

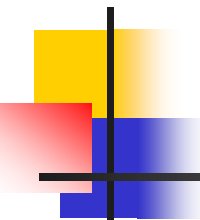
XML - Extensible Markup Language (1)

- XML è un linguaggio di marcatura estensibile (insieme di regole per creare altri linguaggi di marcatura) sviluppato e controllato dal World Wide Web Consortium
- Sottoinsieme di SGML (Standard Generalized Markup Language)
- Non ha tag predefiniti: consente di scrivere altri linguaggi.
- Diverse organizzazioni hanno creato differenti "vocabolari" di marcatura specializzati in determinati settori



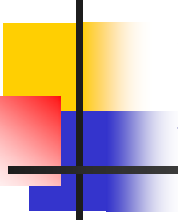
XML (2)

- HTML - è uno standard per la presentazione
- XML - è uno standard per il contenuto e per l'elaborazione dei dati
- La descrizione della struttura di un file XML è fatta tramite DTD (document type definition, definizione del tipo di documento)



XML (3)

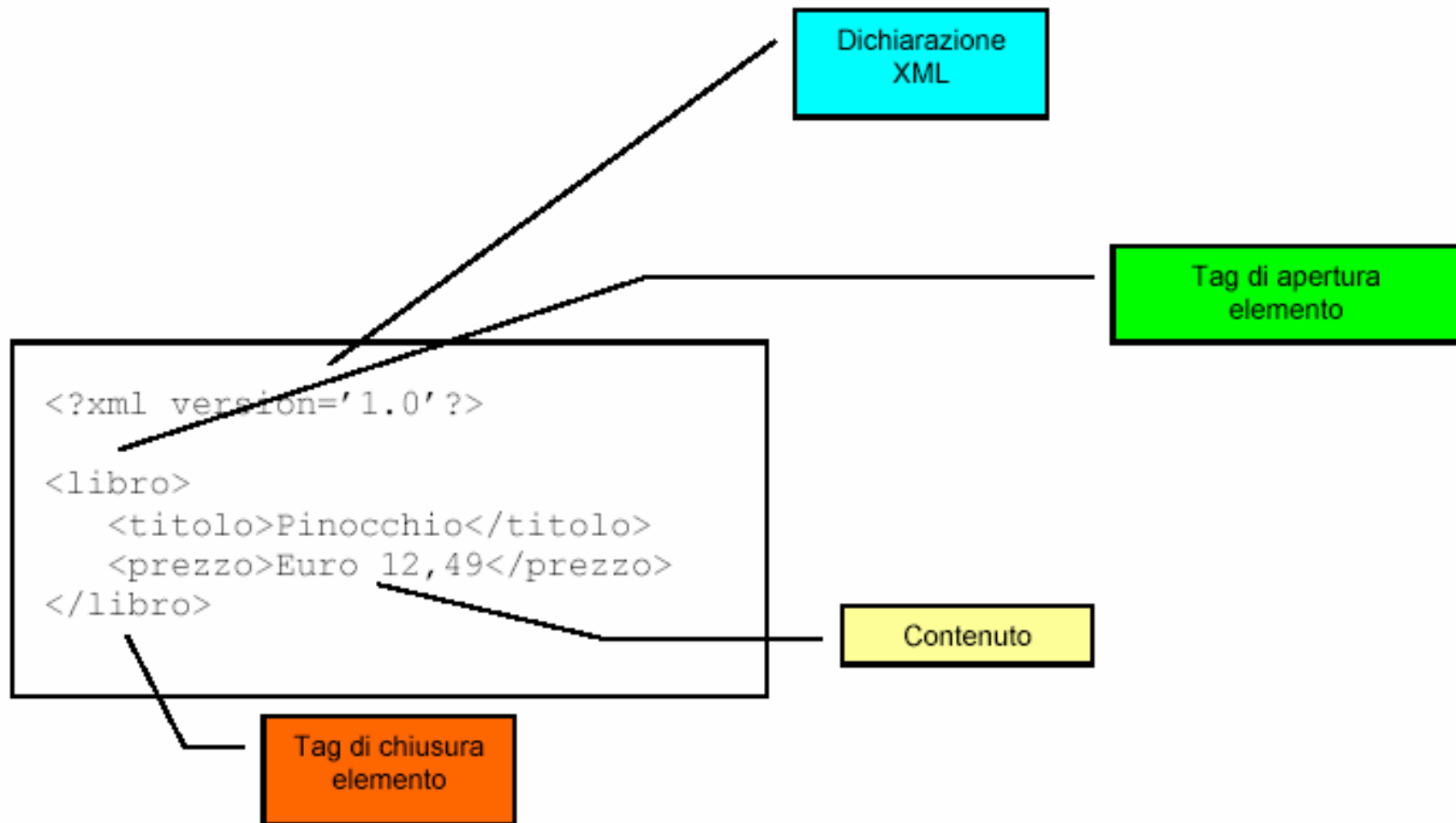
- Obiettivi ufficiali dello standard XML, definiti dal W3C:
 - ⑩ facilità e usabilità su Internet
 - ⑩ supporto di ampie gamme di applicazioni
 - ⑩ Facilità nella scrittura di programmi per l'elaborazione di documenti XML
 - ⑩ leggibilità da parte delle persone e ragionevole chiarezza
 - ⑩ rapidità della fase di disegno
 - ⑩ disegno formale e conciso

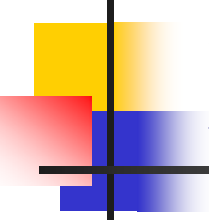


XML – interoperabilità

- Appareti mobili web-enabled
- PC tradizionali
- Web TV
- Lettori vocali
- Stampante/Plotter/Codici a barre

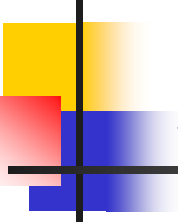
XML - caratteristiche





XML – esempio

```
<?xml version="1.0"?>
  <mezzo_trasporto>
    <categoria tipo="automobile">
      <marca>Fiat</marca>
      <modello>Stilo</modello>
      <motore>benzina</motore>
    </categoria>
  </mezzo_trasporto>
```



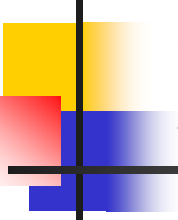
XML – caratteristiche

- Un documento XML può essere:
 - ben formato
 - valido
- Un documento XML è composto da:
 - elementi
 - informazioni di controllo
 - entità



XML - elementi

- *Un solo* elemento radice e un numero variabile di sezioni
- I tag dei vari elementi devono essere nidificati e sono obbligatori i tag di chiusura o la forma abbreviata (
).
- La grammatica di XML è case-sensitive



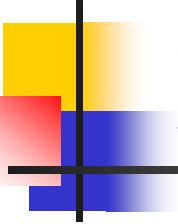
XML – attributi degli elementi

- Contengono valori che descrivono gli elementi ai quali si riferiscono
 - I dati essenziali del problema (ovvero quei dati da sottoporre ad elaborazione diretta) dovrebbero essere dichiarati come elementi
 - Gli attributi sono dati che rivestono significato solo nel contesto o che non variano (significativamente) con il trascorrere del tempo



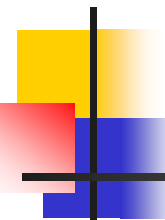
XML – ben formato

- *Un solo* elemento radice e un numero variabile di sezioni
- I tag dei vari elementi devono essere nidificati
- Sono obbligatori i tag di chiusura, se vuoti possono prevedere la forma abbreviata.
- La grammatica di XML è case-sensitive
- I valori degli attributi devono racchiusi tra singoli o doppi apici



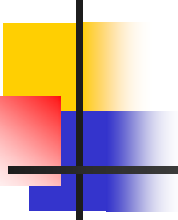
XML – informazioni di controllo (1)

- Le informazioni di controllo sono:
 - commenti
 - istruzioni di elaborazione
 - Document Type Definition
- Commenti (informazioni di controllo)
 - scopo dei commenti è quello di facilitare le operazioni di sviluppo e manutenzione
 - `<!--Questo è un esempio di commento XML -->`



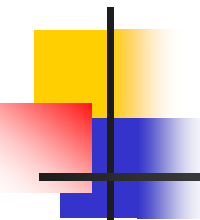
XML – informazioni di controllo (2)

- Istruzioni di elaborazione (informazioni di controllo)
 - lo scopo è quello di “comunicare” con le applicazioni
 - `<?xml version='1.0'?>`
 - `<?xml-stylesheet type="text/css" href="Biblioteca01.css"?>`



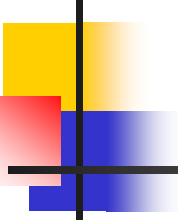
Document Type Definition – DTD (1)

- La DTD contiene la grammatica per una determinata struttura XML
- Parser validatori per determinare la conformità alla grammatica stabilita (xml valido)
- La DTD è opzionale
- La DTD che può essere contenuta nello stesso file XML o può essere specificata in un file esterno



DTD (2)

- Dichiarazioni interne *prevalgono* su quelle esterne.
- Contiene le regole di definizione dei tag (markup)
- Indica gli elementi e il loro ordine nel documento XML



DTD (3)

- Esempio

```
<?xml version='1.0?>
```

```
<!DOCTYPE recipe [
```

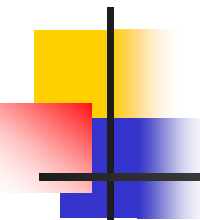
```
  <!ELEMENT cookbook (category+)>
```

```
  <!-- altre istruzioni della DTD -->
```

```
<cookbook>
```

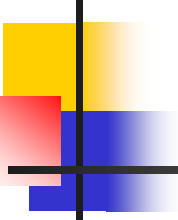
```
  <!-- qui si inserisce il resto del file XML -->
```

```
</cookbook>
```



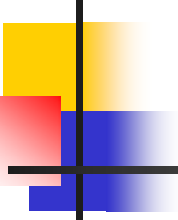
DTD (4)

- Passi per la definizione di una DTD:
- 1) definizione del proprio linguaggio di markup (tag)
- 2) definizione della struttura del documento
- 3) definizione dei vincoli (constraint) e dei valori predefiniti (default) per gli attributi



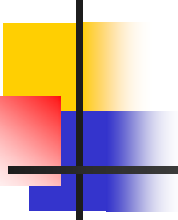
DTD – Elementi (1)

- `<!ELEMENT cookbook (category+)>`:
elemento radice
- Ogni elemento può contenere altri elementi o può definire dei dati
- Ogni sottoelemento può essere accompagnato da un operatore che indica le regole con le quali il sottoelemento deve essere presente.
- `<!ELEMENT biblioteca (libro+, catalogo, chiosco?, puntoristorazione*)>`



DTD – Elementi (2)

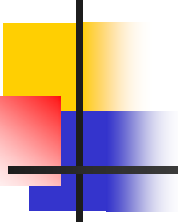
- Operatori
- “*” l’elemento zero o più volte
- “()” racchiude un gruppo di elementi
- “,” l’elemento contiene la sequenza di altri elementi
- “|” indica elementi alternativi, si può presentare uno o l’altro (es.: acceso | spento)
- “+” l’elemento appare almeno una volta
- “?” l’elemento appare una sola volta o nessuna



DTD – Esempio (1)

- `<!ELEMENT articolo(paragrafo+)>`
`<!ELEMENT paragrafo (immagine*, testo+,`
`codice*)>`

`<!ELEMENT immagine EMPTY>`
`<!ELEMENT testo (#PCDATA)>`
`<!ELEMENT codice (#PCDATA)>`

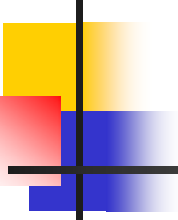


DTD - #PCDATA, CDATA

- #PCDATA (Parsed character data): un set di caratteri analizzato dal parser XML ed è il tipo di dato di default degli elementi.
- CDATA: un set di caratteri che non viene esaminato dal parser XML
- EMPTY: tag vuoto, che non contiene testo

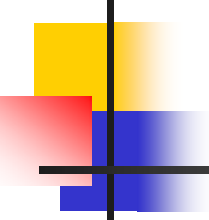
DTD – attributi (1)

- Un elemento XML, oltre a contenere dati o altri elementi, può essere corredato da attributi
- La definizione si esegue con la parola chiave ATTLIST
- <!ATTLIST articolo
titolo CDATA
#REQUIRED>



DTD – attributi (2)

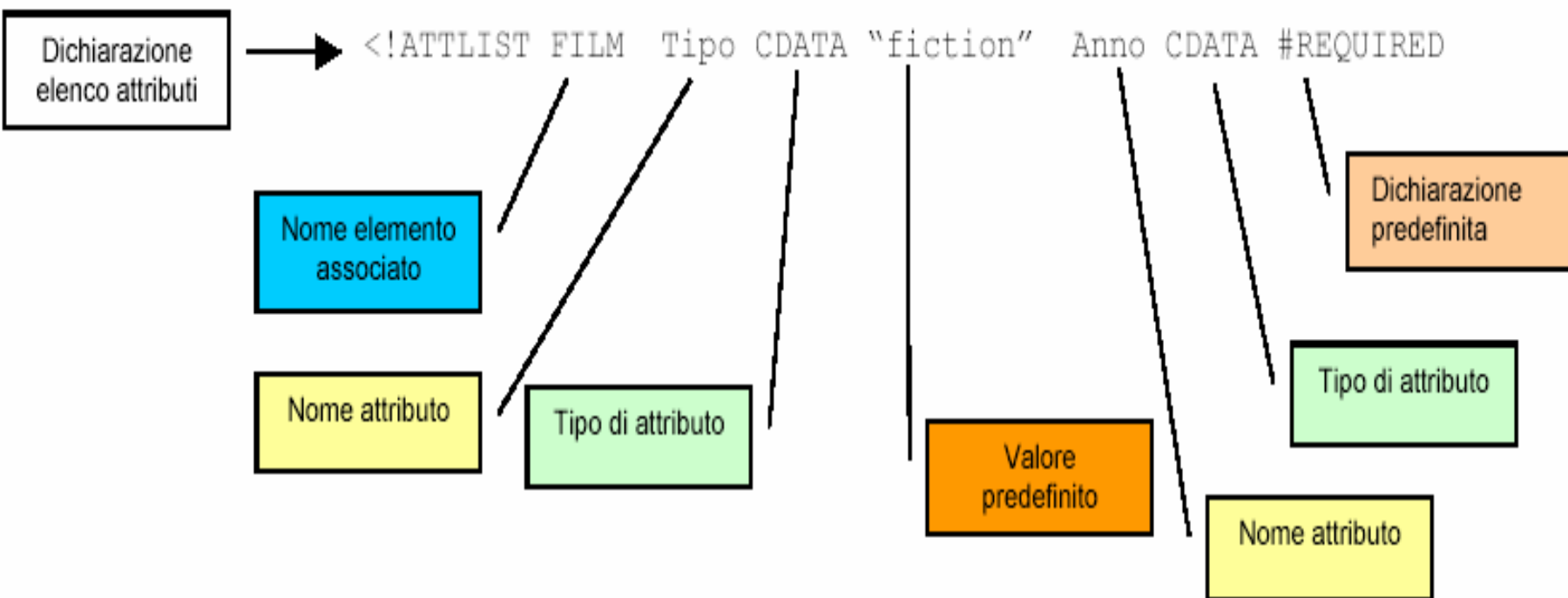
- La definizione di un attributo di un elemento include :
 - • nome attributo dell'elemento
 - • tipo di dato di ogni attributo
 - • eventuale valore di default per ogni attributo
 - • informazioni di controllo

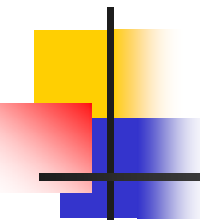


DTD – attributi (3)

- Le informazioni di controllo possono essere:
- REQUIRED
- FIXED valore
- IMPLIED

DTD – attributi (4)



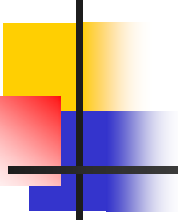


DTD - Esempio

- `<!ELEMENT articolo(paragrafo+)>`
`<!ELEMENT paragrafo (immagine*, testo+, codice*)>`

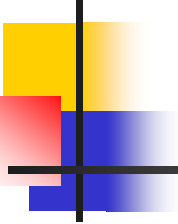
`<!ELEMENT immagine EMPTY>`
`<!ELEMENT testo (#PCDATA)>`
`<!ELEMENT codice (#PCDATA)>`

`<!ATTLIST articolo titolo CDATA #REQUIRED>`
`<!ATTLIST paragrafo`
 titolo CDATA #IMPLIED
 tipo (abstract|bibliografia|note) #IMPLIED
`>`
`<!ATTLIST immagine file CDATA #REQUIRED>`



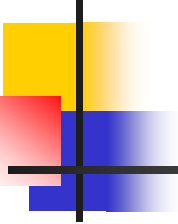
Entità (1)

- Un documento XML può contenere una o più entità
- Le entità sono *oggetti* che sono usati di frequente
- Possono essere di tipo parsed o non parsed
- Con le entità si rappresentano caratteri speciali come &, <, >, ”
- Entità interne ed esterne



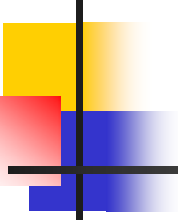
Entità (2)

- `<` Parentesi angolare sinistra (minore)
- `>` Parentesi angolare destra (maggiore)
- `&` e commerciale (`&`)
- `<!ELEMENT TEATRO "Teatro San Carlo">`
 - `<item>` Il tour si conclude con uno spettacolo al `&TEATRO;``</item>`
- `<!ELEMENT logoaziendale SYSTEM "logo.png">`



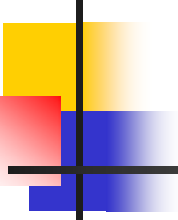
Name Space (spazio dei nomi)

- `<persona>`
 `<nome>Andrea Bertagnoli</nome>`
 `</persona>`
- `<ufficio>`
 `<nome>Informatica</nome>`
 `<ubicazione>Primo piano stanza 9999`
 `</ubicazione>`
 `</ufficio>`



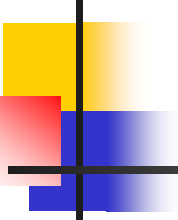
Name Space (2)

- Usando un prefisso si ottiene
- `<a:persona>`
 - `<a:nome>Andrea Bertagnoli</a:nome>`
 - `</a:persona>`
- `<b:ufficio>`
 - `<b:nome>Informatica</b:nome>`
 - `<b:ubicazione>Primo piano stanza 9999`
 - `</b:ubicazione>`
 - `</b:ufficio>`



Name Space (3)

- Usando il name space si ottiene
- `<!xml:namespace
ns=http://URL/namespaces/funzionari
prefix="funz">`
- `<!xml:namespace
ns=http://URL/namespaces/struttura
prefix="org">`



Name Space (4)

- `<funz:persona>`
 - `< funz :nome>Andrea Bertagnoli`
 - `</ funz :nome>`
 - `</ funz :persona>`
- `<org:ufficio>`
 - `<org:nome>Informatica</org:nome>`
 - `<org:ubicazione>Primo piano stanza`
 - `9999 </org:ubicazione>`
 - `</org :ufficio>`