

DTD_e_XSD_approfonditi

XML Schema: DTD e XSD Approfonditi

Appunti universitari – Metodologie di Programmazione

Prof. Lorenzo Rossi – A.A. 2024/2025

Formattato per Obsidian 

Cos'è uno schema XML?

Uno **schema XML** è un insieme di regole che definisce la **struttura**, il **contenuto** e gli **attributi** ammessi in un documento XML.

A cosa serve:

- Convalidare un file XML.
- Descrivere formalmente la struttura dei dati.
- Abilitare l'interoperabilità tra applicazioni.

Esistono principalmente due linguaggi di schema:

-  **DTD** – Document Type Definition (più semplice, non XML)
 -  **XSD** – XML Schema Definition (più potente, scritto in XML)
-

DTD – Document Type Definition

Cos'è una DTD?

Una **DTD** è una specifica che descrive la **struttura legale di un documento XML**, tramite:

- **Elementi** e loro contenuto
 - **Attributi** e vincoli
 - **Entità** e contenuti ricorrenti
-

Dichiarazione di tipo

Un documento XML può dichiarare la sua DTD:

◆ Interna:

```
<!DOCTYPE libro [  
  <!ELEMENT libro (titolo, autore)>  
  <!ELEMENT titolo (#PCDATA)>  
  <!ELEMENT autore (#PCDATA)>  
>
```

◆ Esterna:

```
<!DOCTYPE libro SYSTEM "libro.dtd">
```

Elementi in DTD

Sintassi:

```
<!ELEMENT nomeElemento tipoContenuto>
```

Tipi di contenuto:

- #PCDATA → solo testo
- EMPTY → nessun contenuto
- ANY → qualsiasi contenuto
- (a, b, c) → sottoelementi
- (#PCDATA | bold | italic)* → contenuto misto

Esempio:

```
<!ELEMENT messaggio (#PCDATA)>  
<!ELEMENT nota (titolo, corpo)>
```

Cardinalità e Strutture

Simbolo	Significato
?	opzionale (0 o 1)
*	zero o più
+	una o più
assente	esattamente uno

Esempio:

```
<!ELEMENT lista (item*)>
<!ELEMENT item (#PCDATA)>
```

→ `<lista>` può avere nessun `<item>`, uno o molti.



Alternative

Uso della barra verticale:

```
<!ELEMENT blocco (testo | immagine)>
```

→ `<blocco>` può contenere o `<testo>` o `<immagine>`.



Attributi con ATTLIST

Sintassi:

```
<!ATTLIST elemento nome tipo vincolo>
```

Tipi:

- `CDATA` → stringa
- `(a | b | c)` → enumerazione
- `ID`, `IDREF`, `IDREFS`

Vincoli:

- `#REQUIRED` → obbligatorio

- #IMPLIED → opzionale
- #FIXED "valore" → valore fisso

Esempio:

```
<!ATTLIST persona titolo (Sig | Sig.ra | Dr) "Sig">
```

Entità

Interna:

```
<!ENTITY autore "Douglas Hofstadter">  
<autore>&autore;</autore>
```

Esterna:

```
<!ENTITY testo SYSTEM "introduzione.txt">
```

Esempio completo DTD

```
<!DOCTYPE libro [  
  <!ELEMENT libro (titolo, autore, anno)>  
  <!ELEMENT titolo (#PCDATA)>  
  <!ELEMENT autore (#PCDATA)>  
  <!ELEMENT anno (#PCDATA)>  
<libro>  
  <titolo>GEB</titolo>  
  <autore>Douglas Hofstadter</autore>  
  <anno>1990</anno>  
</libro>
```

DTD - Pro e Contro

Vantaggi:

- Sintassi semplice

- Facile da leggere
- Supportato in molti parser

✗ Svantaggi:

- Nessuna gestione di tipi numerici o date
- Nessuna riusabilità di tipi
- Non scritto in XML → difficile da estendere

XSD – XML Schema Definition

Cos'è XSD?

XSD (XML Schema Definition) è lo standard W3C per definire la struttura di documenti XML. Rispetto a DTD:

- È scritto in XML
- Supporta **tipi di dati forti**
- Consente la **riusabilità** tramite tipi complessi

Struttura base

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- Definizioni -->
</xs:schema>
```

Per collegare lo schema a un file XML:

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

Elementi semplici

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

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

Tipi supportati:

- xs:string , xs:integer , xs:decimal
 - xs:boolean , xs:date , xs:dateTime
-

Restrizioni sui dati

Numeri:

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

Stringhe:

```
<xs:element name="nome">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="2"/>
      <xs:maxLength value="30"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Enumerazioni

```
<xs:element name="stagione">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Primavera"/>
      <xs:enumeration value="Estate"/>
      <xs:enumeration value="Autunno"/>
      <xs:enumeration value="Inverno"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

```
</xs:simpleType>
</xs:element>
```

Elementi complessi

Con `xs:sequence` (ordine richiesto)

```
<xs:element name="persona">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="nome" type="xs:string"/>
      <xs:element name="cognome" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Con `xs:choice` (uno tra molti)

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

Attributi in XSD

```
<xs:attribute name="lingua" type="xs:string" use="optional"/>
```

Valori per `use`: `optional`, `required`, `prohibited`

Tipi personalizzati

Definizione:

```
<xs:complexType name="Persona">
  <xs:sequence>
    <xs:element name="nome" type="xs:string"/>
    <xs:element name="cognome" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

```
</xs:sequence>
</xs:complexType>
```

Uso:

```
<xs:element name="studente" type="Persona"/>
```

Esempio completo XSD

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="libro">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="titolo" type="xs:string"/>
        <xs:element name="autore" type="xs:string"/>
        <xs:element name="anno" type="xs:positiveInteger"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

XSD - Pro e Contro

Vantaggi:

- Sintassi XML
- Tipi di dato forti
- Riusabilità di definizioni
- Estendibile e potente

Svantaggi:

- Verboso
- Più complesso da scrivere rispetto a DTD

Risorse utili

- [XML Validator Online](#)

- [W3C Schema Reference](#)
 - Strumenti: Oxygen XML Editor, XMLSpy, Visual Studio Code (plugin)
-