XML DOM

Nguyễn Hồng Phương
Email: phuong.nguyenhong@hust.vn
Site: http://is.hut.edu.vn/~phuongnh
Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

Nội dung

- Tổng quan XML DOM
- Thao tác với các nút
- Tham chiếu XML DOM

2

Tổng quan XML DOM

3

1. DOM là gì?

- Là một chuẩn W3C
- Định nghĩa chuẩn truy cập tài liệu
- DOM có 3 phần
  - Core DOM: mô hình chuẩn cho các tài liệu có cấu trúc
  - XML DOM: mô hình chuẩn cho các tài liệu XML
  - HTML DOM: mô hình chuẩn cho các tài liệu HTML
- DOM định nghĩa các đối tượng và thuộc tính của tất cả các phần tử tài liệu, và các phương thức truy cập vào chúng.

4

XML DOM

- Là mô hình đối tượng chuẩn cho XML
- Là một giao diện lập trình chuẩn cho XML
- Độc lập với nền và ngôn ngữ
- Định nghĩa một chuẩn cho truy cập và thao tác với tài liệu XML

5

2. Các nút DOM

- Mọi thứ trong một tài liệu XML đều là nút!
  - Toàn bộ tài liệu là nút tài liệu (document node)
  - Mỗi phần tử XML là nút phần tử (element node)
  - Văn bản trong các phần tử XML là nút văn bản (text node)
  - Mỗi thuộc tính là nút thuộc tính (attribute node)
  - Chú thích cũng là nút chú thích (comment node)

6

File book.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
```

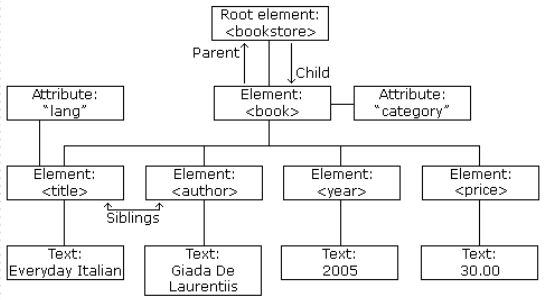

File book.xml (tiếp)

```
  <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book category="web" cover="paperback">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```


File book.xml (tiếp)

- Nút gốc là <bookstore>. Các nút khác trong tài liệu phải nằm trong nút gốc này.
- Dưới nút gốc có 4 nút <book>.
- Nút <book> đầu tiên có các nút con <title>, <author>, <year>, và <price>. Mỗi nút con này chứa các nút text "Everyday Italian", "Giada De Laurentiis", "2005", và "30.00"
- Chú ý: text của nút phần tử được chứa trong nút text.
  - Ví dụ: **<year>2005</year>**, nút phần tử <year> có một nút text có giá trị "2005". "2005" không phải giá trị của nút phần tử year.

3. Cây nút XML DOM

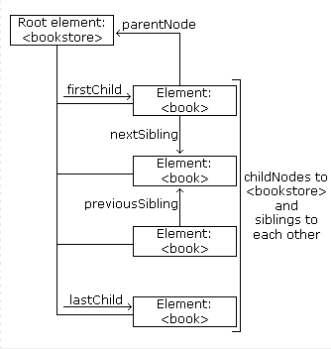


- XML DOM coi một tài liệu XML là một cấu trúc cây, gọi là cây nút.
- Có thể truy cập tới tất cả các nút của cây.
- Có thể thêm mới, sửa, xóa các phần tử.

Các nút cha, con, anh em

- Các nút trong cây có mối quan hệ phân cấp với các nút khác.
  - Nút đỉnh gọi là nút gốc
  - Mỗi nút (trừ nút gốc) đều có 1 nút cha.
  - Một nút có thể có không/một/nhiều nút con
  - Nút lá là nút không có nút con.
  - Các nút anh em là các nút có cùng nút cha.

Các nút cha, con, anh em



13

4. XML DOM Parser

- Hầu hết các trình duyệt đã được tích hợp sẵn XML Parser để đọc và thao tác với XML
- Parser biến đổi XML thành một đối tượng có thể truy cập JavaScript (XML DOM).
- XML DOM chứa các hàm để duyệt cây XML, truy cập, thêm và xóa các nút.
  - Trước khi thực hiện các thao tác với tài liệu XML, cần nạp nó vào đối tượng XML DOM

14

Nạp một tài liệu XML

```
<!DOCTYPE html>
<html>
<body>
<script>
if(window.XMLHttpRequest){
  xhttp=new XMLHttpRequest();
}
else{          // for IE 5/6
xhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xhttp.open("GET","books.xml",false);
xhttp.send();
xmlDoc=xhttp.responseXML;
document.write("XML document loaded into an XML DOM Object.");
</script>
</body>
</html>
```

15

Nạp một chuỗi XML

```
<!DOCTYPE html>
<html>
<body>
<script>
text="<bookstore><book>";
text=text+"<title>Everyday Italian</title>";
text=text+"<author>Giada De Laurentiis</author>";
text=text+"<year>2005</year>";
text=text+"</book></bookstore>";
if(window.DOMParser){
  parser=new DOMParser();
  xmlDoc=parser.parseFromString(text,"text/xml");
}else{      // Internet Explorer
  xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
  xmlDoc.async=false;
  xmlDoc.loadXML(text); }
document.write("XML string is loaded into an XML DOM Object");
</script>
</body>
</html>
```

16

- Vì lí do an toàn, trang web và file XML mà nó muốn nạp phải nằm trên cùng server.

17

5. Các hàm nạp XML DOM

- Hàm loadXMLDoc()
- Hàm loadXMLString()

18

Hàm loadXMLDoc()

```
function loadXMLDoc(dname){
if (window.XMLHttpRequest){
  xhttp=new XMLHttpRequest();
}
else{
  xhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xhttp.open("GET",dname,false);
xhttp.send();
return xhttp.responseXML;
}
```

- File loadxmldoc.js
- File loadxmldoc1.html

19

Hàm loadXMLString()

```
function loadXMLString(txt){
if (window.DOMParser){
  parser=new DOMParser();
  xmlDoc=parser.parseFromString(txt,"text/xml");
}
else{ // Internet Explorer
  xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
  xmlDoc.async=false;
  xmlDoc.loadXML(txt);
 }
return xmlDoc;
}
```

- File loadxmlstring.js
- File loadxmlstring1.html

20

6. Giao diện lập trình

- DOM mô hình XML như là một tập các đối tượng nút.
- Có thể viết các đoạn mã JavaScript, Java, C#,.... để truy cập vào các nút này.
- Giao diện lập trình: xác định thông qua một tập các thuộc tính và phương thức chuẩn.

21

Thuộc tính XML DOM

- Có một số thuộc tính đặc trưng
  - x.nodeName: tên của x
  - x.nodeValue: giá trị của x
  - x.parentNode: nút cha của x
  - x.childNodes: các nút con của x
  - x.attributes: các nút thuộc tính của x

22

Phương thức XML DOM

- x.getElementsByTagName(*name*): lấy về tất cả các phần tử mà tag có tên là name
- x.appendChild(*node*): thêm một nút con vào nút x
- x.removeChild(*node*): loại một nút con ra khỏi nút x

23

Ví dụ:

- Đoạn mã JavaScript lấy về đoạn text của phần tử <title> đầu tiên trong books.xml:

```
txt = xmlDoc.getElementsByTagName("title")[0].
childNodes[0].nodeValue
```

24

7. Truy cập vào các nút

- Có ba cách
  - sử dụng phương thức getElementsByTagName()
  - sử dụng vòng lặp để duyệt cây nút
  - định hướng trong cây, sử dụng mối quan hệ nút.

25

Phương thức getElementsByTagName()

- Trả về tất cả các phần tử có tên thẻ xác định.
- Cú pháp:
  - *node*.getElementsByTagName(*"tagname"*);
- Ví dụ:
  - trả về tất cả các phần tử <title> dưới nút x:
    - x.getElementsByTagName("title");
  - trả về tất cả các phần tử <title> của tài liệu XML
    - xmlDoc.getElementsByTagName("title");

26

Danh sách nút DOM

- Phương thức getElementsByTagName() trả về một danh sách nút – tức là một mảng các nút.
  - Ví dụ:
    - xmlDoc=loadXMLDoc("books.xml");
      x=xmlDoc.getElementsByTagName("title");
- Truy cập vào phần tử thông qua chỉ số. Chỉ số bắt đầu từ 0.
  - Ví dụ: truy cập phần tử <title> thứ 3
    - y=x[2];

27

Chiều dài danh sách nút DOM

- Sử dụng thuộc tính length của danh sách nút: cho biết số nút.
- Sử dụng vòng lặp để duyệt
- Ví dụ:
  - xmlDoc=loadXMLDoc("books.xml");
    x=xmlDoc.getElementsByTagName("title");
    for (i=0;i<x.length;i++){
    document.write(x[i].childNodes[0].nodeValue);
    document.write("
 ");
    }

28

Ví dụ: vidu1.html

```
<!DOCTYPE html>
<html>
<head>
<script src="loadxmldoc.js"></script>
</head>
<body>
<script>
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("title");
for (i=0;i<x.length;i++){
  document.write(x[i].childNodes[0].nodeValue);
  document.write("<br>");
}
</script>
</body>
</html>
```

29

Các kiểu nút

- Thuộc tính documentElement của tài liệu XML là nút gốc.
- Thuộc tính nodeName của nút là tên của nút.
- Thuộc tính nodeType của nút là kiểu của nút

vidu2.html

```
<!DOCTYPE html>
<html>
<head>
<script src="loadxmldoc.js"></script>
</head>
<body>
<script>
xmlDoc=loadXMLDoc("books.xml");
document.write(xmlDoc.documentElement.nodeName);
document.write("<br>");
document.write(xmlDoc.documentElement.nodeType);
</script>
</body>
</html>
```

30

Ví dụ: vidu3.html

```
<!DOCTYPE html>
<html>
<head>
<script src="loadxmldoc.js"></script>
</head>
<body>
<script>
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.documentElement.childNodes;
for (i=0;i<x.length;i++){
if (x[i].nodeType==1){//Process only element nodes (type 1)
  document.write(x[i].nodeName);
  document.write("<br />");
  }
}
</script>
</body>
</html>
```

31

Định hướng quan hệ nút

vidu4.html

```
<!DOCTYPE html>
<html>
<head>
<script src="loadxmldoc.js"></script>
</head>
<body>
<script>
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("book")[0].childNodes;
y=xmlDoc.getElementsByTagName("book")[0].firstChild;
for (i=0;i<x.length;i++){
if (y.nodeType==1){//Process only element nodes (type 1)
  document.write(y.nodeName + "<br>");
}
y=y.nextSibling;
}
</script>
</body>
</html>
```

32

8. Thông tin nút XML DOM

33

Các thuộc tính nút

- Mỗi nút là một đối tượng.
- Đối tượng có phương thức và thuộc tính.
- Ba thuộc tính quan trọng của một nút
  - nodeName
  - nodeValue
  - nodeType

34

Thuộc tính nodeName

- Cho biết tên của 1 nút
- Read-only
- nodeName của một nút phần tử chính là tên thẻ
- nodeName của một nút thuộc tính chính là tên thuộc tính
- nodeName của một nút text là #text
- nodeName của một nút tài liệu là #document

35

Ví dụ: vidu5.html

```
<!DOCTYPE html>
<html>
<head>
<script src="loadxmldoc.js"></script>
</head>
<body>

<script>
xmlDoc=loadXMLDoc("books.xml");

document.write(xmlDoc.documentElement.nodeName);
</script>
</body>
</html>
```

36

Thuộc tính nodeValue

- Cho biết giá trị của nút
- nodeValue của các nút phần tử là không xác định
- nodeValue của nút text chính là text
- nodeValue của nút thuộc tính chính là giá trị thuộc tính
- Lấy về giá trị của một phần tử

37

Ví dụ: vidu6.html

- Lấy về giá trị nút text của phần tử <title> đầu tiên

```
<!DOCTYPE html>
<html>
<head>
<script src="loadxmldoc.js"></script>
</head>
<body>

<script>
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];
txt=x.nodeValue;
document.write(txt);
</script>
</body>
</html>
```

38

Ví dụ: vidu7.html

- Thay đổi giá trị của phần tử <title> đầu tiên

```
<!DOCTYPE html>
<html>
<head>
<script src="loadxmldoc.js"></script>
</head>
<body>
<script>
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];
x.nodeValue="Easy Cooking";
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];
txt=x.nodeValue;
document.write(txt);
</script>
</body>
</html>
```

39

Thuộc tính nodeType

- Cho biết kiểu của nút
- Read-only
- Một số kiểu quan trọng:

Node type	NodeType
Element	1
Attribute	2
Text	3
Comment	8
Document	9

40

Ví dụ: vidu8.html

```
<!DOCTYPE html>
<html>
<head>
<script src="loadxmldoc.js"></script>
</head>
<body>

<script>
xmlDoc=loadXMLDoc("books.xml");
document.write(xmlDoc.documentElement.nodeName);
document.write("<br>");
document.write(xmlDoc.documentElement.nodeType);
</script>
</body>
</html>
```

41

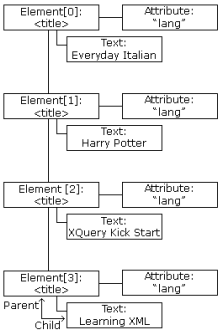
9. Danh sách các nút XML DOM

42

Danh sách nút DOM

- Khi sử dụng thuộc tính hay phương thức (như childNodes hoặc getElementsByTagName) sẽ nhận được đối tượng trả về là một danh sách nút, theo trình tự như trong tài liệu XML.
- Nút có thể được truy cập với chỉ số bắt đầu từ 0

- Danh sách nút <title> trong books.html




```
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("title");
// Nhận về giá trị text của phần tử <title>
// đầu tiên trong danh sách nút
txt=x[0].childNodes[0].nodeValue;
```


Chiều dài danh sách nút

- Khi một phần tử được thêm vào hoặc xóa đi, danh sách sẽ được tự động cập nhật.
- Thuộc tính length của một danh sách nút cho biết số nút có trong danh sách.

```
xmlDoc=loadXMLDoc("books.xml");

//the x variable will hold a node list
x=xmlDoc.getElementsByTagName('title');

for (i=0;i<x.length;i++)
{
document.write(x[i].childNodes[0].nodeValue);
document.write("<br />");
}
```

vidu9.html

Danh sách thuộc tính DOM

- Thuộc tính attributes của một nút phần tử trả về một danh sách các nút thuộc tính.
- Nếu một thuộc tính được thêm vào hay xóa đi, danh sách sẽ dược tự động cập nhật.

```
xmlDoc=loadXMLDoc("books.xml");

x=xmlDoc.getElementsByTagName("book")[0].attributes;

document.write(x.getNamedItem("category").nodeValue);
document.write("<br>" + x.length);
```

vidu10.html

10. Duyệt cây nút

Duyệt cây nút

```
<head>
<script src="loadxmlstring.js"></script>
</head>
<body>
<script>
text="<book>";
text=text+"<title>Everyday Italian</title>";
text=text+"<author>Giada De Laurentiis</author>";
text=text+"<year>2005</year>";
text=text+"</book>";

xmlDoc=loadXMLString(text);
```

Duyệt cây nút

```
// documentElement always represents the root node
x=xmlDoc.documentElement.childNodes;
for (i=0;i<x.length;i++)
  {
  document.write(x[i].nodeName);
  document.write(": ");
  document.write(x[i].childNodes[0].nodeValue);
  document.write("<br>");
  }
</script>
</body>
</html>
```

vidu11.html

49

11. Các trình duyệt DOM

50

Các trình duyệt DOM

- Hầu hết các trình duyệt hiện đại đều hỗ trợ các đặc tả W3C DOM.
- Có một chút khác biệt:
  - IE không coi các kí tự khoảng trắng/ kí tự xuống dòng là các nút text
  - Các trình duyệt khác thì có!

Hãy mở file vidu12.html bằng các trình duyệt khác nhau!

51

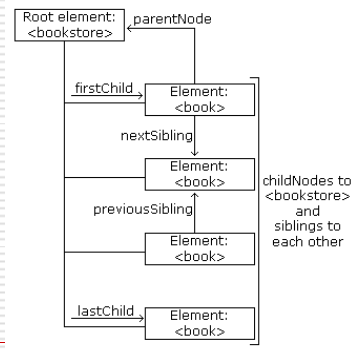
12. Định hướng trong cây

52

Định hướng trong cây

- Truy cập vào các nút thông qua mối quan hệ giữa các nút gọi là "định hướng" trong cây.
- Các thuộc tính của nút
  - parentNode
  - childNodes
  - firstChild
  - lastChild
  - nextSibling
  - previousSibling

53



54

Nút cha

- Tất cả các nút (trừ gốc) đều có chính xác một nút cha.

```
xmlDoc=loadXMLDoc("books.xml");

x=xmlDoc.getElementsByTagName("book")[0];
document.write(x.parentNode.nodeName);
```

vidu13.html

55

Tránh các nút text rỗng

- Một số trình duyệt (như FireFox) coi kí tự trắng và xuống dòng là nút text.
- Vấn đề khi sử dụng các thuộc tính: firstChild, lastChild, nextSibling, previousSibling.
- Để tránh truy cập vào phần tử text rỗng (kí tự trắng và kí tự xuống dòng giữa các nút phần tử), sử dụng hàm kiểm tra kiểu nút

```
function get_nextSibling(n){
y=n.nextSibling;
while (y.nodeType!=1){
  y=y.nextSibling;
}
return y;
}
```

56

Lấy về phần tử con đầu tiên

vidu14.html

```
<html>
<head>
<script src="loadxmldoc.js">
</script>
<script>
//check if the first node is an element node
function get_firstChild(n){
y=n.firstChild;
while (y.nodeType!=1)  {
  y=y.nextSibling;
  }
return y;
}
</script>
</head>
```

```
<body>
<script>
xmlDoc=loadXMLDoc("books.xml");

x=get_firstChild(xmlDoc.getElementsByT
agName("book")[0]);
document.write(x.nodeName);
</script>
</body>
</html>
```

57

- lastChild() vidu15.html
- nextSibling() vidu16.html
- previousSibling() vidu17.html

58

Thao tác với các nút

59

13. Lấy về các giá trị DOM

- Lấy về giá trị của một phần tử
- Lấy về giá trị của một thuộc tính

60

Lấy về giá trị của một phần tử

- Các nút phần tử không có giá trị text
- Text của một nút phần tử được lưu trữ trong nút con, gọi là nút text.
- => để lấy về text của nút phần tử, ta lấy giá trị của nút con (nút text).
- Phương thức getElementsByTagName() trả về một danh sách nút chứa tất cả các phần tử với tên thẻ theo tham số của phương thức này. Trật tự các phần tử theo trật tự xuất hiện của chúng ở trong tài liệu
- Thuộc tính childNodes trả về một danh sách các nút con
- Thuộc tính nodeValue trả về giá trị text của nút text


```
x=xmlDoc.getElementsByTagName("title")[0];
y=x.childNodes[0];
txt=y.nodeValue;
```

vidu18.html

Lấy về giá trị của một thuộc tính

- Trong DOM, các thuộc tính là các nút.
- Nút này có giá trị text.
- Để lấy giá trị text của thuộc tính, sử dụng
  - phương thức getAttribute() trả về một giá trị thuộc tính
  - thuộc tính nodeValue


```
xmlDoc=loadXMLDoc("books.xml");
txt=xmlDoc.getElementsByTagName("title")[0].getAttribute("lang");
document.write(txt);
```

vidu19.html

vidu20.html

- Phương thức getAttributeNode() trả về một nút thuộc tính.

```
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("title")[0].getAttributeNode("lang");
txt=x.nodeValue;
```

vidu21.html

vidu22.html

14. Thay đổi giá trị nút/thuộc tính

- Thay đổi giá trị nút: thuộc tính nodeValue
- Thay đổi giá trị thuộc tính: phương thức setAttribute()

Thay đổi giá trị của nút text

```
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];
x.nodeValue="Easy Cooking";
```

vidu23.html

Thay đổi giá trị của thuộc tính

```
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName('book');
x[0].setAttribute("category","food");
```

vidu24.html

```
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("book")[0]
y=x.getAttributeNode("category");
y.nodeValue="food";
```

vidu25.html

67

15. Loại bỏ nút trong DOM

- Sử dụng phương thức removeChild() để xóa một nút
- Sử dụng phương thức removeAttribute() để xóa một thuộc tính

68

- Xóa bỏ một nút phần tử

```
xmlDoc=loadXMLDoc("books.xml");
y=xmlDoc.getElementsByTagName("book")[0];
xmlDoc.documentElement.removeChild(y);
```

vidu26.html

- Xóa bỏ nút hiện tại

```
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("book")[0];
x.parentNode.removeChild(x);
```

vidu27.html

- Xóa bỏ nút text

```
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("title")[0];
y=x.childNodes[0];
x.removeChild(y);
```

vidu28.html

69

- Xóa text của nút text

```
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];
x.nodeValue="";
```

vidu29.html

- Loại bỏ nút thuộc tính sử dụng tên

```
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("book");
x[0].removeAttribute("category");
```

vidu30.html

- Loại bỏ các nút thuộc tính sử dụng tham số đối tượng

```
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName("book");
for (i=0;i<x.length;i++){
while (x[i].attributes.length>0)  {
  attnode=x[i].attributes[0];
  old_att=x[i].removeAttributeNode(attnode);
  }
}
```

vidu31.html

70

16. Thay thế một nút

- Thay thế một nút phần tử
- Thay thế dữ liệu trong mọt nút text
- Sử dụng thuộc tính nodeValue

71

Thay thế một nút phần tử

- Sử dụng hàm replaceChild()

```
xmlDoc=loadXMLDoc("books.xml");

x=xmlDoc.documentElement;

//create a book element, title element and a text node
newNode=xmlDoc.createElement("book");
newTitle=xmlDoc.createElement("title");
newText=xmlDoc.createTextNode("A Notebook");

//add the text node to the title node,
newTitle.appendChild(newText);
//add the title node to the book node
newNode.appendChild(newTitle);

y=xmlDoc.getElementsByTagName("book")[0]
//replace the first book node with the new node
x.replaceChild(newNode,y);
```

vidu32.html

72

Thay thế dữ liệu trong một nút text

- Sử dụng hàm replaceData(), có 3 tham số:
  - offset: vị trí kí tự đầu tiên sẽ thay thế, bắt đầu từ 0.
  - length: số kí tự cần thay thế.
  - string: chuỗi mới cần chèn vào

```
xmlDoc=loadXMLDoc("books.xml");

x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];

x.replaceData(0,8,"Easy");
```

vidu33.html

73

Sử dụng thuộc tính

- Có thể sử dụng thuộc tính nodeValue để thay thế dữ liệu trong nút text

```
xmlDoc=loadXMLDoc("books.xml");

x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];

x.nodeValue="Easy Italian";
```

vidu34.html

74

17. Tạo nút DOM

- Tạo một nút phần tử mới
- Tạo một nút thuộc tính mới
- Tạo một thuộc tính sử dụng setAttribute()
- Tạo một nút text
- Tạo một nút CDATA Section
- Tạo một nút chú thích

75

Tạo một nút phần tử mới

- Sử dụng phương thức createElement()

```
xmlDoc=loadXMLDoc("books.xml");

newel=xmlDoc.createElement("edition");

x=xmlDoc.getElementsByTagName("book")[0];
x.appendChild(newel);
```

vidu35.html

76

Tạo một nút thuộc tính mới

- Sử dụng phương thức createAttribute()

```
xmlDoc=loadXMLDoc("books.xml");

newatt=xmlDoc.createAttribute("edition");
newatt.nodeValue="first";

x=xmlDoc.getElementsByTagName("title");
x[0].setAttributeNode(newatt);
```

vidu36.html

77

Tạo nút thuộc tính sử dụng setAttribute()

- Phương thức setAttribute() tạo nút thuộc tính mới nếu nó chưa tồn tại.

```
xmlDoc=loadXMLDoc("books.xml");

x=xmlDoc.getElementsByTagName('book');
x[0].setAttribute("edition","first");
```

vidu37.html

78

Tạo một nút text

- Sử dụng phương thức createTextNode()

```
xmlDoc=loadXMLDoc("books.xml");

newel=xmlDoc.createElement("edition");
newtext=xmlDoc.createTextNode("first");
newel.appendChild(newtext);

x=xmlDoc.getElementsByTagName("book")[0];
x.appendChild(newel);
```

vidu38.html

79

Tạo nút CDATA Section

- Sử dụng phương thức createCDATASection()

```
xmlDoc=loadXMLDoc("books.xml");

newCDATA=xmlDoc.createCDATASection("Special Offer & Book Sale");

x=xmlDoc.getElementsByTagName("book")[0];
x.appendChild(newCDATA);
```

vidu39.html

80

Tạo nút comment

- Sử dụng phương thức createComment()

```
xmlDoc=loadXMLDoc("books.xml");

newComment=xmlDoc.createComment("Revised March 2008");

x=xmlDoc.getElementsByTagName("book")[0];
x.appendChild(newComment);
```

vidu40.html

81

18. Thêm nút

- Thêm một nút - appendChild()
- Chèn một nút – insertBefore()
- Thêm một thuộc tính mới
- Thêm text vào nút text - insertData()

82

Thêm một nút - appendChild()

- Dùng phương thức này để thêm một nút con vào nút hiện tại.

```
xmlDoc=loadXMLDoc("books.xml");

newel=xmlDoc.createElement("edition");

x=xmlDoc.getElementsByTagName("book")[0];
x.appendChild(newel);
```

vidu41.html

83

Chèn một nút – insertBefore()

- Sử dụng phương thức này để chèn một nút trước một nút con xác định.
- Phương thức này hữu ích khi quan tâm tới vị trí của nút thêm vào.
- x.insertBefore(newNode,null) và x.appendChild(newNode)

```
xmlDoc=loadXMLDoc("books.xml");

newNode=xmlDoc.createElement("book");

x=xmlDoc.documentElement;
y=xmlDoc.getElementsByTagName("book")[3];

x.insertBefore(newNode,y);
```

vidu42.html

84

Thêm một thuộc tính mới

- Không có addAttribute()!
- Sử dụng phương thức setAttribute()
  - sẽ tạo mới nếu chưa có thuộc tính
  - sẽ ghi đè nếu đã có thuộc tính

```
xmlDoc=loadXMLDoc("books.xml");

x=xmlDoc.getElementsByTagName('book');
x[0].setAttribute("edition","first");
```

vidu43.html

Thêm text vào nút text – insertData()

- insertData() thêm dữ liệu vào nút text đang có.
- Có 2 tham số:
  - offset: chỉ số bắt đầu chèn kí tự (tính từ 0)
  - string: chuỗi cần chèn

```
xmlDoc=loadXMLDoc("books.xml");

x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];

x.insertData(0,"Easy ");
```

vidu44.html

Phương thức cloneNode()

- Tạo ra một bản sao của một nút xác định.
- Có 1 tham số nhận giá trị true/false, cho biết nút sao có bao gồm các thuộc tính và các nút con của nút ban đầu hay không.

```
xmlDoc=loadXMLDoc("books.xml");
oldNode=xmlDoc.getElementsByTagName('book')[0];
newNode=oldNode.cloneNode(true);
xmlDoc.documentElement.appendChild(newNode);
//Output all titles
y=xmlDoc.getElementsByTagName("title");
for (i=0;i<y.length;i++)
{
document.write(y[i].childNodes[0].nodeValue);
document.write("
");
}
```

Kết quả:

Everyday Italian
Harry Potter
XQuery Kick Start
Learning XML
Everyday Italian

vidu45.html

19. Đối tượng XMLHttpRequest

- Đối tượng XMLHttpRequest được sử dụng để trao đổi dữ liệu với server
  - Cập nhật một trang web mà không cần tải lại trang web
  - Yêu cầu dữ liệu từ server sau khi trang đã được tải.
  - Nhận dữ liệu từ server sau khi trang đã được tải.
  - Gửi dữ liệu tới server (in the background)
- Nội dung trình bày:
  - Tạo một đối tượng XMLHttpRequest
  - Gửi một yêu cầu tới server

Tạo một đối tượng XMLHttpRequest

- Tất cả các tình duyệt hiện đại đều tích hợp đối tượng XMLHttpRequest
- Để tạo đối tượng XMLHttpRequest:
  - xmlhttp=new XMLHttpRequest();
- Phiên bản IE5, IE6 sử dụng đối tượng ActiveX:
  - xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

```
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
```

vidu46.html

Gửi yêu cầu tới server

- Sử dụng phương thức open(), send() của đối tượng XMLHttpRequest

```
xmlhttp.open("GET","xmlhttp_info.txt",true);
xmlhttp.send();
```

Phương thức	Mô tả
open(*method,url,async*)	method: kiểu yêu cầu là GET hoặc POST url: vị trí của file trên server async: true (không đồng bộ), false (đồng bộ)
send(*string*)	Gửi request off tới server string: chỉ sử dụng với yêu cầu POST

GET hay POST

- GET đơn giản và nhanh hơn POST, có thể được sử dụng trong hầu hết các trường hợp.
- Sử dụng POST khi:
  - Cập nhật 1 file hoặc CSDL trên server
  - Gửi một lượng lớn dữ liệu tới server (POST không giới hạn kích thước)
  - Gửi dữ liệu nhập của người dùng (có thể chứa các kí tự chưa biết), POST mạnh và an toàn hơn GET.

91

Đồng bộ hay không đồng bộ?

- Gửi yêu cầu không đồng bộ, JavaScript không phải đợi trả lời của server, mà có thể:
  - thực hiện các script khác trong khi đợi trả lời (response) của server.
  - thu nhận trả lời khi trả lời sẵn sàng

92

Async=true

```
xmlhttp.onreadystatechange=function()
 {
 if (xmlhttp.readyState==4 && xmlhttp.status==200)
   {
   document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
   }
 }
xmlhttp.open("GET","xmlhttp_info.txt",true);
xmlhttp.send();
```

vidu47.html

93

Async=false

- Khuyến cáo không nên sử dụng.
- JavaScript sẽ đợi đến khi trả lời của server sẵn sàng, nếu server bận hoặc chậm thì ứng dụng sẽ bị dừng hoặc treo.
- Chú ý: không viết hàm onreadystatechange

```
xmlhttp.open("GET","xmlhttp_info.txt",false);
xmlhttp.send();
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
```

vidu48.html

94

Server Response

- Để nhận đáp ứng từ server, sử dụng thuộc tính responseText hoặc responseXML của đối tượng XMLHttpRequest

Thuộc tính	Mô tả
responseText	Nhận dữ liệu đáp ứng như là một xâu
responseXML	Nhận dữ liệu đáp ứng như là dữ liệu XML

95

Thuộc tính responseText

- Nếu đáp ứng từ server không phải là XML thì sử dụng thuộc tính responseText

```
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
```

vidu49.html

96

Thuộc tính responseXML

- Nếu đáp ứng từ server là XML, để phân tích nó như là một đối tượng XML, ta sử dụng thuộc tính responseXML.

```
xmlDoc=xmlhttp.responseXML;
var txt="";
x=xmlDoc.getElementsByTagName("ARTIST");
for (i=0;i<x.length;i++)
  {
  txt=txt + x[i].childNodes[0].nodeValue + "
";
  }
document.getElementById("myDiv").innerHTML=txt;
```

vidu50.html

97

Sự kiện onreadystatechange

- Khi một yêu cầu được gửi tới server, chúng ta muốn thực hiện một số hành động dựa trên đáp ứng.
- Sự kiện onreadystatechange được kích hoạt mỗi khi readyState thay đổi.
- Thuộc tính readyState nắm trạng thái của XMLHttpRequest.

98

Sự kiện onreadystatechange (tiếp)

- Ba thuộc tính quan trọng của XMLHttpRequest

Thuộc tính	Mô tả
onreadystatechange	Lưu trữ một hàm/tên một hàm sẽ được tự động gọi khi thuộc tính readyState thay đổi
readyState	Nắm giữ trạng thái của đối tượng XMLHttpRequest, thay đổi từ 0 đến 4: 0: yêu cầu không được khởi tạo 1: kết nối server được thành lập 2: đã nhận yêu cầu 3: xử lý yêu cầu 4: yêu cầu đã hoàn tất và đáp ứng đã sẵn sàng
status	200: "OK" 404: Page not found

99

Sự kiện onreadystatechange (tiếp)

- Khi readyState là 4 và status là 200, đáp ứng sẵn sàng:

```
xmlhttp.onreadystatechange=function()
  {
 if (xmlhttp.readyState==4 && xmlhttp.status==200)
   {
   document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
   }
 }
```

vidu51.html

100

Phụ lục: Các kiểu nút

101

1. Các kiểu nút

Kiểu nút	Mô tả	Kiểu nút có thể là nút con
Document	Biểu diễn một văn bản hoàn chỉnh (nút gốc của cây DOM)	Element (tối đa chỉ có 1) ProcessingInstruction, Comment, DocumentType
DocumentFragment	Biểu diễn đối tượng tài liệu nắm giữ một phần tài liệu	Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference
DocumentType	Cung cấp một giao diện tới các thực thể đã định nghĩa trong tài liệu	
ProcessingInstruction	Biểu diễn một chỉ thị xử lý	

102

EntityReference	Biểu diễn một tham chiếu thực thể	Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference
Element	Biểu diễn một phần tử	Element, Text, Comment, ProcessingInstruction, CDATASection, EntityReference
Attr	Biểu diễn một thuộc tính	Text, EntityReference
Text	Biểu diễn nỗi dung text trong một phần tử hoặc một thuộc tính	

CDATASection	Biểu diễn CDATASection trong tài liệu (text không bị kiểm tra bởi trình kiểm ngữ)	
Comment	Biểu diễn một chú thích	
Entity	Biểu diễn một thực thể	Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference
Notation	Biểu diễn một notation đã khai báo trong DTD	

2. Giá trị trả về

Kiểu nút	Tên nút trả về	Giá trị nút trả về
Document	#document	null
DocumentFragment	#document fragment	null
DocumentType	doctype name	null
EntityReference	entity reference name	null
Element	element name	null
Attr	attribute name	attribute value

Kiểu nút	Tên nút trả về	Giá trị nút trả về
ProcessingInstruction	target	content of node
Comment	#comment	comment text
Text	#text	content of node
CDATASection	#cdata-section	content of node
Entity	entity name	null
Notation	notation name	null

3. Kiểu nút – các hằng

Kiểu nút	Hằng
1	ELEMENT_NODE
2	ATTRIBUTE_NODE
3	TEXT_NODE
4	CDATA_SECTION_NODE
5	ENTITY_REFERENCE_NODE
6	ENTITY_NODE

Kiểu nút	Hằng
7	PROCESSING_INSTRUCTION_NODE
8	COMMENT_NODE
9	DOCUMENT_NODE
10	DOCUMENT_TYPE_NODE
11	DOCUMENT_FRAGMENT_NODE
12	NOTATION_NODE

109

Lời hay ý đẹp

"Vì đời chỉ là một câu chuyện, nên điều cần thiết không phải là nó dài hay ngắn, mà là hay hay dở"

Seneca

110