



Introduzione ad XML

e**X**tensible **M**arkup **L**anguage

Sommario

- Che cosa è XML
- Gli obiettivi di XML
- I limiti di HTML
- Un documento XML
- Definizione della struttura del documento (DTD)
- Spazi di nomi (Namespaces)
- Gli Schemi (XML-Schema)
- Lo stile (Stylesheet: CSS, XSL)
- I collegamenti (XML Linking Language)
- Interfacce di accesso al documento XML (DOM, SAX)
- Applicazioni XML
- Software
- Riferimenti
- Conclusioni

Che cosa è XML

XML: acronimo di eXtensible Markup Language

- è un linguaggio estensibile realizzato per poter utilizzare in modo semplice i documenti strutturati
- studiato per il Web, possibilità di utilizzo in ambienti differenti.
- sviluppato dal W3C
- è un sottoinsieme di SGML

XML - eXtensible Markup Language

- prima bozza di XML:
novembre 1996
- specifica attuale:
<http://www.w3.org/TR/1998/REC-xml-19980210>
- traduzione in italiano della specifica attuale:
<http://www.xml.it/REC-xml-19980210-it.html>

XML: gli obiettivi

Questi gli obiettivi progettuali di XML secondo il W3C XML Working Group:

- XML deve essere utilizzato in modo semplice su Internet.
- XML deve supportare un gran numero di applicazioni.
- XML deve essere compatibile con SGML.
- Deve essere facile lo sviluppo di programmi che elaborino XML.
- Il numero di caratteristiche opzionali deve essere mantenuto al minimo possibile, idealmente a zero.
- I documenti XML dovrebbero essere leggibili da un uomo e ragionevolmente chiari.
- La progettazione XML dovrebbe essere rapida.
- La progettazione XML dovrebbe essere formale e concisa.
- I documenti XML devono essere facili da creare.
- Non è di nessuna importanza l'economia nel markup XML.

Markup (marca, etichetta)

- Markup: tutto ciò che ha un significato speciale che deve essere ben caratterizzato, reso esplicito
- Esempi di markup: testo in corsivo, testo sottolineato
- anche identificatore, simbolo o altro espediente per distinguere un elemento da altri simili, può indicare l'inizio o la fine di un oggetto
- In XML tutto ciò che è compreso tra i caratteri "<" e ">" (angled brackets, parentesi angolari) è considerato markup, viene detto anche tag (etichetta), esempio:

<nome>

Anche HTML è un markup language inizialmente definito in SGML.

Estensibilità

- XML non ha tag predefiniti
è estensibile
consente di definire nuovi linguaggi
è un metalinguaggio

Metalinguaggio: nella logica formale, linguaggio impiegato nello studio di un linguaggio oggetto. Può essere o non essere formalizzato e a sua volta può essere oggetto di indagine che si attua mediante un metametalinguaggio.

Detto anche linguaggio di descrizione del linguaggio

- L'insieme delle regole di HTML sono contenute in un documento (separato dal file .html) il DTD HTML (Document Type Definition) incorporato nel browser, è invisibile all'utente.

Metainformazione

A cosa può servirci una metainformazione ?

Esempio: cosa rappresenta la seguente informazione ?

m 13,155 45,126, 115,126, 86,155 x e

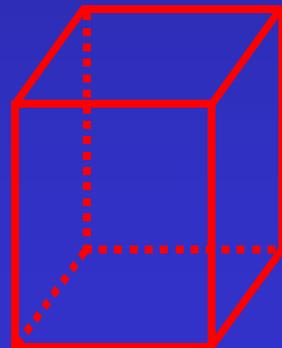
m 13,43 45,13, 115,13, 86,43 x e

m 13,155 13,43 e

m 45,13 45,126 e

m 115,13 115,126 e

m 86,43 86,155 e



Un parallelepipedo !

Perchè un nuovo linguaggio ?

Limiti di HTML

Cosa è questo ?

```
<td> 12 </td>
```

- Il numero civico di una via ?
- Il numero di telefono per ottenere informazioni sugli abbonati ?
- Entrambe le cose ?
- Nessuna delle due ?

Un semplice markup con HTML

```
<p> <b> Sig. Mario Rossi </b>  
<br>  
Via Verdi, 12  
<br>  
56100, Pisa
```

Visualizzazione di markup HTML

Sig. Mario Rossi
Via Verdi, 12
56100, Pisa

Interpretazione di HTML

Il nostro algoritmo per trovare il numero civico:

Se un paragrafo contiene due tag `
`

allora la prima parola dopo la prima virgola dopo il primo tag `
` è il numero civico.

Un semplice markup XML

```
<business-card>  
  <persona>  
    <titolo> Sig. </titolo>  
    <nome> Mario </nome>  
    <cognome> Rossi </cognome>  
  </persona>  
  <indirizzo>  
    <strada> Via Verdi </strada>  
    <numero-civico> 12 </numero-civico>  
    <cap> 56100 </cap>  
    <città> Pisa </città>  
</business-card>
```

Visualizzazione di markup XML

Sig. Mario Rossi
Via Verdi, 12
56100, Pisa

- XML può essere visualizzato nello stesso modo di HTML

Visualizzazione di markup XML

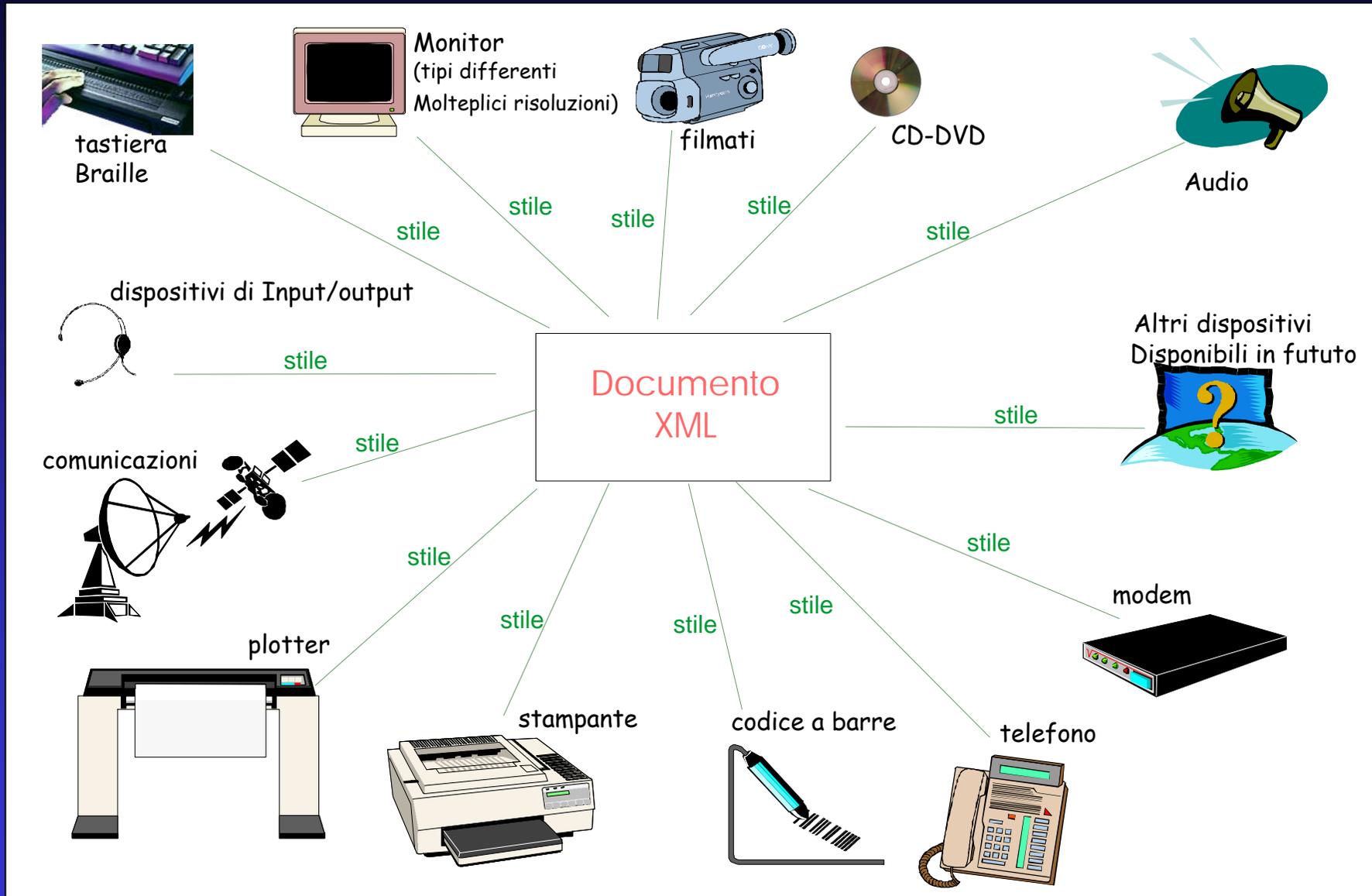
Il markup XML può essere visualizzato anche in questo modo:

Sig. Mario Rossi

Via Verdi, 12

56100, **Pisa**

Visualizzazione di markup XML



Interpretazione di XML

Un algoritmo migliore e più semplice per trovare il numero civico:

il numero civico è il contenuto del tag `<numero-civico>`

Contenuto contro rappresentazione

HTML ci dice come rappresentare un documento ipertestuale su Web
(difficilmente su un altro media, ad esempio su carta)

XML ci dice cosa contiene un documento

Non è sufficiente migliorare HTML

Ricapitolando: limiti di HTML

- non ci dice nulla sul contenuto del documento
- non permette di estendere il linguaggio con tag personali
- limitato come prodotto di pubblicazione
- limitato come ipertesto
- limitato come elaborazione
- non supporta dati strutturati -> inefficiente per i motori di ricerca

Serve un linguaggio semplice, flessibile

HTML non verrà comunque sostituito, almeno nel più immediato futuro, perché offre il metodo più semplice per pubblicare informazioni sul Web

Le componenti di XML

Problema attuale: scambio di documenti
Formati proprietari difficilmente scambiabili

XML studiato per facilitare scambi di dati anche tra applicazioni di tipo diverso (es.: i database e i word processor).

Documento facilmente interpretabile
tre parti fondamentali da tenere distinte:

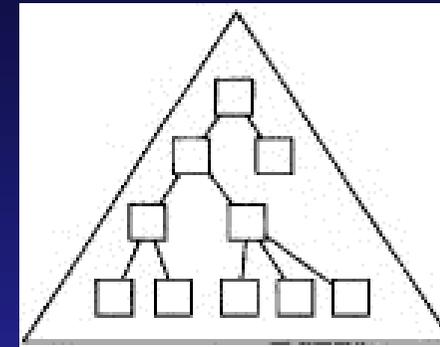
- il contenuto;
- le specifiche relative agli elementi, la struttura (DTD);
- le specifiche relative alla rappresentazione, lo stile (Stylesheet).

Le componenti di XML

Contenuto

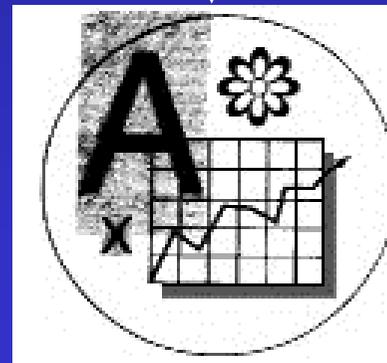
"La Discovery stava accelerando verso Giove lungo un'orbita complessa calcolata alcuni mesi prima dagli astronomi sulla Terra e controllata costantemente da Hal"
A.C. Clarke

Struttura



Specifiche di formattazione dell'output

Formattazione



Rappresentazione (Stile)

Processo completo di codifica XML

Il documento

Uno degli obiettivi di progettazione di XML:
deve essere in un formato leggibile dall'uomo

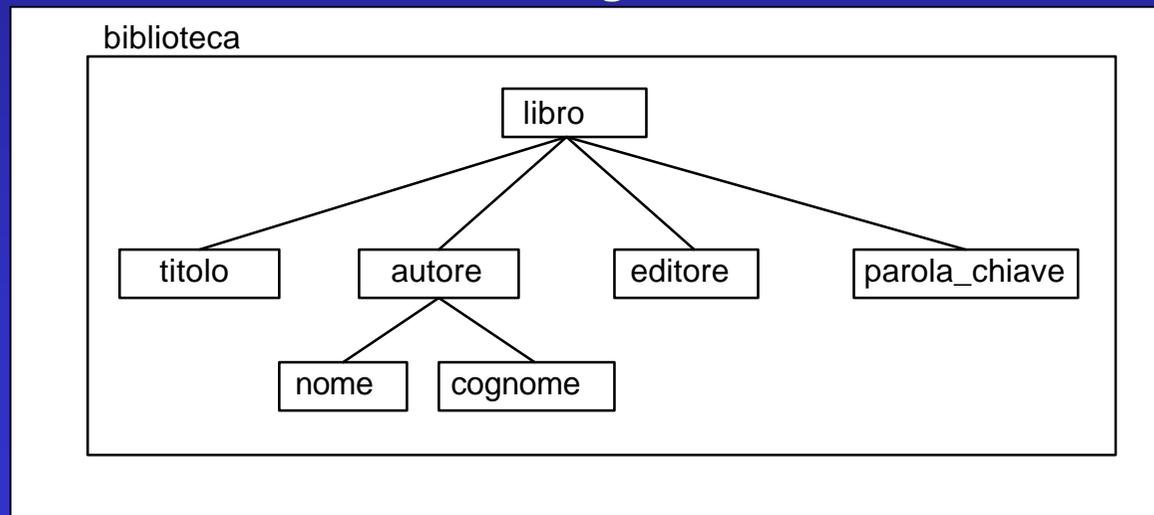
```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro codice="R414">
    <titolo>2001: Odissea nello spazio</titolo>
    <autore>
      <cognome>Clarke</cognome>
      <nome>Arthur Charles</nome>
    </autore>
    <editore>Rizzoli</editore>
    <parola_chiave>romanzo</parola_chiave>
    <parola_chiave>fantascienza</parola_chiave>
  </libro>
</biblioteca>
```

<nome> **1** **<Nome>** **1** **<NOME>**

DTD - Document Type Definition

- contiene le regole di definizione dei tag
- indica gli elementi e il loro ordine all'interno del documento XML
- il suo uso non è obbligatorio; ne è consigliato l'utilizzo
- può essere interna o esterna al documento XML
- il suo nome per convenzione corrisponde a quello dell'elemento radice

Struttura gerarchica



DTD - Document Type Definition

una DTD XML che rappresenta la struttura definita precedentemente:

```
<!DOCTYPE biblioteca [
  <!ELEMENT biblioteca (libro+)>
  <!ELEMENT libro (titolo, autore+, editore, parola_chiave+)>
  <!ATTLIST libro
    codice ID #REQUIRED>
  <!ELEMENT titolo (#PCDATA)>
  <!ELEMENT autore (cognome, nome)>
  <!ELEMENT editore (#PCDATA)>
  <!ELEMENT parola_chiave (#PCDATA)>
  <!ELEMENT cognome (#PCDATA)>
  <!ELEMENT nome (#PCDATA)>
]>
```

Elemento	(1)
Elemento ?	(0,1)
Elemento *	(0,1,+)
Elemento +	(1,+)

Elemento1 | Elemento2
(or)

```
<!ATTLIST nome_elemento
  nome_attributo1 tipo_attributo valore_di_default #REQUIRED
  nome_attributo2 tipo_attributo #IMPLIED
  nome_attributo3 tipo_attributo valore_di_default #FIXED
>
```

(obbligatorio)

(opzionale)

(obbligatorio con valore prefissato)

Regole per i tag

~~Non permesso~~

~~<a>~~

~~~~

~~~~

~~~~

Corretto

<a>

I tag devono essere nidificati

Quando l'elemento è senza contenuto

<tag attr="3"></tag>

<tag attr="3"/>

sono equivalenti

← Nuova sintassi per i tag di chiusura

<tag attributo="valore">contenuto</tag>

Documento ben-formato, valido

Un documento XML si dice "ben formato" quando:

- contiene almeno un elemento;
- esiste un tag unico di apertura e di chiusura contenente l'intero documento;
- tutti i tag sono nidificati
- tutte le entità sono dichiarate.

Un documento si dice "valido" quando

- contiene una DTD e rispetta le regole definite in essa.

document type declaration:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE biblioteca SYSTEM "biblioteca.dtd" >
```

.....

Namespaces

Un documento XML può contenere informazioni relative a più argomenti fonte di problemi: elementi o attributi con lo stesso nome ma in contesti differenti (mouse)

Namespace: insieme di nomi identificati da un URI detti anche vocabolari o vocabolari di markup.

Nome costituito da due parti: il nome locale, un prefisso

Namespace dichiarato con attributo con il prefisso "xmlns" e URI univoco

```
<html:body xmlns:html="http://www.w3.org/TR/REC-html40">  
  <html:h1>text</html:h1>  
</html:body>
```

body e h1 ∈ namespace HTML 4.0.

```
xmlns:html="http://www.w3.org/TR/REC-html40"  
xmlns:dc="http://purl.org/metadata/dublin-core">
```

...

```
<html:p>questo e' un paragrafo html</html:p>  
<dc:Creator>Massimo Martinelli</dc:Creator>
```

Raccomandazione W3C: <http://www.w3.org/TR/1999/REC-xml-names-19990114>
RFC 2396 (URI) : <http://www.ietf.org/rfc/rfc2396.txt>

Namespaces

```
xmlns:persona="http://www.anagrafe.it/persona"  
xmlns:libro="http://www.biblioteca.it/libro"
```

.....

```
<scheda_prestito>  
  <persona:richiedente>  
    <persona:titolo>dottor</persona:titolo>  
    <persona:cognome>Pallino</persona:cognome>  
    <persona:nome>Pinco</persona:nome>  
    <persona:posizione>dipendente</persona:posizione>  
  </persona:richiedente>  
  <materiale>  
    <libro:titolo>Linguaggio C</libro:titolo>  
    <libro:autore>  
      <persona:nome>Brian</persona:nome>  
      <persona:cognome>Kernighan</persona:cognome>  
      <persona:nome>Dennis</persona:nome>  
      <persona:cognome>Ritchie</persona:cognome>  
    </libro:autore>  
    <libro:posizione>S4-R3-P2</libro:posizione>  
  </materiale>  
</scheda_prestito>
```

XML Schema

- Limiti delle DTD:
- non supportano i namespace
 - non sono scritti in XML
 - non sono estensibili
 - non specificano tipi di dato

Proposta di estendere le funzionalità delle DTD: XML-Schema

Metodo estensibile per definire il modello dei dati XML

- definizione di elementi, attributi e relazioni (come le DTD)
- definizione di tipi di dato
- definizione di vincoli (range, lunghezza stringa, precisione decimale,...)

Working Draft: tentativo di unire diverse soluzioni proposte (XML-Data, DCD, SOX, DDML, XDR) in una comune.

Composto da due parti: **strutture** e **tipi di dati**.

<http://www.w3.org/2000/05/06-xmlschema-1>

<http://www.w3.org/2000/05/06-xmlschema-2>

Tipi di dati: **primitivi** e **derivati**.

- Primitivi: oggetti pronti all'uso (numeri, valori logici, caratteri, ...).
- Tipi derivati: oggetti costruibili a partire dai primitivi

XML Schema

```
<?xml version="1.0"?>
<xs:schema
  xmlns:xs="http://www.w3.org/1999/XMLSchema"
  targetNamespace=" http://www.my.org/mySchema"
  version="1.0" >

  <xs:simpleType name="TipoTitolo" base="string">
    <xs:maxLength value="60" />
  </xs:simpleType>

  <xs:simpleType name="TipoEditore" base="string">
    <xs:maxLength value="30" />
  </xs:simpleType>
  <xs:simpleType name="TipoNome" base="string">
    <xs:maxLength value="30" />
  </xs:simpleType>

  <xs:simpleType name=" TipoParola_chiave" base="string">
    <xs:maxLength value="10" />
  </xs:simpleType>

  <xs:complexType name="TipoAutore">
    <xs:sequence>
      <xs:element name="nome" type="TipoNome" minOccurs="1" maxOccurs="1"/>
      <xs:element name="cognome" type="TipoNome" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
```

SEGUE -->

Massimo Martinelli 2000

XML Schema

--> SEGUE

```
<xs:simpleType name="TipoCodice" base="ID">
  <xs:pattern value="[a-zA-Z]{1}\d{3}" />
</xs:simpleType>

<xs:complexType name="TipoLibro">
  <xs:attribute name="codice" type="TipoCodice" minOccurs="1" />
  <xs:sequence>
    <xs:element name="titolo" type="TipoTitolo" minOccurs="1" maxOccurs="1"/>
    <xs:element name="autore" type="TipoAutore" minOccurs="1" maxOccurs="*" />
    <xs:element name="editore" type="TipoEditore" minOccurs="1" maxOccurs="1"/>
    <xs:element name="parola_chiave" type="TipoParola_chiave"
      minOccurs="1" maxOccurs="*" />
  </xs:sequence>
</xs:complexType>

<xs:element name="biblioteca">
  <xs:complexType>
    <xs:element name="libro" type="TipoLibro" minOccurs="1" maxOccurs="*" />
  </xs:complexType>
</xs:element>

</xs:schema>
```

Stylesheet (Foglio di Stile)

Stile separato dal contenuto e dalla struttura:
non più necessario riscrivere tutto il documento ad ogni cambiamento
ma solo gli elementi necessari alla formattazione
modificati per inviare l'output ad altri dispositivi
(es: sintetizzatore vocale, foglio di carta A3) o semplicemente per
presentare su web in modo differente, o elementi diversi.

Specifiche XML: non fanno riferimento a metodi di stile

- utilizzo di CSS (Cascade Style Sheet)
specifiche <http://www.w3.org/TR/REC-CSS2>
- trasformazione dell'output in HTML
problema: i browser HTML interpretano a modo loro i tag
o addirittura utilizzano tag proprietari,
output differente.
- uso di formati proprietari
problema: utilizzazione solo su piattaforme e software specifici

CSS - Cascade Style Sheet

```
titolo { display: block;
text-align: center;
background: blue;
color: white;
font-family: Arial;
font-size: 20pt
}
autore { display: block;
margin-left: 10%;
text-align: left;
color: red;
font-family: Arial;
font-style: italic;
font-size: 14pt
}
```

```
cognome, nome { display: inline; }
editore { display: block;
margin-left: 15%;
color: green;
font-family: Arial;
font-size: 14pt
}
```

```
parola_chiave { display: block;
margin-left: 5%;
color: black;
font-family: "Times New Roman";
text-align: justify;
font-size: 14pt
}
```

Visualizzazione con I.E. 5

Esempio

XSL

Le versioni 2 e 3 di CSS hanno aggiunto nuove possibilità, ma questi Stylesheet sono limitati: non consentono modifiche al documento. Allo studio nuovi stili per XML, in particolare XSL (eXtensible Stylesheet Language), basato sul DSSL (Document Style Semantics and Specification Language), quest'ultimo utilizzato per i documenti SGML.

Le **potenzialità di XSL, nettamente superiori a CSS**, e la semplicità, fanno ritenere che questo possa essere lo stile di riferimento per XML.

XSL

XSL è costituito da 3 parti:

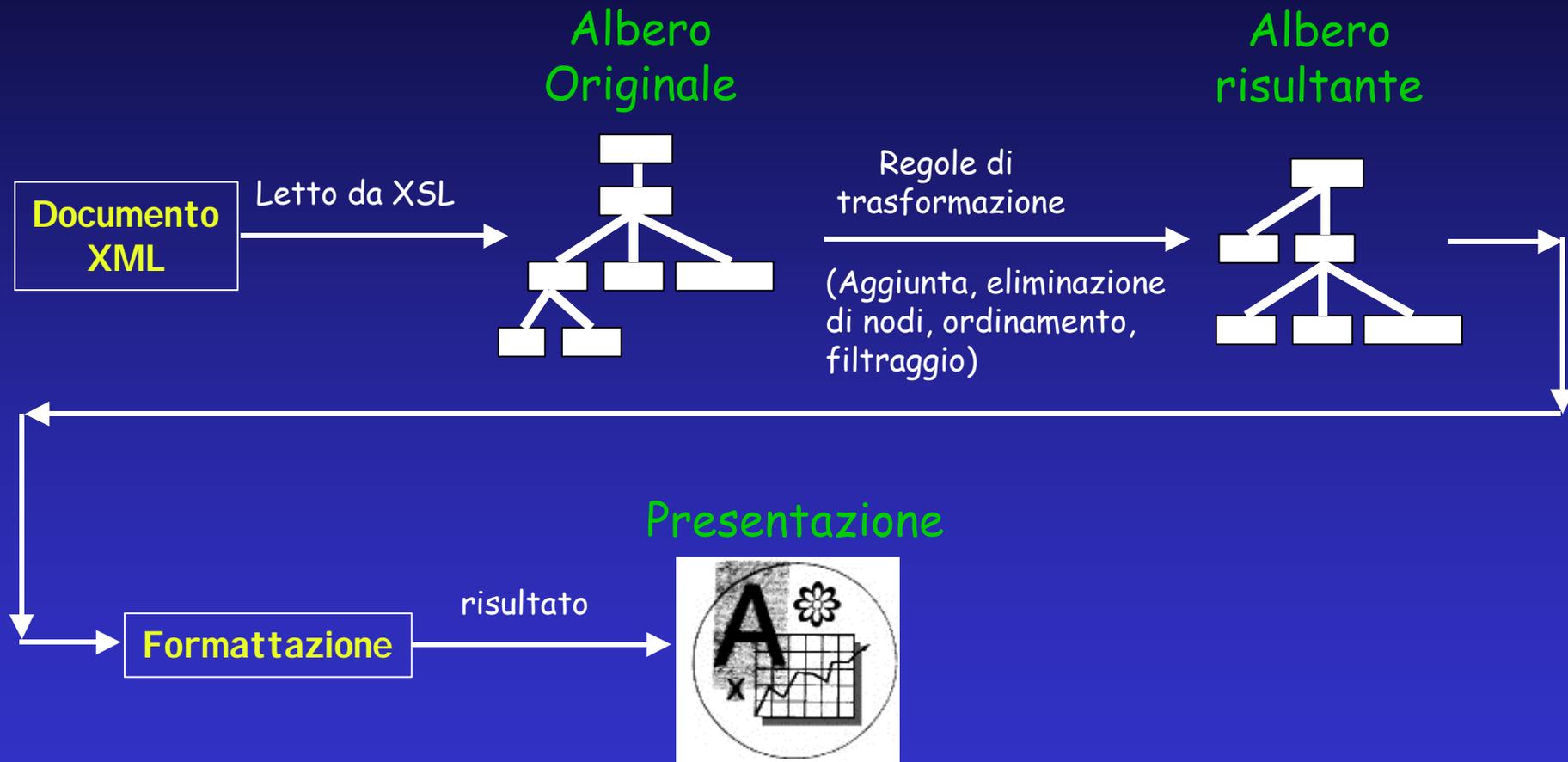
- XSLT
- XSL (F.O.)
- XPath

Working Draft: <http://www.w3.org/TR/2000/WD-xsl-20000327>

Raccomandazione: <http://www.w3.org/TR/xslt>

Raccomandazione: <http://www.w3.org/TR/xpath>

XSL



XSL

Esempi di trasformazione in HTML

[Esempio 1](#)

CON I.E.5
(vedere il file .xsl)

[Esempio 2](#)

XSL

```
<xsl:template match="element">
  <xsl:variable name="var"
                expr="subelement[position() mod 3]"/>
  <xsl:choose>
    <xsl:when test='$var=1'>
      ... do something ...
    </xsl:when>
    <xsl:when test='$var=2'>
      ...do something different ...
    </xsl:when>
    <xsl:otherwise>
      ... do something else ...
    </xsl:otherwise>
  </xsl:choose>
  <xsl:apply-templates/>
</xsl:template>
```

- Notare la dichiarazione di una variabile il cui valore è una espressione matematica

XSL

```
<xsl:template match="biblioteca">
  <xsl:apply-templates select="libro[code >= 'M']" >
    <xsl:sort select="libro/autore/cognome" />
    <xsl:sort select="libro//nome" />
  </xsl:apply-templates>
</xsl:template>
```

Dividiamo gli autori con una virgola:

```
<xsl:template match="autore">
  <xsl:value-of select="./cognome" />
  <xsl:value-of select="./nome" />
  <xsl:apply-templates />
  <xsl:if test="not(position()=last())"> , </xsl:if>
</xsl:template>
```



Introduzione ad XML

XSL F.O.

```

<?xml version='1.0' encoding='UTF-8'?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/TR/WD-xsl"
  xmlns:fo="http://www.w3.org/TR/WD-xsl/FO"
  result-ns="fo">
<xsl:template match="/">
  <fo:page-sequence
    font-family="Times New Roman, Serif"
    font-size="12pt">
    <xsl:apply-templates />
  </fo:page-sequence>
</xsl:template>
<xsl:template match="titolo">
  <fo:block
    background-color="blue"
    color="white"
    font-family="Verdana"
    font-size="18pt"
    font-weight="bold"
    text-align="centered">
    <xsl:apply-templates />
  </fo:block>
</xsl:template>
<xsl:template match="autore">
  <fo:block
    color="red"
    font-family="Arial"
    font-size="14pt"
    font-style="italic"
    start-indent='6pt'
    end-indent='6pt'>
    <xsl:apply-templates />
  </fo:block>
</xsl:template>
<xsl:template match="editore">
  <fo:block
    color="green"
    font-family="Arial"
    font-size="14pt"
    start-indent='20pt'
    end-indent='20pt'>
    <xsl:apply-templates />
  </fo:block>
</xsl:template>
<xsl:template match="parola_chiave">
  <fo:block
    color="black"
    font-family="Arial"
    font-size="12pt"
    start-indent='5pt'
    end-indent='5pt'>
    <xsl:apply-templates />
  </fo:block>
</xsl:template>
</xsl:stylesheet>

```



(Indelv)

I formatting objects ci permettono di indicare, oltre alle caratteristiche che devono avere gli elementi, anche la dimensione della pagina, i margini, salto pagina, intestazione/piè di pagina, note...

DOM - Document Object Model

IL DOM

- Una raccomandazione del W3C

<http://www.w3.org/TR-REC-dom>

- Una interfaccia indipendente dalla piattaforma e dal linguaggio per accedere e modificare documenti XML

(interfaccia codice-documento, accesso a contenuto, struttura e stile del documento)

IL LIVELLO 1 DEL DOM

- Core
- HTML

* il livello 1 consente di accedere e modificare il contenuto di un documento XML e HTML

Il livello 2 consentirà modifiche alla struttura e allo stile del documento.

I livelli successivi forniranno meccanismi per la gestione degli errori e per registrare i documenti su file (per questa ultima cosa attualmente bisogna usare procedure proprietarie).

Gli oggetti DOM

```
<messaggio> benvenuti al seminario </messaggio>
```

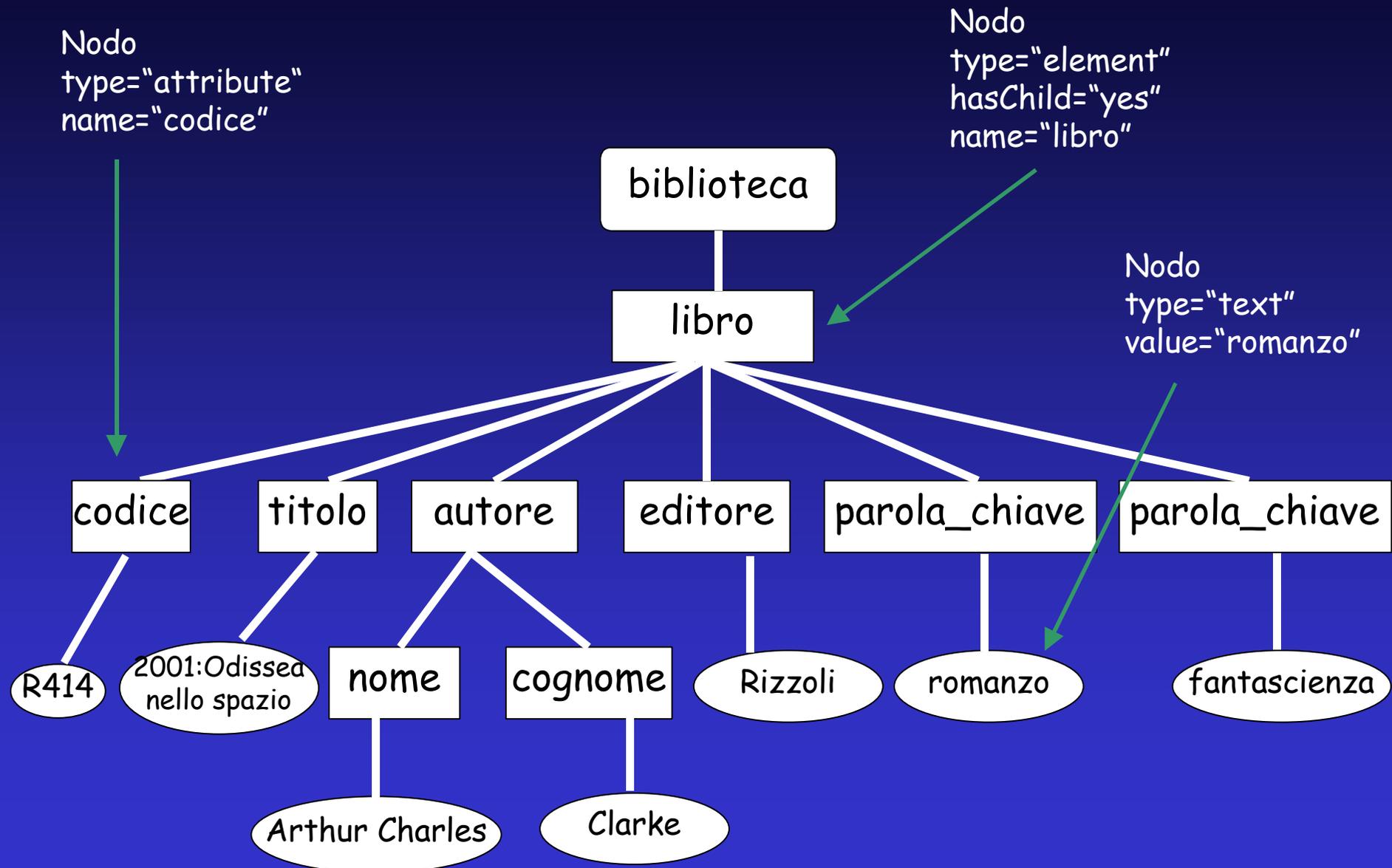
Quanti oggetti vedete ?

Classi di oggetti DOM

Il DOM contiene svariate classi di oggetti

Node	il tipo base del DOM
Element	rappresenta un elemento (il maggiore numero di oggetti con cui avremo a che fare)
Attribute	rappresenta un attributo di un elemento
Text	il contenuto di un elemento o di un attributo
Document	l'intero documento XML
ProcessingInstruction	istruzione di processo
Comment	Commento
CDATASection	sezione CDATA
DocumentFragment	frammento di documento
Entity	entità
EntityReference	riferimento ad entità
DocumentType	commento
Notation	notazione (gif, jpeg, doc, ...)

L'albero DOM



Metodi DOM #1

Ciascun nodo può essere visto come un oggetto che può avere alcuni metodi associati

Alcuni metodo offerti dal DOM:

`Node.getDocumentElement()`

`Node.getFirstChild()`

`Node.getLastChild()`

`Node.getNextSibling()`

`Node.getPreviousSibling()`

`Node.getChildNodes()` --> crea un oggetto di tipo `NodeList`

Metodi DOM #2

Supponendo di voler aggiungere un nuovo elemento a biblioteca.xml:

```
X=Document.createElement(nuovoelemento);
```

```
Document.documentElement.libro[0].appendChild(x);
```

```
X=createTextNode("contenuto del nuovo elemento");
```

```
nuovoelemento[0].appendChild(x);
```

DOM - Document Object Model

Esempio

Tramite DOM (in Javascript) visualizzo una tabella contenente nome, tipo e valore dei nodi di biblioteca.xml

creo un nuovo attributo di libro, lo chiamo posizione e gli assegno il valore "F3-51"

creo un nuovo elemento "pagine" figlio di "libro" e gli assegno il valore "215"

cambio il valore del codice del libro con "B509"

DOM - Document Object Model

The screenshot shows the Multidoc Engine web browser interface. The main content area displays "Selections from French Poetry" with entries for Paul Verlaine and Gérard de Nerval. The left sidebar contains an "Annotation" panel with a "Source Element" field set to "LIGNE", a text area containing "Comme il pleut sur la ville", and another text area for "Enter Annotation Text" containing "Bring your umbrella". Below the text areas are checkboxes for "Note", "Hint", and "Translation", and an "Attach Annotation" button. The status bar at the bottom shows the XML path: "NOTE []<-STROPHE<-POEME<-SECTION<-POESIE".

Multidoc Engine
File Edit MultidocZilla Help

Back Forward Reload Home Search StartPage Print Security Stop

Bookmarks Location: file:///J:/group/MultidocZilla Preview 101298/Demos/annotation/frame.html What's Related

CITEC OASIS HyTime.org

Annotation

Source Element: LIGNE

Comme il pleut sur la ville

Enter Annotation Text

Bring your umbrella

Note Hint Translation

Attach Annotation

NOTE []<-STROPHE<-POEME<-SECTION<-POESIE

Selections from French Poetry

Paul Verlaine
1844-1896
Paul Verlaine varied Bohemian debauchery with religious nostalgia throughout a life marked by tragic decline. His famous affair with Rimbaud culminated in his wounding him with a revolver shot in Brussels and, consequently, being sent to prison. After repenting and backsliding for some twenty years more, he died in a public infirmary. His best volumes of verse are *Fêtes galantes* 1869 and *Romances sans paroles* 1874.

Il pleure dans mon coeur ...
Il pleure dans mon coeur
Comme il pleut sur la ville

Bring your umbrella
Like it is raining on the town

Quelle est cette langueur
Qui pénètre mon coeur?

Gérard de Nerval
1808-1855
Gérard de Nerval wrote a number of excellent short tales, of which *Sylvie* is perhaps the best known. He also translated Goethe's *Faust*, and produced a good deal of journalism.

DOM - Document Object Model

Esempio

I Parser

(interfacce di accesso ai documenti XML)

DOM Parser

Basati sul DOM

Si caricano in memoria l'albero DOM poi si possono realizzare procedure e funzioni

Pro: possibilità di realizzare procedure sofisticate

Contro: uso intensivo di memoria
lentezza

SAX Parser

Ad eventi (Event Driven)

(Es.:Ogni volta che incontri il tag di apertura "nome" esegui una determinata procedura)

Pro: velocità
utile per realizzare procedure semplici

Contro: impossibile accedere a informazioni contenute nel documento a prescindere dal nodo in cui il processo di parsing si trova

Xlink - XML Linking Language

(HTML: possibili solo link unidirezionali tra due risorse)

Xlink fornisce ad XML nuove caratteristiche

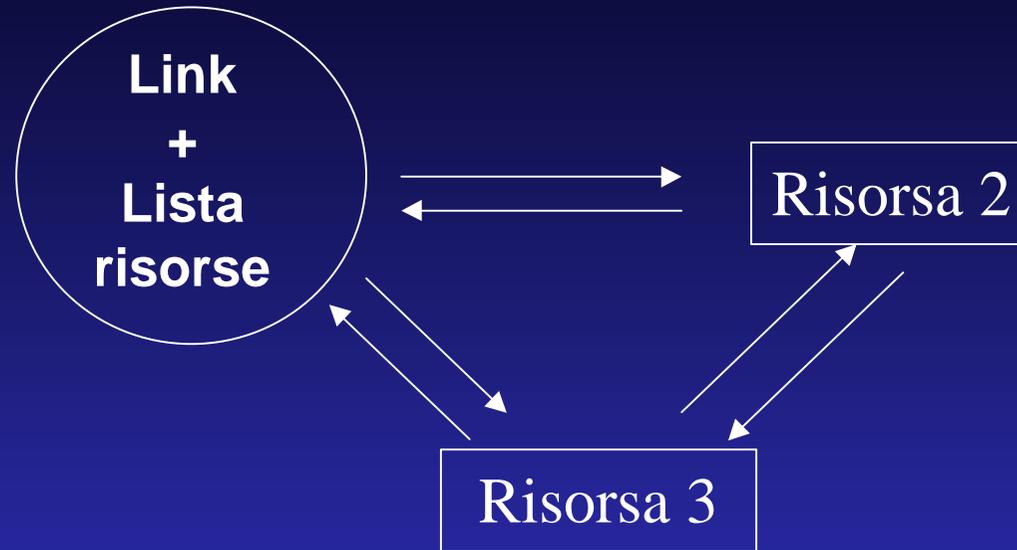
- Consente collegamenti tra più di due risorse
- Inserimento risorsa collegata
 - embed (punto preciso)
 - replace (sostituzione)
 - new (nuova finestra)
- Link bidirezionali (diverso dal link che punta indietro)
- più di una destinazione (link multiplo)
- onRequest, auto.

Working Draft:

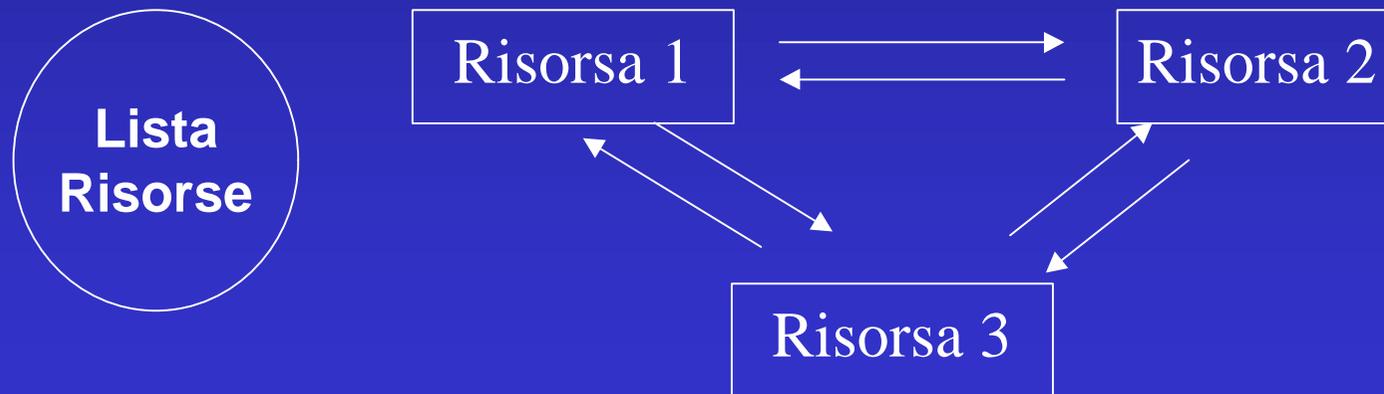
<http://www.w3.org/TR/2000/WD-xlink-20000221>

XLink - XML Linking Language

Inline Link



Out-of-line Link



XLink - XML Linking Language

```
<myns:mioelemento xmlns:xlink="http://www.w3.org/1999/xlink"
  xlink:type="simple"
  xlink:href="#target1"
  xlink:href="#target2" >
</myns:mioelemento>
```

....

```
<name ID="target1"> .... </name>
```

....

```
<title ID="target2"> .... </title>
```

```
<myns:mioelemento id="target1"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xlink:type="extended"
  xlink:href="#target2"
  xlink:href="#target3" >
</myns:mioelemento>
```

....

```
<name ID="target2"> .... </name>
```

....

```
<title ID="target3"> .... </title>
```

XLink - XML Linking Language



XLink - Uso di attributi

```
<myns:myelement>  
  xmlns:xlink="http://www.w3.org/1999/xlink"  
  xlink:type="simple"  
  xlink:href="index.html"  
  xlink:role="index"  
  xlink:title="Indice"  
  xlink:show="new"  
  xlink:actuate="onRequest">  
  Indice  
</myns:myelement>
```

XLink - Link ad arco

```
<directional xlink:from="#target1" xlink:to="#target2"/>
```

```
<extendedlink>
```

```
  <loc xlink:role:"parent" xlink:title="p1"/>
```

```
  <loc xlink:role:"parent" xlink:title="p2"/>
```

```
  <loc xlink:role:"child" xlink:title="c1"/>
```

```
  <loc xlink:role:"child" xlink:title="c2"/>
```

```
  <loc xlink:role:"child" xlink:title="c3"/>
```

```
</extendedlink >
```

```
....
```

```
<go xlink:from="parent" xlink:to="child"/>
```

```
<externallinkset>
```

```
  <linkbase xlink:href="http://linkset1.xml"/>
```

```
  <linkbase xlink:href="http://linkset2.xml"/>
```

```
  <linkbase xlink:href="http://linkset3.xml"/>
```

```
</externallinkset>
```

Alcune applicazioni

I nuovi linguaggi definiti con XML sono detti applicazioni:

Alcuni esempi:

Mathematical Markup Language (MathML)

definisce un linguaggio per la matematica

Chemical Markup Language (CML)

definisce un linguaggio per la chimica

Resource Definition Format (RDF)

strumento per descrivere i metadati

Open Software Description (OSD)

utilizzato per descrivere il software

Synchronized Multimedia Integration Language (SMIL)

utilizzato per descrivere elementi multimediali

Scalable Vector Graphics (SVG)

utilizzato per descrivere grafica vettoriale

XML Query Language (XML-QL)

un linguaggio di interrogazione e trasformazione

VoiceML, ...

MathML

Esempio

C:\PROGRAMMI\AMAYA\doc\amaya\Math.html

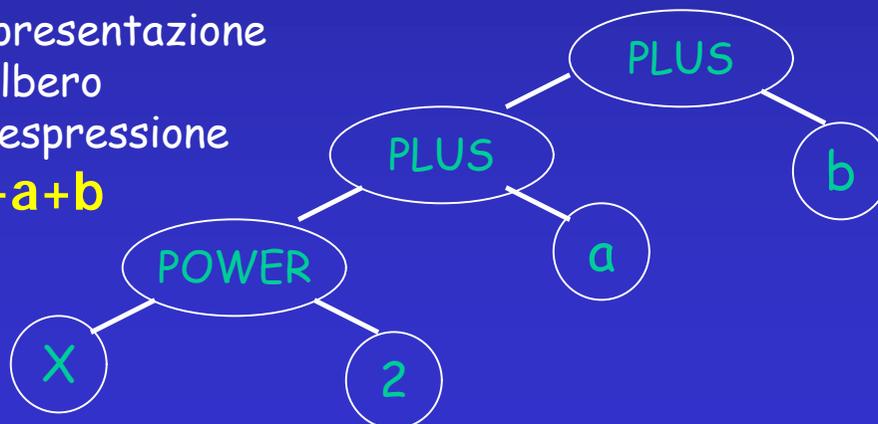
```

<MROW>
  <MI>x</MI>
  <MO>x<TIMES/>x</MO>
  <MN>2</MN>
</MROW>
<MO>x<PLUS/>x</MO>
<MI>a</MI>
<MO>x<PLUS/>x</MO>
<MI>b</MI>

```

Rappresentazione
ad albero
dell'espressione

X^2+a+b



Algebra, aritmetica, insiemi,
funzioni, relazioni, serie,
statistica, trigonometria, ...

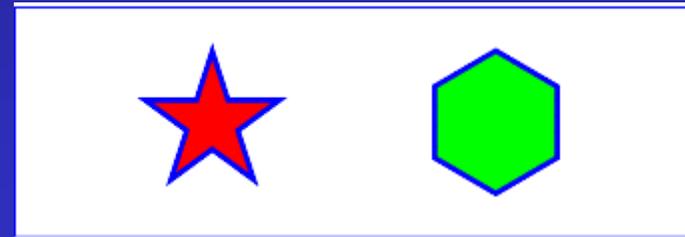
MI = identificatore
MO = operatore matematico
MN = dato numerico
PLUS = addizione
MROW = tipo layout (riga)

TIMES= alla potenza di
MSQRT= radice quadrata
SET=insieme
INTERSECT=intersezione
SUBSET= sottoinsieme
FN=funzione definita dall'utente
MATRIX=matrice
LIMIT x TENDSTO 0 =
limite di x che tende a 0
TAN=tangente
INT=integrale definita
DIFF=derivata
VAR=variazione
SDEV=deviazione standard

Scalable Vector Graphics - SVG

Un linguaggio per descrivere vettori bidimensionali e grafica mista vettoriale/raster in XML
Grafica offerta: Vettori, Poligoni, colori, filtri, animazioni, ...

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG December 1999//EN"
  "http://www.w3.org/Graphics/SVG/SVG-19991203.dtd">
<svg width="12cm" height="4cm" viewBox="0 0 1200 400">
  <desc>Example polygon01 - star and hexagon</desc>
  <polygon style="fill:red; stroke:blue; stroke-width:10" points="350,75 379,161
469,161 397,215 423,301 350,250 277,301 303,215 231,161 321,161" />
  <polygon style="fill:lime; stroke:blue; stroke-width:10" points="850,75 958,137.5
958,262.5 850,325 742,262.6 742,137.5" />
</svg>
```



Esempio

Working Draft: <http://www.w3.org/TR/1999/WD-SVG-19991203/index.html>

Viewers: <http://www.cadcamcenter.com/cadcam/svg.htm>

<http://www.adobe.com> (plugin)

Synchronized Multimedia Integration Language - SMIL

Esempio

Resource Description Framework - RDF

Massimo Martinelli è il creatore della risorsa <http://www.iei.pi.cnr.it/nota-xml>

In RDF descritta da:

Soggetto (Risorsa) <http://www..iei.pi.cnr.it/nota-xml>

Predicato (Proprietà) Creator

Oggetto (letterale) "Massimo Martinelli"

Grafo direzionale etichettato ("diagrammi a nodi e archi")

Nodi (ovali) rappresentando le risorse.

Archi rappresentano le proprietà.

Rettangoli: nodi che rappresentano stringhe letterali.



La direzione della freccia è importante.

Può anche essere letto come

"<http://www.iei.pi.cnr.it/nota-xml> ha creator Massimo Martinelli "

o in generale <soggetto> ha <predicato> <oggetto> ".

```
<rdf:RDF>
```

```
<rdf:Description about="http://www.iei.pi.cnr.it/nota-xml">
```

```
<dc:Creator>Massimo Martinelli</dc:Creator>
```

```
</rdf:Description>
```

```
</rdf:RDF>
```

XML Query Language (XML-QL)

Un linguaggio di interrogazione e di trasformazione per documenti XML

```
WHERE <libro>
    <editore> Rizzoli </editore>
    <titolo> $t </titolo>
    <autore> $a </autore>
</libro> IN "www.iei.pi.cnr.it/Personal/Martinelli/biblioteca.xml"
CONSTRUCT <risultato>
    <autore> $a </autore>
    <titolo> $t </titolo>
</risultato>
```

Cerca tutti gli elementi "libro" nel file "www.iei.pi.cnr.it/.../biblioteca.xml" che abbiano un elemento editore con valore "Rizzoli" e almeno un elemento titolo e un elemento autore

```
<risultato>
    <autore><cognome>Clarke</cognome><nome>Arthur Charles</nome></autore>
    <titolo> 2001: Odissea nello spazio </titolo>
</risultato>
```

II software

- Parser:** MSXML (Microsoft), XML for Java (IBM)
DXP (DataChannel), Oracle, Sun, ...
- Interfacce ai parser:** Sax, DOM
scritte in Java, Perl, Python, Frontier, C++...
- Editor:** Xmetal(Softquad), FrameMaker-SGML (Adobe),
Adept (ArborText), Balise (Chrystal),
XML-Authority, ...
- Database:** Oracle, DB2 (annunciato), Tamino, Excelon, ...
- Browser:** IE5.*, Netscape 6 preview 1, Indelv, Opera,
Jumbo (usato per CML),
Amaya (usato per MathML), ...

Conclusioni: perché utilizzare XML

- Problemi principali da risolvere sul Web: lentezza, difficoltà trovare informazioni.
- XML permette di strutturare i documenti e di associare una sintassi.
- Riduzione del traffico di rete, maggiore sviluppo applicazioni client.
- XML supporta UNICODE, un grande numero di set di caratteri può essere utilizzato.
- XML può essere usato come formato di scambio per documenti elettronici ed applicazioni.
- Indipendente da hardware e software
- Struttura contenente di manipolare i dati in modo più semplice ed efficiente.
- Quando registriamo informazioni vogliamo essere sicuri di poterle riutilizzare in futuro.
(es. word 1, word 2, word per mac, word per pc, incompatibilità, perdita di informazioni)
- Un documento XML è in formato leggibile dall'uomo.
- Documento strutturati e metainformazioni consentono motori di ricerca più accurati
("ALBERO BINARIO")
- Sistemi standard di metadati: Resource Description Framework.
- XML è estensibile, non ha tag predefiniti.
- Consente di creare linguaggi standard ed estendibili, per campi di applicazione
(medicina, elettronica, matematica, musica, ...)
- Con il DOM + semplice scrivere un programma che trova gli elementi e li usa
- I collegamenti offrono nuove possibilità. (punti precisi, più destinazioni, database centralizzato -> maggiore maneggevolezza, più semplice controllare corrispondenza collegamenti).
- Offre una ottima capacità di rappresentare dati complessi (notazioni matematiche, interfacce grafiche)
- Visualizzare documento su media differenti in modi diversi senza doverlo riscriv. ogni volta.
- XSL offre meccanismi per rappresentare e manipolare il documento, buona capacità di rappresentare dati complessi (notazioni matematiche, interfacce grafiche); sequenze, cicli e condizioni
(N-regine)

Dove reperire altre informazioni

Un elenco di indirizzi che può costituire un buon punto di partenza:

<http://www.w3.org/XML> (la home page di XML sul sito del W3C)

<http://www.ucc.ie/xml> (FAQ)

<http://www.oasis-open.org/cover/sgml-xml.html> (SGML/XML group)

<http://www.microsoft.com/xml>

<http://www.mozilla.org>

http://www.gca.org/conf/xml/xml_what.htm

<http://www.arbortext.com/xml.html>

<http://www.xmlu.com>

<http://www.xml.org>

<http://www.netscape.com>

<http://www.iei.pi.cnr.it/Personal/Martinelli/XML/Doc/IntroduzioneXML0205.PDF>

http://www.iei.pi.cnr.it/Personal/Martinelli/XML/Doc/XML-A_Technical_Introduction.PDF

Il Gruppo XML Italia

Il Gruppo XML Italia si è strutturato nell'ottobre del '98 come gruppo di lavoro aperto, basando le proprie procedure sul modello dei gruppi tecnici di Internet.

Scopo primario è quello di aggregare le competenze esistenti a livello nazionale per diffondere in modo organizzato la cultura XML

Le finalità principali:

- trasferire le conoscenze in ambito nazionale
- facilitare la discussione via e-mail e meeting periodici;
- essere un riferimento di informazioni su XML sia per gli sviluppatori sia per gli utenti;
- mostrare l'uso di queste nuove tecnologie tramite prototipi e semplici applicazioni.

II Gruppo XML Italia

<http://www.xml.it> Il sito di XML Italia

Le mailing-list del gruppo XML-Italia:

org@xml.it riservata all'organizzazione

xml@xml.it aperta, dedicata alle attività tecnico-informative

consultabili anche su web:

<http://listserv.xml.it/org.html>

<http://listserv.xml.it/xml.html>

Evento principale organizzato quest'anno:

"II Workshop XML Italia"

Pisa 9/10 Maggio 2000

Gli atti saranno pubblicati on-line sul sito del gruppo

Ringraziamenti

Andrea Marchetti

Piero Maestrini

Anna Vaccarelli

Laura Abba

Vinicio Lami

Stefania Biagioni

Carol Peters

il Gruppo XML Italia

Per l'amicizia,
le opportunità
e i mezzi offerti

Sicuramente mi sto dimenticando qualche altra persona,
nel qual caso spero di rimediare al più presto