

La struttura: DTD

Laura Farinetti

Dip. Automatica e Informatica

Politecnico di Torino

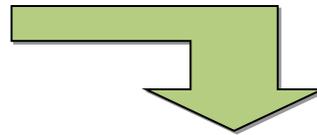
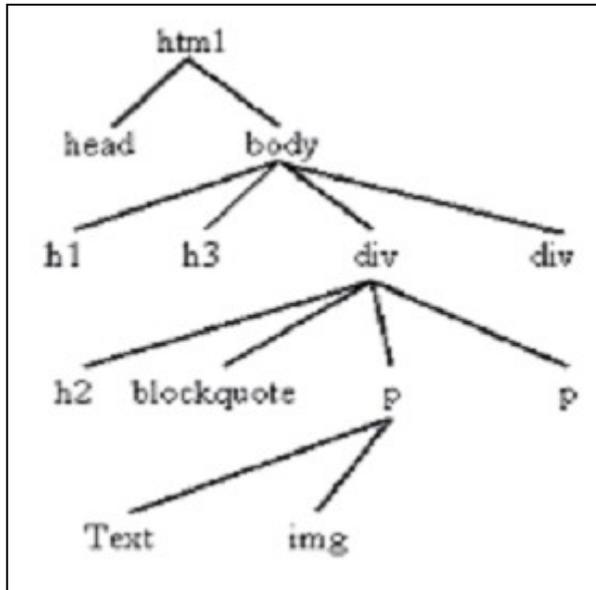
`laura.farinetti@polito.it`



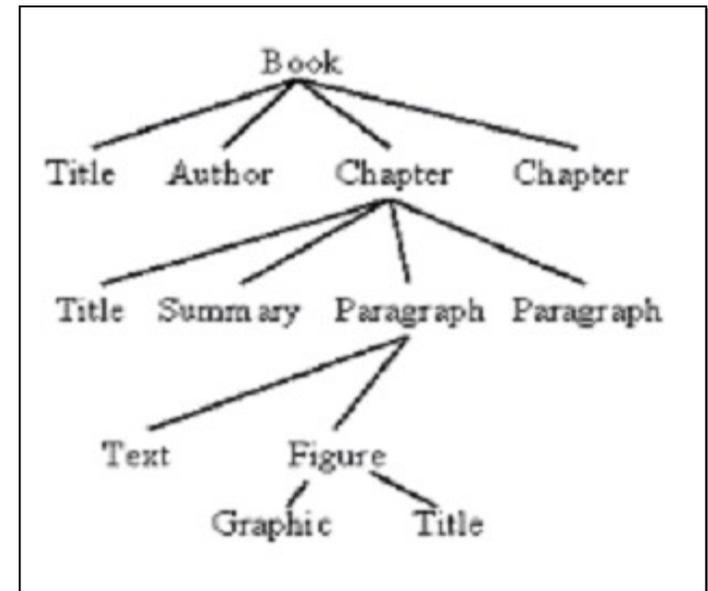
Modello di struttura

- I tag annidati danno origine ad una struttura gerarchica, ad albero
- Ciascun elemento e ciascuna porzione di testo sono rappresentati da un nodo
- Ciascun nodo, se corrisponde ad un elemento non vuoto, ha dei nodi figli

Esempi di struttura ad albero



```
<html>  
<head>...</head>  
<body>  
<h1></h1>  
<h3></h3>  
.....  
</body>  
</html>
```





Document Object Model

- La struttura ad albero può essere rappresentata mediante un modello ad oggetti (DOM)
- La comprensione del DOM sarà essenziale per
 - Trasformazioni XSL
 - Fogli di stile CSS
 - Scripting Javascript



Definizione della struttura

- Regole di definizione dei tag
- Ordine dei tag all'interno del documento XML
- Regole per il contenuto dei tag
- Documento XML valido oltre che documento XML ben formato
- In pratica è la grammatica di un linguaggio basato su XML



Linguaggi per esprimere la struttura

• DTD: Document Type Definition

- eredità di SGML: sintassi non XML
- non è in grado di esprimere tutti i vincoli
- molto diffuso

• XML Schema

- specifica più recente
- sintassi XML
- maggiore ricchezza ed espressività
 - es: supportano diversi tipi di dato: stringhe, numeri reali, interi, decimali, boolean, data e ora, URI, ...

Modalità di specifica del DTD

• DTD esterno

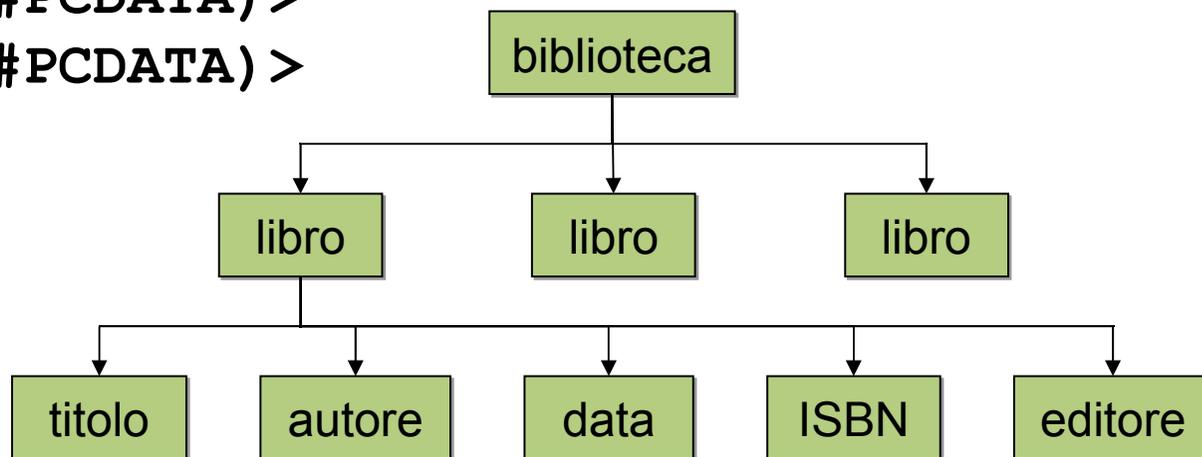
```
<?xml version="1.0"?>  
<!DOCTYPE greeting SYSTEM "hello.dtd">  
<greeting>Hello, world!</greeting>
```

• DTD interno

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE greeting [  
  <!ELEMENT greeting (#PCDATA)>  
>  
<greeting>Hello, world!</greeting>
```

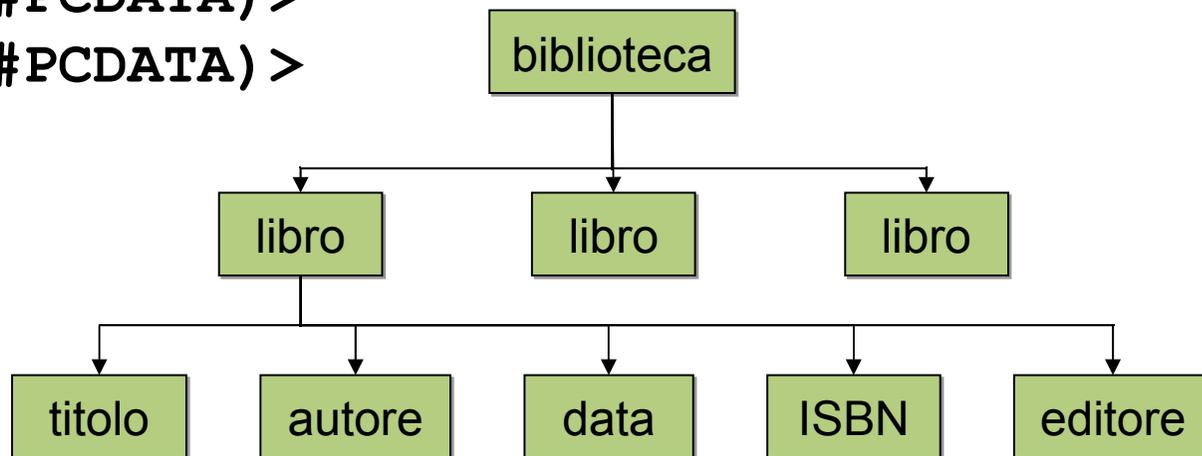
Esempio di DTD (esterno)

```
<?xml version="1.0"?>
<!ELEMENT biblioteca (libro+)>
<!ELEMENT libro (titolo, autore+, data,
                 ISBN, editore)>
<!ELEMENT titolo  (#PCDATA)>
<!ELEMENT autore  (#PCDATA)>
<!ELEMENT data    (#PCDATA)>
<!ELEMENT ISBN    (#PCDATA)>
<!ELEMENT editore (#PCDATA)>
```



Esempio di DTD (interno)

```
<!DOCTYPE biblioteca [  
<!ELEMENT biblioteca (libro+)>  
<!ELEMENT libro (titolo, autore+, data,  
                ISBN, editore)>  
<!ELEMENT titolo (#PCDATA)>  
<!ELEMENT autore (#PCDATA)>  
<!ELEMENT data (#PCDATA)>  
<!ELEMENT ISBN (#PCDATA)>  
<!ELEMENT editore (#PCDATA)>  
>
```





Sintassi dei DTD

- Un DTD è una **grammatica** che definisce quali elementi (tag), attributi ed entità **sono validi** in un documento XML
- Le specifiche XML usano la notazione EBNF (Extended Backus-Naur Form) per definire i costrutti XML validi

Regola di produzione EBNF

```
persona ::= (nome e-mail*)
```

Dichiarazione di un elemento in un DTD

```
<!ELEMENT persona (nome, e-mail*)>
```

Dichiarazione di un elemento non vuoto

DTD

```
<!ELEMENT agenda (contatto*)>  
<!ELEMENT contatto (nome, indirizzo+,  
    telefono*, email, foto?)>  
<!ELEMENT nome (#PCDATA)>  
...
```

XML

```
...  
<nome>Giuliana</nome>  
...
```

Elementi e regole

Definizione	Significato
$A?$	A è opzionale
A^+	Una o più occorrenze di A
A^*	Zero o più occorrenze di A
$A B$	A oppure B ma non entrambi
A, B	A seguito da B, in questo ordine
$(A, B)^+$	Una o più occorrenze di (A seguito da B)
$\#PCDATA$	Stringa di dati

Dichiarazione di un elemento vuoto

DTD

```
<!ELEMENT foto empty>
```

XML

```
...  
<foto />  
...
```

Dichiarazione degli attributi

DTD

```
<!ATTLIST foto
  src  CDATA  #REQUIRED
  desc CDATA  #IMPLIED>
<!ATTLIST telefono
  tipo (casa|ufficio|mobile) "casa">
```

XML

```
...
<foto src="io.jpg"/>
<telefono tipo="mobile">0348.1234567
  </telefono>
```

Tipi di attributo

Attributo	Significato
CDATA	Stringa
ID	Nome unico all'interno del documento: identifica univocamente l'elemento
IDREF	Riferimento ad un elemento che ha un attributo di tipo ID con valore IDREF
IDREFS	Lista di IDREF separati da uno spazio bianco
ENTITY	Nome di un'entità dichiarata nel documento
ENTITIES	Lista di ENTITY separate da uno spazio bianco
MMTOKEN	Stringa di una sola parola
MMTOKENS	Lista di MMTOKEN separati da uno spazio bianco
Enumerazione	Lista di nomi che rappresentano i valori che l'attributo può assumere, separati da virgole e fra parentesi tonde

Valori degli attributi

Valore	Significato
#REQUIRED	Deve sempre essere specificato un valore per l'attributo
#IMPLIED	Non è obbligatorio specificare un valore per l'attributo, e non viene assegnato un valore di default
"valore"	Non è obbligatorio specificare un valore per l'attributo, ma se non lo si specifica viene assegnato un valore di default
#FIXED "valore"	Non è obbligatorio specificare un valore per l'attributo, ma se lo si specifica può solo essere quello prefissato

Dichiarazione delle entità

DTD

```
<!ENTITY xml  
    "eXtensible Markup Language">  
<!ENTITY cap1  
    SYSTEM "/book/capitolo1.xml">
```

Sostituisce la stringa associata

Inserisce il contenuto
del file specificato

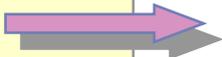
XML

```
...  
... &xml; ...  
... &cap1; ...
```

DTD

Esempio

```
<!-- Dichiarazione della struttura -->
<!ELEMENT agenda (contatto*)
<!ELEMENT contatto (nome, indirizzo+,
    telefono*, e-mail*, foto?)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT indirizzo (via, numero, citta,
    cap, provincia?)
<!ELEMENT via (#PCDATA)>
<!ELEMENT numero (#PCDATA)>
<!ELEMENT citta (#PCDATA)>
<!ELEMENT cap (#PCDATA)>
<!ELEMENT provincia (#PCDATA)>
```



Esempio

DTD



```
<!ELEMENT telefono (#PCDATA)>
<!ATTLIST telefono tipo (casa | ufficio
    mobile) "casa">
<!ELEMENT e-mail (#PCDATA)>
<!ELEMENT foto empty>
<!ATTLIST foto
    src    CDATA    #REQUIRED
    desc   CDATA    #IMPLIED>

<!Dichiarazione di entità -->
<!ENTITY TO "10100">
<!ENTITY BO "40100">
```

Esempio

XML

```
<!-- Intestazione -->
<?XML version="1.0" encoding="UTF-8" ?>

<!-- DTD -->
<!DOCTYPE agenda SYSTEM "agenda.dtd">

<!-- Dati -->
<agenda>
  <contatto>
    <nome>Andrea Bianchi</nome>
    <indirizzo>
      <via>via Rossi</via>
      <numero>10</numero>
```

Esempio

XML



```
<citta>Torino</citta>
  <cap>&T0;</cap>
</indirizzo>
<telefono>011-1234567</telefono>
<telefono tipo="ufficio">011-
  222555</telefono>
<e-mail>andrea.bianchi@mio.org</e-mail>
<foto src="andrea.gif" desc="La mia
  fotografia migliore" />
</contatto>
</agenda>
```



Documenti validi e ben-formati

- Documento XML **ben formato** (“well-formed”)
 - Rispetta le regole di scrittura del markup, e di annidamento dei tag
 - **Non richiede alcun DTD**
- Documento XML **valido**
 - L'ordine dei tag, il relativo annidamento, e la tipologia del testo contenuto negli ELEMENT e negli ATTRIBUTE **rispettano quanto specificato nel DTD**

Esempi di DTD

- DTD specificato internamente
 - es. lettera.xml
- DTD specificato esternamente
 - es. SMIL

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0"  
"http://www.w3.org/2001/SMIL20/SMIL20.dtd">  
<smil xmlns=  
"http://www.w3.org/2001/SMIL20/Language">  
<head>  
    ...  
</head>  
<body>  
    ...  
</body>  
</smil>
```



Esercizio

- Analizzare Invoice.xml ed Invoice.dtd nel progetto XMLExamples
 - Determinare se il file XML è valido e ben formato
 - Aggiungere almeno un elemento, mantenendo il file valido e ben formato
 - Aggiungere almeno un elemento, rendendo il file *non* valido (ma ben formato)
 - Ripetere, *cancellando* un elemento
 - Ripetere, aggiungendo o cancellando un *attributo*



Esercizio

- Due aziende si vogliono scambiare informazioni relative ai prodotti contenuti in magazzino
- Si definisca un DTD atto a rappresentare il contenuto del magazzino
- Si costruisca un esempio di file XML valido e ben formato