# XML

## Tạo một tệp XML đơn giản

Hãy tạo một tệp XML tên **thuvien.xml** mô tả danh sách sách trong một thư viện. Mỗi quyển sách cần có các thông tin sau:

* **Tiêu đề** (title)
* **Tác giả** (author)
* **Năm xuất bản** (year)
* **Thể loại** (genre)

**Yêu cầu:**

* Tạo một tệp XML chứa ít nhất 3 quyển sách.
* Đảm bảo XML có cấu trúc hợp lệ.

## Xây dựng XML có cấu trúc lồng nhau

Hãy tạo một tệp XML **sinhvien.xml** mô tả danh sách sinh viên trong một lớp học. Mỗi sinh viên có:

* **Họ và tên** (name)
* **Mã sinh viên** (id)
* **Danh sách các môn học** (subjects), mỗi môn học gồm:
  + **Tên môn** (subject\_name)
  + **Điểm số** (score)

## Sử dụng XML Attributes để lưu thông tin sản phẩm

Hãy tạo một tệp XML **sanpham.xml** mô tả danh sách sản phẩm của một cửa hàng. Mỗi sản phẩm có các thuộc tính:

* id (mã sản phẩm, duy nhất)
* category (loại sản phẩm, ví dụ: "Điện tử", "Thời trang")
* price (giá sản phẩm)
* currency (đơn vị tiền tệ, ví dụ: "USD", "VND")

**Yêu cầu:**

* Dùng XML **attributes** thay vì XML elements để lưu trữ thông tin sản phẩm.
* Tạo ít nhất 3 sản phẩm.

## Sử dụng XML Namespace để quản lý dữ liệu nhân sự

Tạo file XML **nhansu.xml**

Một công ty có nhiều phòng ban, mỗi phòng ban có nhân viên với thông tin như:

* name (tên nhân viên)
* id (mã nhân viên)
* position (chức vụ)

Để tổ chức dữ liệu tốt hơn, công ty sử dụng **XML Namespace**:

* Namespace xmlns:hr="http://company.com/hr" để chứa thông tin nhân sự.
* Namespace xmlns:dep="http://company.com/department" để chứa thông tin phòng ban.

**Yêu cầu:**

* Sử dụng **namespace prefix** (hr: và dep:) để phân biệt dữ liệu.
* Tạo ít nhất 2 phòng ban và 3 nhân viên.

**Ví dụ XML mẫu:**

<company xmlns:hr="http://company.com/hr" xmlns:dep="http://company.com/department">

<dep:department name="Kế toán">

<hr:employee id="E001">

<hr:name>Nguyễn Văn A</hr:name>

<hr:position>Kế toán trưởng</hr:position>

</hr:employee>

</dep:department>

</company>

## Kết hợp XML Attributes và XML Namespace trong đơn đặt hàng

Tạo file XML **donhang.xml**

Giả sử một công ty có hệ thống quản lý đơn hàng, mỗi đơn hàng gồm:

* id (mã đơn hàng, duy nhất)
* customer (tên khách hàng)
* date (ngày đặt hàng)
* Danh sách **sản phẩm**, mỗi sản phẩm có:
  + id (mã sản phẩm)
  + quantity (số lượng)
  + price (giá)

Sử dụng **XML Namespace** để phân loại dữ liệu:

* Namespace xmlns:order="http://company.com/order" cho thông tin đơn hàng.
* Namespace xmlns:prod="http://company.com/product" cho thông tin sản phẩm.

**Yêu cầu:**

* Sử dụng **namespace** để phân biệt dữ liệu đơn hàng và sản phẩm.
* Dùng **attributes** để lưu thông tin sản phẩm.
* Tạo ít nhất 2 đơn hàng với nhiều sản phẩm.

**Ví dụ XML mẫu:**

<orders xmlns:order="http://company.com/order" xmlns:prod="http://company.com/product">

<order:order id="O001" customer="Nguyễn Văn A" date="2025-03-02">

<prod:item id="P001" quantity="2" price="500">

<prod:name>Điện thoại Samsung</prod:name>

</prod:item>

<prod:item id="P002" quantity="1" price="1000">

<prod:name>Laptop Dell</prod:name>

</prod:item>

</order:order>

</orders>

## Kết hợp nhiều XML Namespace trong một tài liệu XML

Tạo file XML **khoahoc.xml**

Giả sử bạn đang thiết kế một hệ thống quản lý **khóa học trực tuyến**. Dữ liệu cần lưu trữ bao gồm:

* **Thông tin khóa học** (course):
  + id (Mã khóa học)
  + title (Tiêu đề khóa học)
  + category (Danh mục khóa học, ví dụ: "Lập trình", "Thiết kế")
* **Thông tin giảng viên** (instructor):
  + id (Mã giảng viên)
  + name (Tên giảng viên)
  + email (Email giảng viên)
* **Thông tin sinh viên đăng ký** (student):
  + id (Mã sinh viên)
  + name (Tên sinh viên)
  + progress (Tiến độ học tập theo phần trăm)

**Yêu cầu:**

* Sử dụng **3 namespace** để phân loại dữ liệu:
  + xmlns:course="http://example.com/course" cho thông tin khóa học.
  + xmlns:instructor="http://example.com/instructor" cho thông tin giảng viên.
  + xmlns:student="http://example.com/student" cho thông tin sinh viên.
* Kết hợp namespace với XML attributes để lưu thông tin.
* Tạo ít nhất **2 khóa học**, mỗi khóa học có một giảng viên và một danh sách sinh viên.

**Ví dụ XML mẫu:**

<elearning xmlns:course="http://example.com/course"

xmlns:instructor="http://example.com/instructor"

xmlns:student="http://example.com/student">

<course:course id="C001" category="Lập trình">

<course:title>Python cơ bản</course:title>

<instructor:instructor id="I001">

<instructor:name>Nguyễn Văn A</instructor:name>

<instructor:email>nguyenvana@example.com</instructor:email>

</instructor:instructor>

<course:students>

<student:student id="S001" progress="80">

<student:name>Trần Thị B</student:name>

</student:student>

<student:student id="S002" progress="50">

<student:name>Hoàng Minh C</student:name>

</student:student>

</course:students>

</course:course>

</elearning>

## Sử dụng XML Namespace với Default Namespace và Prefixes

Tạo file XML **benhvien.xml**

Bạn đang xây dựng một hệ thống **quản lý bệnh viện**, trong đó có:

* **Thông tin bệnh nhân** (patient):
  + id (Mã bệnh nhân)
  + name (Tên bệnh nhân)
  + dob (Ngày sinh)
* **Thông tin bác sĩ** (doctor):
  + id (Mã bác sĩ)
  + name (Tên bác sĩ)
  + specialty (Chuyên khoa)
* **Thông tin cuộc hẹn** (appointment):
  + id (Mã cuộc hẹn)
  + date (Ngày hẹn)
  + status (Trạng thái: "Đã khám", "Chưa khám")

**Yêu cầu:**

* Sử dụng **default namespace** (xmlns="http://example.com/hospital") cho toàn bộ tài liệu.
* Dùng namespace riêng để phân loại dữ liệu:
  + xmlns:pat="http://example.com/patient" cho bệnh nhân.
  + xmlns:doc="http://example.com/doctor" cho bác sĩ.
  + xmlns:app="http://example.com/appointment" cho cuộc hẹn.
* Kết hợp XML attributes với namespace để lưu thông tin.
* Tạo ít nhất **2 cuộc hẹn**, mỗi cuộc hẹn có 1 bệnh nhân và 1 bác sĩ.

**Ví dụ XML mẫu:**

<hospital xmlns="http://example.com/hospital"

xmlns:pat="http://example.com/patient"

xmlns:doc="http://example.com/doctor"

xmlns:app="http://example.com/appointment">

<app:appointment id="A001" date="2025-03-05" status="Đã khám">

<pat:patient id="P001">

<pat:name>Nguyễn Văn A</pat:name>

<pat:dob>1990-07-15</pat:dob>

</pat:patient>

<doc:doctor id="D001">

<doc:name>Trần Thị B</doc:name>

<doc:specialty>Nội khoa</doc:specialty>

</doc:doctor>

</app:appointment>

</hospital>

# **DTD**

## Xây dựng XML và DTD theo mẫu

### Xây dựng DTD cho danh sách sinh viên

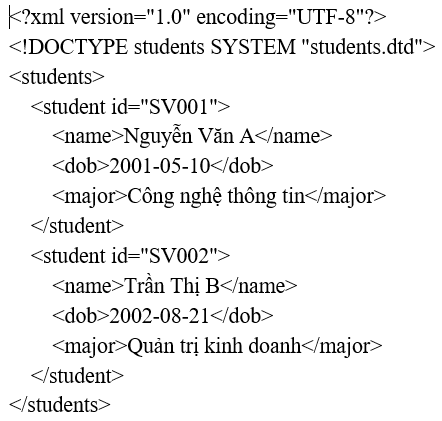
Giả sử bạn cần tạo một tệp XML lưu trữ danh sách sinh viên, trong đó mỗi sinh viên có:

* **Mã sinh viên** (id, dạng thuộc tính)
* **Họ tên** (name)
* **Ngày sinh** (dob)
* **Chuyên ngành** (major)

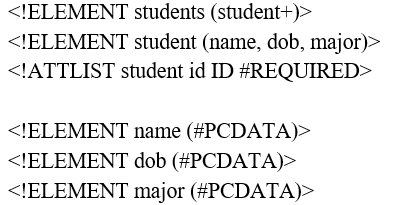
**Yêu cầu:**

1. Viết một tệp XML chứa ít nhất 2 sinh viên.
2. Viết một DTD để kiểm tra tính hợp lệ của XML.

**Ví dụ XML:**



**DTD tương ứng (students.dtd):**



### Xây dựng DTD cho danh sách sách trong thư viện

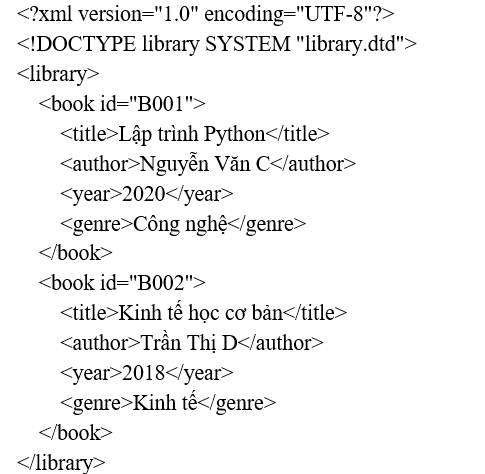
Một thư viện lưu trữ danh sách các cuốn sách, mỗi sách có:

* **ID** (id, dạng thuộc tính)
* **Tiêu đề** (title)
* **Tác giả** (author)
* **Năm xuất bản** (year)
* **Thể loại** (genre)

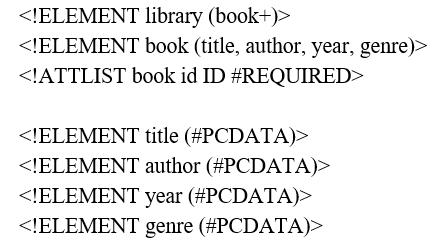
**Yêu cầu:**

1. Viết một tệp XML chứa ít nhất 3 quyển sách.
2. Viết một DTD kiểm tra tính hợp lệ.

**Ví dụ XML:**



**DTD tương ứng (library.dtd):**



### Xây dựng DTD cho hệ thống đơn hàng

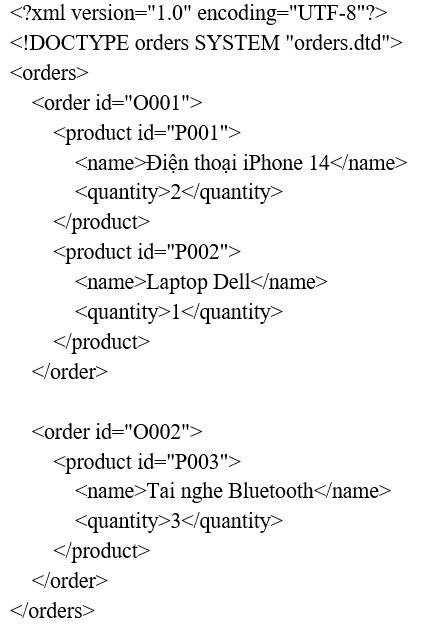
Một công ty quản lý các đơn hàng với thông tin sau:

* **Đơn hàng** (order) gồm nhiều **sản phẩm** (product).
* Mỗi đơn hàng có **ID** (id, dạng thuộc tính).
* Mỗi sản phẩm có:
  + **ID sản phẩm** (id, dạng thuộc tính)
  + **Tên sản phẩm** (name)
  + **Số lượng** (quantity)

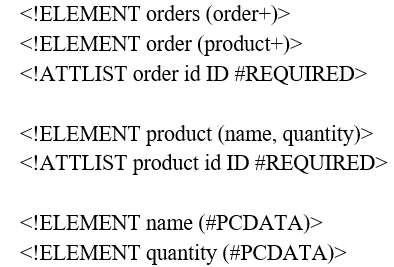
**Yêu cầu:**

1. Viết một tệp XML chứa ít nhất 2 đơn hàng, mỗi đơn hàng có nhiều sản phẩm.
2. Viết một DTD kiểm tra tính hợp lệ.

**Ví dụ XML:**



**DTD tương ứng (orders.dtd):**



## Xây dựng cac DTD cho các bài XML ở buổi 1

## Hoàn chỉnh ERD bài báo cáo.