MỤC LỤC

CHƯƠNG 1. TỔNG QUAN VỀ XML 4

1.1. Giới thiệu 4

1.2. Lợi ích của XML. 5

1.3. Các thành phần của tài liệu XML. 9

1.5. Bộ phân tích XML. 11

1.7. Kết chương 15

CHƯƠNG 2. TẠO TÀI LIỆU XML HỢP KHUÔNG DẠNG 17

2.1. Tài liệu XML hợp khuôn dạng 17

2.2. Các thành phần của tài liệu XML 17

2.2.1. Khai báo XML 20

2.2.2. Chỉ thị xử lý 20

2.2.3. Chú thích 20

2.2.4. Thẻ và các phần tử 21

2.2.5. Thuộc tính.....23

2.3. Xây dựng cấu trúc tài liệu hợp khuôn dạng 27

2.4. Phân đoạn CDATA 28

2.5. Không gian tên (namespace) của XML 29

2.8. Kết chương. 31

CHƯƠNG 3. ĐỊNH NGHĨA KIỂU DỮ LIỆU VÀ KIỂM TRA TÍNH HỢP LỆ CỦA XML 32

3.1. Định nghĩa kiểu tư liệu 32

3.2. Cấu trúc của một DTD 34

3.2.1. Khai báo phần tử trong DTD 34

3.2.2. Khai báo thuộc tính 36

3.2.3. Chú thích 44

3.3. Sử dụng DTD bên trong và bên ngoài tài liệu XML. 44

3.3.1. Khai báo DTD bên trong 44

3.3.2. Khai báo DTD bên ngoài 45

3.4. thực thể 46

3.4.1. Thực thể tổng quát nội 50

3.4.2. Thực thể tổng quát ngoại *52*
3.4.3. Thực thể tham số nội *55*
3.4.4. Thực thể tham số ngoại *56*
CHƯƠNG 4. LƯỢC ĐỒ XML **58**
4.1. Giới thiệu **58**
4.2. So sánh DTD và lược đồ XML **58**
4.3. Các kiểu dữ liệu **59**
4.3.1. Kiểu định nghĩa sẵn *59*
4.3.2. Kiểu đơn giản *60*
4.3.3. Kiểu phức hợp *62*
4.4. Định nghĩa phần tử **67**
4.5. Các phần tử thường dùng trong lược đồ. **68**
CHƯƠNG 5. XPATH và XSL **70**
5.1. Xpath **70**
5.1.1. Giới thiệu *70*
5.1.2. Đường dẫn tuyệt đối *71*
5.1.3. Đường dẫn tương đối *72*
5.1.4. Chọn các phần tử bằng ký tự đại diện *72*
5.1.5. Chọn các phần tử theo điều kiện *72*
5.1.6. Một số hàm thường dùng *73*
5.1.7. Một số toán tử thường dùng *74*
5.2. XSL (eXtensible style sheet) **75**
5.2.1 XSL là gì? *75*
5.2.2 Qui tắc chung *75*
5.2. 3 Một số phần tử(element) thường dùng của XSL *76*
5.2.4. Cách áp dụng Style Sheet vào tài liệu XML *86*
CHƯƠNG 6. DOM VÀ XỬ LÝ XML VỚI JAVASCRIPT **90**
6.1. Mô hình DOM **90**
6.2. Cấu trúc cây của một tài liệu XML **91**
6.3. Các đối tượng trong một tài liệu XML **92**
6.4. Tải một tài liệu XML bằng JavaScript **92**

6.4.1. Nạp dữ liệu XML vào trình duyệt với lớp đối tượng Microsoft.XMLDOM .93
6.4.2. Sử dụng phần tử nạp dữ liệu XML 96
6.4.3. Truy xuất các nút dữ liệu dựa vào tên 99
6.4.4. Truy xuất giá trị thuộc tính 102
TÀI LIỆU THAM KHẢO.....104

CHƯƠNG 1. TỔNG QUAN VỀ XML

1.1. Giới thiệu

Ngôn ngữ định dạng XML (Extensible Markup Language) là ngôn ngữ được định nghĩa bởi tổ chức mạng toàn cầu W3C (World Wide Web Consortium); đây là một tổ chức quốc tế định ra các chuẩn của Web và Internet.

XML là một ngôn ngữ tổng quát dung định nghĩa dữ liệu thong qua các thẻ. Bạn có thể tùy ý định nghĩa mọi thẻ. Trong HTML các thẻ được định nghĩa và quy định trước trong khi XML thì không. Dựa vào một số quy tắc, thế là XML tự tồn tại và phát triển tự thân nó thành các ngôn ngữ định nghĩa khác.

Thực sự thì HTML và XML quan hệ rất gần gũi với nhau; cả hai đều dựa trên chuẩn ngôn ngữ định dạng tổng quát SGML(standard Generalized Markup Language).SGML là ngôn ngữ định dạng rất tổng quát, bao hàm nhiều khả năng to lớn. Bởi vì chúng ta có thể làm được rất nhiều thứ với SGML, cho nên tuy tổng quát nhưng SGML rất khó học và thực tế thì SGMLít được dùng cho một mục đích chung nào cả.. Trong khi đó, XML là tập cpn của SGML nhưng lại dễ dùng, dễ sử dụng ở mức độ tổng quát. Có thể tham khảo thêm về quan hệ của XML và SGML tại địa chỉ www.w3.org/TR/NOTE - sgml-xml.

XML trông ra sao?

Dưới đây là ví dụ về XML chuyển tải trang HTML :

```
<?xml version = “1.0” encoding = “ UTF – 8”?>
<DOCUMENT>
        <GREETING>
                Hello From XML
        </GREETING>
<MESSAGE>
                Welcome to the wild and woolly world of XML
</MESSAGE>
</DOCUMENT>
```

Ví dụ trên thực hiện công việc như sau:

Chúng ta bắt đầu bằng thẻ khai báo chỉ thị xử lý XML

<?xml version = “1.0” encoding = “ UTF – 8”?>

Tất cả các chỉ thị xử lý XML bắt đầu bằng< ? và kết thúc bằng ?> ; theo đó ta chỉ thị rằng ngôn ngữ XML hiện tài liệu sử dụng thoe phiên bản 1.0 và mã hóa ký tự theo UTF – 8 (phiên bản thu gọn 8 – bít của mã Unicode). Tiếp đến ta tạo thẻ mới <DOCUMENT> ; tuy nhiên bạn có thể tạo bất kỳ thẻ nào không riêng gì tên DOCUMENT. Tên đặt cho thẻ có thể bắt đầu bằng ký tự gạch chân, ký số, ký tự, dấu chấm, dấu gạch nối nhưng không được có khoảng trắng. Trong XML thẻ luôn bắt đầu bằng ký tự < và kết thúc bằng ký tự >.

Ngôn ngữ định dạng (Markup languages)

Ngôn ngữ định dạng là tất cả những gì dùng để mô tả nội dung một tài liệu. Đó là cách nội dung của tài liệu được diễn dịch. Ngôn ngữ định dạng mà mọi người quen nhất là HTML. HTML là ngôn ngữ cho phép tạo các trang web.

XML và HTML rất giống nhau, cả hai đều dựa trên chuẩn ngôn ngữ định dạng tổng quát SGML (Standard Geeneralized Marup Language). SGML là ngôn ngữ định dạng tổng quát bao hàm nhiều khả năng to lớn. Bởi vì SGML làm được rất nhiều thứ, cho nên tuy tổng quát nhưng khó học và thực thế thì SGML ít được dùng. Trong khi đó XML là tập con của SGML nhưng lại dễ dùng, dễ sử dụng ở mức tổng quát.

Ngôn ngữ HTML không đủ để biểu diễn các thông tin cho mục đích riêng. Ví dụ, chúng ta muốn xây dựng một mô hình xe máy và chúng ta muốn trao đổi những đặc tả về chiếc tàu với các đồng nghiệp? HTML không hề chứa thẻ định nghĩa với các tên như <SINHVIEN>, <COLOR>, ... hay những thẻ mà muốn mô tả về xe máy. Chúng ta cần thêm các thẻ để mô tả thông tin về xe máy.

1.2. Lợi ích của XML.

XML trở lên phổ biến có rất nhiều nguyên do và chúng ta xem xét các nguyên do này trong phần tổng quan dưới đây. Xml cần gì cho chúng ta hiện tại và tương lai?

Điểm quan trọng nhất là XML cho phép dễ dàng xử lý, chuyển tải và trao đổi dữ liệu giữa rất nhiều ứng dụng và tài liệu người dùng với các định dạng khác nhau

Dễ dàng trao đổi dữ liệu: Trong XML**,** dữ liệu và định dạng được lưu ở dạng text và có thể dễ dàng cấu hình cũng như thay đổi chúng bằng các trình soạn thảo thông thường. Nếu thích chúng ta có thể dùng một trình soạn thảo XML chuyên nghiệp để soạn thảo nội dung XML. Tuy nhiên trường hợp đơn giản hơn là chúng ta có

thể dùng ngay những trình soạn thảo kiểu notepad trong Windows để chỉnh sửa file XML. Dữ liệu và các thẻ trong XML cũng không mã hóa theo một thuật giải đòi hỏi bản quyền nào cả....

Thêm vào đó XML đã được chuẩn hóa. Mọi người đều có thể hiểu và dùng được nó.

Tùy biến ngôn ngữ định dạng: Chúng ta có thể tạo ra các ngôn ngữ định dạng tùy biến dựa trên XML, đây là một trong những khả năng mạnh nhất của XML . Hàng trăm ngôn ngữ định dạng chuyên dụng dựa trên XML đã rađời như:

+ Ngôn ngữ văn phòng về kỹ thuật nghiệm vụ ngân hàng.

+ Trao đổi dữ liệu tài chính.

+ Hệ thống thanh toán qua internet của nghiệp vụ ngân hàng

+ Định dạng trao đổi viễn thông.

+ Thư viện kinh doanh thông dụng.

+ Khởi đầu XML kinh doanh điện tử.

+ Ngôn ngữ định dạng dữ liệu sản phẩm.

+ Giao thức trao đổi thông tin tài chính.

Dữ liệu tự mô tả:Dữ liệu trong tài liệu XML tự mô tả nội dung và ý nghĩa của nó. Ví dụ:

```
<?xml version = “1.0” encoding = “ UTF – 8”?>
<DOCUMENT>
        <GREETING>
                Hello From XML
        </GREETING>
<MESSAGE>
                Welcome to the wild and woolly world of XML
</MESSAGE>
</DOCUMENT>
```

Chỉ dựa vào tên thẻ được đặt cho các phần tử XML, chúng ta có thể hình daung ra nội dung và cách dữ liệu muốn thể hiện: Tài liệu này chứa thông tin là một câu chào cùng với thông điệp là một chuỗi văn bản

Dữ liệu có cấu trúc và tích hợp:Với trang tài liệu HTML nếu bạn viết sai một thẻ mà trình duyệt không hiểu, trình duyệt sẽ bỏ qua và hiển thị các thẻ hợp lệ khác. Với XML thì không. Tài liệu được kiểm tra rất nghiêm ngặt; trình duyệt không chấp nhận tài liệu XML viết sai cúa pháp và nó sẽ không phân tích tiếp tài liệu XML cho đến khi tài liệu XML hoàn toàn đúng khuôn dạng và cú pháp. Bạn phải tự bắt lỗi tài liệu thay cho trình duyệt. Bằng cách này việc xử lý tài liệu XML của trình duyệt sẽ nhanh và hiệu quả hơn.

Trình duyệt kiểm tra tài liệu XML bằng hai cách chính: Kiểm tra tính hợp khuôn dạng của tài liệu XML và kiểm tra tính hợp lệ của tài liệu XML

Tài liệu XML hợp khuôn dạng (well-form)

Mặc dù bạn có thể đặt ra bao nhiêu thẻ cũng được, nhưng mỗi tài liệu XML cần phải theo một số qui luật để được xem là **Well-Formed**. Nếu một tài liệu XML không Well-Formed thì coi như không sử dụng đuợc, không có chương trình xử lý nào sẽ làm việc với dữ liệu bên trong file đó. Do đó một tài liệu XML cần phải theo đúng các qui luật sau đây:

- Tài liệu XML phải bắt đầu bằng câu khai báo XML: <?xml version="1.0"?>
- Mỗi phần tử phải nằm giữa một cặp thẻ.

```
<GREETING>Hello from XML</GREETING>
```

- Nếu thẻ nào không chứa gì ở giữa thì phải chấm dứt bằng "/>", thí dụ như <BR/> hay <HR/>.
- Một tài liệu XML phải có một thành phần duy nhất chứa tất cả các thành phần khác. Đó là gốc của cây biểu diễn tài liệu XML (ví dụ: <DOCUMENT>).
- Các cặp thẻ không được xen kẻ nhau, ví dụ:

```
<NAME>John Stanmore<ADDRESS>25 King Street</NAME></ADDRESS>
```

là bất hợp lệ vì <ADDRESS> nằm trong cặp thẻ name).

Và thêm một vài qui luật về cách dùng các mẫu tự đặc biệt. Ngoài ra các cặp thẻ phải đánh vần đúng y như nhau kể cả chữ hoa, chữ thường, (thí dụ: <STUDent> và </STUDENT> là bất hợp lệ) và tất cả giá trị các thuộc tính đều phải nằm giữa hai ngoặc kép (thí dụ: standalone=YES là bất hợp lệ, phải dùng standalone="yes" mới được.)

Tài liệu XML hợp lệ (valid Document)

Hầu hết các trình duyệt trước hết kiểm tra xem tài liệu XML của bạn có hợp khuôn dạng không. Tiếp theo một vài trình duyệt có thẻ kiểm tra thêm tính hợp lệ của tài liệu.

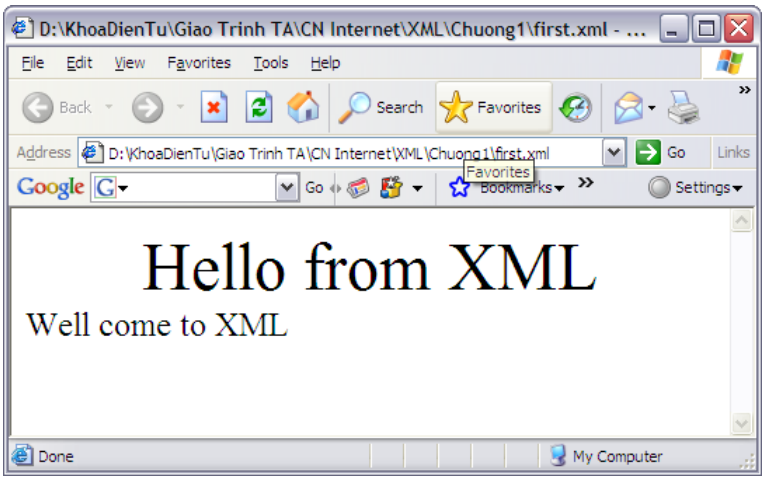
Một tài liệu XML được gọi là hợp lệ khi nó là tài liệu được kết hợp với định nghĩa kiểu tư liệu DTD (DOCUMENT TYPE DEFINITION) và tuân theo chuẩn DTD. DTD có thể được chứa trong một file tách biệt hoặc chứa ngay trong chính tài liệu. DTD sử dụng thẻ <!DOCTYPE>. Dưới đây là một ví dụ về định nghĩa kiểu tư liệu DTD được chúng ta thêm vào file tài liệu XML ở phần trên.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="greeting.css"?>
<!DOCTYPE DOCUMENT[
<!ELEMENT DOCUMENT (GREETING, MESSAGE)>
<!ELEMENT GREETING (#PCDATA)>
<!ELEMENT MESSAGE (#PCDATA)>
]>
<DOCUMENT>
        <GREETING>Hello from XML</GREETING>
        <MESSAGES>Well come to XML</MESSAGES>
</DOCUMENT>
```

Nội dung file greeting.css:

```
GREETING{
        display: block;
        font-SIZE:36pt;
        color:#FF000;
        text-align:center}
MESSAGES{
        display: block;
        font-SIZE:18pt;
        color:#00000;}
```

Chúng ta thấy DTD chỉ định có hai thẻ <GREETING>, <MESSAGES> được đặt nằm trong thẻ <DOCUMENT>. Định nghĩa cho thấy <DOCUMENT> là phần tử gốc. Và <GREETING>, <MESSAGES> là các phần tử mang theo văn bản. Nội dung tài liệu greeting.xml khi hiển thị trên IE như sau:



Hình 1 Nội dung tài liệu greeting.xml khi hiển thị trên IE

1.3. Các thành phần của tài liệu XML.

Một file tài liệu XML có phần mở rộng là xml, ví dụ này ta đặt tên file là greeting.xml.

Dưới đây là ví dụ chuyển tải file HTML trong ví dụ trên:

```
<?xml version="1.0" encoding="UTF-8"?>
<DOCUMENT>
        <GREETING>Hello from XML</GREETING>
        <MESSAGES>Well come to XML</MESSAGES>
</DOCUMENT>
```

Chúng ta bắt đầu bằng thẻ khai báo chỉ thị xử lý XML

```
<?xml version="1.0" encoding="UTF-8"?>
```

Tất cả các chỉ thị XML bắt đầu bằng <? và kết thúc bằng ?>. Tiếp theo ta chỉ định ngôn ngữ XML hiện tài liệu sử dụng theo phiên bản 1.0 và mã hóa ký tự theo UTF-8 (phiên bản thu gọn 8-bít của bảng mã unicode).

Tiếp đến ta tạo ra thẻ mới mang tên <DOCUMENT>, có thể sử dụng bất kỳ tên gì để đặt tên thẻ. Tên đặt cho thẻ có thể bắt đầu bằng ký tự ghạch chân (_), ký tự số, ký tự

(a, b, c, …), dấu chấm, dấu ghách nối nhưng không được có khoảng trắng. Trong XML thẻ luôn bắt đầu bằng < và kết thúc bằng >.

Tài liệu XML được hình thành từ các phần tử XML. Tương tự HTML bạn tạo phần tử XML khởi đầu, ví dụ <DOCUMENT> tiếp đến là dữ liệu của các phần tử nếu có, như dữ liệu văn bản hay các phần tử XML khác cuối cùng là thẻ đóng có tên trùng với thẻ mở. Tuy nhiên thẻ đóng phải bắt đầu bằng </ (ví dụ </DOCUMENT>). Cần có thẻ đóng và thẻ mở cho toàn bộ tài liệu, phần tử này gọi là phần tử gốc. Ví dụ DOCUMENT được xem là phần tử gốc của tài liệu.

```
<?xml version="1.0" encoding="UTF-8"?>
<DOCUMENT>
...............
</DOCUMENT>
```

Bên trong phần tử gốc ta thêm vào thành phần thẻ khác mang tên <GREETING>. Thẻ này có chứa dữ liệu sau phần tên thẻ. Dữ liệu của thẻ là "Hello from XML".

```
<?xml version="1.0" encoding="UTF-8"?>
<DOCUMENT>
<GREETING>Hello from XML</GREETING>
.
.
</DOCUMENT>
```

Kế đến ta thực hiện thêm một thẻ khác vào phần tử gốc. Thẻ mới thêm vào mang tên <MESSAGES>, tên này do ta tự đặt. Thẻ này mang theo dữ liệu thể hiện là chuỗi văn bản như sau:

```
<?xml version="1.0" encoding="UTF-8"?>
<DOCUMENT>
        <GREETING>Hello from XML</GREETING>
        <MESSAGES>Well come to XML</MESSAGES>
</DOCUMENT>
```

Cuối cùng là phần tử gốc </DOCUMENT>.