

XML Schema

Nguyễn Hồng Phương
 Email: phuong.nguyenhong@hust.edu.vn
 Site: <http://is.hut.edu.vn/~phuongnh>
 Bộ môn Hệ thống thông tin
 Viện Công nghệ thông tin và Truyền thông
 Đại học Bách Khoa Hà Nội

1

Nội dung

- Giới thiệu
- Các kiểu đơn giản
- Các kiểu phức tạp
- Kiểu dữ liệu

2

1. Giới thiệu

- Là một giải pháp thay thế cho DTD
- Một XML Schema mô tả cấu trúc một tài liệu XML
- Ngôn ngữ lược đồ XML được biết đến là XML Schema Definition (XSD)

3

□ Một lược đồ XML

- định nghĩa các phần tử có thể xuất hiện trong tài liệu.
- định nghĩa các thuộc tính có thể xuất hiện trong tài liệu.
- định nghĩa phần tử con
- định nghĩa thứ tự các phần tử con
- định nghĩa số lượng các phần tử con
- định nghĩa một phần tử là rỗng hay chứa text
- định nghĩa kiểu dữ liệu cho phần tử/thuộc tính
- định nghĩa giá trị mặc định cho phần tử/thuộc tính

4

□ XML Schema là một sự thay thế cho DTD?

- Dễ mở rộng
- Giàu tính biểu đạt hơn DTD
- Được viết trong XML
- Hỗ trợ các kiểu dữ liệu
- Hỗ trợ các không gian tên
- Là khuyến cáo của W3C

5

XML Schema hỗ trợ các kiểu dữ liệu

- Một trong những điểm mạnh của XML Schema là hỗ trợ các kiểu dữ liệu
 - mô tả nội dung tài liệu được phép
 - hợp thức hóa tính đúng đắn của dữ liệu
 - làm việc với dữ liệu của một CSDL
 - xác định các khía cạnh dữ liệu (các giới hạn)
 - xác định các mẫu dữ liệu (format dữ liệu)
 - chuyển đổi dữ liệu giữa các kiểu dữ liệu khác nhau

6

XML Schema sử dụng cú pháp XML

- Lược đồ XML được viết trong XML
- Lợi ích:
 - Không phải học ngôn ngữ mới
 - Có thể soạn thảo file schema bằng bộ soạn thảo XML
 - Có thể sử dụng XML parser để parse file schema
 - Có thể thao tác lược đồ với XML DOM
 - Có thể chuyển đổi lược đồ với XSLT

7

XML Schema truyền dữ liệu thống nhất

- Kiểu ngày "03-02-2013"
 - 03 Feb 2013
 - 02 March 2013
- Thống nhất bên gửi và bên nhận
 - <date type="date">2013-02-03</date>
 - Kiểu dữ liệu date yêu cầu định dạng yyyy-mm-dd

8

Lược đồ XML có thể mở rộng

- Vì được viết trong XML nên có thể mở rộng
- Với một định nghĩa lược đồ có thể mở rộng, chúng ta có thể:
 - Tái sử dụng lược đồ trong các lược đồ khác
 - Tạo ra kiểu dữ liệu riêng dẫn xuất từ kiểu chuẩn
 - Tham chiếu nhiều lược đồ trong cùng tài liệu

9

Dạng chuẩn thô chưa đủ!

- Tài liệu XML dạng chuẩn tuân theo luật cú pháp XML
- Dạng chuẩn vẫn có thể chứa lỗi và có thể gây hậu quả nghiêm trọng.

10

Một ví dụ

- Quan sát lại file note.xml

```
<?xml version="1.0"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

- Quan sát lại file note.dtd

```
<!ELEMENT note (to, from, heading, body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

11

Một ví dụ

- File note.xsd

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.w3schools.com"
  xmlns="http://www.w3schools.com"
  elementFormDefault="qualified">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

12

Một ví dụ

- ❑ Tài liệu XML có một tham chiếu tới DTD

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM
"http://www.w3schools.com/dtd/note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

13

Một ví dụ

- ❑ Tài liệu XML có một tham chiếu tới XSD

```
<?xml version="1.0"?>
<note
xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3schools.com note.xsd">
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

14

Phần tử <schema>

- ❑ Là phần tử gốc của mọi lược đồ XML
- ❑ Có thể chứa một số thuộc tính

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">
...
</xs:schema>
```

15

- ❑ Đoạn `xmlns:xs=http://www.w3.org/2001/XMLSchema` chỉ ra rằng các phần tử và kiểu dữ liệu sử dụng trong lược đồ lấy từ không gian tên <http://www.w3.org/2001/XMLSchema> (chúng nên gắn kèm tên xs)
- ❑ `targetNamespace=http://www.w3schools.com` chỉ ra rằng các phần tử xác định bởi lược đồ này (note, to, from, heading, body) lấy từ không gian tên <http://www.w3schools.com>
- ❑ `xmlns=http://www.w3schools.com` : không gian tên mặc định

16

- ❑ Tham khảo tới một lược đồ trong tài liệu XML:

```
<?xml version="1.0"?>
<note xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3schools.com note.xsd">
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

17

2. Các kiểu đơn giản

- ❑ Phần tử XSD
- ❑ Thuộc tính XSD
- ❑ Giới hạn XSD

18

2.1. Phần tử đơn giản XSD

- Phần tử đơn giản là phần tử chỉ chứa text, không chứa phần tử hay thuộc tính khác.
- Text có thể là nhiều kiểu khác nhau: boolean, string, date,...

19

Cú pháp định nghĩa phần tử đơn giản

- `<xs:element name="xxx" type="yyy"/>`
 - xxx là tên phần tử
 - yyy là kiểu dữ liệu của phần tử
- Một số kiểu thông dụng
 - xs:string
 - xs:decimal
 - xs:integer
 - xs:boolean
 - xs:date
 - xs:time

20

Ví dụ

```
<lastname>Refsnes</lastname>
<age>36</age>
<dateborn>1970-03-27</dateborn>
```

```
<xs:element name="lastname" type="xs:string"/>
<xs:element name="age" type="xs:integer"/>
<xs:element name="dateborn" type="xs:date"/>
```

21

Giá trị của các phần tử đơn giản

- Giá trị mặc định

```
<xs:element name="color" type="xs:string" default="red"/>
```

- Giá trị gán sẵn

```
<xs:element name="color" type="xs:string" fixed="red"/>
```

22

2.2. Thuộc tính XSD

- Nếu một phần tử có các thuộc tính, thì phần tử đó có kiểu phức tạp.
- Cú pháp định nghĩa thuộc tính

```
<xs:attribute name="xxx" type="yyy"/>
```

- Một số kiểu thông dụng

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

23

Ví dụ

```
<lastname lang="EN">Smith</lastname>
```

```
<xs:attribute name="lang" type="xs:string"/>
```

24

□ Giá trị mặc định và giá trị gán sẵn

```
<xs:attribute name="lang" type="xs:string" default="EN"/>
```

```
<xs:attribute name="lang" type="xs:string" fixed="EN"/>
```

□ Mặc định, thuộc tính là tùy chọn. Để chỉ ra là thuộc tính bắt buộc, viết như sau:

```
<xs:attribute name="lang" type="xs:string" use="required"/>
```

25

2.3. Các ràng buộc trên giá trị

□ Giới hạn trên các giá trị

- giá trị <0 hoặc >120

```
<xs:element name="age">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="120"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

26

□ Giới hạn trên tập các giá trị

- các giá trị có thể chấp nhận: Audi, Golf, BMW

```
<xs:element name="car">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Audi"/>
      <xs:enumeration value="Golf"/>
      <xs:enumeration value="BMW"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

```
<xs:element name="car" type="carType"/>
<xs:simpleType name="carType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Audi"/>
    <xs:enumeration value="Golf"/>
    <xs:enumeration value="BMW"/>
  </xs:restriction>
</xs:simpleType>
```

27

□ Ràng buộc trên một chuỗi giá trị

□ Ví dụ:

- phần tử tên là letter chỉ nhận giá trị là 1 ký tự thường a-z

```
<xs:element name="letter">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-z]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

28

- phần tử initials chỉ chấp nhận giá trị là 3 ký tự hoa từ A-Z

```
<xs:element name="initials">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Z][A-Z][A-Z]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- phần tử initials chỉ chấp nhận giá trị là 3 ký tự hoa hoặc thường từ a-z hoặc A-Z

```
<xs:element name="initials">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z][a-zA-Z][a-zA-Z]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

29

```
<xs:element name="choice">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[xyz]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

```
<xs:element name="prodid">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:pattern value="[0-9][0-9][0-9][0-9]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

30

```
<xs:element name="letter">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-z]*"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Có không hoặc nhiều ký tự thường a-z

```
<xs:element name="letter">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z]+"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Có một hoặc nhiều cặp ký tự: ký tự thường đứng trước ký tự hoa

31

```
<xs:element name="gender">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="male|female"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Chỉ nhận giá trị male hoặc female

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9]{8}/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Chỉ nhận giá trị chính xác là 8 ký tự thường/hoa/chữ số

32

Ràng buộc ký tự trắng

```
<xs:element name="address">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="preserve"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Giá trị được thiết lập là "preserve". Bộ xử lý XML sẽ không bỏ bất kỳ ký tự trắng nào

```
<xs:element name="address">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="replace"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Bộ xử lý XML sẽ thay thế ký tự trắng (line feeds, tabs, spaces, carriage return) bằng spaces

33

```
<xs:element name="address">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="collapse"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Bộ xử lý XML sẽ thay line feeds, tabs, spaces, carriage returns bằng spaces; leading and trailing spaces được loại bỏ; nhiều spaces sẽ thu gọn lại 1 space

34

Giới hạn chiều dài

Để giới hạn chiều dài giá trị của phần tử, sử dụng ràng buộc length, maxLength, minLength

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Chiều dài chính xác là 8 ký tự

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="5"/>
      <xs:maxLength value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Tối thiểu 5 ký tự, tối đa 8 ký tự

35

Tổng hợp các giới hạn kiểu dữ liệu

Ràng buộc	Mô tả
enumeration	Định nghĩa danh sách các giá trị có thể nhận
fractionDigits	Số tối đa vị trí thập phân cho phép (>=0)
length	Xác định số lượng chính xác các ký tự hoặc các khoản mục được phép (>=0)
maxExclusive	Xác định cận trên cho giá trị số (giá trị của phần tử phải nhỏ hơn giá trị cận trên này)
maxInclusive	Xác định cận trên cho giá trị số (giá trị của phần tử phải nhỏ hơn hoặc bằng giá trị cận trên này)
maxLength	Xác định số ký tự hoặc khoản mục cực đại được phép (>=0)

36

Tổng hợp các giới hạn kiểu dữ liệu (tiếp)

Ràng buộc	Mô tả
minExclusive	Xác định cận dưới cho giá trị số (giá trị của phần tử phải lớn hơn giá trị cận trên này)
minInclusive	Xác định cận dưới cho giá trị số (giá trị của phần tử phải lớn hơn hoặc bằng giá trị cận trên này)
minLength	Xác định số kí tự hoặc khoản mục tối thiểu được phép (>=0)
pattern	Định nghĩa chính xác chuỗi kí tự được phép
totalDigits	Xác định chính xác số chữ số được phép (>0)
whiteSpace	Xác định cách thức xử lý với các kí tự trắng

37

3. Các kiểu phức tạp

38

3.1. Phần tử phức tạp

- ❑ Là phần tử chứa phần tử khác và/hoặc các thuộc tính
- ❑ Có 4 loại phần tử phức tạp
 - Phần tử rỗng
 - Phần tử chỉ chứa các phần tử khác
 - Phần tử chỉ chứa text
 - Phần tử chứa các phần tử khác và text

39

Ví dụ

`<product pid="1345"/>` → Phần tử product rỗng

`<employee>
<firstname>John</firstname>
<lastname>Smith</lastname>
</employee>` → Phần tử employee chỉ chứa các phần tử khác

`<food type="dessert">Ice cream</food>` → Phần tử food chỉ chứa text

`<description>
It happened on <date lang="norwegian">03.03.99</date>
</description>` → Phần tử description chứa cả phần tử và text

40

Định nghĩa phần tử phức tạp

- ❑ Quan sát ví dụ:
- ❑ Cách 1: định nghĩa trực tiếp bằng cách đặt tên phần tử

```
<employee>  
<firstname>John</firstname>  
<lastname>Smith</lastname>  
</employee>
```

```
<xs:element name="employee">  
<xs:complexType>  
<xs:sequence>  
<xs:element name="firstname" type="xs:string"/>  
<xs:element name="lastname" type="xs:string"/>  
</xs:sequence>  
</xs:complexType>  
</xs:element>
```

41

Định nghĩa phần tử phức tạp (tiếp)

- ❑ Cách 2: Định nghĩa thông qua tên kiểu phức tạp

```
<xs:element name="employee" type="personinfo"/>  
<xs:complexType name="personinfo">  
<xs:sequence>  
<xs:element name="firstname" type="xs:string"/>  
<xs:element name="lastname" type="xs:string"/>  
</xs:sequence>  
</xs:complexType>
```

```
<xs:element name="employee" type="personinfo"/>  
<xs:element name="student" type="personinfo"/>  
<xs:element name="member" type="personinfo"/>
```

```
<xs:complexType name="personinfo">  
<xs:sequence>  
<xs:element name="firstname" type="xs:string"/>  
<xs:element name="lastname" type="xs:string"/>  
</xs:sequence>  
</xs:complexType>
```

42

□ Có thể định nghĩa 1 kiểu phức tạp dựa trên 1 kiểu phức tạp

```

<xs:element name="employee" type="fullpersoninfo"/>
<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="fullpersoninfo">
  <xs:complexContent>
    <xs:extension base="personinfo">
      <xs:sequence>
        <xs:element name="address" type="xs:string"/>
        <xs:element name="city" type="xs:string"/>
        <xs:element name="country" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
    
```

43

3.2. Phần tử rỗng

□ Phần tử rỗng: `<product prodid="1345" />`

□ Định nghĩa phần tử rỗng:

```

<xs:element name="product">
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base="xs:integer">
        <xs:attribute name="prodid" type="xs:positiveInteger"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
    
```

44

Định nghĩa phần tử rỗng (tiếp)

Cách ngắn gọn

```

<xs:element name="product">
  <xs:complexType>
    <xs:attribute name="prodid" type="xs:positiveInteger"/>
  </xs:complexType>
</xs:element>
    
```

Cách khác

```

<xs:element name="product" type="prodtype"/>
<xs:complexType name="prodtype">
  <xs:attribute name="prodid" type="xs:positiveInteger"/>
</xs:complexType>
    
```

45

3.3. Phần tử chỉ chứa phần tử

□ Ví dụ:

```

<person>
  <firstname>John</firstname>
  <lastname>Smith</lastname>
</person>
    
```

□ Định nghĩa phần tử này trong lược đồ:

```

<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
    
```

46

Phần tử chỉ chứa phần tử (tiếp)

□ Định nghĩa cách khác:

```

<xs:element name="person" type="persontype"/>
<xs:complexType name="persontype">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
    
```

47

3.4. Phần tử chỉ chứa text

□ Chứa text và các thuộc tính

Định nghĩa

```

<xs:element name="somename">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="basetype">
        ....
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
    
```

```

<xs:element name="somename">
  <xs:complexType>
    <xs:simpleContent>
      <xs:restriction base="basetype">
        ....
      </xs:restriction>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
    
```

48

Ví dụ

```
<shoesize country="france">35</shoesize>
```

Định nghĩa

```
<xs:element name="shoesize">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:integer">
        <xs:attribute name="country" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

49

Ví dụ (tiếp)

Định nghĩa khác

```
<xs:element name="shoesize" type="shoetype"/>
<xs:complexType name="shoetype">
  <xs:simpleContent>
    <xs:extension base="xs:integer">
      <xs:attribute name="country" type="xs:string" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

50

3.5. Phần tử hỗn hợp

- Chứa cả text và phần tử khác

```
<letter>
  Dear Mr.<name>John Smith</name>.
  Your order <orderid>1032</orderid>
  will be shipped on <shipdate>2001-07-13</shipdate>.
</letter>
```

```
<xs:element name="letter">
  <xs:complexType mixed="true">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="orderid" type="xs:positiveInteger"/>
      <xs:element name="shipdate" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

51

Phần tử hỗn hợp (tiếp)

- Cách khác:

```
<xs:element name="letter" type="lettertype"/>
<xs:complexType name="lettertype" mixed="true">
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="orderid" type="xs:positiveInteger"/>
    <xs:element name="shipdate" type="xs:date"/>
  </xs:sequence>
</xs:complexType>
```

52

3.6. Indicator

- Có 7 indicator:

- Indicator trình tự
 - All
 - Choice
 - Sequence
- Indicator xuất hiện
 - maxOccurs
 - minOccurs
- Indicator nhóm
 - Group name
 - attributeGroup name

53

3.6.1. Indicator trình tự

- Indicator All: phần tử con có thể xuất hiện theo bất kỳ trật tự nào và chỉ xuất hiện 1 lần

```
<xs:element name="person">
  <xs:complexType>
    <xs:all>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:all>
  </xs:complexType>
</xs:element>
```

54

3.6.1. Indicator trình tự (tiếp)

- Indicator Choice: hoặc 1 phần tử này hoặc 1 phần tử kia xuất hiện

```
<xs:element name="person">
  <xs:complexType>
    <xs:choice>
      <xs:element name="employee" type="employee"/>
      <xs:element name="member" type="member"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

55

3.6.1. Indicator trình tự (tiếp)

- Indicator Sequence: các phần tử con phải xuất hiện theo đúng trật tự

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

56

3.6.2. Indicator xuất hiện

- Cho biết tần suất xuất hiện của phần tử
- Indicator maxOccurs: số lần tối đa một phần tử có thể xuất hiện

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="full_name" type="xs:string"/>
      <xs:element name="child_name" type="xs:string" maxOccurs="10"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Để cho phần tử xuất hiện không giới hạn số lần, để maxOccurs="unbounded"

57

3.6.2. Indicator xuất hiện (tiếp)

- Indicator minOccurs: số lần tối thiểu một phần tử có thể xuất hiện

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="full_name" type="xs:string"/>
      <xs:element name="child_name" type="xs:string"
        maxOccurs="10" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Mặc định xuất hiện tối thiểu 1 lần (minOccurs = 1), nếu không đề cập

58

Ví dụ: Myfamily.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<persons xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="family.xsd">
  <person>
    <full_name>Hege Refsnes</full_name>
    <child_name>Cecilie</child_name>
  </person>
  <person>
    <full_name>Tove Refsnes</full_name>
    <child_name>Hege</child_name>
    <child_name>Stale</child_name>
    <child_name>Jim</child_name>
    <child_name>Borge</child_name>
  </person>
  <person>
    <full_name>Stale Refsnes</full_name>
  </person>
</persons>
```

59

family.xsd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="persons">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="person" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="full_name" type="xs:string"/>
              <xs:element name="child_name" type="xs:string"
                minOccurs="0" maxOccurs="5"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

60

3.6.3. Indicator nhóm

- Định nghĩa các tập hợp phần tử có quan hệ
- Các nhóm phần tử

```
<xs:group name="groupname">
...
</xs:group>
```

- Phải định nghĩa phần tử All, Choice, Sequence bên trong khai báo nhóm.

```
<xs:group name="persongroup">
<xs:sequence>
<xs:element name="firstname" type="xs:string"/>
<xs:element name="lastname" type="xs:string"/>
<xs:element name="birthday" type="xs:date"/>
</xs:sequence>
</xs:group>
```

61

3.6.3. Indicator nhóm (tiếp)

- Sau khi định nghĩa nhóm, có thể tham khảo đến nó trong định nghĩa khác

```
<xs:group name="persongroup">
<xs:sequence>
<xs:element name="firstname" type="xs:string"/>
<xs:element name="lastname" type="xs:string"/>
<xs:element name="birthday" type="xs:date"/>
</xs:sequence>
</xs:group>
```

```
<xs:element name="person" type="personinfo"/>
<xs:complexType name="personinfo">
<xs:sequence>
<xs:group ref="persongroup"/>
<xs:element name="country" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

62

3.6.3. Indicator nhóm (tiếp)

- Các nhóm thuộc tính:

- định nghĩa với khai báo như sau:

```
<xs:attributeGroup name="groupname">
...
</xs:attributeGroup>
```

- ví dụ:

```
<xs:attributeGroup name="personattrgroup">
<xs:attribute name="firstname" type="xs:string"/>
<xs:attribute name="lastname" type="xs:string"/>
<xs:attribute name="birthday" type="xs:date"/>
</xs:attributeGroup>
```

63

3.6.3. Indicator nhóm (tiếp)

- Sau khi định nghĩa nhóm thuộc tính, có thể tham khảo đến nó trong định nghĩa khác

```
<xs:attributeGroup name="personattrgroup">
<xs:attribute name="firstname" type="xs:string"/>
<xs:attribute name="lastname" type="xs:string"/>
<xs:attribute name="birthday" type="xs:date"/>
</xs:attributeGroup>
```

```
<xs:element name="person">
<xs:complexType>
<xs:attributeGroup ref="personattrgroup"/>
</xs:complexType>
</xs:element>
```

64

3.7. Phần tử <any>

- Cho phép mở rộng tài liệu XML với các phần tử không được xác định trong lược đồ.
- Ví dụ: khai báo phần tử person. Bằng cách sử dụng phần tử <any>, chúng ta có thể mở rộng nội dung của person (sau <lastname>) với bất kỳ phần tử nào

family.xsd

```
<xs:element name="person">
<xs:complexType>
<xs:sequence>
<xs:element name="firstname" type="xs:string"/>
<xs:element name="lastname" type="xs:string"/>
<xs:any minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
```

65

children.xsd

- Chúng ta muốn mở rộng phần tử person với một phần tử children

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">
```

```
<xs:element name="children">
<xs:complexType>
<xs:sequence>
<xs:element name="childname" type="xs:string"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
```

```
</xs:schema>
```

66

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<persons xmlns="http://www.microsoft.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.microsoft.com family.xsd
http://www.w3schools.com children.xsd">
  <person>
    <firstname>Hege</firstname>
    <lastname>Refsnes</lastname>
    <children>
      <childname>Cecilie</childname>
    </children>
  </person>
  <person>
    <firstname>Stale</firstname>
    <lastname>Refsnes</lastname>
  </person>
</persons>
```

67

3.8. Phần tử <anyAttribute>

- Cho phép chúng ta mở rộng tài liệu XML với các thuộc tính không được xác định bởi lược đồ.

68

- Ví dụ: file family.xsd
- Có thể thêm bất kỳ số lượng thuộc tính vào phần tử person

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
    <xs:anyAttribute/>
  </xs:complexType>
</xs:element>
```

69

- File attribute.xsd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">
  <xs:attribute name="gender">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="male|female"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:schema>
```

70

- File Myfamily.xml sau sử dụng các thành phần từ 2 file family.xsd và attribute.xsd:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<persons xmlns="http://www.microsoft.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:SchemaLocation="http://www.microsoft.com family.xsd
http://www.w3schools.com attribute.xsd">
  <person gender="female">
    <firstname>Hege</firstname>
    <lastname>Refsnes</lastname>
  </person>
  <person gender="male">
    <firstname>Stale</firstname>
    <lastname>Refsnes</lastname>
  </person>
</persons>
```

Hợp lệ vì file family.xsd cho phép thêm thuộc tính vào phần tử person

71

- Như vậy, phần tử <any> và <anyAttribute> được sử dụng để làm cho tài liệu có thể mở rộng. Chúng cho phép thêm các phần tử chưa được khai báo trong XML schema chính.

72

3.9. Phần tử thay thế

- ❑ Với lược đồ XML, một phần tử có thể thay thế một phần tử khác.
- ❑ Ví dụ: có 2 người dùng ở 2 quốc gia khác nhau: Anh và Na Uy. Lược đồ cho phép người Anh sử dụng tên phần tử bằng tiếng Anh, và người Na Uy sử dụng tên phần tử bằng tiếng Na Uy.
- ❑ Quan sát ví dụ:

Khai báo phần tử đầu tiên là name

```
<xs:element name="name" type="xs:string"/>
<xs:element name="navn" substitutionGroup="name"/>
```

Định nghĩa substitutionGroup: phần tử navn có thể thay thế cho phần tử name

Ví dụ

```
<xs:element name="name" type="xs:string"/>
<xs:element name="navn" substitutionGroup="name"/>

<xs:complexType name="custinfo">
  <xs:sequence>
    <xs:element ref="name"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="customer" type="custinfo"/>
<xs:element name="kunde" substitutionGroup="customer"/>
```

```
<customer>
  <name>John Smith</name>
</customer>

<kunde>
  <navn>John Smith</navn>
</kunde>
```

Hai tài liệu hợp lệ

Không cho phép thay thế phần tử

- ❑ Để ngăn không cho phép thay thế phần tử, sử dụng thuộc tính block:

```
<xs:element name="name" type="xs:string" block="substitution"/>
```

Ví dụ:

```
<xs:element name="name" type="xs:string" block="substitution"/>
<xs:element name="navn" substitutionGroup="name"/>

<xs:complexType name="custinfo">
  <xs:sequence>
    <xs:element ref="name"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="customer" type="custinfo" block="substitution"/>
<xs:element name="kunde" substitutionGroup="customer"/>
```

Không cho phép thay thế phần tử (tiếp)

```
<customer>
  <name>John Smith</name>
</customer>
```

Hợp lệ theo lược đồ

```
<kunde>
  <navn>John Smith</navn>
</kunde>
```

Không hợp lệ theo lược đồ

Chú ý

- ❑ Phần tử đầu và phần tử thay thế phải có cùng kiểu
- ❑ Tất cả các phần tử đầu và phần tử thay thế nó phải được khai báo là phần tử toàn cục.
- ❑ Phần tử toàn cục là phần tử con trực tiếp của phần tử <schema>. Các phần tử lồng trong phần tử khác là các phần tử cục bộ.

Bài tập

- ❑ Xây dựng lược đồ cho tài liệu XML sau:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<shiporder orderid="889923"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="shiporder.xsd">
  <orderperson>John Smith</orderperson>
  <shipto>
    <name>Ola Nordmann</name>
    <address>Langgt 23</address>
    <city>4000 Stavanger</city>
    <country>Norway</country>
  </shipto>
  <item>
    <title>Empire Burlesque</title>
    <note>Special Edition</note>
    <quantity>1</quantity>
    <price>10.90</price>
  </item>
  <item>
    <title>Hide your heart</title>
    <quantity>1</quantity>
    <price>9.90</price>
  </item>
</shiporder>
```

Cách 1:

□ File shiporder.xsd:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="shiporder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="orderperson" type="xs:string"/>
      <xs:element name="shipto">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="address" type="xs:string"/>
            <xs:element name="city" type="xs:string"/>
            <xs:element name="country" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

79

Cách 1:

```
<xs:element name="item" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="note" type="xs:string" minOccurs="0"/>
      <xs:element name="quantity" type="xs:positiveInteger"/>
      <xs:element name="price" type="xs:decimal"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:attribute name="orderid" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>

</xs:schema>
```

80

Cách 2:

□ Cách khác: định nghĩa các phần tử, các thuộc tính, rồi tham chiếu đến chúng

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<!-- definition of simple elements -->
<xs:element name="orderperson" type="xs:string"/>
<xs:element name="name" type="xs:string"/>
<xs:element name="address" type="xs:string"/>
<xs:element name="city" type="xs:string"/>
<xs:element name="country" type="xs:string"/>
<xs:element name="title" type="xs:string"/>
<xs:element name="note" type="xs:string"/>
<xs:element name="quantity" type="xs:positiveInteger"/>
<xs:element name="price" type="xs:decimal"/>

<!-- definition of attributes -->
<xs:attribute name="orderid" type="xs:string"/>
```

81

Cách 2:

```
<!-- definition of complex elements -->
<xs:element name="shipto">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="name"/>
      <xs:element ref="address"/>
      <xs:element ref="city"/>
      <xs:element ref="country"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="item">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="title"/>
      <xs:element ref="note" minOccurs="0"/>
      <xs:element ref="quantity"/>
      <xs:element ref="price"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

82

Cách 2:

```
<xs:element name="shiporder">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="orderperson"/>
      <xs:element ref="shipto"/>
      <xs:element ref="item" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute ref="orderid" use="required"/>
  </xs:complexType>
</xs:element>

</xs:schema>
```

83

Cách 3: định nghĩa kiểu

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:simpleType name="stringtype">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<xs:simpleType name="inttype">
  <xs:restriction base="xs:positiveInteger"/>
</xs:simpleType>

<xs:simpleType name="dectype">
  <xs:restriction base="xs:decimal"/>
</xs:simpleType>

<xs:simpleType name="orderidtype">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]{6}"/>
  </xs:restriction>
</xs:simpleType>
```

84

Cách 3:

```
<xs:complexType name="shiptotype">
  <xs:sequence>
    <xs:element name="name" type="stringtype"/>
    <xs:element name="address" type="stringtype"/>
    <xs:element name="city" type="stringtype"/>
    <xs:element name="country" type="stringtype"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="itemtype">
  <xs:sequence>
    <xs:element name="title" type="stringtype"/>
    <xs:element name="note" type="stringtype" minOccurs="0"/>
    <xs:element name="quantity" type="inttype"/>
    <xs:element name="price" type="dectype"/>
  </xs:sequence>
</xs:complexType>
```

85

Cách 3:

```
<xs:complexType name="shipordertype">
  <xs:sequence>
    <xs:element name="orderperson" type="stringtype"/>
    <xs:element name="shipto" type="shiptotype"/>
    <xs:element name="item" maxOccurs="unbounded" type="itemtype"/>
  </xs:sequence>
  <xs:attribute name="orderid" type="orderidtype" use="required"/>
</xs:complexType>

<xs:element name="shiporder" type="shipordertype"/>

</xs:schema>
```

86

4. Các kiểu dữ liệu

- Kiểu chuỗi
- Kiểu Date
- Kiểu số

87

4.1. Kiểu chuỗi

- Chứa các kí tự, kí tự xuống dòng, kí tự về đầu dòng, kí tự tab
- Ví dụ:
 - khai báo


```
<xs:element name="customer" type="xs:string"/>
```
 - phần tử trong tài liệu XML có thể:


```
<customer>John Smith</customer>
```

```
<customer> John Smith </customer>
```
 - Khi sử dụng kiểu chuỗi, bộ xử lý XML sẽ không biến đổi giá trị!

88

Kiểu dữ liệu chuỗi chuẩn hóa

- Dẫn xuất từ kiểu chuỗi
- Chứa các kí tự
- Bộ xử lý XML sẽ loại bỏ các kí tự xuống dòng, về đầu dòng, thay thế tab bằng khoảng trắng
- Khai báo:

```
<xs:element name="customer" type="xs:normalizedString"/>
```

89

Kiểu dữ liệu Token

- Dẫn xuất từ kiểu chuỗi
- Chứa các kí tự
- Bộ xử lý XML sẽ **loại bỏ** các kí tự xuống dòng, kí tự về đầu dòng, kí tự tab, leading and trailing spaces, multiple spaces.
- Khai báo:

```
<xs:element name="customer" type="xs:token"/>
```

90

Tổng hợp các kiểu dữ liệu chuỗi

Tên	Mô tả
ENTITIES	
ENTITY	
ID	Chuỗi biểu diễn thuộc tính ID trong XML
IDREF	Chuỗi biểu diễn thuộc tính IDREF trong XML
IDREFS	
language	Chuỗi chứa id ngôn ngữ hợp lệ
Name	Chuỗi chứa tên XML hợp lệ

91

Tổng hợp các kiểu dữ liệu chuỗi

Tên	Mô tả
NCName	
NMTOKEN	Chuỗi biểu diễn thuộc tính NMTOKEN trong XML
NMTOKENS	
normalizedString	Chuỗi không chứa kí tự xuống dòng, về đầu dòng, tab
QName	
string	
token	Chuỗi không chứa kí tự xuống dòng, về đầu dòng, tab, leading or trailing space, multiple space

92

Các giới hạn trên kiểu chuỗi

- enumeration
- length
- maxLength
- minLength
- pattern (NMTOKENS, IDREFS, and ENTITIES cannot use this constraint)
- whitespace

93

4.2. Kiểu date, time

- Kiểu dữ liệu date
 - Được sử dụng để xác định ngày
 - Dạng: YYYY-MM-DD (phải đầy đủ tất cả)
 - Khai báo trong lược đồ:


```
<xs:element name="start" type="xs:date"/>
```
 - Phần tử trong tài liệu XML:


```
<start>2002-09-24</start>
```

94

Time zones

- Để xác định vùng thời gian theo UTC, thêm chữ Z sau date


```
<start>2002-09-24Z</start>
```
- Xác định offset: thêm thời gian dương/âm


```
<start>2002-09-24-06:00</start>
```

```
<start>2002-09-24+06:00</start>
```

95

Kiểu dữ liệu Time

- Dùng để xác định thời gian
- Dạng: "hh:mm:ss" (phải có đầy đủ)
- Khai báo


```
<xs:element name="start" type="xs:time"/>
```

```
<start>09:00:00</start>
```

```
<start>09:30:10.5</start>
```

96

Time zones

- Tương tự kiểu date, thêm chữ 'Z'

```
<start>09:30:10Z</start>
```

- Hoặc thêm giờ dương/âm:

```
<start>09:30:10-06:00</start>
```

```
<start>09:30:10+06:00</start>
```

97

Kiểu dữ liệu DateTime

- Để xác định date và time
- Dạng: "YYYY-MM-DDThh:mm:ss"
- Phải đầy đủ các thành phần trên
- Khai báo trong schema:

```
<xs:element name="startdate" type="xs:dateTime"/>
```

- Trong tài liệu XML, có thể như sau:

```
<startdate>2002-05-30T09:00:00</startdate>
```

```
<startdate>2002-05-30T09:30:10.5</startdate>
```

98

Time zones

- Tương tự, thêm Z vào đằng sau hoặc thêm giờ dương/âm

```
<startdate>2002-05-30T09:30:10Z</startdate>
```

```
<startdate>2002-05-30T09:30:10-06:00</startdate>
```

```
<startdate>2002-05-30T09:30:10+06:00</startdate>
```

99

Kiểu dữ liệu Duration

- Dùng để xác định khoảng thời gian
- Dạng: "PnYnMnDTnHnMnS"
 - P: chu kỳ
 - nY: số năm
 - nM: số tháng
 - nD: số ngày
 - T: ngăn cách phần date với phần time
 - nH: số giờ
 - nM: số phút
 - nS: số giây

100

Kiểu dữ liệu Duration (tiếp)

- Khai báo trong lược đồ:

```
<xs:element name="period" type="xs:duration"/>
```

- Phần tử trong tài liệu

- Chu kỳ 5 năm:

```
<period>P5Y</period>
```

- Chu kỳ 5 năm 2 tháng 10 ngày:

```
<period>P5Y2M10D</period>
```

- Chu kỳ 5 năm, 2 tháng, 10 ngày 15 giờ:

```
<period>P5Y2M10DT15H</period>
```

- Chu kỳ 15 giờ:

```
<period>PT15H</period>
```

- Thêm dấu trừ: chu kỳ âm

```
<period>-P10D</period>
```

101

Các kiểu dữ liệu Date Time

Tên	Mô tả
date	Định nghĩa 1 giá trị date
dateTime	Định nghĩa 1 giá trị datetime
duration	Định nghĩa 1 khoảng thời gian
gDay	Định nghĩa phần của date - the day (DD)
gMonth	Định nghĩa phần của date - the month (MM)
gMonthDay	Định nghĩa phần của date - the month and day (MM-DD)
gYear	Định nghĩa phần của date - the year (YYYY)
gYearMonth	Định nghĩa phần của date - the year and month (YYYY-MM)
time	Định nghĩa giá trị time

102

Các ràng buộc trên kiểu dữ liệu thời gian

- enumeration
- maxExclusive
- maxInclusive
- minExclusive
- minInclusive
- pattern
- whiteSpace

103

4.3. Kiểu dữ liệu thập phân

- Dùng để xác định kiểu số
- Khai báo trong schema:

```
<xs:element name="prize" type="xs:decimal"/>
```

- Ví dụ:

```
<prize>0</prize>      <prize>999.50</prize>
<prize>14</prize>     <prize>+999.5450</prize>
                        <prize>-999.5230</prize>
```

- Số chữ số tối đa là 18

104

Kiểu dữ liệu số nguyên

- Khai báo trong lược đồ:

```
<xs:element name="prize" type="xs:integer"/>
```

- Ví dụ:

```
<prize>999</prize>    <prize>0</prize>
<prize>+999</prize>  <prize>-999</prize>
```

105

Kiểu dữ liệu numeric

Tên	Mô tả
byte	Số nguyên 8 bit có dấu
decimal	Giá trị thập phân
int	Số nguyên 32 bit có dấu
integer	Giá trị số nguyên
long	Số nguyên 64 bit có dấu
negativeInteger	Số nguyên âm
nonNegativeInteger	Số nguyên không âm

106

Tên	Mô tả
nonPositiveInteger	Số nguyên không dương
positiveInteger	Số nguyên dương
short	Số nguyên 16 bit có dấu
unsignedLong	Số nguyên 64 bit không dấu
unsignedInt	Số nguyên 32 bit không dấu
unsignedShort	Số nguyên 16 bit không dấu
unsignedByte	Số nguyên 8 bit không dấu

107

Các ràng buộc trên kiểu numeric

- enumeration
- fractionDigits
- maxExclusive
- maxInclusive
- minExclusive
- minInclusive
- pattern
- totalDigits
- whiteSpace

108

4.4. Các kiểu dữ liệu khác

- Boolean
- Binary
- AnyURI

109

Kiểu boolean

- Khai báo trong schema:

```
<xs:attribute name="disabled" type="xs:boolean"/>
```

- Giá trị hợp lệ gồm có: true, false, 1(true), 0(false)

- Ví dụ:

```
<prize disabled="true">999</prize>
```

110

Kiểu binary

- Có 2 kiểu
 - base64Binary
 - hexBinary
- Khai báo trong schema:

```
<xs:element name="blobsrc" type="xs:hexBinary"/>
```

111

Kiểu AnyURI

- Khai báo trong schema:

```
<xs:attribute name="src" type="xs:anyURI"/>
```

- Nếu URI có kí tự cách, thay thế nó bởi %20

- Ví dụ:

```
<pic src="http://www.w3schools.com/images/smiley.gif" />
```

112

Một số kiểu

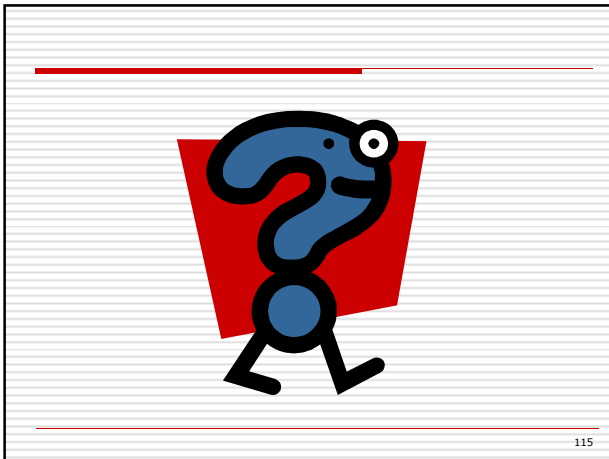
- anyURI
- base64Binary
- boolean
- double
- float
- hexBinary
- NOTATION
- QName

113

Các giới hạn trên những kiểu này

- enumeration (kiểu boolean không sử dụng ràng buộc này)
- length (kiểu boolean không sử dụng ràng buộc này)
- maxLength (kiểu boolean không sử dụng ràng buộc này)
- minLength (kiểu boolean không sử dụng ràng buộc này)
- pattern
- whiteSpace

114



Lời hay ý đẹp

"Hãy lấy sự khôn ngoan làm hành trang cho bạn trong hành trình của cuộc đời, vì đó là sự nâng đỡ đáng tin cậy hơn mọi tài sản khác"

G. Bizet

116