

Contenuti del Seminario "XML – Una introduzione"
6 Luglio 2001 - Comune di Livorno

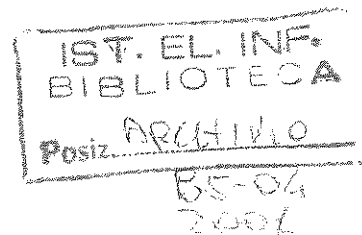
Massimo Martinelli - M.Martinelli@iei.pi.cnr.it
Consiglio Nazionale delle Ricerche
Istituto di Elaborazione dell'Informazione

La necessità di utilizzare e rappresentare informazioni complesse, informazioni che derivano dalle nuove tecnologie e dai nuovi media, ed inoltre la necessità di recuperare ed elaborare tali informazioni.

Una panoramica iniziale (da zero, senza prerequisiti) su XML (eXtensible Markup Language), un linguaggio estensibile realizzato per poter utilizzare in modo semplice i documenti strutturati, studiato per il Web e per superare i limiti di HTML (Hyper Text Markup Language), ma con possibilità di utilizzo in altri ambienti.

Le componenti fondamentali del nuovo linguaggio, che cosa è, a che cosa serve, alcuni cenni sulla vasta famiglia dei linguaggi XML.

Alcuni esempi di possibili utilizzi attuali, suggerimenti per applicazioni future.





Comune di Livorno



XML eXtensible Markup Language

Introduzione

Massimo Martinelli

6/7/2001

M.Martinelli@iei.pi.cnr.it



Sommario



- **Che cosa è XML**
- **A cosa può servire**
- **I limiti di HTML**
- **Un documento XML**
- **Definizione della struttura del documento (DTD)**
- **Gli Schemi (XML - Schema)**
- **Lo stile (Stylesheet CSS, XSL)**
- **Interfacce di accesso al documento XML**
- **Applicazioni XML**
- **Riferimenti**



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
 - prima bozza di XML: novembre 1996
 - Extensible Markup Language (XML) 1.0 (Second Edition)
<http://www.w3.org/TR/REC-xml>
 - traduzione in italiano: <http://www.xml.it/REC-xml-19980210-it.html>



Markup

(marca, etichetta)

- Markup: tutto ciò che ha un significato speciale che deve essere ben caratterizzato, reso esplicito
- anche identificatore, simbolo o altro espediente per distinguere un elemento da altri simili, può indicare l'inizio o la fine di un oggetto
- Esempi di markup: testo in corsivo, testo sottolineato
- 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



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



Comune di Livorno

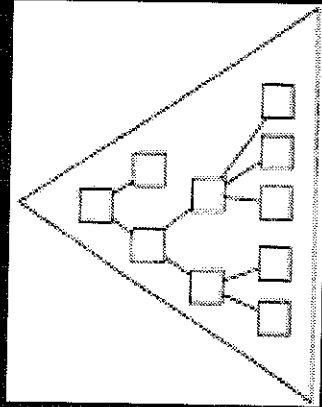


Le componenti di XML

Contenuto

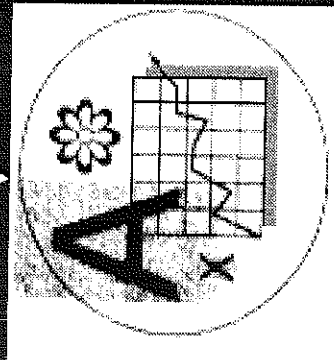
"La Discovery stava accelerando verso Giove lungo un'orbita complessa calcolata alcuni mesi prima dagli astronomi sulla Terra e controllata costantemente da Hal"
A.C. Clarke

Struttura



Specifiche di formattazione dell'output

Formattazione



Processo completo di codifica XML

Rappresentazione (Stile)
6/7/2001

Massimo Martinelli

M.Martinelli@iei.pi.cnr.it



Comune di Livorno

Un semplicemarkup con HTML



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

Rappresentazione:

Sig. Mario Rossi
Via Verdi, 12
56100, Pisa

Elaborazione ?

Interpretazione ?

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>
```

```
</indirizzo>
```

```
</business-card>
```

Rappresentazione:

Sig. Mario Rossi

Via Verdi, 12

56100, Pisa

Sig. Mario Rossi

Via Verdi, 12

56100, Pisa

Interpretazione di XML

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



Il documento

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

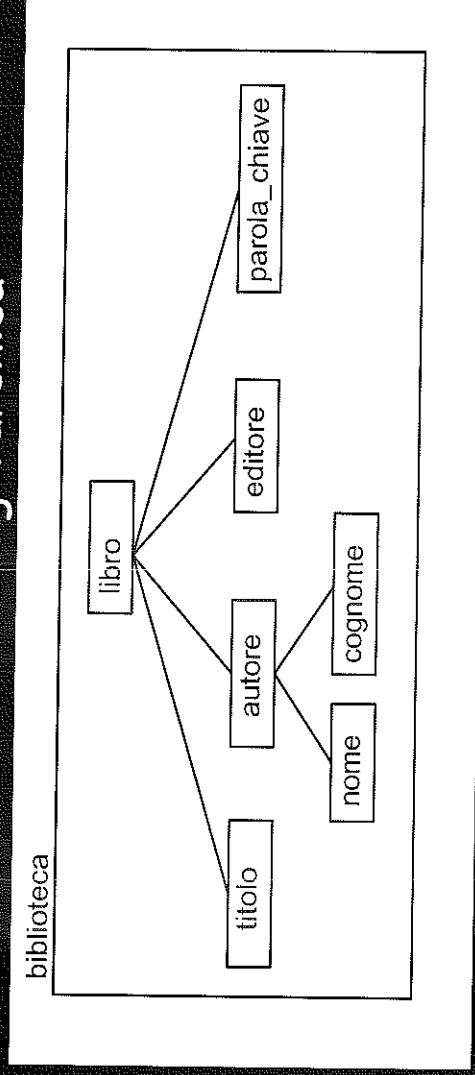
```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro codice="R414">
    <titolo>2001: Odissea nello spazio</titolo>
    <autore>
      <cognome>Clarke</cognome>
      <nome>Arthur Charles</nome>
    </autore>
    <editore>Rizzoli</editore>
    <parola_chiave>romanzo</parola_chiave>
    <parola_chiave>fantascienza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

<nome> ≠ <Nome> ≠ <NOME>



DTD - Document Type Definition

- contiene le regole di definizione dei tag
 - indica gli elementi e il loro ordine all'interno del documento XML
 - può essere interno o esterno al documento XML
 - il suo nome per convenzione corrisponde a quello dell'elemento radice
- Struttura gerarchica



- 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.



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)>
]>
```



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.



Elemento senza contenuto

<tag attre="3" > </tag>

<tag attre="3"/ >

Sintassi equivalente

Nuova sintassi

x i tag

di chiusura

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

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" >



XML Schema

- non supportano i namespace
- non sono scritti in XML
- non sono estensibili
- non specificano tipi di dato

Limiti delle DTD:

Soluzione che estende 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,...)

Raccomandazione: unione diverse soluzioni proposte (XML-Data, DCD, SOX, DDML, XDR) in una comune. Composta da due parti: strutture e tipi di dati.

Tipi di dati: primitivi, derivati e complessi,

- Primitivi (intero, stringa, ...)
- Tipi derivati
- Tipi complessi



Comune di Livorno

XML Schema



```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:simpleType name="TipoTitolo">
    <xsd:restriction base="string">
      <xsd:maxLength value="60" />
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="TipoNome">
    <xsd:restriction base="string">
      <xsd:maxLength value="30" />
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="TipoParola_chiave" base="string">
    <xsd:restriction base="string">
      <xsd:maxLength value="10" />
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="TipoAutore">
    <xsd:sequence>
      <xsd:element name="nome" type="TipoNome" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="cognome" type="TipoNome" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>

```

Massimo Martinelli

SEGUITE/2001

M.Martinelli@iei.pi.cnr.it



Comune di Livorno

--> SEQUE

XML Schema



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

<xsd:complexType name="TipoLibro">
<xsd:attribute name="codice" type="TipoCodice" use="required" />
<xsd:sequence>
<xsd:element name="titolo" type="TipoTitolo" minOccurs="1" maxOccurs="1"/>
<xsd:element name="autore" type="TipoAutore" minOccurs="1" maxOccurs="unbounded" />
<xsd:element name="editore" type="TipoNome" minOccurs="1" maxOccurs="1"/>
<xsd:element name="parola_chiave" type="TipoParola_chiave"
minOccurs="1" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>

<xsd:element name="biblioteca">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="libro" type="TipoLibro" minOccurs="1" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

Massimo Martinelli

6/7/2001

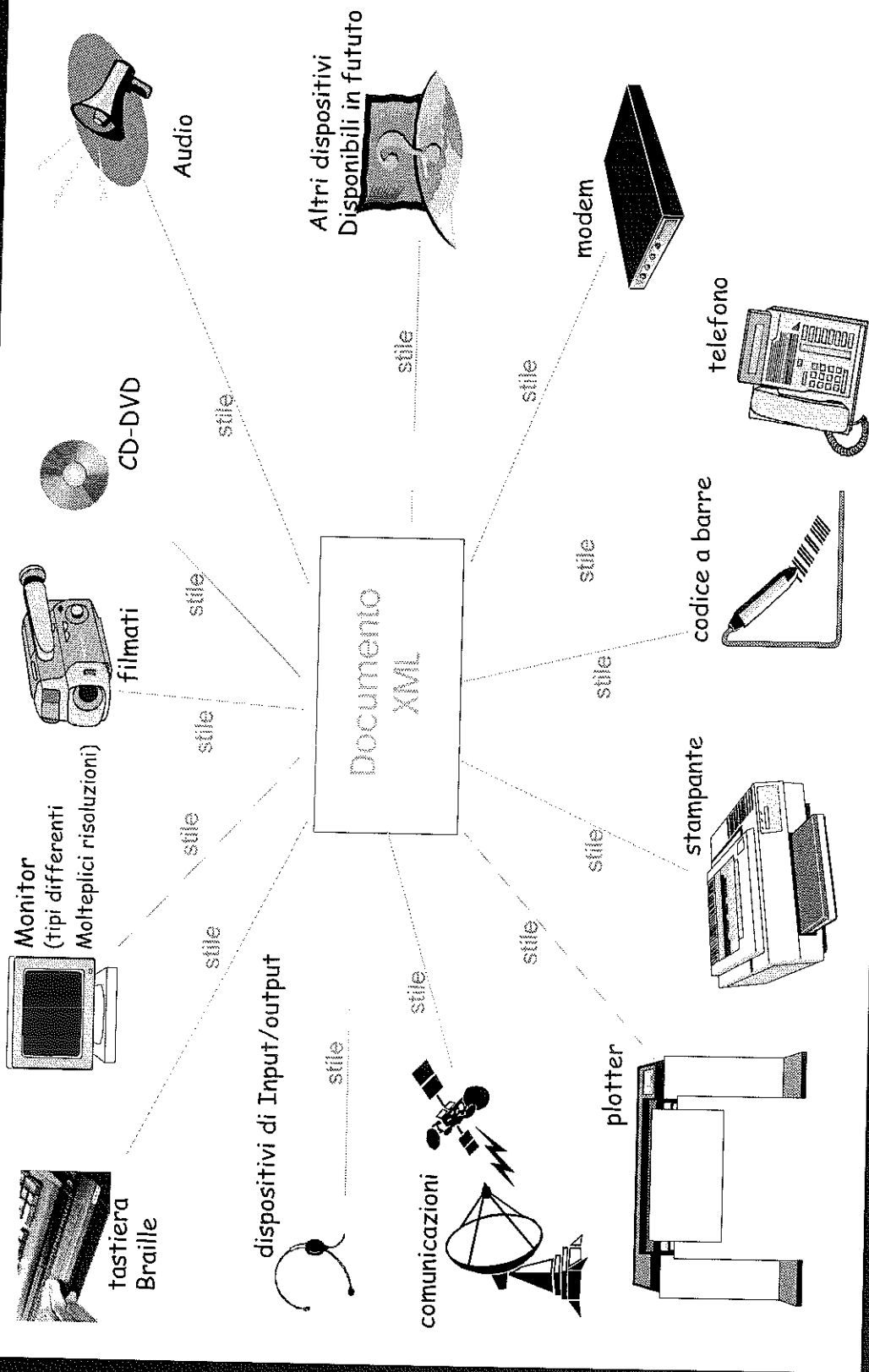
M.Martinelli@iei.pi.cnr.it



Comune di Livorno



Stylesheet (Foglio di Stile) Rappresentazione di markup XML



Massimo Martinelli

6/7/2001

M.Martinelli@iei.pi.cnr.it



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
}
```

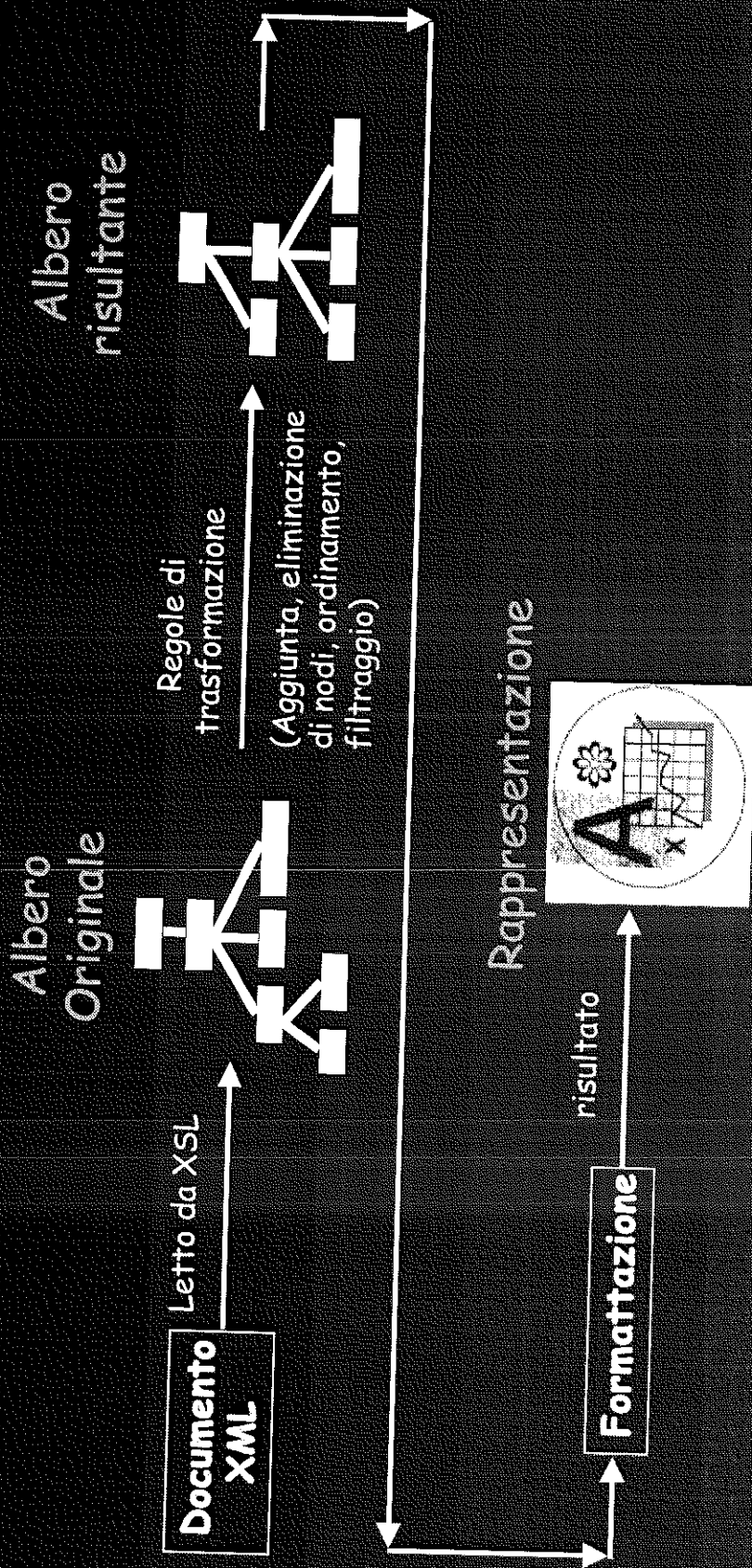
```
cognome, nome { display: inline; }
editore { display: block;
margin-left: 15%;
color: green;
font-family: Arial;
font-size: 14pt
}
parola_chiave { display: block;
margin-left: 5%;
color: black;
font-family: "Times New Roman";
text-align: justify;
font-size: 14pt
}
```




Comune di Livorno



XSL - extensible Stylesheet Language (XSLT, Xpath, XSL-FO)



Massimo Martirelli

6/7/2001

M.Martirelli@iei.pi.cnr.it



XSLT



```

<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>

<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:text> </xsl:text>
  <xsl:value-of select="/nome" />
  <xsl:apply-templates />
  <xsl:if test="not(position()=last())">,</xsl:if>
</xsl:template>

```

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



Comune di Livorno

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

I livelli successivi consentiranno 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).

6/7/2001

Massimo Martinelli

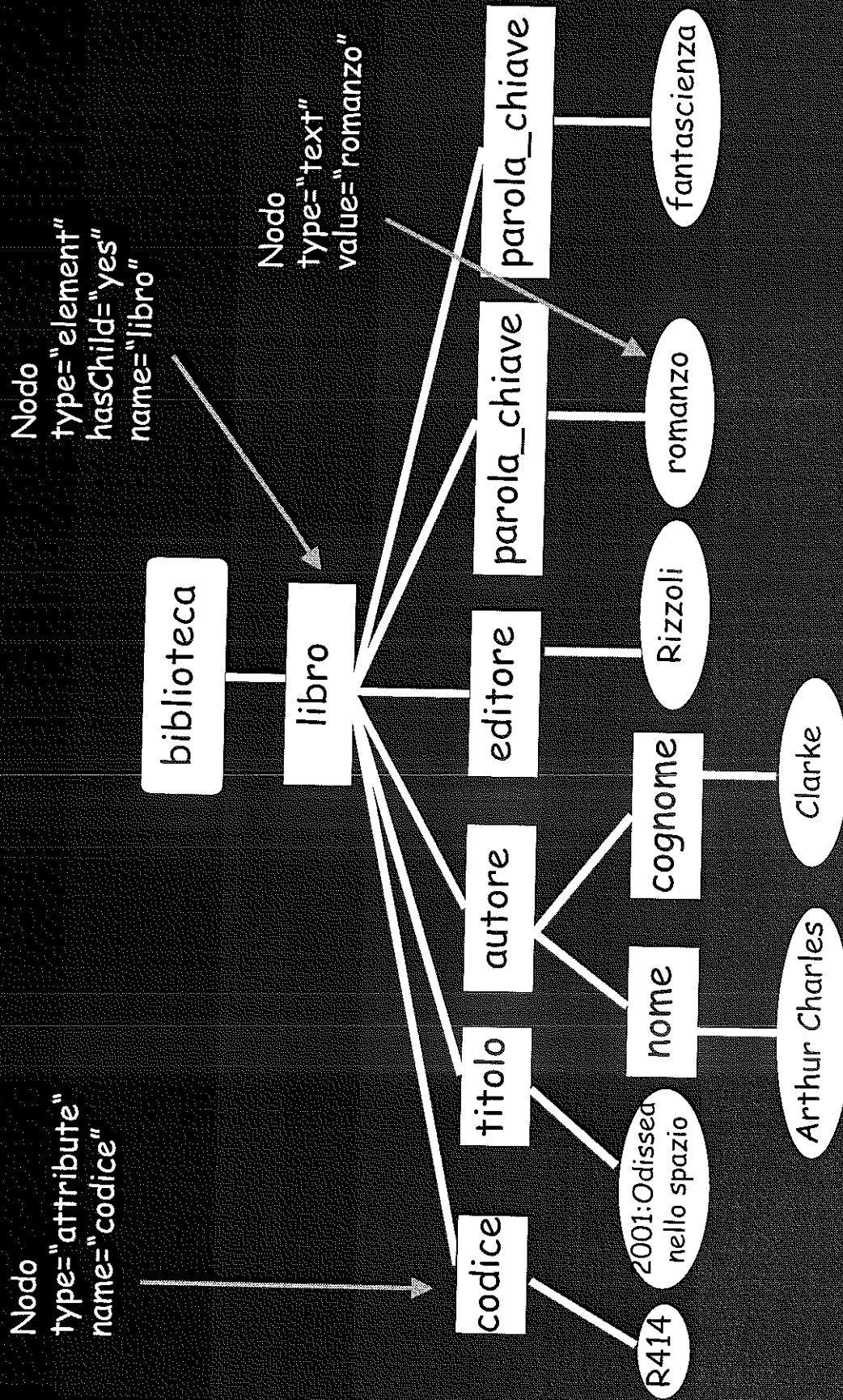
M.Martinelli@iei.pi.cnr.it



Comune di Livorno



L'albero DOM



Massimo Martinelli

6/7/2001

M.Martinelli@iei.pi.cnr.it



Metodi DOM #1

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

Alcuni metodo offerti daDOM:

```
Node.getDocumentElement()
Node.getFirstChild()
Node.getLastChild()
Node.getNextSibling()
Node.getPreviousSibling()
Node.getChildNode()
```

--> crea un oggetto di tipo NodeList

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

```
X=document.createElement(nuovoelemento);
document.documentElement.appendChild(X);
X=createTextNode("contenuto del nuovo elemento");
nuovoelemento.appendChild(X);
```




Comune di Livorno



Le principali tecnologie associate ad XML

DOM	SVG	XLink
MathML	UNICODE	XML Protocol
Micropayments	Voice XML	XML Query
Mobile	WAI	XML Schema
PICS	WebCGM	XML Signature
Privacy, P3P	Web Services	XPath
RDF	XForms	Xpointer
Semantic Web	XHTML	XSL, XSLT
SMIL	XML Encryption	...

Massimo Martignelli

6/7/2001

M.Martignelli@iei.pi.cnr.it



Comune di Livorno

Alcune applicazioni

- Mathematical Markup Language (MathML)
definisce un linguaggio per la matematica
- Chemical Markup Language (CML)
definisce un linguaggio per la chimica
- Resource Definition Framework (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
- XML Digital Signature (XML-DSig)
utilizzato per poter utilizzare la firma elettronica
- VoiceML, ...

Massimo Martinelli

6/7/2001

M.Martinelli@iei.pi.cnr.it



Comune di Livorno

- link

SMIL



Massimo Martinelli

6/7/2001

M.Martinelli@iei.pi.cnr.it



Comune di Livorno



Resource Description Framework RDF

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

In RDF descrittta 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) rappresentano 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 <oggetto> 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>

Massimo Martinelli

6/7/2001

M.Martinelli@iei.pi.cnr.it



Comune di Livorno

Riferimenti



<http://www.w3org/XML>

<http://www.w3org>

<http://www.w3org/TR/>

<http://www.xml.org>

<http://www.semanticweb.org>

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

<http://www.mozilla.org>

<http://www.xml.com>

<http://www.netscape.com>

<http://www.xmlsoftware.com>

<http://listserv.xml.it/xml.html> (xml@xml.it) gruppo XML-Italia

<http://www.iei.pi.cnr.it/Personal/Martinelli/XML/Doc/IntroduzioneXML060701.PDF>

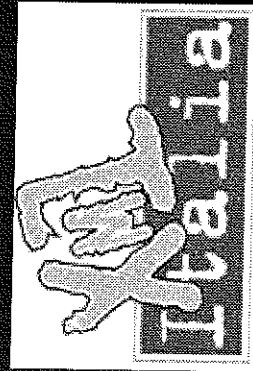
http://www.iei.pi.cnr.it/Personal/Martinelli/XML/Doc/Nota_XML.html

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

Massimo Martinelli

6/7/2001

M.Martinelli@iei.pi.cnr.it





Comune di Livorno

Vantaggi XML

- Estensibile
- Si descrive da se
- Semplice, flessibile
- Facile da elaborare
- Formato leggibile dall'uomo
- Indipendente dalla piattaforma
- Riusabile

Massimo Martinelli

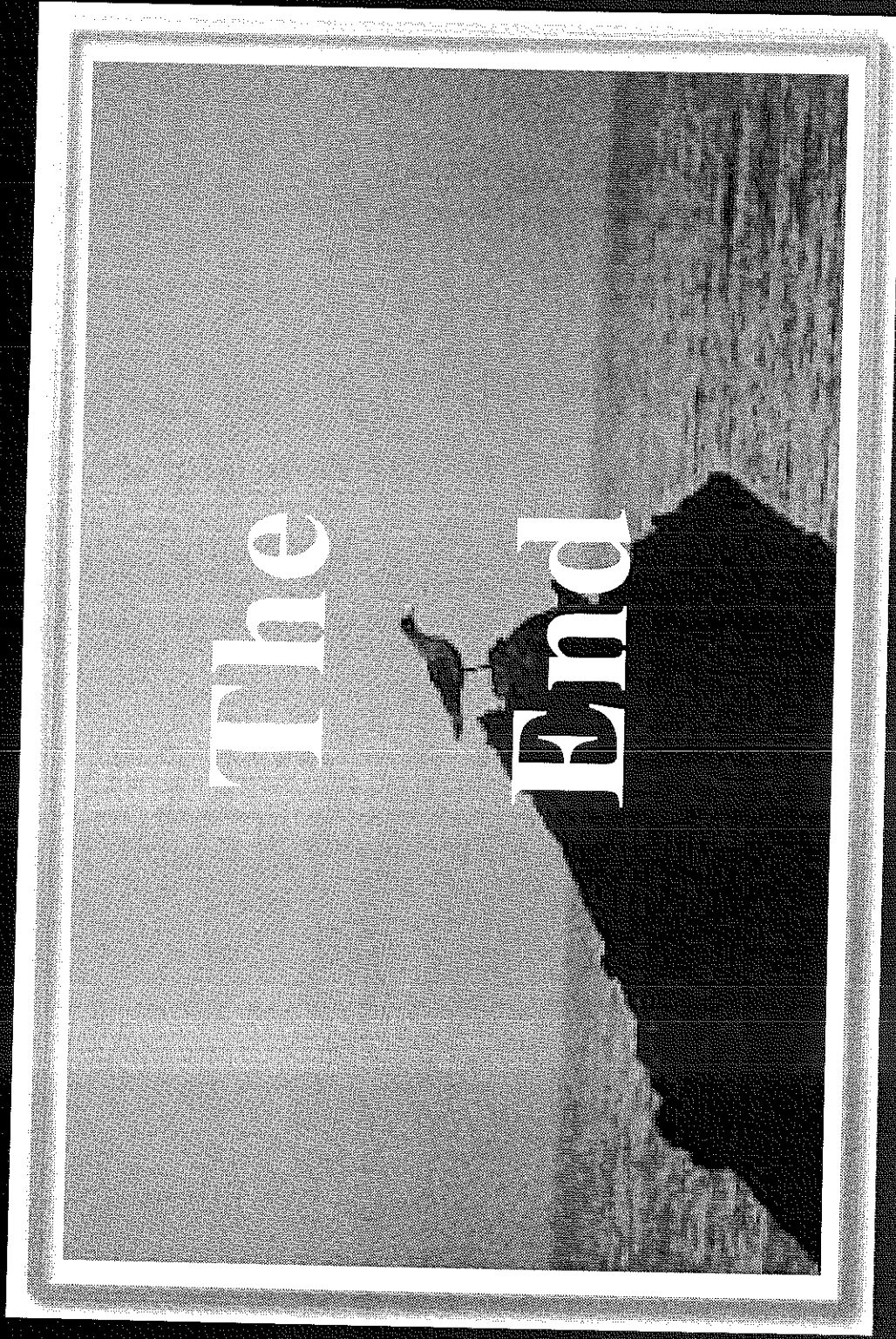
6/7/2001

M.Martinelli@iei.pit.cnr.it





Comune di Livorno



Massimo Martinelli

6/7/2001

M.Martinelli@iei.pi.cnr.it