

# Lab1: Data mapping and exchange

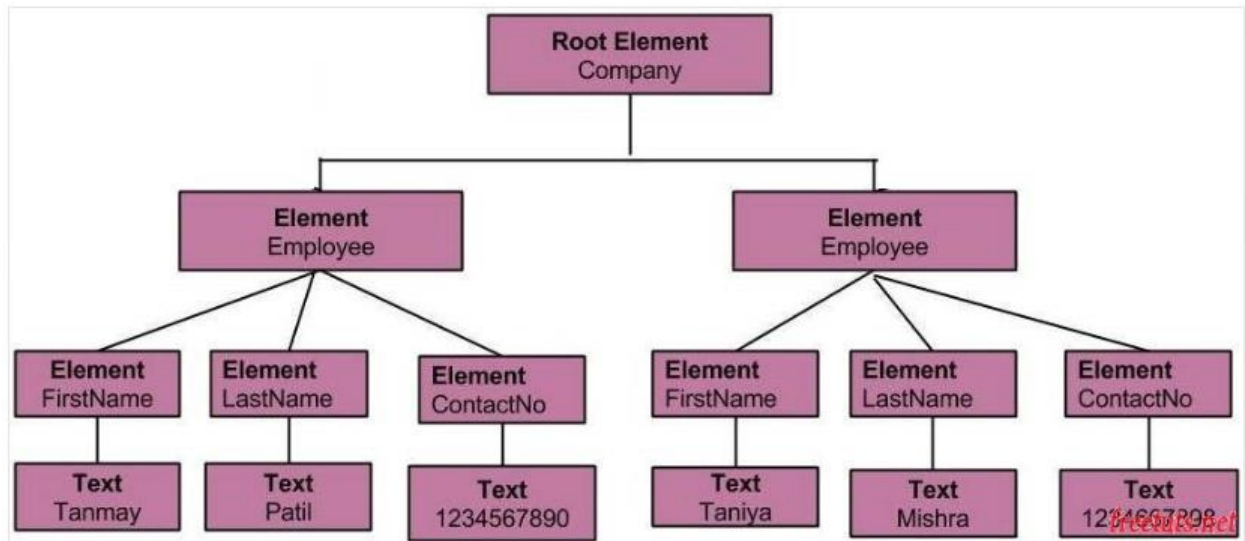
## Contents

XML .....	2
1. XML là gì? Cú pháp căn bản của XML.....	2
2. Cấu trúc cây trong XML.....	2
3. XML Elements .....	3
4. XML Attributes .....	4
5. XML Namespace.....	6
Cú pháp tạo XML namespace: .....	7
6. CDATA .....	8
DOM .....	10
Xpath.....	10
Tài liệu tham khảo.....	11

## XML

1. XML là gì? Cú pháp căn bản của XML
2. Cấu trúc cây trong XML

Tạo tài liệu XML cho sơ đồ cấu trúc cây lưu trữ thông tin nhân viên công ty sau:



```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <company>
3      <employee>
4          <firstname>
5              Tanmay
6          </firstname>
7          <lastname>
8              Patil
9          </lastname>
10         <contactno>
11             123456789
12         </contactno>
13     </employee>
14     <employee>
15         <firstname>
16             Taniya
17         </firstname>
18         <lastname>
19             Mishra
20         </lastname>
21         <contactno>
22             123456789
23         </contactno>
24     </employee>
25     <employee>
26         <firstname>
27             Cuong
28         </firstname>
29         <lastname>
30             Nguyen
31         </lastname>
32         <contactno>
33             0979306603
34         </contactno>
35     </employee>
36 </company>

```

Lấy ví dụ ở phần trên và bổ sung thêm yêu cầu sau: Mỗi nhân viên lưu trữ thêm **danh sách người thân của nhân viên đó** (chỉ cần lưu trữ tên, mối quan hệ).

### 3. XML Elements

Ví dụ ta cần viết một file XML để lưu trữ danh sách sinh viên gồm các thông tin:

- Tên sinh viên (tensv)
- Năm sinh (namsinh)
- Giới tính (gioitinh)

Lúc này ta sẽ có ba thẻ chứa dữ liệu đơn giản đó là tensv, namsinh và gioitinh. Có một thẻ chứa dữ liệu phức tạp đó là thẻ **sinhvien**. Nhưng chưa dừng lại ở đó, như ở bài tìm hiểu [xml là gì](#) chúng ta cần

xác định thêm một thẻ Root nữa. Phân tích kỹ thì ta thấy mỗi sinh viên sẽ được mô tả bởi thẻ `sinhvien` và tài liệu sẽ chứa nhiều sinh viên, vì vậy mình sẽ đặt tên cho thẻ root là `ds-sinhvien`.

Như vậy dưới đây là [cấu trúc cây XML](#) của chúng ta.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ds-sinhvien>
3   <sinhvien>
4     <tensv>[Data]</tensv>
5     <namsinh>[Data]</namsinh>
6     <gioitinh>[Data]</gioitinh>
7   </sinhvien>
8 </ds-sinhvien>
```

Lưu hai sinh viên với các thông tin như sau:

SV1: Tên – Năm sinh – Giới tính

SV2: Tên – Năm sinh – Giới tính – Mô tả

## 4. XML Attributes

```
1 <element_name property_name="property_value">
2
3 </element_name>
```

Cách 1: Sử dụng thuộc tính

```
<?xml version="1.0" encoding="UTF-8"?>
<coders>
  <person skill="PHP Javascript jQuery" master_at="PHP">
    <name>Nguyễn Văn Cường</name>
    <website>Freetuts.net</website>
  </person>
</coders>
```

Cách 2: Sử dụng thẻ XML

```
<?xml version="1.0" encoding="UTF-8"?>
<coders>
  <person>
    <master_at>PHP</master_at>
    <skill>PHP Javascript jQuery</skill>
    <name>Nguyễn Văn Cường</name>
    <website>Freetuts.net</website>
  </person>
</coders>
```

## XML Attributes vs. XML Properties

Như ở ví dụ so sánh cách sử dụng XML Attributes thay thế cho XML Elements ở trên thì bạn đã thấy được sự chuyển đổi đơn giản giữa hai khái niệm này, tuy nhiên chúng ta vẫn còn một số lưu ý nữa mà bạn nên đọc để hiểu thêm.

### XML Multi value:

Thứ nhất: Mỗi một thuộc tính XML chỉ có thể chứa một giá trị duy nhất cho dù bạn nhập gì bên trong nó đi nữa thì XML vẫn tính là một giá trị, bởi vì XPath (tìm hiểu sau) sẽ không tính nó là nhiều giá trị như bạn nghĩ.

Thứ hai: Các thuộc tính XML chỉ chứa giá trị dạng text, number chứ không thể chứa một thẻ XML khác.

Từ hai tính chất trên ta thấy khi sử dụng thuộc tính XML thì rất khó mở rộng tài liệu, ví dụ sau này bạn cần bổ sung thông tin là con của thuộc tính nào đó thì không thể làm được, nhưng với element thì hoàn toàn làm được.

Ví dụ: Trường hợp attributes khó nâng cấp và mở rộng

```
<?xml version="1.0" encoding="UTF-8"?>
<coders>
  <person master_at="PHP" skill="PHP Javascript jQuery"
    name="Nguyễn Văn Cường" website="Freetuts.net">
  </person>
</coders>
```

Giả sử giờ mình muốn lấy họ của lập trình viên thì phải xử lý tách chuỗi, Nhưng với cách thiết kế dưới đây thì không cần

```
<?xml version="1.0" encoding="UTF-8"?>
<coders>
  <person>
    <master_at>PHP</master_at>
    <skill>PHP Javascript jQuery</skill>
    <name>
      <firstname>Cường</firstname>
      <middlename>Văn</middlename>
      <lastname>Nguyễn</lastname>
    </name>
    <website>Freetuts.net</website>
  </person>
</coders>
```

### Metadata trong XML

Tới phần này bạn mới thấy Attributes thực sự có công dụng. Chúng ta thường sử dụng Attributes để lưu trữ những thông tin chính như là khóa chính của thẻ XML đó.

Ví dụ

```
<?xml version="1.0" encoding="UTF-8"?>
<coders>
  <person id="1">
```

```

    <master_at>PHP</master_at>
    <skill>PHP Javascript jQuery</skill>
    <name>
      <firstname>Cường</firstname>
      <middlename>Văn</middlename>
      <lastname>Nguyễn</lastname>
    </name>
    <website>Freetuts.net</website>
  </person>
  <person id="2">
    <master_at>Javascript</master_at>
    <skill>PHP Javascript AngularJS</skill>
    <name>
      <firstname>Tinh</firstname>
      <middlename>Thị Thu</middlename>
      <lastname>Vũ</lastname>
    </name>
    <website>Freetuts.net</website>
  </person>
</coders>

```

Ngoài cách sử dụng thuộc tính ra thì ta cũng có thể tạo thêm một thẻ `id` để lưu trữ, tuy nhiên lời khuyên cho bạn nên sử dụng thuộc tính cho trường hợp này.

## 5. XML Namespace

### XML Namespace là gì?

Cấu trúc một tài liệu XML được xây dựng bởi các lập trình viên, do đó họ có thể tự đặt tên thẻ XML (*XML Element*), điều này dẫn đến xung đột nếu trong một file bị đặt trùng tên dẫn đến không phân biệt được thẻ nào dùng cho ứng dụng nào.

**Ví dụ:** Bị trùng lặp thẻ Title.

```

<?xml version="1.0" encoding="UTF-8"?>
<ShopOrder>
  <Order>
    <Customer>
      <Title>Nguyễn Văn Cường</Title>
      <Address>Buôn Ma Thuật - Đắk Lắk</Address>
    </Customer>
    <Product>
      <Title>Dép thái cao cấp</Title>
      <Qua>20</Qua>
      <Price>200.000 VNĐ</Price>
    </Product>
  </Order>
</ShopOrder>

```

Trong tài liệu này có hai thẻ `Title`, thẻ thứ nhất là tên của khách hàng và thẻ thứ hai là tên của sản phẩm => điều này ta gọi là xung đột thẻ XML.

Nếu bạn nghĩ điều này không có gì đáng lo ngại thì đúng rồi đấy, bởi vì trong ví dụ chỉ nói đến một tài liệu XML. Nhưng bạn có nghĩ trường hợp ta sử dụng kết hợp nhiều tài liệu XML với nhau thì sẽ dẫn đến sự nhầm lẫn tệ hại không? Vì vậy để giải quyết vấn đề này thì ta phải sử dụng **XML Namespace**.

Cú pháp tạo XML namespace:

Để tạo một namespace thì ta sẽ khai báo một thuộc tính `xmlns:uri`.

- Trong đó `xmlns` là viết tắt của từ XML namespace.
- `uri` là đường dẫn URL của namespace, có thể là một địa chỉ nào đó trên internet hoặc một địa chỉ nào đó đều được nhưng phải đảm bảo rằng nó là duy nhất trong file tài liệu XML của bạn.

**Ví dụ:**

```
<?xml version="1.0" encoding="UTF-8"?>
<ShopOrder>
  <Order>
    <cus:Customer xmlns:cus="https://freetuts.net/customer">
      <cus:Title>Nguyễn Văn Cường</cus:Title>
      <cus:Address>Buôn Ma Thuột - Đắk Lắk</cus:Address>
    </cus:Customer>
  </Order>
</ShopOrder>
```

Như vậy các thẻ XML chúng ta cũng phải bổ sung tên của namespace theo cú pháp `namespace_name:element_name`

### Namespace mặc định

Nếu tài liệu của chúng ta chỉ sử dụng một namespace thì bạn có thể khai báo namespace mặc định cho nó bằng cách chỉ sử dụng từ khóa `xmlns` và bỏ đi tên của namespace.

**Ví dụ:**

```
<?xml version="1.0" encoding="UTF-8"?>
<ShopOrder xmlns="https://freetuts.net/customer">
  <Order>
    <Customer>
      <Title>Nguyễn Văn Cường</Title>
      <Address>Buôn Ma Thuột - Đắk Lắk</Address>
    </Customer>
    <Customer>
      <Title>Nguyễn Văn Kinh</Title>
      <Address>Buôn Ma Thuột - Đắk Lắk</Address>
    </Customer>
  </Order>
</ShopOrder>
```

Nhưng giả sử chúng ta có hai namespace mặc định trong một tài liệu XML như trường hợp dưới đây:

```
<?xml version="1.0" encoding="UTF-8"?>
<ShopOrder>
  <Order>
    <Customer xmlns="https://freetuts.net/customer">
      <Title>Nguyễn Văn Cường</Title>
      <Address>Buôn Ma Thuột - Đắk Lắk</Address>
    </Customer>
```

```

    <Product xmlns="https://freetuts.net/product">
      <Title>Dép thái cao cấp</Title>
      <Qua>20</Qua>
      <Price>200.000 VNĐ</Price>
    </Product>
  </Order>
</ShopOrder>

```

Trường hợp này nếu ta có nhiều đơn hàng thì phải tạo nhiều namespace mặc định. Điều này khá phiền toái nếu ta có 1000 đơn hàng phải không nào? Để giải quyết vấn đề này thì ta khai báo namespace ngay tại thẻ Root.

```

<?xml version="1.0" encoding="UTF-8"?>
<ShopOrder
  xmlns:cus="https://freetuts.net/customer"
  xmlns:pro="https://freetuts.net/product">
  <Order>
    <cus:Customer>
      <cus:Title>Nguyễn Văn Cường</cus:Title>
      <cus:Address>Buôn Ma Thuột - Đắk Lắk</cus:Address>
    </cus:Customer>
    <pro:Product>
      <pro:Title>Dép thái cao cấp</pro:Title>
      <pro:Qua>20</pro:Qua>
      <pro:Price>200.000 VNĐ</pro:Price>
    </pro:Product>
  </Order>
</ShopOrder>

```

## 6. CDATA

### CDATA trong XML

Một tài liệu XML được xây dựng bởi các thẻ XML do lập trình viên tự định nghĩa ra, và đương nhiên lập trình viên phải tuân thủ các quy tắc mà XML đề ra (xem bài [XML Element](#)). Tuy nhiên có một số trường hợp dữ liệu chứa trong các thẻ XML lại có những ký tự đặc biệt dẫn đến tài liệu XML bị đảo lộn.

**Ví dụ:**

```

<?xml version="1.0" encoding="UTF-8"?>
<Tutorials>
  <Post>
    <Title>
      Học CDATA trong XML miễn phí
    </Title>
    <Code>
      <?xml version="1.0" encoding="UTF-8"?>
      <Persons>
        <Person>
          Nguyễn Văn Cường
        </Person>
      </Persons>
    </Code>
  </Post>
</Tutorials>

```



Bạn thấy nội dung bên trong thẻ `Code` là một đoạn mã XML khác, có nghĩa đây là trường hợp giá trị của thẻ XML lại là một đoạn mã XML khác. Nếu chạy lên sẽ bị lỗi như hình dưới đây:

### This page contains the following errors:

error on line 8 at column 18: XML declaration allowed only at the start of the document

### Below is a rendering of the page up to the first error.

Để khắc phục trường hợp này ta sẽ sử dụng CDATA như sau:


```
<?xml version="1.0" encoding="UTF-8"?>
<Tutorials>
  <Post>
    <Title>
      Học CDATA trong XML miễn phí
    </Title>
    <Code>
      <![CDATA[
        <?xml version="1.0" encoding="UTF-8"?>
        <Persons>
          <Person>
            Nguyễn Văn Cường
          </Person>
        </Persons>
      ]]>
    </Code>
  </Post>
</Tutorials>
```

Như vậy cú pháp của CDATA là `<![CDATA[nội dung]]>`

## Sử dụng Special Character trong XML

Special character (đôi lúc ta gọi là thực thể) là những ký tự đặc biệt được chuyển đổi thành mỗi dãy các chữ cái và dấu `&`;

Như ở trong phần một ta sử dụng CDATA để khắc phục lỗi nội dung XML chứa ký tự đặc biệt. Nhưng cũng có một cách khác là bạn chuyển những ký tự đặc biệt đó sang Special characters như bảng dưới đây.

Thực thể	Mô tả
&apos;	Dấu nháy đơn ' 
&amp;	Dấu & 
&gt;	Dấu > 
&lt;	Dấu < 
&quot;	Dấu nháy kép " 

**Ví dụ:** Viết lại ví dụ trên bằng cách sử dụng Special character.

```
<?xml version="1.0" encoding="UTF-8"?>
<Tutorials>
  <Post>
    <Title>
      Học CDATA trong XML miễn phí
    </Title>
    <Code>
      &lt;?xml version="1.0" encoding="UTF-8"?&gt;
      &lt;Persons&gt;
        &lt;Person&gt;
          Nguyễn Văn Cường
        &lt;/Person&gt;
      &lt;/Persons&gt;
    </Code>
  </Post>
</Tutorials>
```

## DOM

[https://www.w3schools.com/xml/xml\\_dom.asp](https://www.w3schools.com/xml/xml_dom.asp)

[https://www.w3schools.com/xml/dom\\_examples.asp](https://www.w3schools.com/xml/dom_examples.asp)

## Xpath

Businesscard.xml

```
<?xml version="1.0"?>
<BusinessCard>
  <Name>Joe Marini</Name>
  <phone type="mobile">(415) 555-4567</phone>
  <phone type="work">(800) 555-9876</phone>
  <phone>(510) 555-1234</phone>
```

```
<email>joe@joe.com</email>
</BusinessCard>
```

Items.xml

```
<?xml version="1.0"?>
<items>
  <item id="item1">
    <name>Mocha</name>
    <type>Coffee</type>
    <photo>photos/candles.jpg</photo>
  </item>
  <item id="item2">
    <name>Decaf</name>
    <type>Coffee</type>
    <photo>photos/teacup.jpg</photo>
  </item>
  <item id="item3">
    <name>Raspberry Zinger</name>
    <type>Tea</type>
    <photo>photos/teapot_white.jpg</photo>
  </item>
  <item id="item4">
    <name>Earl Grey</name>
    <type>Tea</type>
    <photo>photos/teapot_bone.jpg</photo>
  </item>
</items>
```

## Tài liệu tham khảo

<https://freetuts.net/>