



Maurizio Cozzetto
Itis Castelli
Brescia

Linguaggio di markup



- E' un linguaggio che permette di aggiungere mediante *tag* informazioni a un documento
- Il documento è costituito da
 - Contenuto
 - Sistema di contrassegno



- Nasce intorno alla fine del 1996
- XML sta per *eXtensible Markup Language*
- Sottinsieme di *SGML* pensato per la creazione di *documenti su web*



- SGML sta per *Standard Generalized Markup Language*
- E' un *metalinguaggio* per la creazione di linguaggi di markup

Usi del linguaggio XML



- Descrizione dei documenti (alternativa all' *HTML*)
- Descrizione dei dati (alternativa ai *database*)



- Per creare linguaggi di markup serve una DTD
- DTD sta per *Document Type Definition*
- La DTD definisce i *tag*, gli *attributi* e le *regole* di utilizzo dei tag

Documenti validi



Un documento XML che segue le regole definite in una DTD si dice *valido* rispetto alla DTD

Documenti ben formati



Un documento XML che *non* utilizza una DTD ma rispetta le regole dei documenti XML si dice *ben formato (well formed)*

Parser XML (1)



- Per visualizzare o elaborare un documento XML è necessario un programma detto *parser XML*
- Esegue la *convalida* rispetto alla DTD, se disponibile
- Rende disponibili le *informazioni* del documento XML

Parser XML (2)



- Nel browser è incluso un parser XML
 - DOM (*Document Object Model*)
 - Javascript
- Java dispone di due API complementari per XML
 - SAX (*Simple API per XML*)
 - DOM (*Document Object Model*)
- Con il JDK viene fornito anche un parser XML



Un linguaggio di markup che segue le specifiche XML è detto *applicazione XML*

Applicazioni XML (1)



- XHTML
- XSL
- Xlink
- MathML
 - *<http://www.w3.org/Math/>*

Applicazioni XML (2)



- SVG
- VRML/X3D
 - *<http://www.web3d.org/x3d/>*
- WML
 - *protocollo WAP (telefoni cellulari)*
- MusicXML
 - *<http://www.musicxml.org>*

Un documento XML



- è un *file di testo* con estensione *.xml*
- viene trasformato in un file *HTML* se visualizzato in un browser (*eccetto Netscape*) con un foglio di stile di default

Un documento XML è *ben formato* se



- Tutti i tag *aperti* vanno *chiusi* e i tag *vuoti* vanno chiusi con /
 - `<tag>...</tag>`
 - `<tag ... />`
- I tag vanno *nidificati* correttamente
 - `<tag1><tag2>...</tag2></tag1>`
- I valori degli attributi vanno indicati tra *doppie virgolette*
 - `<tag attr1="val1" attr2="val2" ...>...</tag>`

Formato dei documenti XML (1)



- Un documento XML è formato da
 - Prologo
 - Elemento Radice

Formato dei documenti XML (2)



- Prologo
 - Dichiarazione xml
 - `<?xml version="1.0"?>`
 - Commenti
 - `<!-- ... -->`
 - Dichiarazione della DTD
 - `<!DOCTYPE nomeRadice [definizione_DTD]>`
 - Istruzioni di elaborazione
 - `<?xml-stylesheet ...?>`



- Elemento Radice
 - `<nomeRadice>`
 - `<nomeElemento>...</nomeElemento>`
 - `<nomeElemento>...</nomeElemento>`
 - `</nomeRadice>`

Esempio



Apri `orderlist.xml` nel browser



- I nomi in un documento XML possono entrare in conflitto
 - Es. `<table><tr><td>...</td>...</tr>...</table>`
 - `<table><name>...</name><width>...</width><height>...</height></table>`
- Il prefisso risolve i conflitti dei nomi
 - Es. `<h:table>....<h:table>`
 - `<f:table>...</f:table>`

Esempio di documento XML



```
<?xml version="1.0" ?>
<OrderList>
  <Customer>
    <Name>
      <FirstName>Bill</FirstName>
      <LastName>King</LastName>
    </Name>
    <CardInfo>
      <CardNum>99999999</CardNum>
      <Validate>08/31/2002</Validate>
    </CardInfo>
  </Customer>
  <Item>
    <ItemNo>S-1000</ItemNo>
    <Title>Shoes</Title>
    <Look>
      <Color>Blue</Color>
      <Size unit="cm">22.5</Size>
    </Look>
    <Price unit="yen">15,000</Price>
  </Item>
</OrderList>
```

Processing instruction, indicates XML document

La DTD definisce



- un'applicazione XML
- un vocabolario XML
- una classe di documenti

La DTD può essere



- Interna
 - al documento XML
- Esterna
 - definita in un file con estensione .dtd
 - `<!DOCTYPE nomeRadice SYSTEM "nomeFile.dtd">`

Esempio di DTD



```
<?xml version="1.0" ?>
<!DOCTYPE OrderList[
<!ELEMENT OrderList(Customer, Item+)>
<!ELEMENT Customer(Name, CardInfo+)>
<!ELEMENT Name (FirstName, MiddleName?, LastName)>
<!ELEMENT FirstName (#PCDATA)>
<!ELEMENT MiddleName (#PCDATA)>
<!ELEMENT LastName (#PCDATA)>
<!ELEMENT CardInfo (CardNum, Validate)>
<!ELEMENT CardNum (#PCDATA)>
<!ELEMENT Validate (#PCDATA)>

<!ELEMENT Item (ItemNo, Title, Look, Price)*>
<!ELEMENT ItemNo (#PCDATA)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Look (Color*, Size)>
<!ELEMENT Color (#PCDATA)>
<!ELEMENT Size (#PCDATA)>
<!ATTLIST Size unit CDATA "0">
<!ELEMENT Price (#PCDATA)>
<!ATTLIST Price unit CDATA "0">
]>
<OrderList>
...
</OrderList>
```

Dichiarazione di elemento



`<!ELEMENT Customer (Name, CardInfo+)>` — Include the **Name** and **CardInfo**, tags in the **Customer** tag.

`<!ELEMENT Line EMPTY>` — The **Line** element is an empty element tag, and **EMPTY** is a fixed specification.

`<!ELEMENT Other ANY>` — The contents of any element can be included in the **Other** tag.

`<!ELEMENT FirstName (#PCDATA)>` — Mixed-content declaration.

Dichiarazione di attributo



<code><!ATTLIST</code>	<code>Size</code>	<code>unit</code>	<code>CDATA</code>	<code>"0"></code>
	Element	Attribute	Attribute	Default
	name	name	type	value

Indicatori di molteplicità



Symbol	Multiplicity
+	Repeat one or more times
*	Repeat zero or more times
?	Indicates an element that is optional; it occurs once or not at all.

Tipi predefiniti



Symbol	Description
#PCDATA	The character data to be analyzed. Child elements and character data can be included in elements. It is also called <i>parsed character data</i> . It is processed (parsed) by an XML parser.
CDATA	Represents character data that is not analyzed (not parsed).



- è un documento XML con estensione `.xsd` che usa lo *spazio dei nomi* <http://www.w3.org/2001/XMLSchema>
- è definito mediante il linguaggio *XML Schema*
- permette di definire le regole in modo più *flessibile* (il parser deve supportare la validazione tramite XML Schema)