
Tạo tài liệu XML

TS. Phạm Thị Thu Thúy
Khoa CNTT – Trường ĐH Nha Trang
thuthuy@ntu.edu.vn

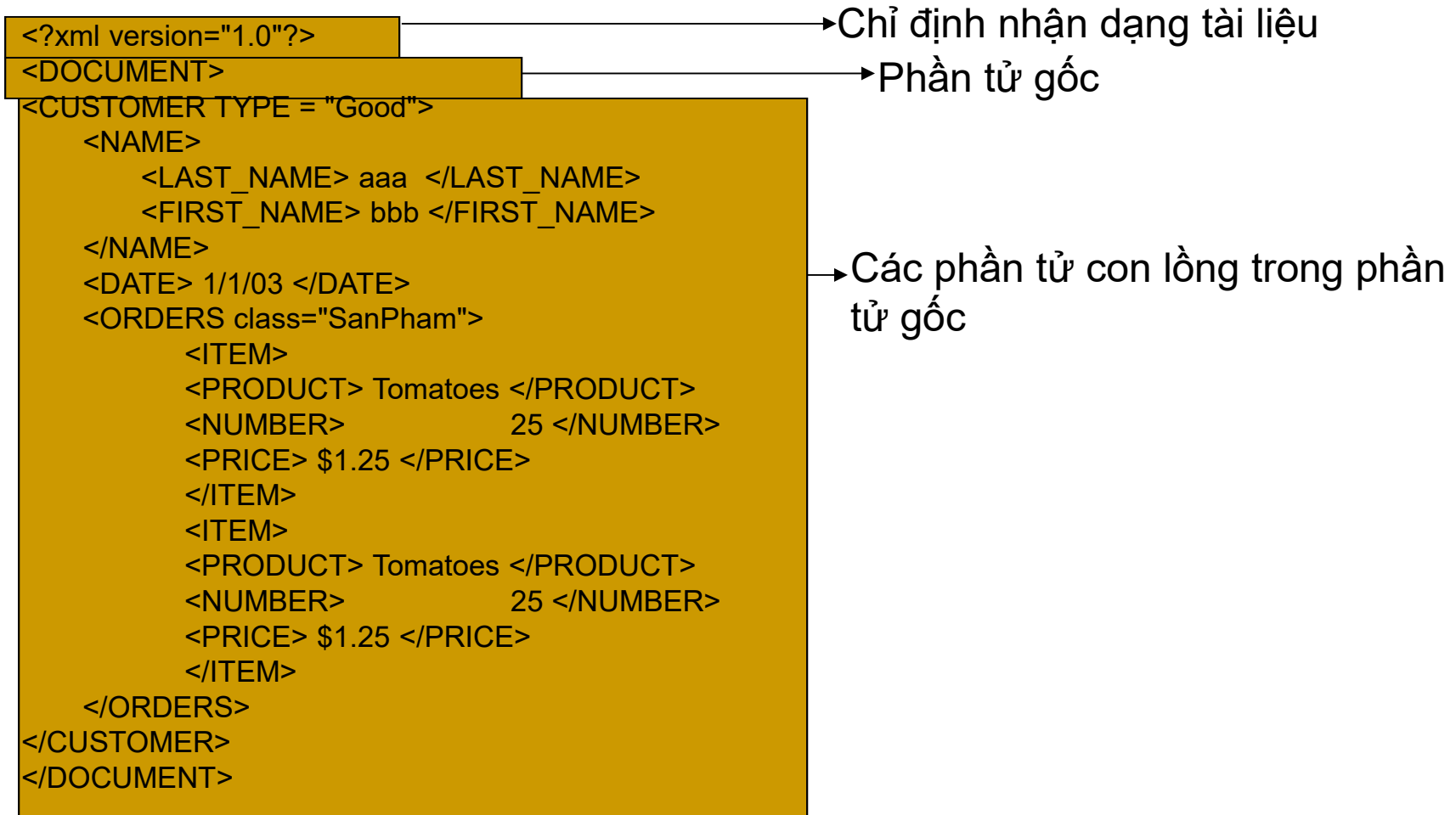
Ví dụ

```
<?xml version="1.0"?>
<DOCUMENT>
<CUSTOMER TYPE = "Good">
  <NAME>
    <LAST_NAME> aaa </LAST_NAME>
    <FIRST_NAME> bbb </FIRST_NAME>
  </NAME>
  <DATE> 1/1/03 </DATE>
  <ORDERS class="SanPham">
    <ITEM>
      <PRODUCT> Tomatoes </PRODUCT>
      <NUMBER>      25 </NUMBER>
      <PRICE> $1.25 </PRICE>
    </ITEM>
    <ITEM>
      <PRODUCT> Tomatoes </PRODUCT>
      <NUMBER>      25 </NUMBER>
      <PRICE> $1.25 </PRICE>
    </ITEM>
  </ORDERS>
</CUSTOMER>
</DOCUMENT>
```

Xây dựng tài liệu XML hợp khuôn dạng

- Một tài liệu XML hợp khuôn dạng:
 - Chỉ định dấu hiệu nhận dạng thông tin về nội dung tài liệu.
`<?xml version="1.0"?>`
 - Có phần tử gốc root.
 - Tạo các phần tử con lồng nhau trong phần tử gốc.
 - Một tài liệu XML có thể có nhiều phần. Các phần được gọi là một thực thể(entity).
 - Một thực thể có thể tham chiếu đến một thực thể khác. Khi đó thực thể tham chiếu sẽ được đưa vào tài liệu.
-

Xây dựng tài liệu XML hợp khuôn dạng



Khoảng trắng

- Các kí tự spacebar, backspace, Tab, kí tự xuống dòng đều được xem là khoảng trắng đối với trình phân tích XML.
- Ví dụ

```
<?xml version="1.0"?>
<CUSTOMER TYPE = "Good">
  <NAME>
    aaa
  </NAME>
  <ORDERS>
    <PRODUCT>
      Tomatoes
    </PRODUCT>
    <NUMBER>
      6
    <NUMBER>
  </ORDERS>
</CUSTOMER>
```

```
<?xml version="1.0"?>
<CUSTOMER TYPE = "Good">
  <NAME> aaa </NAME>
  <ORDERS> <PRODUCT> Tomatoes </PRODUCT>
    <NUMBER> 6 <NUMBER>
  </ORDERS>
</CUSTOMER>
```

Định dạng và dữ liệu kí tự

- Định dạng bao gồm:
 - Thẻ bắt đầu.
 - Thẻ kết thúc
 - Các phần tử thẻ rỗng
 - Các tham chiếu thực thể
 - Các tham chiếu kí tự
 - Lời chú thích
 - Phân đoạn CDATA khai báo kiểu tài liệu và chỉ thị xử lý.
 - Các dữ liệu còn lại trong tài liệu không được định dạng thì được xem là dữ liệu kí tự
-

Phần khởi đầu tài liệu XML

- Khai báo XML
 - Lời chú thích về tài liệu
 - Chỉ thị xử lý
 - Định nghĩa DTD (DTD nội).
-

Phần khởi đầu tài liệu XML

- Khai báo XML

- Khai báo phiên bản

- <?xml version="1.0"?>

- Khai báo mã hóa.

- Mặc định là bản mã UTF-8.

- Mã Unicode, USC-2, USC-4.

- Khai báo thực thể độc lập (standalone)

- Yes: nếu không tham chiếu đến các thực thể khác

- No: ngược lại.

- Ví dụ: <?xml version="1.0" encoding="UTF-8" standalone="yes"?>

Phần khởi đầu tài liệu XML

- Chú thích
 - Bắt đầu bằng <!--
 - Kết thúc bằng -->
 - Ví dụ: <!-- bắt đầu -->
 - Quy tắc:
 - Không được đặt trước khai báo
 - Không đặt vào bên trong phần định dạng
 - Không được dùng – vào bên trong dòng chú thích.
-

Phần khởi đầu tài liệu XML

- Chỉ dẫn cho bộ phân tích cách xử lý tài liệu XML.
- Chỉ thị bắt đầu bằng <? và kết thúc bằng ?>
- Ví dụ: chỉ thị yêu cầu bộ phân tích kết hợp dữ liệu của XML với bảng định kiểu XSLT như sau:

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="vd.xsl" ?>

Thẻ và các phần tử (element)

- Thẻ bắt đầu bằng < và kết thúc bằng >
 - Tên thẻ
 - Bắt đầu phải bằng kí tự, hoặc dấu _ hoặc dấu :
 - Kí tự kế tiếp có thể là kí tự, kí số, gạch chân, gạch nối, dấu chấm, dấu :
 - Không được dùng khoảng trắng.
-

Thẻ và phần tử (element)

■ Phần tử rỗng:

- ❑ Phần tử chỉ có một thẻ duy nhất: phần tử rỗng không kèm theo dữ liệu.
- ❑ Ví dụ: trong html thẻ là một thẻ rỗng
- ❑ Khi sử dụng thẻ rỗng, bộ phân tích không tìm đến thẻ đóng để xác minh tính hợp lệ

■ Phần tử gốc (root element):

- ❑ Tài liệu XML được coi là hợp khuôn dạng khi có một phần tử gốc duy nhất
 - ❑ Phần tử gốc chứa các phần tử và các cặp thẻ khác trong tài liệu.
-

Nội dung phần tử

1. Phần tử có thể bao gồm một văn bản đơn giản:

`<first>John</first>`

Nội dung phần tử được gọi là Parsed
Character Data (PCDATA)

2. Phần tử có thể bao gồm các phần tử khác

`<name>`

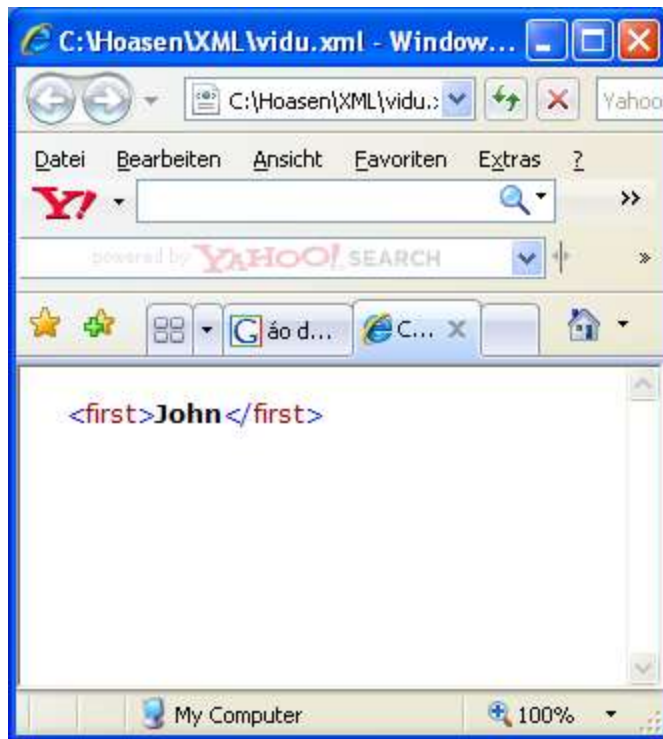
`<first>John</first>`

`<last>Doe</last>`

`</name>`

Được miêu tả theo hệ phân cấp

Phần tử (Element)



- **<first>** là một thẻ gán đầu
- **</first>** là một thẻ gán cuối
- **<first>John</first>** là một phần tử
- Văn bản giữa thẻ gán đầu và thẻ gán cuối của một thành được gọi là nội dung thành phần

Nội dung thành phần

Thành phần có thể chứa cả văn bản lẫn các phần tử khác

`<section>`

Text

`<subsection>... </subsection>`

Text

`</section>`

Như vậy ta nói chúng có **nội dung hỗn hợp**
(mixed content)

Nội dung thành phần

4. Phần tử rỗng:

<name>

<first>John</first>

<middle></middle>

<last>Doe</last>

</name>

Chú ý: Sự khác biệt so với phần tử không tồn tại

Phần tử rỗng <tag-name></tag-name> có thể được viết tắt là <tag-name/>

<name>

<first>John</first>

<middle/>

<last>Doe</last>

</name>

PCDATA

- Đối với XML có vài ký tự như `<` và `&` không thể gộp trong PCDATA
- Có 2 cách mà có thể khắc phục:
 - Thoát các ký tự
 - Bao văn bản trong một đoạn CDATA

Thoát các ký tự

- Thay ký tự ``<`` bằng `<` và ký tự ``&`` bằng `&`
- `<` và `&` được xem là tham chiếu thực thể (entity references)
- Các thực thể sau đây được định nghĩa trong XML:
 - `&` cho ``&``
 - `<` cho ``<``
 - `>` cho ``>``
 - `&apos` cho ``'``
 - `"` cho ``"``

CDATA

- Nếu có khá nhiều ký tự `<` và `&` cần thoát thì sử dụng các đoạn CDATA tốt hơn
- Dùng các đoạn CDATA có thể báo cho trình phân ngữ XML không phân ngữ văn bản, nhưng cứ để tất cả đi cho đến khi nó dừng đến cuối đoạn
- Mọi thứ bắt đầu sau `<![CDATA[` và kết thúc tại `]]>` đều được bỏ qua bằng trình phân ngữ

```
<comparison>  
    <![CDATA[6 is < 7 & 7 > 6] ]>  
</comparison>
```

Quy tắc cho các thành phần

- Mọi thẻ gán đầu phải có một thẻ gán cuối so khớp
- Các thẻ gán không thể phủ chồng
- Các tư liệu XML chỉ có thể có một thành phần gốc
- Các tên thành phần phải tuân thủ các quy ước đặt tên XML
- XML có phân biệt chữ hoa chữ thường
- XML sẽ giữ khoảng trắng trong văn bản

Mọi thẻ gán đầu phải có một thẻ gán cuối

- Trong XML cần có thẻ gán cuối và phải so khớp chính xác với thẻ gán đầu
- Quy tắc này không bắt buộc trong HTML

```
<html>
<body>
<P>Here is some text in an HTML paragraph
<BR>
Here is some more text in the same paragraph
<P>And here is some text in another HTML paragraph</p>
</body>
</html>
```
- Phần tử P đầu tiên kết thúc ở đâu?
- Cú pháp rõ ràng hơn -> Cấu trúc của XML

Các thẻ gán không thể phủ chồng

- Trong XML các phần tử không được phủ chồng
- Trong HTML các thẻ gán được phép phủ chồng
<P>Some
formatted
text
, but
no grammar no good!
</P>
- Quy tắc rõ ràng hơn

Một tư liệu XML chỉ có thể có một thành phần gốc

- Một tư liệu XML phải có một và chỉ một thành phần gốc
- Phải có thành phần gốc cho dù không có nội dung
- Ví dụ, một số thành phần gốc:

```
<name>John</name>
```

```
<name>Jane</name>
```

Phải bổ sung thêm thành phần cấp đỉnh:

```
<names>
```

```
  <name>John</name>
```

```
  <name>Jane</name>
```

```
</names>
```

Các tên thành phần

- Các tên có thể bắt đầu bằng các mẫu tự (bao gồm các ký tự phi La tinh) hoặc ký tự `_`, nhưng không có các con số hoặc các ký tự dấu chấm câu khác
- Sau ký tự đầu tiên, ta được phép dùng các con số, cũng như các ký tự `-` và `.`
- Các tên không thể chứa các khoảng cách
- Các tên không thể chứa ký tự `;`
- Các tên không thể bắt đầu bằng các mẫu tự `xml`, bằng chữ hoa, chữ thường, hoặc hỗn hợp
- Không thể có một dấu cách sau ký tự `<` mở, tên của thành phần phải đi ngay sau đó. Tuy nhiên, có thể có dấu cách trước ký tự `>` đóng, nếu muốn

Các tên thành phần

- `<résumé>` hợp lệ
- `<xml-tag>` bắt đầu bằng `xml`
- `<123>` bắt đầu bằng một con số
- `<fun=xml>` dấu `=` là không hợp lệ
- `<my tag>` chứa một dấu cách

Phân biệt chữ hoa chữ thường

- Các thẻ gán trong XML đều phân biệt chữ hoa chữ thường
- Sự khác biệt lớn so với HTML, nó không phân biệt chữ hoa chữ thường, ví dụ giữa `<P>` và `<p>`
- `<first>` khác với `<FIRST>`, khác với `<First>`

Khoảng cách trắng trong PCDATA

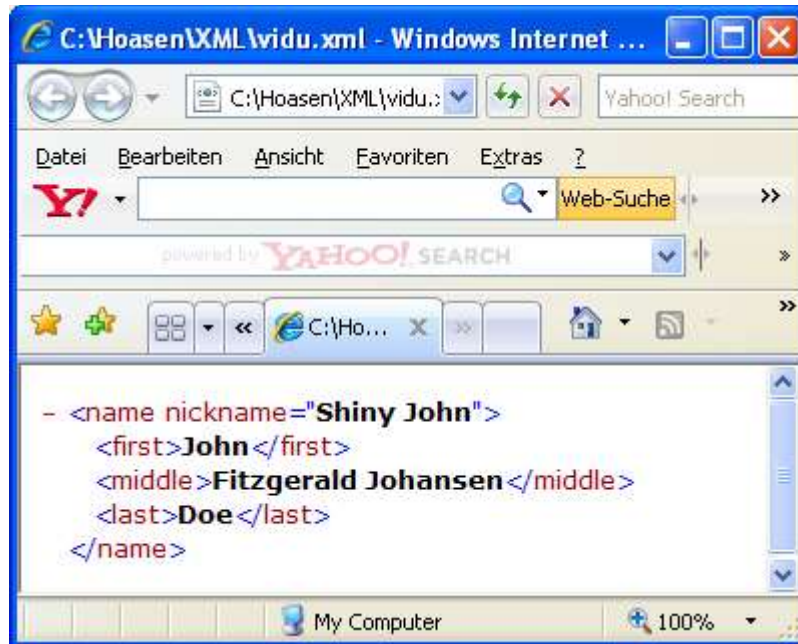
- Ví dụ:
<P>This is a paraprgraph. It has a whole bunch of space.</P>
- Trong HTML, mọi khoảng cách trắng được xem là vô nghĩa sẽ bị cắt bỏ ra khỏi tư liệu khi nó được xử lý



Khoảng cách trắng trong PCDATA

- Trong XML, không có tính năng cắt bỏ khoảng trắng đối với PCDATA
- Ví dụ:
`<paragraph>This is a parapraph. It has a whole bunch of space.</paragraph>`
- Chú ý: trong IE 5 và IE 6 khoảng cách trắng sẽ được cắt bỏ hoặc có vẻ như vậy
- Nguyên nhân: XML được biến đổi thành HTML để hiển thị

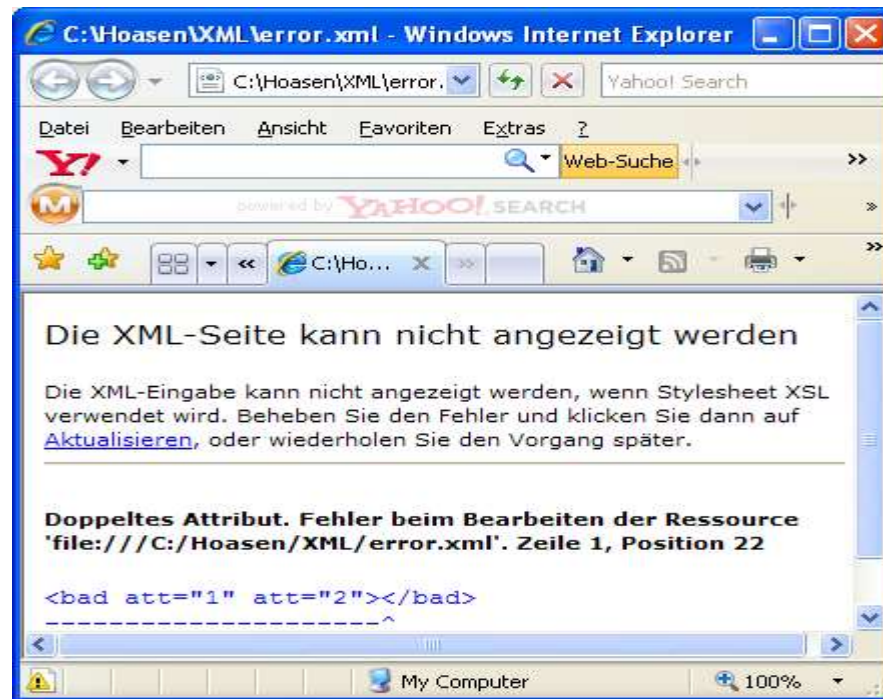
Thuộc tính



- Ngoài các thẻ gán và thành phần, các tư liệu XML còn bao gồm các thuộc tính
- Các thuộc tính là các cặp tên/giá trị đơn giản kết hợp với một thành phần
- Giá trị của thuộc tính phải nằm trong dấu nháy
- Một thành phần có thể có nhiều thuộc tính

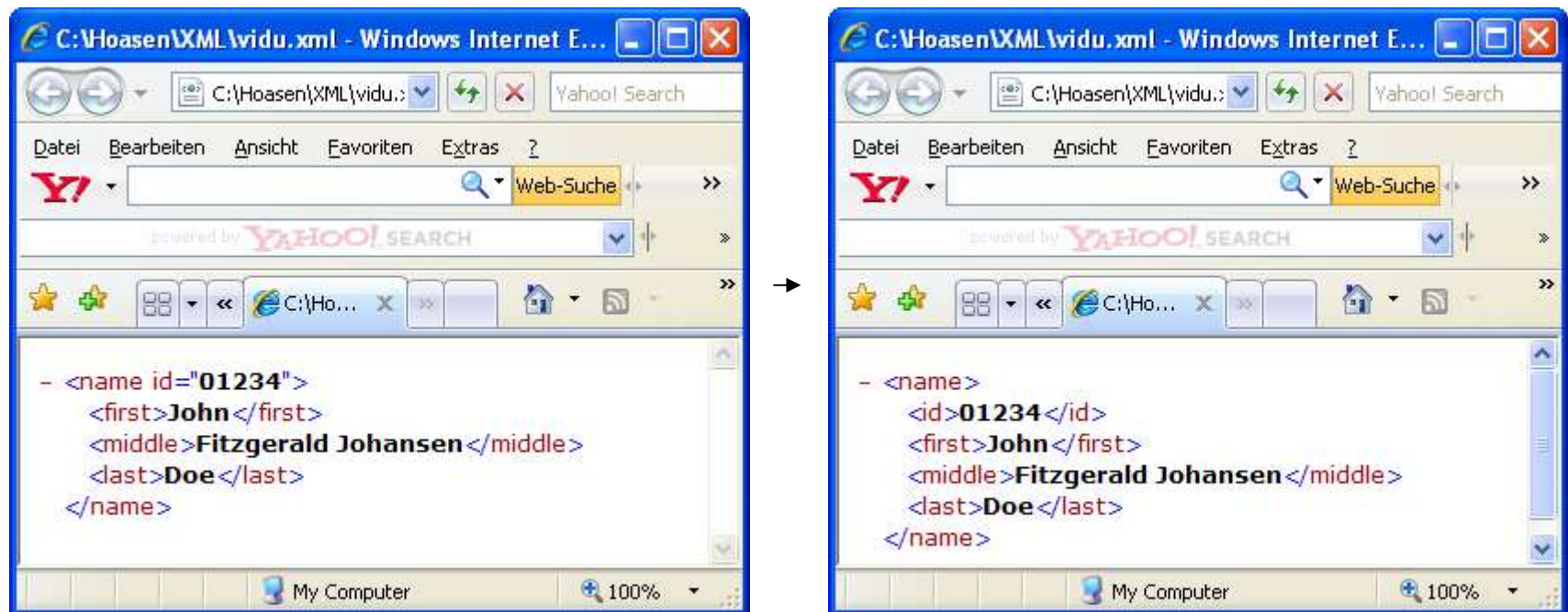
Thuộc tính

- Không thể có nhiều thuộc tính trùng tên trên một thành phần



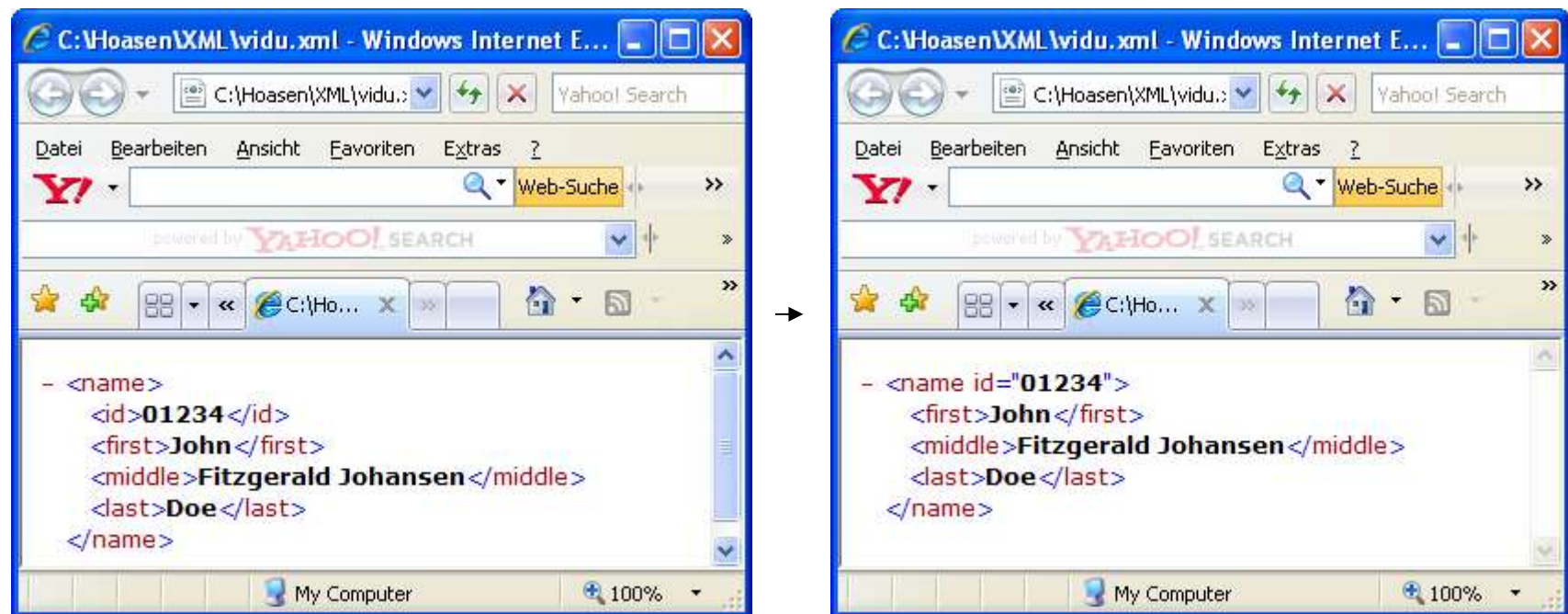
Thành phần thay thế thuộc tính

- Một thuộc tính có thể được hiển thị như một thành phần con



Thuộc tính thay thế thành phần

- Một thành phần con có thể hiển thị như một thuộc tính



Khai báo XML

- Chứa những thông tin cho bộ xử lý, đặc biệt là sử dụng XML-Version và Mã hóa
- Phải luôn xuất hiện ở dòng đầu tiên của tư liệu

```
<?xml version='1.0' encoding='UTF-16' standalone='yes'?>
<name nickname='Shiny John'>
  <first>John</first>
  <!--John lost his middle name in a free-->
  <middle/>
  <last>Doe</last>
</name>
```

Khai báo XML

- version(bắt buộc): Sử dụng XML-Version, version hiện tại là `1.0`
- standalone(tuỳ chọn): nó phải là yes hoặc no
- encoding(tuỳ chọn): Sử dụng mã hoá của tư liệu XML

```
<?xml version='1.0' encoding='UTF-16'  
standalone='yes'>
```

Khai báo XML: Mã hoá


- ASCII (American Standard Code for Information Interchange) là lược đồ mã hoá 7 bit và 8 bit.
- 7-bit-ASCII phủ 128 ký tự với chỉ một phần nhỏ của những ký tự sử dụng trên thế giới
- ASCII 8-bit dùng một byte (8 bit) cho mỗi ký tự
- Chỉ có thể lưu trữ 256 giá trị khác nhau, do đó giới hạn ASCII ở mức 256 ký tự.
- Tổng hợp: ASCII chỉ có thể điều quản một tập con nhỏ trong số các ngôn ngữ trên thế giới

Khai báo XML: Unicode


- Unicode phủ 65356 ký tự cho tất cả những ký tự được sử dụng trên thế giới
- Có 2 phương pháp mã hoá ký tự chính cho Unicode: UTF-16 và UTF-8
- UTF-8 hay UTF-16 bao hàm đủ các ký tự khả dĩ để chứa tất cả các ký tự trong bất kỳ ngôn ngữ nào của loài người, UTF-8 sử dụng mã hoá thông minh hơn
- 7-bit ASCII là một phần của UTF-8
- Với các văn bản bằng tiếng anh, ở đó hầu hết các ký tự hợp với phương pháp mã hoá ký tự ASCII, UTF-8 có thể cho ra các kích cỡ tập tin nhỏ hơn, nhưng với văn bản trong các ngôn ngữ khác, UTF-16 thường nhỏ hơn

Khai báo XML: Mã hoá

- Tất cả các bộ xử lý XML phải dùng Unicode nội bộ
- Thuộc tính encoding chỉ định, với trình phân ngữ XML, phương pháp mã hoá ký tự mà văn bản tuân thủ
- Nếu không có phương pháp mã hoá nào chỉ định, UTF-16 hoặc UTF-8 được mặc nhận
- Nếu không có phương pháp mã hoá nào được chỉ định, và tư liệu không thuộc UTF-8 hoặc UTF-16, điều đó sẽ dẫn đến một lỗi



Không gian tên (Namespace)



Nội dung

- Cần cho không gian tên
- Cú pháp không gian tên
- Không gian tên mặc định
- Bộ xử lý (Parser) và không gian tên

Cần cho không gian tên

- Một tư liệu XML có thể dùng nhiều ngữ vựng XML
- Ví dụ:
 - XHTML có thể chứa cả hai phần tử SVG và MathML
 - XSL stylesheet có thể chứa cả hai phần tử XSLT và kết quả cây
- Như thế nào để tránh sự va chạm tên?
 - Cả hai SVG và MathML “set” phần tử

Cần cho không gian tên

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<catalog>

  <RDF>
    <Description about="http://ibiblio.org/examples/impressionists.xml">
      <!-- title of a webpage -->
      <title> Impressionist Paintings </title>
      <creator> Elliotte Rusty Harold </creator>
      <description>
        A list of famous impressionist paintings organized
        by painter and date
      </description>
      <date>2000-08-22</date>
    </Description>
  </RDF>
```

Tiếp tục

```
<painting>
  <!-- title of a painting -->
  <title>Memory of the Garden at Etten</title>
  <artist>Vincent Van Gogh</artist>
  <date>November, 1888</date>
  <description>
    Two women look to the left. A third works in her garden.
  </description>
</painting>

<painting>
  <title>The Swing</title>
  <artist>Pierre-Auguste Renoir</artist>
  <date>1876</date>
  <description>
    A young girl on a swing. Two men and a toddler watch.
  </description>
</painting>

<!-- Many more paintings... -->

</catalog>
```

Cần cho không gian tên

- Thay đổi tên phần tử (để tránh sự va chạm) không là một sự tùy chọn quy ước
- Một vài sự va chạm không thể tránh được:
 - Nếu cả hai ngữ vựng tiêu chuẩn
 - SVG “set” với MathML “set”
- Một nhóm tên sẽ hữu dụng trong bất kỳ trường hợp nào
 - Bộ máy xử lý XSLT cần biết loại nào là lệnh XSLT và loại nào là phần tử kết quả cây

Ví dụ không gian tên

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<catalog>

  <rdf:RDF xmlns:rdf="http://www.w3.org/TR/REC-rdf-syntax#"
    <rdf:Description xmlns:dc="http://purl.org/dc/"
      about="http://ibiblio.org/examples/impressionists.xml">
        <dc:title> Impressionist Paintings </dc:title>
        <dc:creator> Elliotte Rusty Harold </dc:creator>
        <dc:description>
          A list of famous impressionist paintings organized
          by painter and date
        </dc:description>
        <dc:date>2000-08-22</dc:date>
      </rdf:Description>
    </rdf:RDF>
  ...
```

Cú pháp không gian tên

- Hai phần:
 - Mô tả không gian tên
 - Thành phần và Thuộc tính

Mô tả không gian tên

- Một *prefix* được liên kết với một URI
- Sự liên kết này được định nghĩa như một thuộc tính trong một thành phần
 - *xmlns:prefix*
- xmlns là một từ khóa không gian tên. Prefix là người dùng định nghĩa

```
<classes xmlns:XMLclass="http://www.brandeis.edu/rseg-0151-g">
  <XMLclass:syllabus>
    ...
    <XMLclass:syllabus>
</classes>
```

Mô tả không gian tên

- Có thể mô tả trong thành phần gốc hoặc trong thành phần phân cấp thấp hơn
- Cùng prefix có thể được tái định nghĩa trong cùng một tư liệu
 - Mô tả scope của không gian tên thì nằm trong thành phần nơi nó được định nghĩa

Thành phần và thuộc tính với prefix không gian tên

■ Ví dụ

- ❑ XMLclass:syllabus
- ❑ svg:set
- ❑ mathml:set

■ *prefix:local part*

- ❑ prefix đồng nhất không gian tên một thành phần và thuộc tính phụ thuộc
- ❑ local part đồng nhất một phần thành phần hay thuộc tính trong không gian tên
- ❑ Tên hạn định

Thành phần và thuộc tính với prefix không gian tên

■ Prefix

- ❑ Có thể được hình thành từ bất kỳ ký tự quy tắc tên của XML ngoại trừ “:”
- ❑ “xml” (trong bất kỳ trường hợp kết hợp nào) được dự phòng

■ Local part

- ❑ Không thể chứa “:”

Không gian URI

- URI không thể là prefix
 - “/”, “%”, và “~” không hợp quy tắc trong tên phần tử
 - URI có thể được chuẩn hóa trong khi prefix chỉ là quy ước
- URI chỉ là đồng nhất
 - URI không phải là dạng “http”
 - URI không phải được giải quyết
 - Nó thì như “một giá trị hằng”

Không gian tên mặc định

- Biểu thị với xmlns thuộc tính với không prefix
- Chỉ cung cấp đến thành phần unprefix và những thành phần con cháu của nó
- Chỉ cung cấp đến những thành phần không thuộc tính

Không gian tên mặc định

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:xlink="http://www.w3.org/1999/xlink">
  <head><title>Three Namespaces</title></head>
  <body>
    <h1 align="center">An Ellipse and a Rectangle</h1>
    <svg xmlns="http://www.w3.org/2000/svg"
         width="12cm" height="10cm">
      <ellipse rx="110" ry="130" />
      <rect x="4cm" y="1cm" width="3cm" height="6cm" />
    </svg>
    <p xlink:type="simple" xlink:href="ellipses.html">
      More about ellipses
    </p>
    <p xlink:type="simple" xlink:href="rectangles.html">
      More about rectangles
    </p>
    <hr/>
    <p>Last Modified May 13, 2000</p>
  </body>
</html>
```

Bộ xử lý và không gian tên

- Không gian tên là sự suy nghĩ sau của XML 1.0
- Tình trạng tương thích đã qua
 - SAX 1.0 và bộ xử lý DOM bậc 1 (1.0) không nhận biết được không gian tên
 - Chúng nó vẫn đọc không gian tên-khả năng tự liệu XML
- Bây giờ SAX 2.0 và bộ xử lý DOM bậc 2 nhận biết được không gian tên

Tóm lại: Các quy tắc tạo tài liệu XML hợp khuôn dạng

- Các khai báo đặt ngay đầu tiên của tài liệu
 - Tài liệu có phần tử gốc, các phần tử con khác nếu có phải là con của phần tử gốc.
 - Mọi phần tử XML khác rỗng phải có đầy đủ thẻ đóng và thẻ mở.
 - Đóng phần tử rỗng với />
 - Mọi phần tử trong tài liệu XML khác phần tử gốc (root) đều phải nằm giữa cặp thẻ gốc
-

Bài tập

**Bài tập Chương 2 (1, 2, 3) Sách “Giáo trình Công nghệ XML & UĐ” –
Phạm Thị Thu Thúy**