通称 0级DOM（DOM level 0）或DOM0的一些历史既成的编程接口当中的一员。DOM0 是一套“被某些浏览器所支持”的编程接口（事实上，市场上不存在不支持DOM0的浏览器，只有在某些软件爱好者的收藏品中才能见得到），不属于任何DOM 标准规范。

"Href of this a element is : " +
一 个字符串字面量和一个字符串链接符。JavaScript 的东西。

anchorTags[i].href
href 是 DOM1 HTML 规范中定义的 HTMLAnchorElement 接口的属性，返回链接（<a>）元素的 href 属性的值。

在此我们用了像 anchorTags[i]这样的用法，这和 JavaScript 里访问第 i 个数组项的语法是一样的。语言中性（language-neutral，与具体语言无关）的所谓“DOM 方式”访问某个节 点列表中的一个项目的办法是使用在 NodeList 接口中定义的 item() 方法：anchorTags.item(1).href。但是大多数 JavaScript 实现程序都允许你使用这种简单的类似于数组的语法，而这也是大多 数人实际在用的方式。

+ "\n");
又一个字符串连接。在字符串的末尾加入一个回车符。

}
“for 循环”结束。

旧 一篇:sqlserver 常用函数/存储过程/数据库角色 | 新 一篇:ASP.NET 页面跳转几种方式

什么是 DOM

文 件对象模型（Document Object Model，DOM）是给 HTML 与 XML 文件使用的一组 API。它提供了文件的结构表述（representation），让你可以更动其中的内容及可见物。其本质是建立网页与 Script 或程序语言沟通的桥梁。

所有网页设计师可操作及建立文件的属性、方法及事件都以[对象]来展现（例如，document 就代表「文件本身」这个对象，table 对象则代表 HTML 的表格对象等等）。这些对象可以由当今大多数的浏览器以 Script 来取用。

DOM 最常被用以[与 JavaScript 沟通]，也就是说虽然程序以 JavaScript? 写成，但使用 DOM 来存取页面及其元素。无论如何，DOM 本身是设计为一种独立的程序语言，以一致的 API 存取文件的结构表述是以虽然本站的焦点放在 JavaScript? 上，但 DOM 其实可以与[任何程序语言]共同运作。

[全球信息网协会]（World Wide Web Consortium，W3C）建立了 [DOM] 的标准，称之为「W3C DOM」。在当今主要浏览器都已正确实作的情况下，W3C DOM 使强大、跨浏览器的应用程序成真。这是众网页设计师在 Netscape 4 与 MSIE 多不相容的时代从未梦想过的事情。

DOM的结 构：

在DOM 中，我们将代表XML 文件的程序设计对象，称为节点（nodes）。当Internet Explorer 5 处理被链接的XML 文件并储存于DOM 中时，它会为XML 文件的每一个基本组件建立一个节点。这些基本组件包括了元素、属性，与处理指令DOM 会使用不同形态的节点来代表不同形态的XML 组件。例如，元素是储存在Element 节点中，而属性则是储存在Attribute 节点中。表格1 列出了这些节点类型最重要的部分。

节点形态	节点对象所代表的XML 文件组件	节点名称（nodeName 对象属性）	节点的值（nodeValue 对象属性）
文 件（Document）	文件阶层中的根节点（代表整个XML 文件）	#document	Null
元 素（Element）	元素	元素形态名称(例如，BOOK)	null（包含在元素中的 （Element）任何字符数据，是位在一个或多个子文字节点中）
描述范 例attributes 该节点的所有非属性的子节点的NamedNodeMap 集合AttributeNode=Element.attributes.getNamedItem ("Binding");属 性（Attribute）属性 (以及其它的名值对，像处理指令中的名字与值)属性名称 (如Binding)属 性值（例如hardcover）处理指令（Processing	属于由这个节 点的父节点所代表的元素，属性及实体的文字。	#text	父XML 组件的文字

Instruction ）处 理指令（XML 宣告或自订的处理指令）处理指令的目标（例如 xml）除了目标之外整个处理指令的内容（例 如，Version ″1.0″）批注（Comment）批注#comment 在批注符号中的文字 CDATA 区段（CDATASection）CDATA 区段 #cdata-sectionCDATA 区段中的内容文件类型（Document Type）文件形态宣告出 现在 DOCTYPE 宣告中的根元素的名字(例如 INVENTORY)Null 实体（Entity）DTD 中的实体宣告实体名称（例如 image）null （实体值是位在子文字节点中）标签 DTD 中的标签宣告标签名称（例如 BMP）>null （卷标的系统 literal（Notation）是位在名为 SYSTEM 的 子 Attribute node 中） 上表中用来表示不同 XML 文件组件的基本节点形态。这些类型的每一个节点都是一个程序设计对象，提供了存取相关组件的属性与方法。		

你可以从节点中的 nodeName 属性获得每个节点的名称（详列于表格 9-1 中的第三栏）。这个名称是以字符#起始，代表那些未在文件中命名的 XML 组件节点的标准名称。（例如，在 XML 文件中的批注并未命名。因此，DOM 将使用标准名称 #comment。）其它节点的名称则是由指定到 XML 文件中相对应组件的名称衍生而来。（例如，代表形态 BOOK 元素的元素节点也可以命名为 BOOK。） 文字（Text） 你可以从节点的 nodeValue 属性取得每个节点的节点值（列于表格 9-1 中最后一栏）。如果 XML 组件拥有一个相关的值（例如，属性），该值将会被储存于节点的节点值中。如果 XML 组件并没有节点值（例如，元素），则 DOM 将会把节点值设成 null。在本章稍后，你将学到更多有关列于表格 9.1 中各种节点类型的相关知识。 DOM 会将 XML 文件的节点建构成树状的阶层结构，反映出 XML 文件本身的阶层结构。DOM 将会建立一个单一		

文件节点来表示整个 XML 文件，并将其视为阶层结构的根节点。注意，XML 元素的逻辑阶层结构，包含了整个 XML 文件，结构中的根节点，只是 DOM 节点的阶层结构的一个分枝。 每 个节点，就像可程序化的对象，提供了属性和方法，让你可以存取、显示、管理，和取得对应到 XML 组件上的信息。例如，nodeName 和 nodeValue 属性（表 1 所示）提供了元素的名称及内含值。 所有形态的节点共同分享一组公共的属性与方法。这些属性与方 法一般是设计来偕同节点一起运作。表格 2 列出了一些比较有用的共同属性。在本章稍后你将获得有关这些属性的更多信息及范例。 属性		
childNodes	该 节点的所有非属性的子节点的 NodeList 集合	FirstNode =Element.childNodes (0);
dataType	该 节点的数据类型（只适用于某些类型 Attribute 节	AttributeType =Attribute.dataType;

	点）	
firstChild	该 节点的第一个非属性的子节点	FirstChildNode =Element.firstChild;
lastChild	该 节点的最后一个非属性的子节点	LastChildNode =Element.lastChild;
nextSibling	与 本节点位于同一层级的后继前一节点	NextElement =Element.nextSibling;
nodeName	节 点的名称	ElementName =Element.nodeName;
nodeType	表 示该节点类型的数值码	NodeTypeCode =Node.nodeType;
nodeTypeString	包 含该节点类型的字符串，以小写字母撰写（例如，"element"或"attribute"）	NodeTypeString=Node.nodeTypeString;
nodeValue	该 节点的值（如果不含值则为 null）	AttributeValue =Attribute.nodeValue;
ownerDocument	包 含本节点的文件的根 Document 节点	Document =Node.ownerDocument;
parentNode	该 节点的父节点（并不适用于 Attribute 节点）	ParentElement =Element.parentNode;
previousSibling	与 本节点位于同一层级的先前节点	PreviousElement =Element.previousSibling;
text	该 节点与其后裔节点的全部文字内容	AllCharacterData =Element.text;
xml	该 节点与其后裔节点的全部 XML 内容	XMLContent =Element.xml;