

XML

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE dssk [
<ELEMENT dssk (sk)+ >
<!ATTLIST dssk ten CDATA #IMPLIED>
<ELEMENT sk (dsvdv, kq) >
<!ATTLIST sk ten CDATA #REQUIRED>
<ELEMENT dsvdv (vdv)+ >
<!ELEMENT vdvd (vitri)? >
<!ATTLIST vdvd ten CDATA #REQUIRED>
<!ATTLIST vdvd quocgia CDATA #REQUIRED>
<!ATTLIST vdvd ten CDATA #REQUIRED>
<ELEMENT kq (nhath, nhi, ba) >
<!ELEMENT nhath (vdvd)* >
<!ELEMENT nhi (vdvd)* >
<!ELEMENT ba (vdvd)* >
]>
<dssk>
    <sk ten="dienkinh">
        <dsvdv>
            <vdvd ten="ulsanbolt" quocgia="VN">
            </vdvd>
        </dsvdv>
        <kq>
            <nhath>
            </nhath>
            <nhi>
            </nhi>
            <ba>
            </ba>
        </kq>
    </sk>
</dssk>
```

1. Giới thiệu XML

```
<?xml version="1.0" encoding="utf-8"?>
<student>
    <name>Đàm Minh Tiến</name>
    <age>21</age>
    <class>CNTT</class>
    <description>dz</description>
</student>
```

Văn bản có cấu trúc XML cho phép biểu diễn thông tin về các đối tượng trong thực tế. XML dùng để phục vụ cho việc mô tả dữ liệu (thông tin lưu trữ bao gồm những gì, lưu trữ ra sao) để các hệ thống khác nhau có thể đọc và sử dụng những thông tin này một cách thuận tiện. Các thẻ tag của XML thường không được định nghĩa trước mà chúng được tạo ra theo quy ước của người theo những quy ước riêng. XML sử dụng các khai báo kiểu dữ liệu DTD (Document Type Definition) hay lược đồ Schema để mô tả dữ liệu.

Định chuẩn XML

Có ba kiểu tài liệu XML là:

- Well-formed Document: tài liệu XML chuẩn cú pháp
- DTD - Constrained Document: Tạo XML có khai báo DTD (Document type definition) để mô tả cấu trúc dữ liệu trong XML
- XML-Schema - Constrained Document: Tạo XML có sử dụng lược đồ Schema để kiểm tra tính hợp lệ của XML.

2. Well-formed XML Document

Các quy ước:

- Có duy nhất một phần tử thuộc cấp cao nhất trong tài liệu, còn gọi là nút gốc
- Mỗi một thẻ mở đều phải có thẻ đóng và tên thẻ là phân biệt hoa thường

- Các thẻ khi đóng phải theo đúng trình tự (mở sau đóng trước)
- Tên thẻ không nên có khoảng trống, không nên bắt đầu bằng “xml”
- Các thuộc tính (attributes) của một thẻ luôn luôn tồn tại theo cặp quy ước <tên> = “giá trị”; không nên đặt tên thuộc tính trùng nhau và giá trị của thuộc tính phải đặt trong cặp dấu nháy kép hoặc nháy đơn. Tên của thuộc tính sẽ theo quy luật như tên thẻ.
- Các thẻ (tag) trong XML có thể lồng nhau.

3. DTD (Document Type Definition)

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE BIEU_THUC [
  <!ELEMENT BIEU_THUC (PHAN_SO|TICH_SO)+ >
    <ATTLIST BIEU_THUC Ten CDATA #IMPLIED
    <!-- Ten : A_String -->
  >
  <!ELEMENT PHAN_SO EMPTY >
    <ATTLIST PHAN_SO
      Tu_so CDATA #REQUIRED
      <!-- Tu_so : A_Int -->
      Mau_so CDATA #REQUIRED
      <!-- Mau_so : A_Int // > 0 -->
    >
  <!ELEMENT TICH_SO(PHAN_SO)+ >|
```

Ví dụ đặc tả một biểu thức phân số

DTD định nghĩa các thành phần - thuộc tính và cấu trúc của tài liệu XML.

Các phần trong DTD:

- Elements: các thẻ chính của tài liệu XML
- Attributes: thuộc tính của elements
- Entities: các ký tự có ý nghĩa đặc biệt trong tài liệu XML, ví dụ như “<” đổi thành <
- PCDATA: các đoạn text sẽ được parser chuyển đổi thành dữ liệu
- CDATA: các đoạn text sẽ được parser bỏ qua

Cú pháp đặc tả chung cấu trúc của 1 thẻ

```
<!ELEMENT Ten_the Bieu_thuc_dac_ta_cau_truc>
```

Bieu_thuc sẽ chứa các thẻ con trong Ten_the. Với A, B là 2 thẻ con của thẻ X:

A,B A, B sắp xếp tuần tự A đến B A* A có thể lặp lại ít nhất 0 lần (≥ 0) B+ B có thể lặp lại ít nhất 1 lần (≥ 1) A? A có thể xuất hiện 0 hoặc 1 lần (0 hoặc 1) A|B Có thể sử dụng A hoặc B

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```

Ví dụ trên W3S

4. XML Schema (XSD)

XML Schema hay XML Schema Definition miêu tả cấu trúc tài liệu XML 1 cách chi tiết hơn. Ví dụ 1 tài liệu XML

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

XML well-formed

DTD File:

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

DTD

XSD File:

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

XML Schema File

XML Schema có 3 loại kiểu chính:

- Kiểu định nghĩa sẵn (Built-in Type)
- Kiểu đơn giản (simpleType)
- Kiểu phức hợp (complexType)

Kiểu định nghĩa sẵn:

Được sử dụng khi:

- Thẻ không có thuộc tính
- Thẻ không chứa thẻ khác và có miền giá trị thích hợp với kiểu

Các loại kiểu: string, int, integer, float, double, boolean, date, month, ID, binary
Cú pháp

```
<xs:element name="Ten_the" type="Ten_kieu_co_so" .../><xs:attribute name="Ten_thuoc_tinh" type="Ten_kieu_co_so" .../>
```

Kiểu đơn giản:

Được sử dụng khi:

- Thẻ không có thuộc tính
- Thẻ không chứa thẻ khác và có miền giá trị là tập con của miền giá trị của 1 kiểu cơ sở nào đó

Cú pháp:

```
<xs:simpleType name="Ten_kieu"><xs:restriction base="Ten_kieu_co_so"> Giới hạn trên miền giá trị</xs:restriction></xs:simpleType>
```

Khai báo cận dưới: minInclusive (cho phép sử dụng biên), minExclusive (không sử dụng biên) Khai báo cận trên: maxInclusive (cho phép sử dụng biên), maxExclusive (không sử dụng biên)

Ngoài ra còn có kiểu liệt kê:

```
<xs:simpleType name="HOC_LUC"><xs:restriction base="string"> <xs:enumeration value="Xuất sắc"/><xs:enumeration value="Giỏi"/></xs:restriction></xs:simpleType>
```

Kiểu phức hợp:

Được sử dụng khi:

- Thẻ có thuộc tính
- Thẻ chứa thẻ khác

Các dạng

- Tuần tự
- Tùy chọn
- Lặp

Tuần tự:

```
<xs:complexType name="Ten_kieu"> <xs:sequence>
  Thanh_phan_1
  Thanh_phan_2
  ...
</xs:sequence>
...
</xs:complexType>
```

Tùy chọn:

```
<xs:complexType name="Ten_kieu"> <xs:choice>
  Thanh_phan_1
  Thanh_phan_2
  ...
</xs:choice>
...
</xs:complexType>
```

Lặp:

```
<xs:complexType name="Ten_kieu"> <xs:sequence>    ...
  <xs:element name="Ten_the_con"
    type="Kieu_the_con"          minOccurs="So_lan_lap_min"
    maxOccurs="So_lan_lap_max"
  />
  ...
</xs:sequence>
...
</xs:complexType>
```

Thuộc tính của thẻ:

```
<xs:attribute name="Ten_thuoc_tinh"          type="Kieu_thuoc_tinh"
  Tu_khoa_1="Gia_tri_1"          Tu_khoa_2="Gia_tri_2"/>
```

Một số tính chất thông dụng:

- default: giá trị mặc định
- fixed: giá trị cố định
- use: tùy chọn sử dụng (có hay không)