<?xml version="1.0"?>

Introduzione ad XML

Introduzione ad XML

- eXtensible Markup Language
- Le specifiche sono presenti sul sito http://www.w3.org/XML

- Metalinguaggio per la definizione di linguaggi di markup
- Permette di definire la struttura di documenti e dati

- Software: Notepad++, Brackets, Atom, Visual Studio Code, ecc

Introduzione ad XML

- XML è un insieme di regole sintattiche per modellare la struttura di documenti e dati:
  - Le regole sono standard. Garantiscono l'indipendenza da una specifica piattaforma hardware e software
  - Non permettono di specificare altre caratteristiche come il tipo o la presentazione dei dati o documenti

- Descrive i dati e non la loro rappresentazione!

A cosa serve XML

- Data Interchange
  - Scambio di informazioni e dati tra più programmi
- Document publishing
  - lo stesso documento XML può essere usato e trasformato per la stampa, il Web, il cellulare, ecc
- Interazione tra database eterogenei
- Android Manifest, interfacce delle activity …

Perché XML

- Documenti autodescrittivi
  - La scelta dei nomi può essere fatta per facilitarne la comprensione
- Struttura navigabile dei documenti
  - La struttura ad albero rendono semplice la navigazione
- Platform independence
  - XML è uno standard aperto
- Facile convertibilità
  - La conversione tra formati anche di diversa natura è semplice

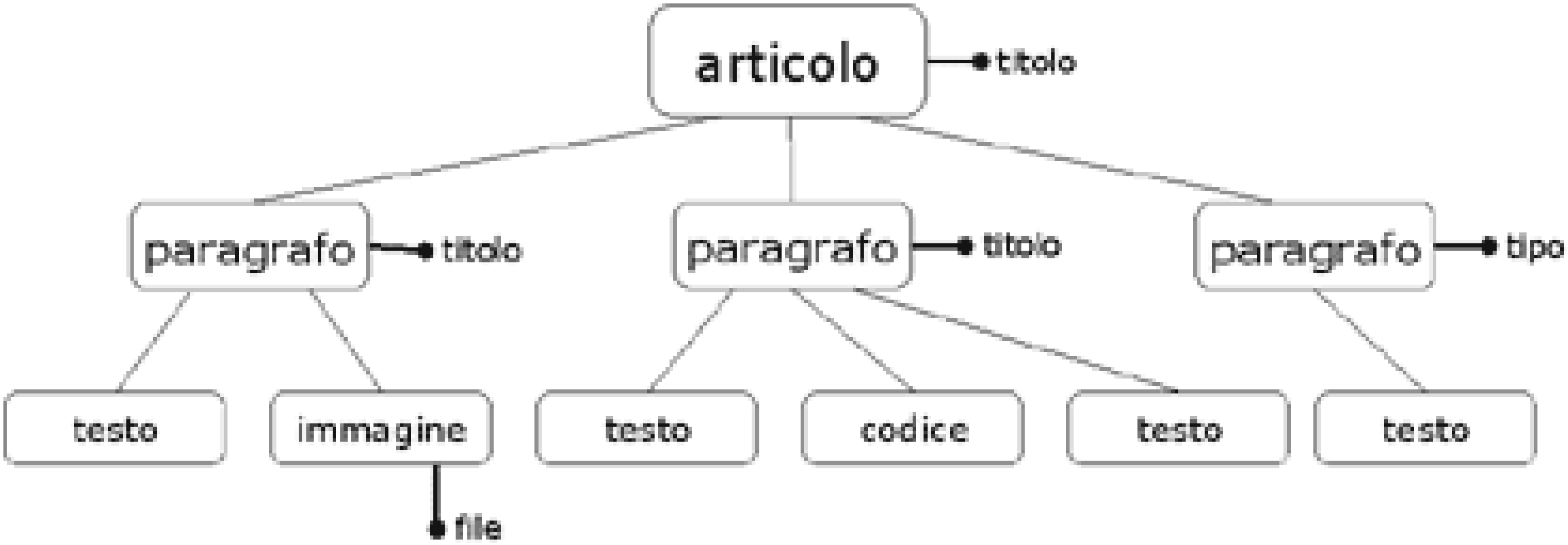
Struttura di un file XML

- Un file XML è un file di testo contenente tag, attributi e testi secondo regole sintattiche ben precise
- Ha un formato aperto e leggibile, simile all'HTML
- Contrariamente all'HTML, però, l'XML è ESTENSIBILE
  - Possiamo creare qualsiasi tag e qualsiasi attributi

Struttura di un file XML

- La struttura è gerarchica: albero, detto document tree
- Ogni componente logica viene detta elemento
- Essendo gerarchico, ogni elemento può contenere sottoelementi
- Gli elementi possono avere delle proprietà, dette attributi
- L'elemento principale viene detto ROOT

Esempio di file XML



Esempio di file XML

```
1  <?xml version="1.0" ?>
2  <articolo titolo="Titolo dell'articolo">
3      <paragrafo titolo="Titolo del primo paragrafo">
4          <testo>
5              Blocco di testo del primo paragrafo
6          </testo>
7          <immagine file="immagine1.jpg" />
8      </paragrafo>
9      <paragrafo titolo="Titolo del secondo paragrafo">
10         <testo>
11             Blocco di testo del secondo paragrafo
12         </testo>
13         <codice>
14             Esempio di codice
15         </codice>
16         <testo>
17             Altro blocco di testo
18         </testo>
19     </paragrafo>
20     <paragrafo tipo="bibliografia">
21         <testo>
22             Riferimento ad un articolo
23         </testo>
24     </paragrafo>
25 </articolo>
```

Specifica il tipo di documento e la versione

Root: è il primo elemento

Elemento paragrafo

Sintassi abbreviata

Elemento testo

Attributo

Sintassi standard

Struttura di un file XML

- La prima riga del codice è sempre costituita dalla dichiarazione del tipo di file, dalla versione ed eventualmente dall'encoding:

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

- Ogni elemento è costituito da tag:

```
<testo>
    Blocco di testo del secondo paragrafo
</testo>
```

- Gli attributi si definiscono in questo modo:

```
<paragrafo tipo="bibliografia">
```


Struttura di un file XML

- La struttura deve essere well-formed
- Ogni file ha solo una root
- XML è case sensitive
- I valori degli attributi devono essere sempre tra " o '
- La sintassi per i commenti è

<!-- qui il commento -->

XML e grammatica

- XML offre la possibilità di definire i tag a seconda delle necessità, ma per evitare confusione è necessario un meccanismo che ne vincoli l'utilizzo all'interno dei documenti: grammatica.

- La grammatica è un insieme di regole che indica quali vocaboli possono essere utilizzati e con che struttura è possibile comporre frasi.

- Un documento può essere valido rispetto ad una grammatica, ma non rispetto ad un'altra!

XML Schema

- Come si costruiscono le grammatiche?
  - DTD – Document Type Definition
  - XML Schema
- Entrambi forniscono la descrizione formale di una grammatica per XML.

XML Schema

- XML Schema utilizza la sintassi XML per definire la grammatica.

- La struttura generale è la seguente:

```
1  <?xml version="1.0"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3  ...
4  </xs:schema>
```

- Qual è l'elemento root?
- Che rappresenta la xs:?

XML Schema

- Come vengono definiti gli elementi?

```
<xs:element name="note">
```

- Come specifichiamo il tipo di dato da inserire?

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

- In XML esistono tipi di dati semplici e complessi

XML Schema: tipi di dato

- In XML esistono due categorie di tipi di dati:
  - Semplici (sotto la lista dei più usati)

Codifica XML Schema	Descrizione
xs:string	Stringa di caratteri
xs:integer	Numero intero
xs:decimal	Numero decimale
xs:boolean	Valore booleano: vero o falso
xs:date	Data
xs:time	Orario
xs:uriReference	Inserimento URL

  - Complessi

XML Schema: tipi di dato semplici

- XML Schema permette di definire tipi di dati semplici, ma personalizzati.
- Ad esempio, possiamo definire l'età di una persona come:

```
<xs:element name="eta" >
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="1" />
            <xs:maxInclusive value="130" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```


XML Schema: tipi di dato complessi

- I tipi di dato complessi si riferiscono ad elementi che possono contenere altri elementi e possono avere attributi.

- Definire un elemento di tipo complesso corrisponde a definire la relativa struttura.

XML Schema: tipi di dato complessi

- Lo schema generale per la definizione di un elemento di tipo complesso è il seguente:

```
<xs:element name="NOME_ELEMENTO">
    <xs:complexType>
        ... Definizione del tipo complesso ...
        ... Definizione degli attributi ...
    </xs:complexType>
</xs:element>
```


XML Schema: tipi di dato complessi

- In XML Schema i costruttori di tipi complessi previsti sono:

Costruttori	Descrizione
<xs:sequence>	Consente di definire una sequenza ordinata di sottoelementi
<xs:choice>	Consente di definire un elenco di sottoelementi alternativi
<xs:all>	Consente di definire una sequenza non ordinata di sottoelementi

- Per ciascuno di questi costruttori e per ciascun elemento è possibile definire il numero di occorrenze previste utilizzando gli attributi minOccurs e maxOccurs.

XML Schema: tipi di dato complessi

- Esercizio: dato il seguente file *note.xml* creare l'XML Schema corrispondente. Salvare il file con nome *note.xsd*

```
1  <?xml version="1.0"?>
2  <note>
3    <to>Tove</to>
4    <from>Jani</from>
5    <heading>Reminder</heading>
6    <body>Don't forget me this weekend!</body>
7  </note>
```


XML Schema: tipi di dato complessi

- In XML Schema gli attributi vengono considerati come un tipo di dato complesso:

```
<xs:attribute name="titolo" type="xs:string" use="required" />
```

  - L'attributo use permette di indicare se l'attributo è obbligatorio (required) o se ha un valore predefinito (default). In quest'ultimo caso occorre inserire anche l'attributo value

```
<xs:attribute name="titolo" type="xs:string" use="default" value="test" />
```


XML Schema: combinazione di grammatiche

- Una delle caratteristiche principali dell'XML Schema è la possibilità di integrare elementi derivanti da grammatiche diverse.
- Questa caratteristica consente di riutilizzare parti di grammatiche già definite evitando di dover rifare parte di lavoro già fatto in altri ambiti.
- La composizione di linguaggi pone almeno due tipi di problemi:
  - la validazione: a quale schema si deve fare riferimento per validare un documento XML "ibrido"?
  - due linguaggi potrebbero avere tag ed attributi con lo stesso nome

XML: Namespace

- Un namespace è un insieme di nomi di elementi e nomi di attributi identificati univocamente da un identificatore.
- L'identificatore univoco individua l'insieme dei nomi distinguendoli da eventuali omonimie in altri namespace.
- In un documento XML si fa riferimento ad un namespace utilizzando un attributo speciale (xmlns) associato al root element

```
<note xmlns="http://www.w3schools.com/xml/note" >
```