

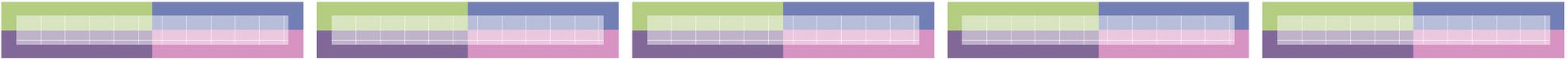
# XML: eXtensible Markup Language

Laura Farinetti

Dip. Automatica e Informatica

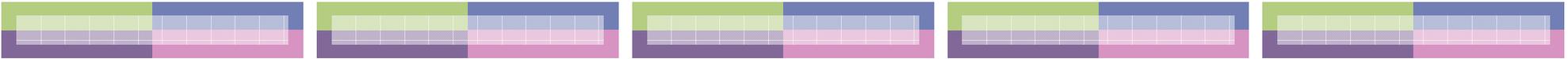
Politecnico di Torino

`laura.farinetti@polito.it`



# Introduzione

- XML: eXtensible Markup Language
- Linguaggio estensibile realizzato per elaborare i documenti strutturati
- Pensato per il Web e per superare i limiti dell'HTML
- Sottinsieme di SGML: Standard Generalized Markup Language
- Sviluppato dal [W3C](#), World Wide Web Consortium



# Le specifiche

- La prima bozza di XML risale al 1996

- Attuale specifica: 1.1 (4 febbraio 2004)

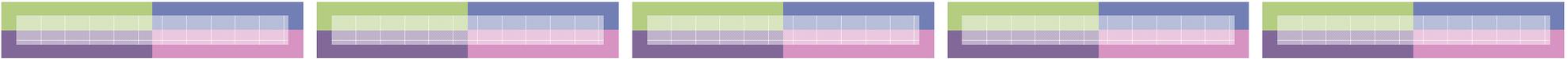
`http://www.w3.org/TR/REC-xml`

- Traduzione in **italiano** (1.0):

`http://www.xml.it:23456/XML/REC-xml-19980210-it.html`

- World Wide Web Consortium:

`http://www.w3.org/`



## II W3C

- Il W3C non produce standard, ma **rapporti tecnici** o specifiche ("raccomandazioni")
- Ciclo di vita di un documento
  - **Nota:** documento, idea o commento sottomesso al W3C secondo alcune regole, per il quale non esiste un impegno da parte del W3C a lavorare sull'argomento



## II W3C

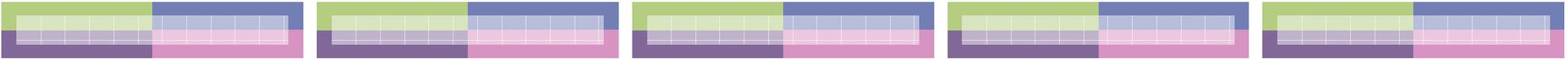
### ● Ciclo di vita di un documento

- **Working draft:** relativo a specifiche in corso di sviluppo da parte del W3C; i documenti possono essere modificati o sostituiti da altri documenti in qualunque momento
- **Candidate recommendation:** specifica rivista ed approvata dal gruppo tecnico coinvolto, e inviata ad altri gruppi per un'ulteriore revisione



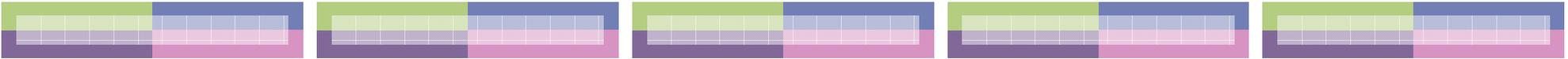
## II W3C

- **Ciclo di vita di un documento**
  - **Proposed recommendation:** rappresenta il consenso nel gruppo tecnico che l'ha creata e tiene conto della revisione degli altri gruppi; proposta al Director of the Advisory Committee per un'ultima revisione
  - **Recommendation:** approvata dal Director; è il grado più elevato a cui può giungere una specifica W3C



# XML, HTML e SGML

- HTML è un linguaggio predefinito
- XML è un metalinguaggio
  - non ha tag predefiniti
  - consente di definire nuovi metalinguaggi
- Anche SGML è un metalinguaggio
  - molto complesso
- XML è un sottoinsieme di SGML
  - un documento XML valido è conforme allo standard SGML



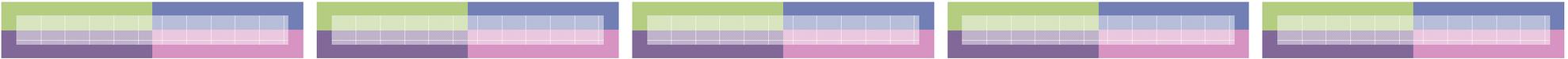
# Perché utilizzare XML?

- Chi scrive in XML può focalizzarsi solo sul contenuto del documento
- Si possono scrivere linguaggi "ad hoc" per specifiche comunità di utenti (matematici, chimici, musicisti, ...)
- È facile passare dall'SGML all'XML, e quindi al Web
- XML facilita lo scambio di dati fra applicazioni di tipo diverso



# Linguaggi basati su XML

- SMIL (Synchronized Multimedia Integration Language)
  - integrazione e sincronizzazione di flussi multimediali
- SVG (Scalable Vector Graphics)
  - grafica vettoriale
- XHTML
  - sostituirà HTML
- MathML (Math Markup Language)
  - formule e funzioni matematiche
- VoiceXML
  - applicazioni che integrano sintesi/riconoscimento vocale
- ... e moltissimi altri
- Definiti in modo formale con i rispettivi XML DTD/Schema:  
specifica del linguaggio



# Esempio

```
<?xml version="1.0"?>
<!-- esempio di file XML -->
<article num="3">
  <title>Esempio di XML</title>
  <date>Settembre 2005</date>
  <author>
    <firstName>Mario</firstName>
    <lastName>Rossi</lastName>
    <initials value="MR"/>
  </author>
  <summary>XML e' facile</summary>
  <content>Il funzionamento del markup di XML non e'
molto diverso da HTML, ma si presta a gestire
informazione piuttosto che visualizzarla</content>
</article>
```

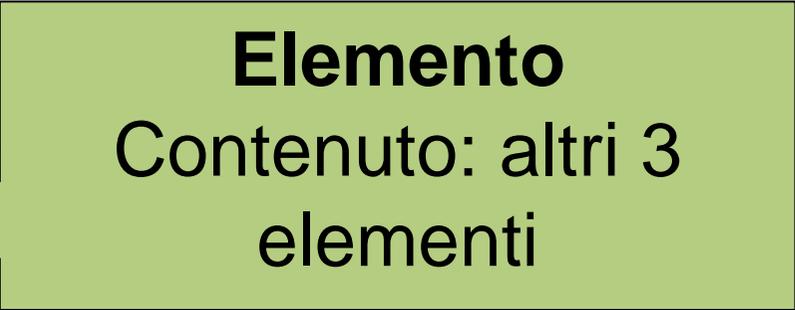
# Esempio

```
<?xml version="1.0"?>
<!-- esempio di file XML -->
<article num="3">
  <title>Esempio di XML</title>
  <date>Settembre 2005</date>
  <author>
    <firstName>Mario</firstName>
    <lastName>Rossi</lastName>
    <initials value="MR" />
  </author>
  <summary>XML e' facile</summary>
  <content>Il funzionamento del markup di XML non e'
molto diverso da HTML, ma si presta a gestire
informazione piuttosto che visualizzarla</content>
</article>
```

**Elemento**  
Apertura <date>  
Chiusura </date>  
Contenuto: testo

# Esempio

```
<?xml version="1.0"?>
<!-- esempio di file XML -->
<article num="3">
  <title>Esempio di XML</title>
  <date>Settembre 2005</date>
  <author>
    <firstName>Mario</firstName>
    <lastName>Rossi</lastName>
    <initials value="MR" />
  </author>
  <summary>XML e' facile</summary>
  <content>Il funzionamento del markup di XML non e'
molto diverso da HTML, ma si presta a gestire
informazione piuttosto che visualizzarla</content>
</article>
```



**Elemento**  
Contenuto: altri 3  
elementi

# Esempio

```
<?xml version="1.0"?>
<!-- esempio di file XML -->
<article num="3">
  <title>Esempio di XML</title>
  <date>Settembre 2005</date>
  <author>
    <firstName>Mario</firstName>
    <lastName>Rossi</lastName>
    <initials value="MR" />
  </author>
  <summary>XML e' facile</summary>
  <content>Il funzionamento del markup di XML non e'
molto diverso da HTML, ma si presta a gestire
informazione piuttosto che visualizzarla</content>
</article>
```

## Elemento vuoto

Auto-chiusura <x/>

Equivalente a <x></x>

# Esempio

```
<?xml version="1.0"?>
<!-- esempio di file XML -->
<article num="3">
  <title>Esempio di XML </title>
  <date>Settemb</date>
  <author>
    <firstName>F</firstName>
    <lastName>R</lastName>
    <initials v</initials v
  </author>
  <summary>XML e' facile</summary>
  <content>Il funzionamento del markup di XML non e'
molto diverso da HTML, ma si presta a gestire
informazione piuttosto che visualizzarla</content>
</article>
```

## Attributo

Nome = "valore"  
(virgolette obbligatorie)

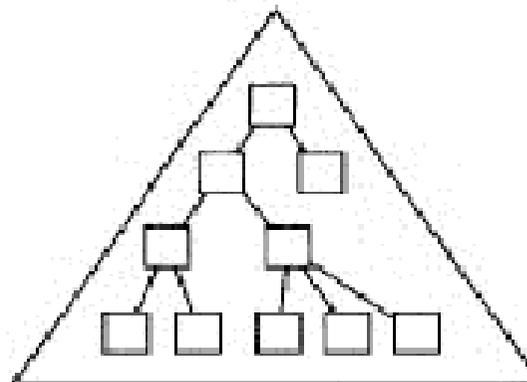


# Domande ...

- Un browser recente **supporta XML** ma...
  - che senso ha per lui il tag `<firstName>`?
  - in che modo visualizza il testo fra `<firstName>` e `</firstName>`?
- Un documento XML contiene solo i dati, non specifica né la loro **struttura**
  - es: `<author>` ha due componenti, `<firstName>` e `<lastName>`
- né come questi debbano essere **visualizzati**
  - es: Rossi deve essere in grassetto

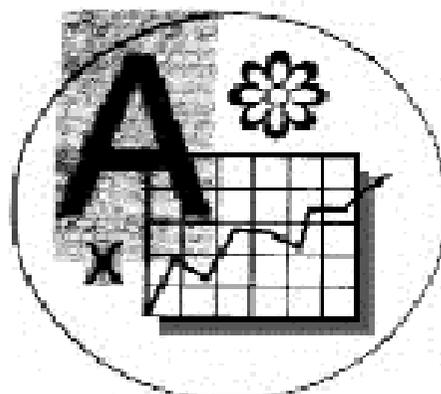
# Le componenti di XML

**"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**

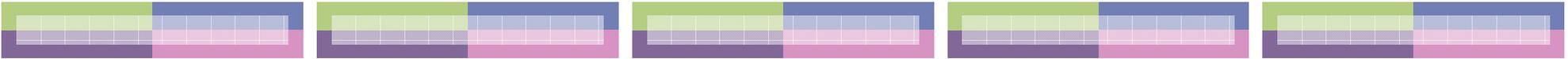


**Contenuto**

**Struttura**



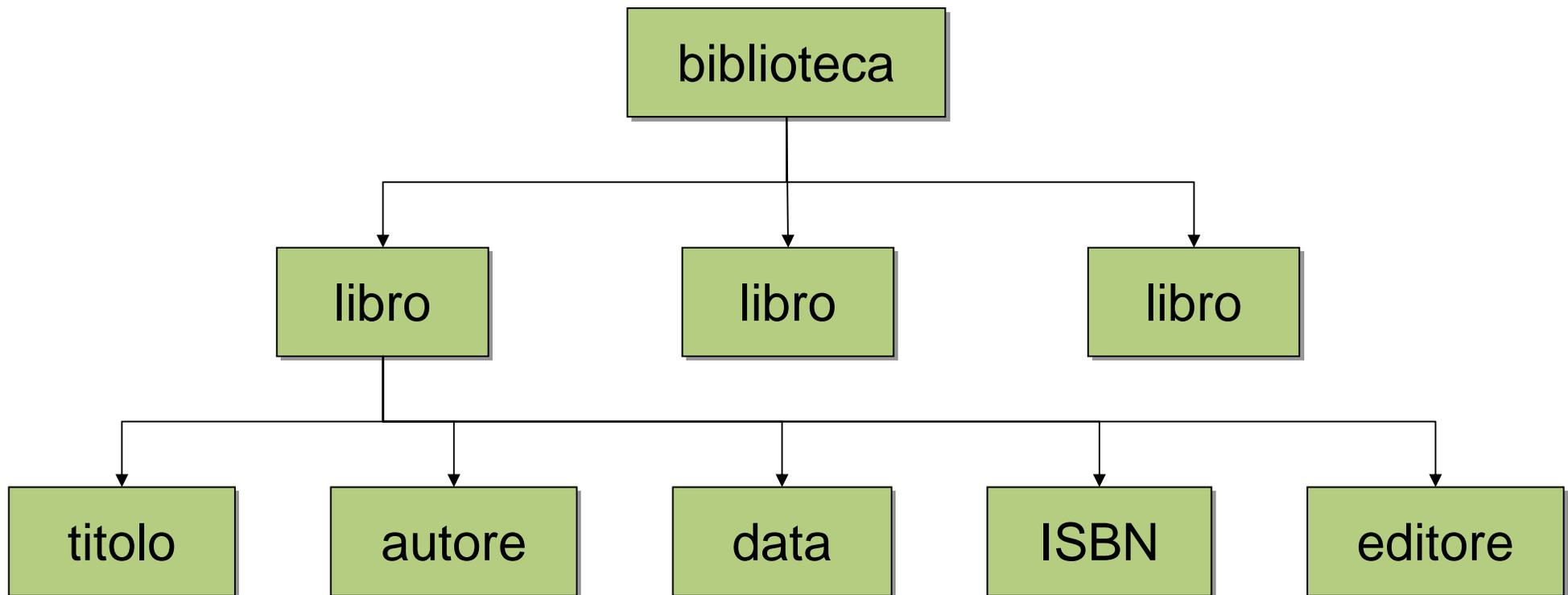
**Stile**



# Esempio di contenuto

```
<?xml version="1.0"?>
<biblioteca>
  <libro>
    <titolo>2001: Odissea nello spazio</titolo>
    <autore>Arthur Charles Clarke</autore>
    <data>1969</data>
    <ISBN>88-304-1885-4</ISBN>
    <editore>Longanesi</editore>
  </libro>
  <libro>
    ...
  </libro>
</biblioteca>
```

# Esempio di struttura

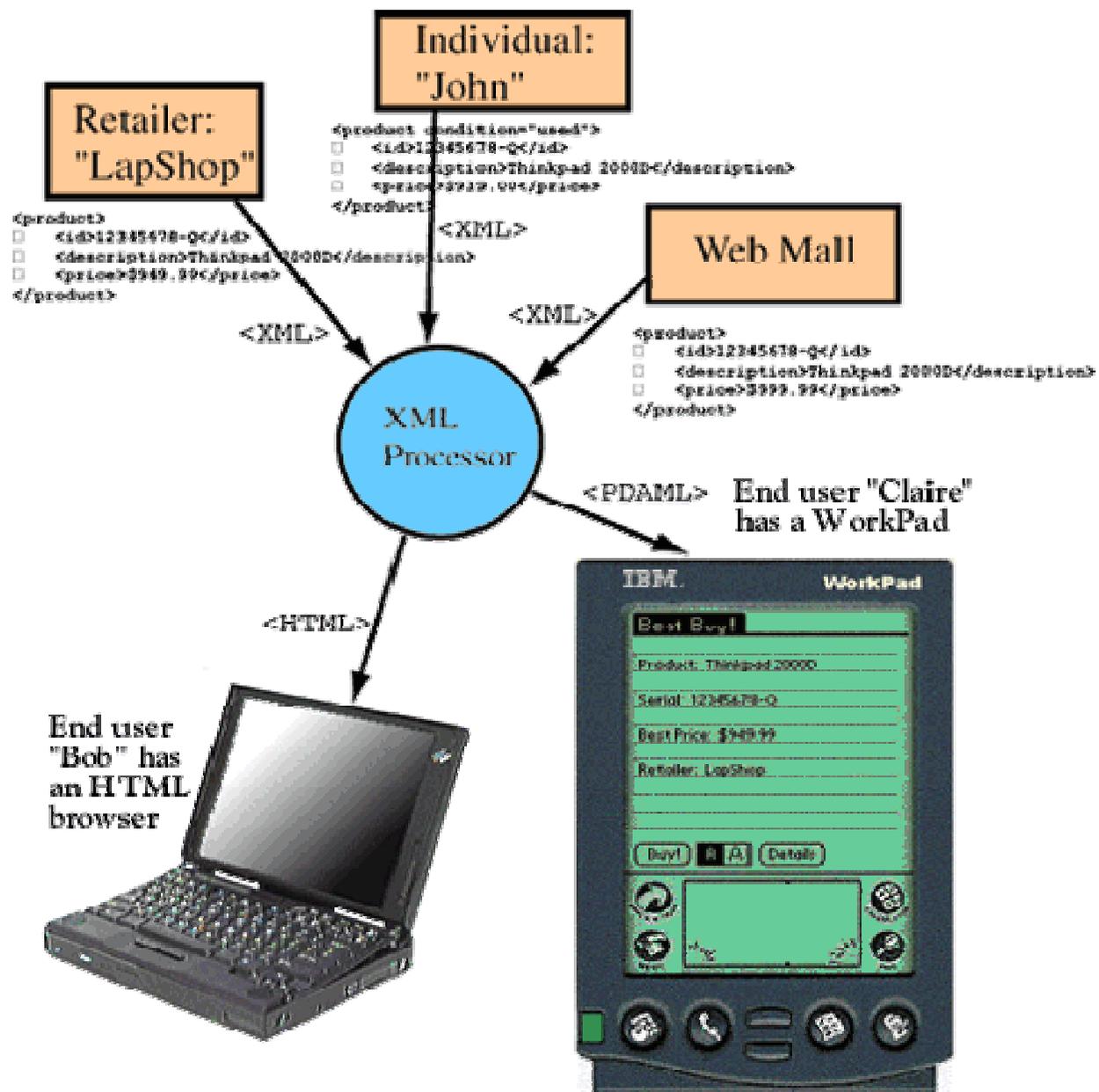




# Esempio di stile

## La mia biblioteca

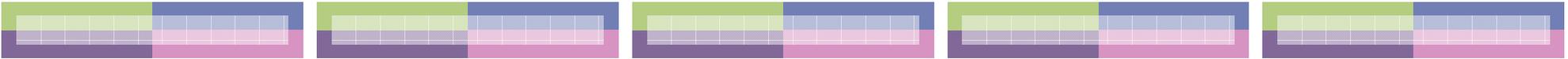
<b>Titolo</b>	<b>Autore</b>	<b>Editore</b>	<b>ISBN</b>
2001: Odissea nello spazio	Arthur Charles Clarke	Longanesi	88-304-1885-4
XSLT: a programmer's reference	Michael Kay	Wrox Press Ltd.	1-861003-12-9
Structuring XML documents	David Megginson	Prentice Hall	0-13-642299-3
Designing XML Internet applications	Michael Levental & altri	Prentice Hall	0-13-616822
XML by example: building e-commerce applications	Sean McGrath	Prentice Hall	0-13-960162-7





# SMIL: esempio

```
<smil>
<head>
  <layout>
    <region id="topleft" width="100" height="100" fit="meet"/>
    <region id="topright" left="120" width="100" height="100"
      fit="meet"/>
    <region id="bottom" top="120" width="220" height="20" fit="meet"/>
  </layout>
</head>
<body>
  <par>
    
    
    
  </par>
</body>
</smil>
```



# Le componenti di XML

- Tre parti fondamentali di un documento che dovrebbero essere gestite separatamente
  - Il **contenuto** (XML)
  - Le specifiche relative agli elementi:  
la **struttura** (DTD: Document Type Definition o XSD: XML Schema)
  - Le specifiche relative alla visualizzazione:  
lo **stile** (XSL)

# Il contenuto: la sintassi XML



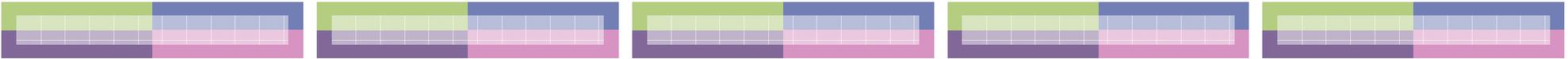
# Sintassi XML

- Simile a (X)HTML, a parte
  - la possibilità di definire a piacere tag e attributi
  - alcune regole sintattiche
- Un documento che segue le regole sintattiche si dice “ben-formato”



# Intestazione

- Tutti i file XML devono iniziare con una *XML Declaration*
  - `<?xml version="1.0"?>`



# Commenti

- Il formato dei commenti XML è il seguente:
  - `<!-- ... commento ... -->`



# Documento ben formato

- Se rispetta tre regole
- Bilanciamento dei marcatori (tag)
  - ad ogni tag di apertura deve corrisponderne uno di chiusura

`<tag>.....</tag>`

`<tagvuoto/>`

- Valori degli attributi fra virgolette o apici

``

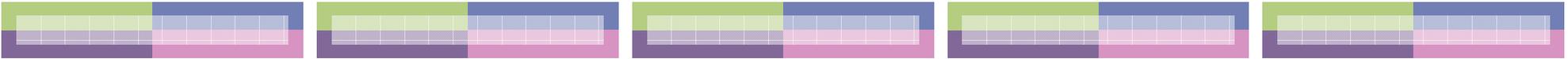
`<img src='immagine.jpg' width='50' border='0' />`



# Documento ben formato

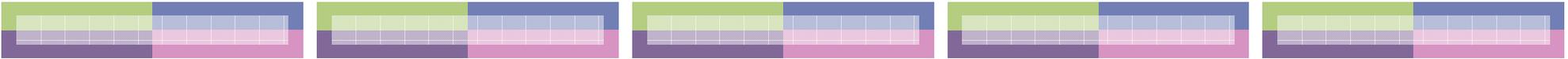
- Uso di **entità dichiarate** (es. costanti)
  - l'uso di entità è consentito solo se la dichiarazione è presente all'inizio del documento XML o nel DTD associato
  - eccezione:  
**entità predefinite**

<code>&amp;amp;</code>	<code>&amp;</code>
<code>&amp;lt;</code>	<code>&lt;</code>
<code>&amp;gt;</code>	<code>&gt;</code>
<code>&amp;apos;</code>	<code>'</code>
<code>&amp;quot;</code>	<code>"</code>



# Esempio: XML ben formato

```
<?xml version="1.0"?>
<!DOCTYPE ricetta [
  <!ENTITY g "grammi">
]>
<ricettario>
  <ricetta numero="1">
    <titolo>...</titolo>
    <ingred>200 &g; di ...</ingred>
    <ingred>...</ingred>
    <passo numero="1">...</passo>
    <passo numero="2">...</passo>
  </ricetta>
</ricettario>
```



# Esempio: XML NON ben formato

```
<?xml version="1.0"?>
<!DOCTYPE ricetta>

<ricettario>
  <ricetta numero="1">
    <titolo>...
    <ingred>200 &g; di ...</ingred>
    <ingred>...</ingred>
    <passo numero=1>...</passo>
    <passo numero=2>...</passo>
  </ricetta>
</ricettario>
```



# Esercizio: definizione formale di well-formedness

- Leggere

- <http://www.w3.org/TR/xml/#sec-well-formed>

- Verificare se coincide con le definizioni precedenti

- Illustrare la definizione dei termini: root, child, parent



# References

- Extensible Markup Language (XML) 1.0 (Fourth Edition) - W3C Recommendation
  - <http://www.w3.org/TR/xml/>
- Namespaces in XML 1.0 (Second Edition) - W3C Recommendation
  - <http://www.w3.org/TR/REC-xml-names/>