

Consegna esercizio

Elaborare in XML una gestione libri che contenga almeno i dati di seguito esposti in forma tabellare:

libro
titolo
edizione

autori (da 1 all'infinito)
cognome
nome

ISBN (almeno 1 con max 13 caratteri)
casaeditrice
anno (intero positivo)

scontistica (max 1 volta)
sconto (intero positivo)
data di inizio (date)
data di fine (date)

disponibilita (intero positivo)

Completare tale tabella con campi propri personalizzati.

Dato lo schema prima esposto creare

- il relativo file XSD

- un file xml valido contenente almeno due record di libri secondo l'XSD

- un file css che permetta la visualizzazione a schermo in modo comprensibile del contenuto del file xml.

I tre file vanno compressi in un unico file zip denominato con il proprio COGNOME e NOME e spedito a g.pistorio@gferraris.it entro sabato 5 aprile 2025.

Ripetere l'intera sequenza di operazioni per l'esercizio iniziato in classe su un ipotetico ordine Amazon dove i campi XSD vanno elaborati in modo personalizzato.

Il file XML deve essere valido assieme al suo schema XSD, che ne definisce la grammatica = funziona!

Sito: <https://www.liquid-technologies.com/online-xsd-validator>

XML (eXtensible Markup Language)

Cos'è XML?

XML è un linguaggio di markup che definisce un insieme di regole per la codifica di documenti in un formato leggibile sia da umani che da macchine. È progettato per memorizzare e trasportare dati in modo indipendente dal software e dall'hardware.

Caratteristiche principali di XML

- **Estensibilità:** consente di definire tag personalizzati per descrivere precisamente i dati
- **Struttura ad albero:** organizza i dati in una gerarchia di elementi
- **Separazione di dati e presentazione:** i dati XML possono essere visualizzati in diversi modi
- **Auto-descrizione:** i tag descrivono il significato del contenuto
- **Standard W3C:** è uno standard gestito dal World Wide Web Consortium
- **Indipendenza dalla piattaforma:** funziona su qualsiasi piattaforma e con qualsiasi linguaggio di programmazione

Sintassi di base XML

1. **Dichiarazione XML:** specifica la versione XML e l'encoding

```
<?xml version="1.0" encoding="UTF-8"?>
```

2. **Elementi:** unità fondamentali, costituiti da tag di apertura, contenuto e tag di chiusura

```
<libro>Clean Code</libro>
```

3. **Attributi:** forniscono informazioni aggiuntive sugli elementi

```
<libro id="123" disponibile="true">Clean Code</libro>
```

4. **Elementi nidificati:** creano una struttura gerarchica

```
<libro>  
  <titolo>Clean Code</titolo>  
  <autore>Robert C. Martin</autore>  
</libro>
```

5. **Elementi vuoti:** elementi senza contenuto

```
<immagine src="copertina.jpg" />
```

6. **Commenti:** per annotazioni nel codice

```
<!-- Questo è un commento -->
```

Regole di sintassi XML

- XML è case-sensitive: `<libro>` e `<Libro>` sono tag diversi
- Tutti gli elementi devono avere tag di chiusura o essere auto-chiusi
- Gli elementi devono essere correttamente nidificati
- I valori degli attributi devono essere racchiusi tra virgolette
- I nomi degli elementi devono iniziare con una lettera o underscore
- I caratteri speciali (`<` , `>` , `&` , `'` , `"`) devono essere sostituiti con le entità corrispondenti
- Un documento XML deve avere un solo elemento radice

Documenti XML ben formati e validi

- **Ben formato:** rispetta tutte le regole sintattiche di XML
- **Valido:** oltre ad essere ben formato, è conforme a una definizione di struttura (DTD, XSD, ecc.)

XSD (XML Schema Definition)

Cos'è XSD?

XSD è un linguaggio di schema per XML che definisce la struttura, i vincoli e la tipizzazione dei dati in un documento XML. Consente di verificare che un documento XML sia valido rispetto a un modello predefinito.

Vantaggi di XSD rispetto a DTD

- **Supporto per i tipi di dati:** consente di specificare tipi come integer, string, date, ecc.
- **Supporto per i namespace:** permette di combinare definizioni di schema diverse
- **Estensibilità:** consente di creare tipi di dati personalizzati
- **Più espressivo:** permette di definire vincoli più dettagliati
- **Sintassi XML:** è esso stesso un documento XML
- **Supporto per l'ereditarietà:** permette relazioni di derivazione tra tipi

Struttura di base di un XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- Definizioni di elementi e tipi -->
</xs:schema>
```

Componenti principali di XSD

1. Elementi

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

2. Attributi

```
<xs:attribute name="id" type="xs:ID"/>
```

3. Tipi semplici

Puoi inserire dati di vario tipo (es. stringa, valore., etc.) usando "restriction" per indicare che tipo è (es. intero, stringa) per un singolo attributo

```
<xs:simpleType name="tipoISBN">
  <xs:restriction base="xs:string">
    <xs:attribute name="input" type="xs:string">
  </xs:restriction>
</xs:simpleType>
```

4. Tipi complessi

Complesso = racchiude una serie di attributi al suo interno = avere più "element".

```
<xs:complexType name="tipoLibro">
  <xs:sequence>
    <xs:element name="titolo" type="xs:string"/>
    <xs:element name="autore" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID"/>
</xs:complexType>
```

5. Gruppi di elementi

```
<xs:group name="infoEditore">
  <xs:sequence>
    <xs:element name="nome" type="xs:string"/>
    <xs:element name="sede" type="xs:string"/>
  </xs:sequence>
</xs:group>
```

```
</xs:sequence>
</xs:group>
```

Vincoli comuni in XSD

1. **Cardinalità**: specificare quante volte un elemento può apparire

```
<xs:element name="autore" minOccurs="1" maxOccurs="unbounded"/>
```

2. **Restrizioni sui valori**:

```
<xs:restriction base="xs:integer">
  <xs:minInclusive value="1"/>
  <xs:maxInclusive value="100"/>
</xs:restriction>
```

3. **Pattern di espressioni regolari**:

```
<xs:pattern value="[A-Z][0-9]{5}"/>
```

4. **Enumerazioni**:

```
<xs:restriction base="xs:string">
  <xs:enumeration value="Disponibile"/>
  <xs:enumeration value="Esaurito"/>
  <xs:enumeration value="Prenotabile"/>
</xs:restriction>
```

Compositori in XSD

1. **sequence**: gli elementi devono apparire nell'ordine specificato

```
<xs:sequence>
  <xs:element name="titolo" type="xs:string"/>
  <xs:element name="autore" type="xs:string"/>
</xs:sequence>
```

2. **choice**: solo uno degli elementi può apparire

```
<xs:choice>
  <xs:element name="isbn10" type="xs:string"/>
```

```
<xs:element name="isbn13" type="xs:string"/>
</xs:choice>
```

3. **all**: tutti gli elementi devono apparire, ma in qualsiasi ordine

```
<xs:all>
  <xs:element name="titolo" type="xs:string"/>
  <xs:element name="autore" type="xs:string"/>
</xs:all>
```

CSS per XML

Cos'è CSS per XML?

CSS (Cascading Style Sheets) può essere utilizzato per formattare e visualizzare documenti XML in un browser web.

Come collegare CSS a un documento XML

```
<?xml-stylesheet type="text/css" href="stile.css"?>
```

Selettori CSS per XML

- **Selettore di elemento**: corrisponde a un elemento XML

```
libro {
  display: block;
  margin-bottom: 20px;
}
```

- **Selettore di attributo**: corrisponde a elementi con un certo attributo

```
libro[disponibile="true"] {
  color: green;
}
```

- **Pseudo-classi e pseudo-elementi**: per selezionare stati o parti di elementi

```
libro:first-child {
  font-weight: bold;
}
```

Proprietà CSS importanti per XML

- **display:** definisce come un elemento deve essere visualizzato

```
elemento {  
  display: block; /* o inline, table, ecc. */  
}
```

- **Proprietà di contenuto:** aggiunge contenuto prima o dopo un elemento

```
elemento::before {  
  content: "Testo prima dell'elemento";  
}
```

- **Proprietà di bordo, margine e padding:** per il layout

```
elemento {  
  border: 1px solid black;  
  margin: 10px;  
  padding: 5px;  
}
```

- **Proprietà di colore e sfondo:** per l'aspetto visivo

```
elemento {  
  color: #333;  
  background-color: #f5f5f5;  
}
```

Validazione dei documenti XML

Metodi di validazione

1. **Controllo di buona formazione:** verifica la sintassi XML
2. **Validazione con XSD:** verifica la conformità a uno schema
3. **Strumenti di validazione:** editor XML, validatori online, librerie di programmazione

Processo di validazione

1. Caricare il documento XML
2. Caricare lo schema XSD
3. Eseguire la validazione
4. Gestire gli errori trovati

Attributi per collegare uno schema a un documento XML

```
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="schema.xsd">
  <!-- Contenuto XML -->
</root>
```