

Il Metalinguaggio XML

STRUTTURA, SINTASSI, ESEMPI ED UTILIZZI

Francesco Gobbi

16 gennaio 2026

Cos'è XML

- XML = eXtensible Markup Language.
- È un **metalinguaggio**, cioè serve a definire altri linguaggi.
- Progettato per descrivere dati, non per visualizzarli.
- È indipendente da piattaforma e tecnologia.
- Basato su una sintassi rigorosa utile allo scambio di informazioni.

Obiettivi principali di XML:

- **Risolvere il problema della condivisione** dei dati tra piattaforme eterogenee.
- Permettere la **definizione della struttura dei dati** in modo standard.
- Favorire l'**interoperabilità tra sistemi e applicazioni**.
- Rendere i documenti leggibili sia da umani che da programmi.

Attenzione! XML non è...

- **HTML**: XML non definisce grafica o presentazione.
- **Java**: non esegue istruzioni, non contiene logica.
- **HTTP**: non è un protocollo di comunicazione.
- **MySQL**: non memorizza dati, li descrive.

XML come linguaggio formale

- Basato su una **grammatica formale** che definisce le regole sintattiche.
- Non possiede tag predefiniti: piena libertà di definizione.
- La struttura dei dati è descritta dall'utente.
- I tag hanno significato semantico, non grafico.

XML come metalinguaggio:

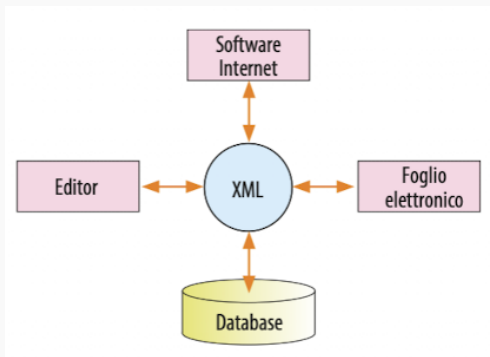
- Può essere usato per creare linguaggi personalizzati.
- Organizza i dati in maniera gerarchica.
- Permette rappresentazioni complesse tramite nested elements.
- Offre una sintassi universale per descrivere contenuti.

Caratteristiche dei file XML

- XML utilizza un **formato di file testuale**:
 - sia il markup sia il contenuto sono stringhe di caratteri;
 - il file è leggibile anche senza software specifici.
- I documenti XML si basano sulla codifica dei caratteri **ISO 10646 / Unicode**.
- Questo garantisce un forte supporto alla **internazionalizzazione**:
 - è possibile scrivere documenti che mescolano alfabeti diversi;
 - non servono convenzioni o parametri esterni per riconoscere la lingua;
 - la codifica dei caratteri identifica automaticamente l'alfabeto usato.
- Un documento XML è **leggibile da un utente umano** anche senza la mediazione di applicazioni dedicate.

Utilizzi principali dell'XML

- Scambio dati Web → Server/Client.
- Integrazione applicazioni aziendali.
- Configurazione software (es. `web.xml` di Java EE).
- Memorizzazione strutturata all'interno dell'HTML (data islands).
- Generazione di documenti tramite XSLT.



Vantaggi per lo scambio dei dati

- Riduce incompatibilità tra software.
- Evita conversioni multiple tra formati proprietari.
- Standard de facto dello scambio commerciale su Internet.
- Compatibile con database, fogli elettronici, editor.

Struttura di un documento XML: la Sintassi

- **Prologo:** dichiarazione XML + eventuali riferimenti.
- **Corpo:** contenuto strutturato tramite tag.
- Organizzazione gerarchica → albero.
- Un solo elemento radice (root).

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/css" href="greco.css"?>
```

} Prologo

```
<radice>
```

Marcatore di inizio

```
  <!-- Questo è un commento -->
```

```
  <figlio>
```

```
  ...
```

```
  </figlio>
```

```
  <figlio>
```

```
  ...
```

```
  </figlio>
```

Marcatore di fine

```
</radice>
```

} Corpo

XML Declaration: il Prologo

- Indica versione XML: es. `version="1.0"`.
- Indica codifica dei caratteri: es. `encoding="UTF-8"`.
- Parametro `standalone` specifica se è richiesto un DTD esterno (DTD = Document Type Definition).

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

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

Versione di XML

Codifica
dei caratteri

Indica se il documento è
conforme a una sintassi
esterna (`standalone="no"`)

- In XML esistono **regole sintattiche**:
 - definiscono **come** scrivere correttamente un documento XML;
 - se non rispettate, il documento **non è XML valido per un parser**.
- Possono esistere anche **regole semantiche** (opzionali):
 - definiscono **cosa** è consentito scrivere (struttura attesa, elementi ammessi, ordine, tipi);
 - vengono fornite da una **grammatica** (es. XML Schema).

Ben formato (well-formed) vs Valido (valid)

Definizioni

- Un documento XML che rispetta le **regole sintattiche** è **ben formato**.
- Un documento XML che rispetta le **regole sintattiche** + **regole semantiche** è **valido**.

Relazione fondamentale

Un documento **valido** è **sempre ben formato**, ma un documento **ben formato** può **non essere valido**.

Un documento XML è ben formato se:

- contiene una **dichiarazione XML corretta** (se presente);
- il corpo ha **un unico elemento radice** (root), esclusi eventuali commenti;
- ogni elemento ha **tag di apertura e tag di chiusura**:
 - se l'elemento è vuoto si può usare la forma abbreviata `<nomeTag/>`.
- i tag sono **correttamente nidificati**:
 - chiusura in ordine inverso rispetto all'apertura (niente sovrapposizioni).

Un documento XML è ben formato se:

- i nomi dei tag di apertura e chiusura **coincidono perfettamente**
 - anche per **maiuscole/minuscole** (XML è **case-sensitive**).
- i nomi dei tag:
 - **non iniziano** con underscore (`_`) o con un numero;
 - **non contengono spazi**.
- i valori degli attributi sono sempre racchiusi tra **apici singoli o doppi**
 - es: `tipo="casa"` oppure `tipo='casa'`.

Errori tipici (ben-formazione)

- Tag non chiuso: `<nome>`
- Tag sovrapposti:
 - `<a>`
- Maiuscole/minuscole diverse:
 - `<Titolo>...</titolo>`
- Più elementi radice:
 - `<a/> `
- Attributo senza virgolette:
 - `<x tipo=casa>`