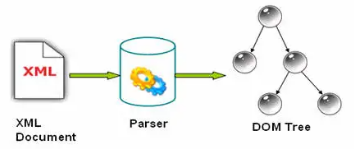
Nội dung

* Giới thiệu DOM
* Định nghĩa các mô hình của DOM
* Mô tả cấu trúc cây của một tài liệu XML
* Tạo một đối tượng tài liệu XML
* Thao tác trên một tài liệu XML sử dụng DOM

Giới thiệu DOM

* Ứng với mỗi cấu trúc tài liệu XML phải có một cách thức để truy xuất nên khó phát triển ứng dụng => Cần xây dựng một thư viện tổng quát sử dụng cho mọi tài liệu => DOM được sử dụng để thao tác lên tất cả các tài liệu XML.
* DOM (Document Object Model), mô hình đối tượng tài liệu, là một giao diện lập trình ứng dụng (API). Thông thường DOM có dạng một cây cấu trúc dữ liệu, được dùng để truy cập và thao tác các tài liệu dạng HTML và XML: "*The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*



* Mô hình DOM độc lập với hệ điều hành và dựa theo kỹ thuật lập trình hướng tượng để mô tả tài liệu.
* DOM không có giới hạn về cấu trúc dữ liệu của một tài liệu.
* DOM có nhiều phiên bản: DOM Level 1, DOM Level 2, DOM Level 3 (draft).

    -  Level 0: Bao gồm mọi chi tiết DOM theo quy định của từng nhà phát triển riêng biệt tồn tại trước khi DOM Level 1 ra đời.

    -  Level 1: Cách duyệt qua tài liệu (cấu trúc cây) của DOM (HTML và XML) và thao tác nội dung bao gồm cả việc thêm thành phần. Các thành phần riêng cho HTML cũng được đưa vào.

    -  Level 2: Hỗ trợ không gian tên XML, khung nhìn có lọc và các sự kiện DOM.

    -  Level 3: Nhằm nâng cao các khả năng của DOM với việc cho phép nạp, xử lý và lưu lại mô hình cho nội dung tài liệu.

Định nghĩa các mô hình của DOM

**1. CORE DOM**

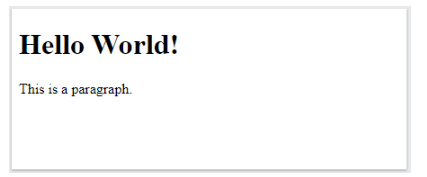
* CORE DOM là mô hình chuẩn cho các tài liệu có cấu trúc.

**2. HTML DOM**

* HTML DOM xác định cách thức chuẩn để thao tác và truy xuất tài liệu HTML. Kết quả hiển thị tài liệu HTML theo cấu trúc cây (tree-structure).
  + Ví dụ 1: Sử dụng HTML DOM thay đổi nội dung của phần tử <h1> từ "This is a Heading" thành "Hello World!"



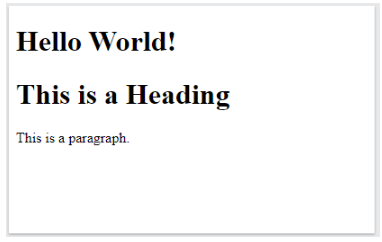
    Kết quả:



* + Ví dụ 2: Sử dụng HTML DOM thay đổi nội dung của phần tử <h1> đầu tiên từ "This is a Heading" thành "Hello World!"



    Kết quả:



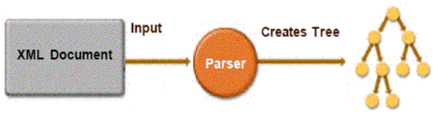
**3. XML DOM**

* XML DOM là một tiêu chuẩn của W3C, xác định cách thức thao tác và truy xuất tài liệu XML. Kết quả hiển thị tài liệu XML theo cấu trúc cây.
  + Ví dụ: Đoạn mã JavaScript sau truy xuất giá trị văn bản của phần tử <title> đầu tiên trong tài liệu XML:

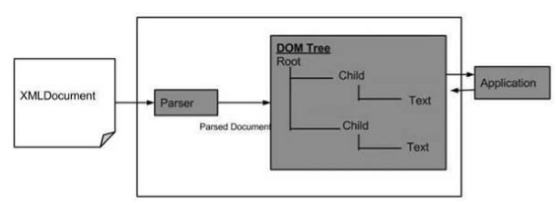


Mô tả cấu trúc cây của tài liệu XML

* **Quy trình phân tích tài liệu XML của DOM**:



* + Một tài liệu XML sau khi được phân tích bởi mô hình DOM thì một cây sẽ được tạo ra trong bộ nhớ mang thông tin của tài liệu đó.
  + XML DOM cũng cung cấp API cho phép nhà phát triển thêm, chỉnh sửa, di chuyển hoặc xóa các nút tại bất kỳ điểm nào trên cây để tạo ứng dụng.
  + Sau đây là sơ đồ cho cấu trúc DOM, mô tả quá trình phân tích cú pháp đánh giá một tài liệu XML dưới dạng cấu trúc DOM bằng cách duyệt qua từng nút:



* Theo mô hình DOM, tài liệu XML là một cây bao gồm tập hợp các nút (node). Nội dung những nút này có thể chứa phần tử (*element*), dữ liệu (*text*), thuộc tính (*attribute*),… và các nút con khác. Có tất cả 12 loại nút trong DOM:



* **Các Node thông dụng trong XML DOM**:

+ **Document Node (Nút tài liệu)**: Cấu trúc tài liệu XML hoàn chỉnh là một nút tài liệu.

+ **Element Node (Nút phần tử)**: Mỗi phần tử XML là một nút phần tử. Đây cũng là loại nút duy nhất có thể có thuộc tính.

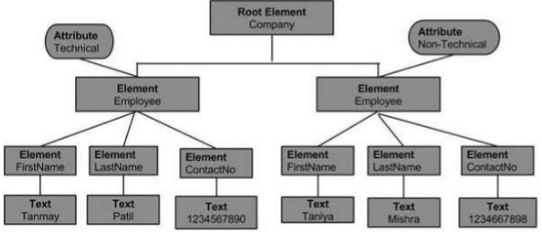
+ **Attribute Node (Nút thuộc tính)**: Mỗi thuộc tính được coi là một nút thuộc tính. Nó chứa thông tin về một nút phần tử, nhưng không thực sự được coi là nút con của phần tử.

+ **Text Node (Nút văn bản)**: Các văn bản tài liệu được coi là nút văn bản. Nó có thể bao gồm nhiều thông tin hơn hoặc chỉ chứa khoảng trắng.

* Ví dụ: Với tài liệu **note.xml** sau:



Mô hình Đối tượng Tài liệu (DOM) của tài liệu XML ở trên là:



Giải thích:

+ Nút trên cùng của cây được gọi là gốc. Nút gốc là <Company> lần lượt chứa hai nút <Employee>. Các nút này được gọi là các nút con.

+ Nút con <Employee> của nút gốc <Company>, lần lượt bao gồm nút con của chính nó (<FirstName>, <LastName>, <ContactNo>).

+ Hai nút con <Employee> có các giá trị thuộc tính *technical* và *non-technical*, được gọi là các nút thuộc tính.

+ Văn bản trong mỗi nút được gọi là nút văn bản.

**Giao diện lập trình**:

* DOM mô hình hóa XML dưới dạng một tập hợp các đối tượng nút. Các nút có thể được truy cập bằng JavaScript hoặc các ngôn ngữ lập trình khác (C#, VB.NET,...). Trong chủ đề này chúng ta sử dụng JavaScript để minh họa.
* Giao diện lập trình cho DOM được xác định bởi một tập hợp các thuộc tính và phương thức chuẩn.
* **Các thuộc tính đặc trưng của DOM** (giả sử x là một đối tượng node):
  + 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
* **Một số phương thức của DOM**(giả sử x là một đối tượng node):
  + x.getElementsByTagName(name) - lấy tất cả các phần tử có tên thẻ được chỉ định
  + x.appendChild(node) - chèn một nút con vào x
  + x.removeChild(node) - xóa nút con khỏi x