XML

eXtensible Markup Language

- LINGUAGGIO PER LO SCAMBIO DI DATI FRA APPLICAZIONI DIVERSE E DBMS DIVERSI
- NASCE COME METALINGUAGGIO (DERIVA DA SGML→STANDARD GENERALIZAD MARKUP LANGUAGE)
- E' UNO STANDARD DEL W3C (WORLD WIDE WEB CONSORTIUM)

- STORIA → GUERRA DEI BROWSER (ANNI ’90)

DIFFERENZA FRA XML E HTML

HTML

Linguaggio di markup i cui tag sono interpretati dai browser per la visualizzazione di contenuti multimediali e ipertestuali

XML

Linguaggio di markup i cui tag sono finalizzati alla memorizzazione e interscambio di dati fra applicazioni software diverse e DBMS di produttori diversi

- XML NON E’ UN LINGUAGGIO CHE «FA» QUALCOSA, E’ UN LINGUAGGIO CHE CONSENTE DI RAPPRESENTARE DEI DATI.

- QUALCUNO PUO’ SCRIVERE UN SOFTWAR PER INTERPRETARE QUESTI DATI (PARSING)

- UN DOCUMENTO XML VIENE MEMORIZZATO COME FILE DI TESTO CON ESTENSIONE .XML

POICHE' LA STRUTTURA DEI TAG NEI LINGUAGGI DI MARKUP E' GERARCHICA (UNO DENTO L'ALTRO), UN DOCUMENTO XML CONSENTE DI RAPPRESENTARE DATI STRUTTURATI SECONDO LA STRUTTURA AD ALBERO

ESEMPIO

Il seguente file di testo è un documento XML che rappresenta gli indirizzi di posta elettronica e le *password* degli utenti di un sistema informatico:

```
<?xml version="1.0" ?>
<users>
  <!-- Nota: le password sono cifrate. -->
  <user>
    <username>Pippo</username>
    <email>pippo32@disney.com</email>
    <password>0A1B2C3D4E5F</password>
  </user>
  <user>
    <username>Pluto</username>
    <email>pluto30@disney.com</email>
    <password>A9B8C7D6E5F4</password>
  </user>
</users>
```

PERCHE' E' DETTO

EXTENSIBLE ?

ESEMPIO

Il documento dell'esempio precedente presenta la struttura ad albero di FIGURA **1**, che deriva dalla struttura di nidificazione dei *tag* con l'elemento che racchiude tutti gli altri e che diviene il nodo radice dell'albero.

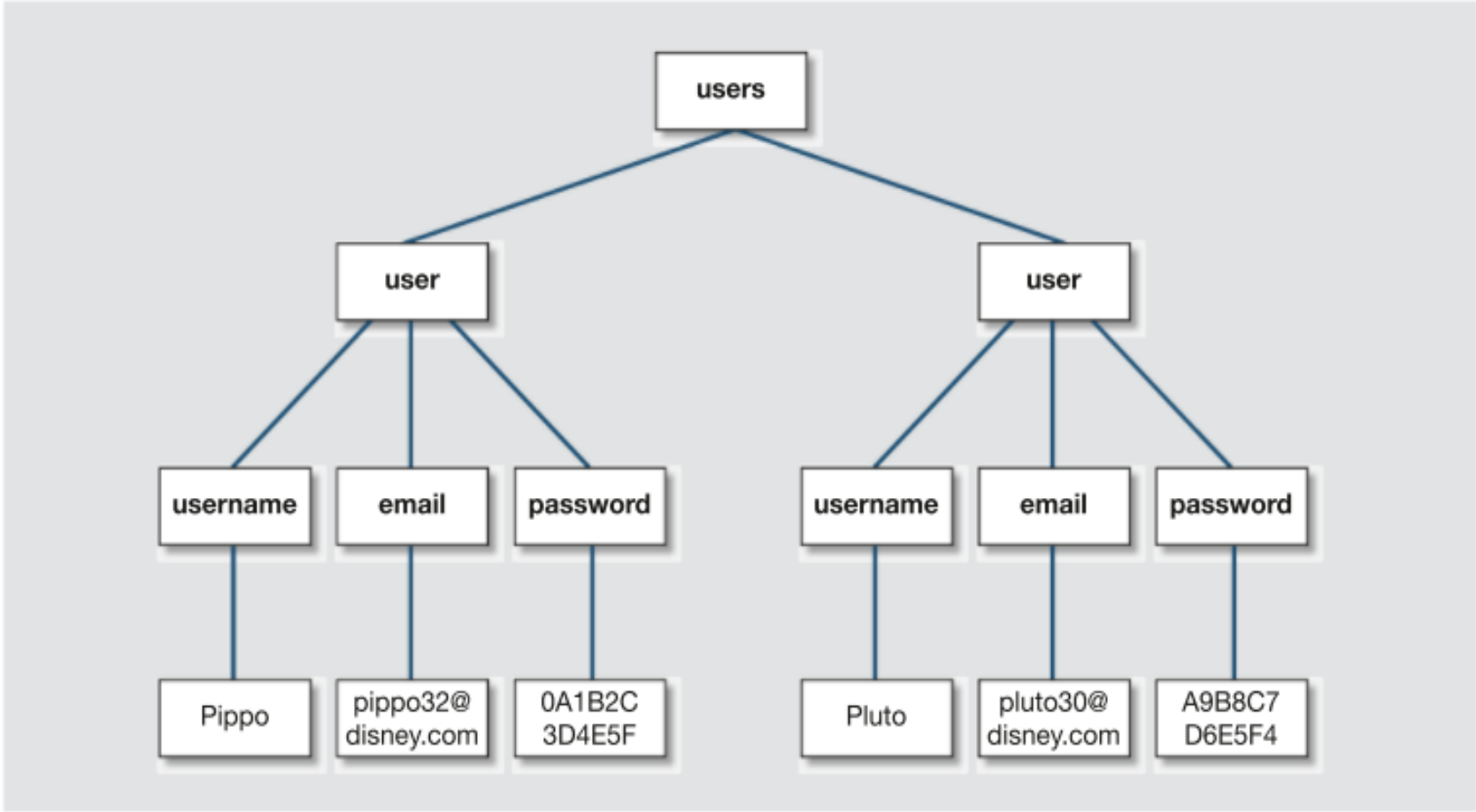


FIGURA **1**

SINTASSI XML: REGOLE DI BASE

- PROLOGO <?xml version=″1.0″ ?>

  Con aggiunta eventualmente della codifica

<?xml version=″1.0″ encoding= ″ISO-8859-1″ ?>

UTF-8 è LA CODIFICA PREDEFINITA DI XML

ISO-8859-1	Codifica standard a 8 bit dei caratteri dell'Europa occidentale
UTF-8	Codifica Unicode che utilizza da 8 a 32 bit per il singolo carattere
UTF-16	Codifica Unicode che utilizza 16 bit per il singolo carattere ed è adottata dal linguaggio Java

SINTASSI XML: REGOLE DI BASE

Case sensitive

NOMI DEI TAG

Contenuto dei nomi

- Caratteri alfanumerici
- _ - .
- Possono iniziare solo con lettere o "_" (no numeri)
- Non possono contenere "xml"

SINTASSI XML: REGOLE DI BASE

TUTTI I TAG DEVONO ESSERE

- Aperti e chiusi (/ per chiudere). Per i tag vuoti o con solo attributi è permessa a chiusura: <tag_vuoto/>
  <tag attributo= ″….. ″/>
- Rispettosi della gerarchia e con un elemento Root che contiene tutti gli altri tag

SINTASSI XML: REGOLE DI BASE

ATTRIBUTI

- Sono indicati all'interno del tag e il valore deve essere sempre "quotato" (fra virgolette)

COMMENTI

- <!-- commento -->

SPAZI

- Gli spazi all'interno degli elementi vengono mantenuti (diversamente da HTML)

SINTASSI XML: REGOLE DI BASE

ENTITA’

- I simboli chiave >, <, &, ″, ‘

Sono rappresentati mediante le rispettive entità:

Carattere	Entità
&	&amp;
<	&lt;
>	&gt;
"	&quot;
'	&apos;

SINTASSI XML: GLI ELEMENTI

Un **elemento** XML è tutto ciò che è compreso fra un tag di apertura e un tag di chiusura **(INCLUSI!).**

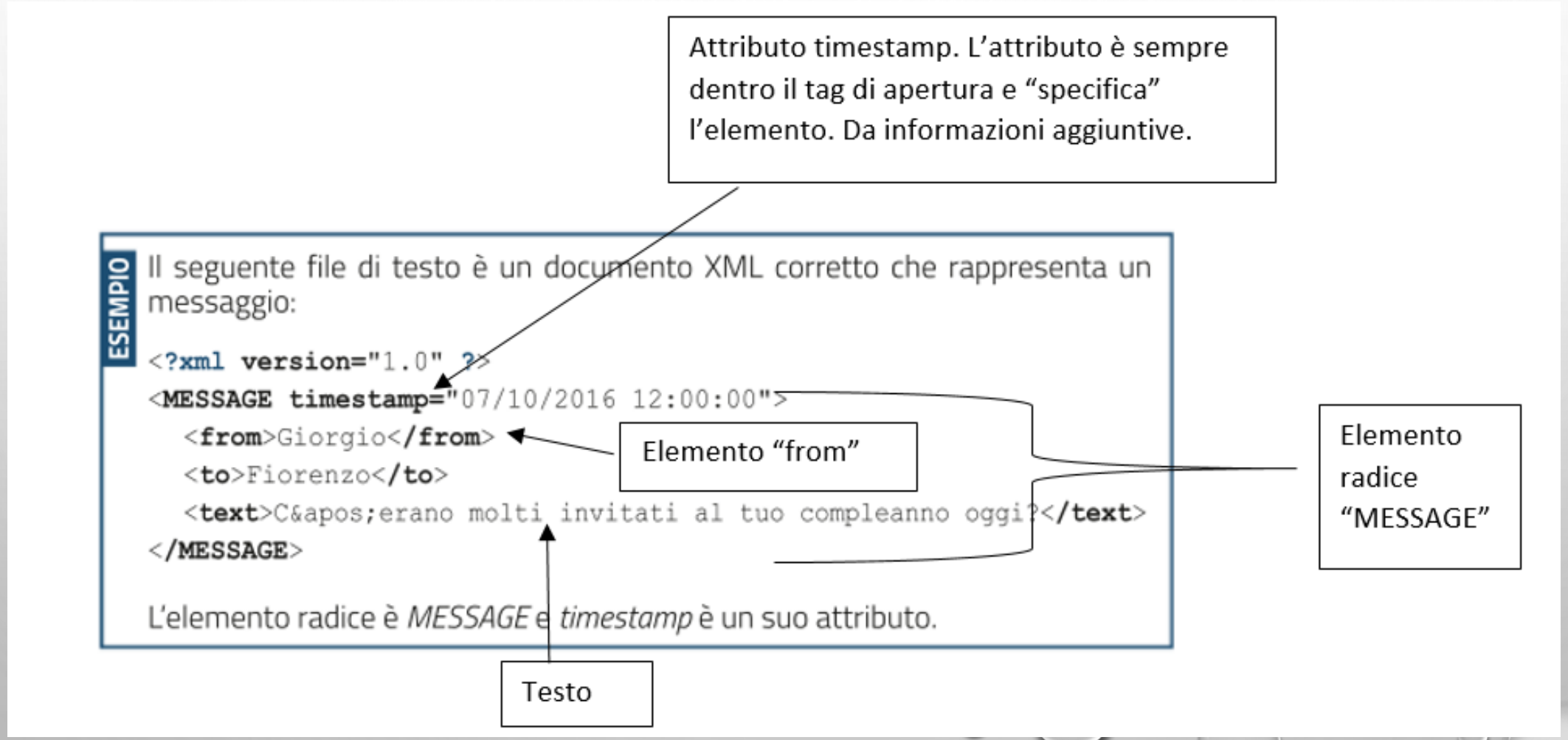
Un elemento può contenere:

- del **testo**
- degli **attributi**
- altri **elementi**.

oppure può essere **vuoto**

SINTASSI XML: GLI ELEMENTI

Esempio:



ATTRIBUTI VS ELEMENTI

Ma che differenza c'è fra attributi ed elementi? Quando uso uno e quando l'altro?

La regola è che i **DATI** vanno negli elementi, i **METADATI** (dati che "spiegano i dati") vanno negli attributi.

Regola empirica: non esagerare con gli attributi perché non consentono di creare struttura ad albero (non possono esistere attributi con dentro attributi) e non sono facilmente espandibili.

:

ATTRIBUTI PREDEFINITI

- Gli attributi di un TAG possono essere definiti da chi costruisce il documento XML, ma esistono alcuni attributi predefiniti validi per ogni TAG ai quali è possibile assegnare un valore.

L'attributo xml:lang
Indica la lingua del testo contenuto in un elmento

ESEMPIO

Il documento XML dell'esempio precedente dovrebbe riportare l'attributo per l'indicazione del linguaggio:

```
<?xml version="1.0" ?>
<MESSAGE timestamp="07/10/2016 12:00:00">
  <from>Giorgio</from>
  <to>Fiorenzo</to>
  <text xml:lang="it">
    C&apos;erano molti invitati al tuo compleanno oggi?
  </text>
</MESSAGE>
```

Altri valori:
en, fr, de, es....

ATTRIBUTI PREDEFINITI

Altri attributi predefinti che incontereremo più avanti:

- default= (per indicare valori di default di un elemento)
- fixed= (per indicare un elemento con valore costante)

SINTASSI SPECIALE

IL QUALIFICATORE CDATA (Character DATA)

Abbiamo detto che un elemento XML può contenere altri elementi XML. I software che gestiscono i documenti XML analizzano l'intero contenuto del documento per individuare gli elementi. Se si vuole che il contenuto di un elemento non venga analizzato, per consentire di inserire qualunque sequenza di caratteri (ad esempio del codice HTML o XML), si deve qualificare tale contenuto come Character DATA utilizzando l'apposito tag:

<![CDATA[

testo…. ad esempio altro codice XML

]]>

SINTASSI SPECIALE

ESEMPIO

Il seguente file di testo è un documento XML corretto che rappresenta un messaggio:

```
<?xml version="1.0" ?>
<MESSAGE timestamp="07/10/2016 12:00:00">
  <from>Giorgio</from>
  <to>Fiorenzo</to>
  <text>
    <![CDATA[
      Ti invio un esempio di XML:
      <?xml version="1.0" ?>
      <users>
        <user>
          <username>Pippo</username>
        <email>pippo32@disney.com</email>
        </user>
        <user>
          <username>Pluto</username>
          <email>pluto30@disney.com</email>
        </user>
      </users>
    ]]>
  </text>
</MESSAGE>
```

ESEMPIO DI UN DOCUMENTO XML: LIBRO

La struttura di un documento XML corretto si può sempre rappresentare con una struttura ad albero dove il nodo radice è l'elemento radice dell'albero e gli altri nodi sono:

- altri elementi
- attributi degli elementi
- contenuti testuali degli elementi
- eventuali commenti

Scriviamo il documento con notepad++ e verifichiamo se è «ben formato»

Oppure verifichiamolo qui:
http://www.utilities-online.info/xsdvalidation/#.W_LljuhKjIU

ESEMPIO

Il seguente file di testo è un documento XML corretto che rappresenta un libro:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<libro genere="fantascienza" xml:lang="it">
   <autore>Stefano Benni</autore>
   <titolo>Terra!</titolo>
   <editore>Feltrinelli</editore>
   <anno>1983</anno>
</libro>
```

Esso è rappresentato dal diagramma ad albero di FIGURA 2.

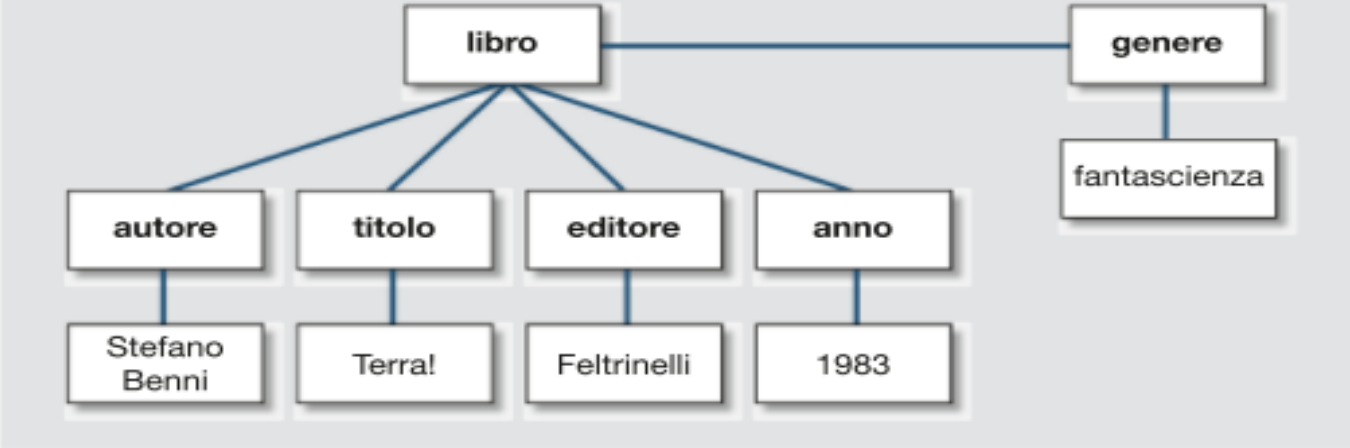


FIGURA 2

DOCUMENTO XML BEN FORMATO

E’ un documento XML che rispetta le regole sintattiche

Esistono software o risorse online che consento di verificare se un documento è ben formato o meno

DOCUMENTO XML BEN FORMATO

Un documento XML potrebbe essere ben formato ma utilizzare i tag in maniera **non corretta**

ESEMPIO

Il seguente documento XML che rappresenta una collezione di libri è ben formato:

```
<?xml version="1.0" ?>
<libri>
  <libro genere="fantascienza">
    <autore>Stefano Benni</autore>
    <titolo>Terra!</titolo>
    <editore>Feltrinelli</editore>
    <anno>83</anno>
  </libro>
  <libro genere="romanzo">
    <scrittore>Mark Twain</scrittore>
    <titolo>Le avventure di Huckleberry Finn</titolo>
    <anno>1884</anno>
  </libro>
  <libro>
    <autore>Herman Melville</autore>
    <titolo>Moby Dick</titolo>
    <editore/>
    <anno>milleottocentocinquantuno</anno>
  </libro>
</libri>
```

XSD SCHEMA

- Poiché il documento XML viene utilizzato per trasmettere dati fra applicazioni diverse, è importante che esso «rispetti delle regole» condivise fra le due applicazioni (quanti e quali elementi sono contenuti, i tipi di dato….)

- Uno **schema XSD** è un documento che contiene le «regole» che devono essere rispettate da un documento XML affinché esso sia considerato **valido**.

SCHEMA XSD

- **Schema XSD:** è un documento di testo (file con estensione .xsd) scritto in linguaggio **XSD (XML Schema Definition)**

- Il linguaggio XSD è a sua volta un linguaggio definito con XML (i suoi tag sono definiti in XML)

SCHEMA XSD

- **VALIDAZIONE:** verifica che un documento XML rispetta i requisiti stabiliti nell'apposito schema XSD
- La **VALIDAZIONE** di un documento XML deve sempre essere fatta rispetto ad un schema XSD. Si dice che un documento XML «è valido» o «non è valido» **rispetto ad uno schema XSD**
- La **VALIDAZIONE** viene eseguita con appositi strumenti software (notepad++ o http://www.utilities-online.info/xsdvalidation/#.XAVkxWhKjIV)

SCHEMA XSD

Nello schema XSD si definiscono:

- Gli elementi che deve contenere un documento XML e la loro relazione gerarchica
- Gli eventuali attributi di ogni elemento
- Il numero e l'ordinamento degli elementi
- Gli eventuali elementi vuoti
- Il tipo di dato contenuto negli elementi e negli attributi
- Eventuali valori predefiniti o costanti degli elementi e degli attributi.

SCHEMA XSD

Ogni schema XSD ha il seguente elemento radice:

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

SCHEMA XSD

Ogni documento XML che fa riferimento ad uno schemaXSD deve contenere il riferimento allo schema in un apposito attributo predefinito dell'elemento radice, come nel seguente esempio:

```
<?xml version="1.0" ?>
<libro
        xmlns:xsi="http//www.w3.org/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="libro.xsd">
    <titolo>Pierone è il migliore</titolo>
    <autore>Pierone</autore>
</libro>
```

Questo attributo indica al parser che il ddocumento XML deve essere validato riapetto ad uno schema XSD

Questo attributo specifica al parser il path name del file .xsd che contiene lo schema XSD. Può essere anche un URL

SCHEMA XSD ESEMPIO

ESEMPIO

Un possibile schema XSD per un documento XML che rappresenta un singolo libro potrebbe essere il seguente:

```
<?xml version="1.0" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="libro">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="autore" type="xs:string"/>
        <xs:element name="titolo" type="xs:string"/>
        <xs:element name="editore" type="xs:string"/>
        <xs:element name="anno" type="xs:integer"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Non lo indichiamo quando utilizziamo il validatore online

Un documento XML deve riferire lo schema a cui si attiene nella definizione dell'elemento radice, come nel seguente documento XML, in cui si assume che lo schema precedente sia stato salvato nel file «libro.xsd»:

```
<?xml version="1.0" ?>
<libro
  xmlns:xsi="http://www.w3.org/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="libro.xsd">
    <autore>Stefano Benni</autore>
    <titolo>Terra!</titolo>
    <editore>Feltrinelli</editore>
    <anno>1983</anno>
</libro>
```