

文档对象模型(DOM)及其应用

何 标^{1,2}

(1. 绍兴文理学院学科建设处, 浙江 绍兴 312000; 2. 上海同济大学, 上海 200092)

摘 要:主要介绍了文档对象模型(DOM)和 DOM 应用的实例, 主要包括在节点树中按照节点之间的关系来检索出各个节点, 通过 id 属性返回指定的对象。

关键词:DOM; XML; HTML; 节点

中图分类号:TP18

文献标识码:A

文章编号:1672-3198(2009)11-0291-02

1 引言

文档对象模型(Document Object Model, DOM)最初是 W3C 为了解决浏览器混战时代不同浏览器环境之间的差别而制定的模型标准, 主要是针对 Internet Explorer 和 Netscape Navigator。W3C 解释为:“文档对象模型(DOM)是一个能够让程序和脚本动态访问和更新文档内容、结构和样式的语言平台, 提供了标准的 XML 和 HTML 对象集, 并有一个标准的接口来访问并操作它们”。它使得程序员可以很便捷地访问 XML 和 HTML 页面上的标准组件, 如元素、样式表、脚本等并作相应的处理。

2 DOM

2.1 DOM 概述

DOM 是对 XML 或 HTML 文档进行应用开发、编程的应用程序接口(API)。作为 W3C 公布的一种跨平台、与语言无关的接口规范, DOM 提供了在不同环境和应用中的标准程序接口, 可以用任何语言实现。

DOM 采用对象模型和一系列的接口来描述 XML 或 HTML 文档的内容和结构, 即利用对象把文档模型化。这种对象模型实现的基本功能包括:

描述文档表示和操作的接口; 接口的属性和方法; 接口之间的关系以及互操作。

DOM 可对结构化的 XML 或 HTML 文档进行解析, 文档中的指令、元素、实体、属性等所有内容个体都用对象模型表示, 整个文档被看成是一个有结构的信息树, 而不是简单的文本流, 生成的对象模型就是树的节点, 对象同时包含了方法和属性。因此, 对文档的所有操作都是在对象树上的进行。在 DOM 中, 树中的一切都是对象, 不管是根节点还是实体的属性。

利用 DOM, 开发人员可以动态地创建 XML 或 HTML 文档, 遍历结构, 添加、修改、删除内容等。其面向对象的特性, 使人们在处理 XML 或 HTML 解析相关的事务时节省大量的精力, 是一种符合代码重用思想的强有力编程工具。

2.2 DOM 中的节点含义

DOM 将文档表示为一颗枝繁叶茂的家谱树, 如果把文档元素想象成家谱树上的各个节点的话, 可以用同样的记

号来描述文档结构模型, 从这种意义上讲, 将文档看成一棵“节点树”更为准确。

所谓节点(node), 表示某个网络中的一个连接点, 换句话说, 网络是节点和连线的集合。在 W3C DOM 中, 每个容器、独立的元素或文本块都被看做一个节点, 节点是 W3C DOM 的基本构建块。当一个容器包含另一个容器时, 对应的节点之间有父子关系。同时该节点树遵循 XML 或 HTML 的结构化本质, 如<html>元素包含<head>, 前者又包含<title>, 后者包含各种块元素等。

DOM 节点树中的节点有元素节点、文本节点和属性节点三种不同的类型, 下面在 HTML 文档中分别予以介绍。

2.2.1 元素节点(element node)

在 HTML 文档中, 各 HTML 元素如<body>、<p>、等构成文档结构模型的一个元素对象。在节点树中, 每个元素对象又构成了一个节点。元素可以包含其他的元素, 例如在下面的对象中:

```
<ul id="qindan">
  <li>乒乓球</li>
  <li>羽毛球</li>
  <li>足球</li>
</ul>
```

所有的列表元素都包含在无序清单元素内部。值得注意的是, <html>元素是节点树的根节点。

2.2.2 文本节点(text node)

在节点树中, 元素节点构成树的枝条, 而文本则构成树的叶子。如果一份文档完全由空白元素构成, 它将只有一个框架, 本身并不包含什么内容。没有内容的文档是没有价值的, 而绝大多数内容由文本提供。看下面的代码:

```
<p>欢迎来到<em>体育</em>世界</p>
```

包含“欢迎来到”、“体育”、“世界”3 个文本节点。在 HTML 中, 文本节点总是包含在元素节点的内部, 但并非所有的元素节点都包含或直接包含文本节点, 如“qindan”中, 元素节点并不包含任何文本节点, 而是包含着另外的元素节点, 后者包含着文本节点。所以说, 有的元素节点只是间接包含文本节点。

2.2.3 属性节点(attribute node)

HTML 文档中的元素或多或少都有一些属性,既便于准确、具体地描述相应的元素,又便于进行进一步的操作,如下所示:

```
<h1 class="sample">欢迎来到体育世界</h1>
<ul id="qindan">...</ul>
```

这里 class="sample"、id="qindan"都属于属性节点。因为所有的属性都是放在元素标签里,所以属性节点总是包含在元素节点中。

3 DOM 应用

3.1 在节点树中按照节点之间的关系来检索出各个节点

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.0//EN"
```

```
"http://www.w3.org/TR/REC-html140/strict.dtd"
>
```

```
<html>
<head>
<meta http-equiv=content-type content="text/html; charset=gb2312">
<title>sample page! </title>
</head>
<body>
<p id="p1">welcome to<B>DOM</B> world!
</p>
```

```
<script language="JavaScript" type="text/javascript">
<!--
var temp="";
if(node.nodeName!=null)
{
temp+="nodeName:"+node.nodeName+"\n";
}
else temp+="nodeName:null!\n";
if(node.nodeType!=null)
{
temp+="nodeType:"+node.nodeType+"\n";
}
else temp+="nodeType:null!\n";
if(node.nodeValue!=null)
{
temp+="nodeValue:"+node.nodeValue+"\n\n";
}
else temp+="nodeValue:null!\n\n";
return temp;
}
```

//处理并输出节点信息

//返回 id 属性值为 p1 的元素节点

```
var currentElement = document.getElementById("p1");
```

```
var msg=nodeStatus(currentElement);
//返回 p1 的第一个孩子,即文本节点"Welcome to"
currentElement=currentElement.firstChild;
msg+=nodeStatus(currentElement);
//返回文本节点"Welcome to"的下一个同父节点,即元素节点 B
```

```
currentElement=currentElement.nextSibling;
msg+=nodeStatus(currentElement);
//返回文本节点 B 的第一个孩子,即文本节点"DOM"
currentElement=currentElement.firstChild;
msg+=nodeStatus(currentElement);
//返回文本节点"DOM"的父节点,即元素节点 B
currentElement=currentElement.parentNode;
msg+=nodeStatus(currentElement);
//返回文本节点 B 同父节点,即文本节点"Welcome to"
currentElement=currentElement.previousSibling;
msg+=nodeStatus(currentElement);
//返回文本节点"Welcome to"的父节点,即元素节点 P
currentElement=currentElement.parentNode;
msg+=nodeStatus(currentElement);
//返回文本节点 P 的最后一个孩子,即文本节点"World!"
```

```
currentElement=currentElement.lastchild;
msg+=nodeStatus(currentElement);
alert(msg);
-->
</script>
</body>
</html>
```

程序运行后,弹出警告框,如果出现 null 表示某个节点对应的属性。

3.2 通过 id 属性返回指定对象

DOM 提供 getElementById() 方法,用于返回与制定 id 属性值的元素节点相对应的对象,对应着文档里一个特定的元素节点(元素对象)。下例是通过 id 属性返回指定对象的代码。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.0//EN"
```

```
"http://www.w3.org/TR/REC-html140/strict.dtd"
>
```

```
<html>
<head>
<meta http-equiv=content-type content="text/html; charset=gb2312">
<title>sample page! </title>
</head>
<body>
```

电子节能灯制造关键技术研究

林 澜

(温州电子技术研究所, 浙江 温州 325000)

摘 要:电子节能灯是一种节能减排的新型照明设备。介绍了电子节能灯在制造过程中的几项关键技术, 为相关工程技术人员提供参考。

关键词:电子节能灯; 制造; 关键技术

中图分类号: TS956.2

文献标识码: A

文章编号: 1672-3198(2009)11-0293-02

1 引言

节能减排是当前我国十分关注的问题。随着社会的发展和人们生活水平的提高, 照明用电量逐年增加, 照明在能源消耗和环境污染上的问题已引起人们的重视。在保证照明质量的前提下如何节约用电、减少污染、实现绿色照明已成为照明行业的发展主题。电子型自镇流荧光灯, 即一体化电子节能灯, 已越来越多地代替白炽灯泡而广泛用于家庭、商场、宾馆等各种室内照明场合。

2 电子节能灯的制造关键技术

2.1 节能灯元器件的选择

(1) 灯管是电子节能灯的重要部件。技术指标主要有光电参数, 其次有外形尺寸和外观以及机械强度等, 再次是低温启动性能和热态参数稳定特性等。光通量是否合格将直接关系到节能灯的能效等级能否达到标准的要求。光通维持率是产品申请“能源之星”认证时必须考核的指标, 要求节能灯燃点至有效寿命 40% 时, 其光通量不得小于额定值的 80%。此外还有相关色温、显色性、色坐标和色容差等指标, 也是应该重视的。尤其是色容差, 色容差超标往往会给予不合格的判定。至于荧光粉品质的分选, 可用验钞灯作对比检验。

(2) 整流二极管。

选择整流二极管时, 应注意以下参数。最大正向整流电流该参数与灯功率有关, 所选二极管的额定电流值至少应是交流母线中峰值电流的 3 倍以上。对大功率灯而言, 整流二极管不可直接并联使用, 因为二极管的热电阻有差异, 会使可靠性降低, 最好用一组电流大的二极管。

峰值反向截止电压, 因工作温度高, 一般要选用 1200V 的产品。滤波电解电容器, 电解电容因会受到高频脉动电流的冲击和工作温度升高, 选用高性价比的电解电容器就显得很重要。电容量及容差、额定电压、耐纹波电流、串联等效电阻(ESR)、允许温度等都是重要的技术指标, 质量的好坏直接关系到使用寿命。电解电容的耐纹波电流值应越大越好。如果电解电容的耐纹波电流值达不到线路要求, 则会严重影响其使用寿命, 选用时应特别注意。纹波电流流经 ESR, 会产生热量引起电解内部温度升高, 目前制造商对电容器在额定工作温度下工作的纹波电流的确定, 一般均遵循 5℃ 原则。电容量的选择与输入电流中的谐波含量和灯电流的波峰因数以及镇流器的效率有关。电解电容的标记温度, 必须大于实际工作温度, 并留出一定的差值。

(3) 功率晶体管。

</html>

4 结论

本文主要介绍了文档对象模型(DOM)和 DOM 应用的实例, DOM 是处理 XML 和 HTML 的标准 API 之一。虽然它占用内存大、速度慢, 并且冗长, 但它易用性强, 使用 DOM 时, 将把所有的 XML 文档信息都存于内存中, 并且遍历简单, 支持 XPath, 增强了易用性, 所以对于很多应用程序来说, 它仍然是最佳选择。

参考文献

- [1] 陈奇. DOM 的现状及其发展趋势[J]. 计算机工程, 2001, (10).
- [2] 张军, 王红玲. DOM 技术在制作 HTML 页面菜单中的应用[J]. 中国科技信息, 2005, (12).

```
<ul id="chases">
<li>beans</li>
<li>cheese</li>
<li>milk</li>
</ul>
<script language="JavaScript" type="text/javascript">
<!--
Document.write(typeof document.getElementById
("chases"));
-->
</script>
</body>
```