Nội dung

* Cấu trúc cây của XML DOM
* Các thuộc tính và phương thức trong XML DOM
* Truy xuất tài liệu XML sử dụng DOM

Cấu trúc cây của XML DOM

1. Các node trong XML DOM

* Trong DOM, mọi thứ trong tài liệu XML là một nút (node):
  + Toàn bộ tài liệu là 1 nút tài liệu (Document node)
  + Mỗi phần tử XML là 1 nút phần tử (Element node)
  + Text trong các phần tử XML là các nút text (Text node)
  + Mỗi thuộc tính là 1 nút thuộc tính (Attribute node)
  + Các chú thích là nút chú thích (Comment node)

Ví dụ: Hãy xem tập tin XML sau (*books.xml*):



Giải thích:

- Nút gốc (root) trong file XML ở trên có tên là <bookstore>. Tất cả các nút khác trong tài liệu được chứa bởi nút <bookstore>.

- Nút gốc <bookstore> chứa 4 nút <book> (4 cuốn sách khác nhau).

- Nút <book> đầu tiên chứa 4 nút: <title>, <author>, <year>, và <price>; mỗi nút bao gồm 1 nút text, “Everyday Italian”, “Giada De Laurentiis”, “2005”, và “30.00”.

Lưu ý: Một lỗi phổ biến trong xử lý DOM là cho rằng nút phần tử chứa text. Tuy nhiên, text của nút phần tử được đặt trong nút text.

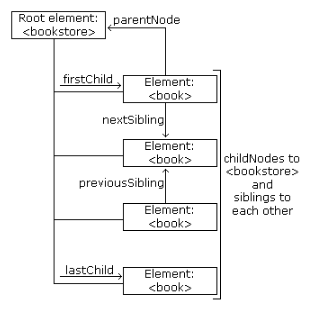
- Trong ví dụ này: <year>2005</year>, nút phần tử <year> chứa 1 nút text với giá trị là “2005”.

- “2015” không phải là giá trị của phần tử <year> !

2. Sơ đồ cây của XML DOM

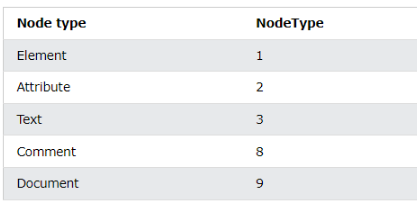
* XML DOM xem tài liệu XML như một cây cấu trúc. Cây cấu trúc này được gọi là node-tree.
* Tất cả các nút có thể được truy cập thông qua cây. Nội dung của chúng có thể được thay đổi, được xóa và các phần tử mới có thể được tạo.
* Node-tree thể hiện một tập các nút và sự kết nối giữa chúng. Cây bắt đầu với nút gốc (root node) và các nhánh tới các nút text ở cấp độ thấp nhất của cây.
* Các nút trong cây có mối quan hệ phân cấp với nhau.
* Các từ parent, child và sibling được sử dụng để mô tả mối quan hệ giữa các nút trong cây. Nút cha có các nút con. Các nút con có cùng cấp độ (cùng cha) được gọi là siblings (anh chị em ruột).
  + Trong 1 node tree, nút trên đỉnh được gọi là nút gốc (root).
  + Mọi nút ngoại trừ nút gốc đều chắc chắn có 1 nút cha.
  + Một nút có thể có nhiều nút con.
  + Nút lá là nút không có con.
  + Siblings (các nút anh/chị em ruột) là các nút cùng cha.

Ví dụ: Hình sau giải thích một phần của node-tree và mối quan hệ giữa chúng:



Một số phương thức và thuộc tính của XML DOM

* Trong XML DOM, mỗi nút là một đối tượng.
* Các đối tượng có các phương thức và thuộc tính, có thể được truy cập và thao tác bằng JavaScript.
* Các thuộc tính nút quan trọng là:  
  + **documentElement**: xác định nút gốc của tài liệu XML.
  + **nodeName**: trả về tên của một nút.
  + **nodeValue**: trả về giá trị của một nút (trả về giá trị văn bản của nút văn bản).
  + **nodeType**: xác định loại nút. Bảng sau liệt kê các giá trị của NodeType và kiểu nút tương ứng:



* + **childNodes**: trả về danh sách các nút con của một phần tử.
* Các phương thức là:
  + **getElementById**: trả về tất cả các phần tử có ID được chỉ định.
  + **getElementsByTagName**: trả về một danh sách nút gồm tất cả các phần tử, với tên thẻ đã chỉ định, theo cùng thứ tự như chúng xuất hiện trong tài liệu nguồn..
  + **getElementsByName**: trả về tất cả các phần tử có tên được chỉ định.
  + **getAttribute**: trả về giá trị của một thuộc tính.
  + **getAttributeNode**: trả về một node thuộc tính.

Ví dụ:



    Kết quả:

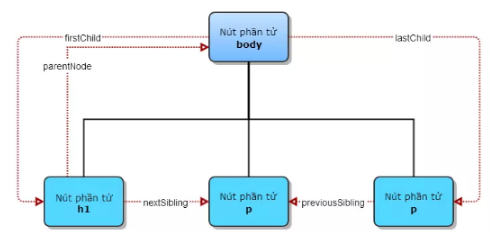


Truy xuất tài liệu XML sử dụng DOM

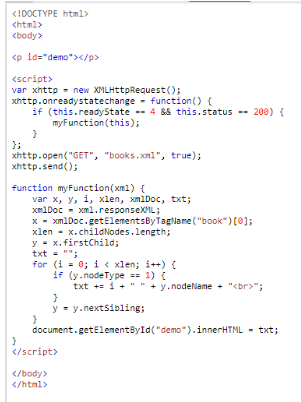
**1. Truy xuất gián tiếp**

* Thao tác truy xuất các nút gián tiếp theo vị trí trên cây DOM còn được gọi là duyệt cây DOM (DOM Tree traversal)
* Mỗi nút trên cây DOM đều có 6 thuộc tính quan hệ, từ đó ta có thể truy xuất gián tiếp theo vị trí của nút:
  + **Node.parentNode**: tham chiếu đến nút cha của nút hiện tại, và nút cha này là duy nhất cho mỗi nút. Do đó, nếu  cần tìm nguồn gốc sâu xa của 1 nút, ta phải nối thuộc tình nhiều lần, ví dụ Node.parentNode.parentNode.
  + **Node.childNodes**: tham chiếu đến các nút con trực tiếp của nút hiện tại, và kết quả là 1 mảng các đối tượng. Lưu ý rằng, các nút con không bị phân biệt bởi loại nút, nên kết quả mảng trả về có thể bao gồm nhiều loại nút khác nhau.
  + **Node.firstChild**: tham chiếu đến nút con đầu tiên của nút hiện tại, và tương đương với việc gọi Node.childNodes[0].
  + **Node.lastChild**: tham chiếu đến nút con cuối cùng của nút hiện tại, và tương đương với việc gọi Node.childNodes[Element.childNodes.length-1].
  + **Node.nextSibling**: tham chiếu đến nút anh em nằm liền kề sau với nút hiện tại.
  + **Node.previousSibling**: tham chiếu đến nút anh em nằm liền kề trước với nút hiện tại.

Ví dụ:



Ví dụ sau điều hướng cây nút bằng cách sử dụng các mối quan hệ nút:



Kết quả:



Giải thích:

1. Giả sử ta đã tải tài liệu "books.xml" vào xmlDoc

2. Lấy các nút con của phần tử <book> đầu tiên

3. Đặt biến "y" thành nút con đầu tiên của phần tử <book> đầu tiên

4. Đối với mỗi nút con (bắt đầu với nút con đầu tiên "y"):

5. Kiểm tra loại nút. Nếu loại nút là "1" thì đó là nút phần tử

6. Xuất tên của nút nếu đó là nút phần tử

7. Đặt biến "y" thành nút anh em tiếp theo và chạy lại vòng lặp.

**2. Truy xuất trực tiếp**

* Truy xuất trực tiếp nhanh hơn, và đơn giản hơn khi ta không cần phải biết nhiều về quan hệ và vị trí của nút. Có 3 phương thức để truy xuất trực tiếp được hỗ trợ ở mọi trình duyệt:
  + **document.getElementById('id\_cần\_tìm')**
  + **document.getElementsByTagName('div')**
  + **document.getElementsByName('tên\_cần\_tìm')**
* Phương thức getElementsByTagName() trả về tất cả các phần tử có tên thẻ được chỉ định.

Cú pháp:



    Ví dụ sau trả về tất cả các phần tử <title> bên dưới phần tử x:



    Ví dụ sau đây trả về tất cả các phần tử <title> trong tài liệu XML, trong đó xmlDoc chính là nút tài liệu (document node):  


* Phương thức getElementsByTagName() trả về một danh sách nút, chính là một mảng các nút.

    Ví dụ:



    Các phần tử <title> trong x có thể được truy cập bằng số chỉ mục (index). Ví dụ, để truy cập phần tử <title> thứ 3, ta có thể viết: y = x[2];

    Ví dụ sau lấy giá trị text node của phần tử <title> đầu tiên:



    Kết quả:



Truy xuất giá trị một nút

**1. Truy xuất giá trị một nút từ file XML**

* Tài liệu **books.html** sau kết hợp Javascript đọc file "books.xml" thành *xmlDoc* và truy xuất giá trị văn bản của phần tử <title> đầu tiên trong *books.xml*:

<!DOCTYPE html>  
<html>  
<body>  
  
    <p id="demo"></p>  
  
    <script>  
        var xhttp = new XMLHttpRequest();  
        xhttp.onreadystatechange = function() {  
            if (this.readyState == 4 && this.status == 200) {  
                myFunction(this);  
            }  
        };  
        xhttp.open("GET", "books.xml", true);  
        xhttp.send();  
  
        function myFunction(xml) {  
            var xmlDoc = xml.responseXML;  
            document.getElementById("demo").innerHTML = xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;  
        }  
    </script>  
  
</body>  
</html>

Kết quả:



Giải thích:

* xmlDoc - đối tượng XML DOM được tạo bởi trình phân tích cú pháp.
* getElementsByTagName("title")[0] - lấy phần tử <title> đầu tiên
* childNodes[0] - phần tử con đầu tiên của phần tử <title> (nút văn bản)
* nodeValue - giá trị của nút (là văn bản)

**2. Truy xuất giá trị một nút từ chuỗi XML**

* Ví dụ sau nạp một chuỗi văn bản thành một đối tượng XML DOM và trích xuất các thông tin XML với JavaScript:

<html>

    <head><title></title></head>

    <body>

         <p id="demo"></p>

        <script>

            var text, parser, xmlDoc;

            text = "<bookstore><book>" +

                    "<title>Everyday Italian</title>" +

                    "<author>Giada De Laurentiis</author>" +

                    "<year>2005</year>" +

                    "</book></bookstore>";

            parser = new DOMParser();

            xmlDoc = parser.parseFromString(text,"text/xml");

          document.getElementById("demo").innerHTML = xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;

        </script>

    </body>

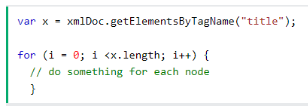
</html>

Kết quả:



Truy xuất một danh sách node

* Thuộc tính length xác định độ dài của danh sách nút (số lượng nút).
* Ta có thể lặp qua danh sách nút bằng cách sử dụng thuộc tính length. Ví dụ:



* Ví dụ sau duyệt và in ra toàn bộ nội dung các phần tử <title> trong tài liệu "books.xml":



    Kết quả:  
    