

Codifica di Testi - Introduzione XML Markup a.a. 2024-2025

Angelo Mario Del Grosso

`angelo.delgrosso@ilc.cnr.it`

CNR-ILC

Istituto di Linguistica Computazionale “A. Zampolli”,
11th March 2025

Contenuto della lezione

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

1 I linguaggi di codifica

2 Fondamenti del linguaggio XML

3 Validare un documento XML e Definire uno Schema

4 Conclusioni

Progress status

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

1 I linguaggi di codifica

2 Fondamenti del linguaggio XML

3 Validare un documento XML e Definire uno Schema

4 Conclusioni

I linguaggi di codifica

introduzione

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Definizione di codifica digitale del testo

Per **codifica** digitale dei testi intendiamo la *rappresentazione formale* di un **testo** ad un qualche livello descrittivo, su di un supporto digitale, in un formato utilizzabile da un elaboratore (*Machine Readable Form*) mediante un opportuno **linguaggio informatico** (F. Ciotti).

I linguaggi di codifica

Riassumendo

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Impostazione teorico-pratica

- un testo è molto di **più della sequenza di caratteri** che lo compongono
- per mezzo della codifica vogliamo **rendere esplicite le caratteristiche** che vogliamo analizzare
- solo quello che è esplicito può essere **interpretato ed elaborato dal computer**
- vogliamo codificare il **testo per quello che è**, non per quello che sembra
- codifica da effettuare mediante **linguaggio di markup**

I linguaggi di codifica

Linguaggi di marcatura

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Il markup

Il termine **markup** è stato utilizzato in passato per denotare i **segni grafici** che accompagnavano un testo apposti sul documento per **indicare correzioni o modalità grafiche di stampa**.

I linguaggi di codifica

Linguaggi di marcatura

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

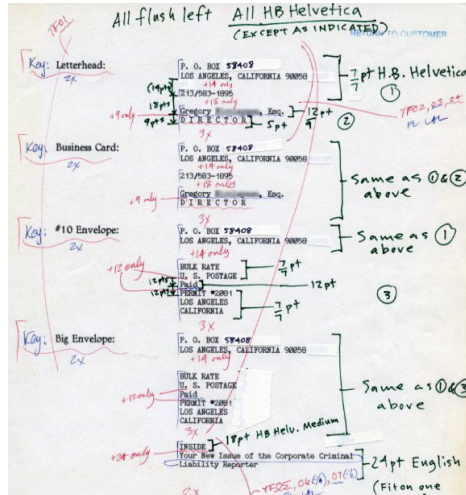
A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni



I linguaggi di codifica

Linguaggi di marcatura

Codifica di
Testi -
Introduzione
XML Markup
a.a.




















































2024-2025
A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

| | | | | | | | | |
|---|---|----------------------------------|---|---|----------------------------|--|--|-------------------------|
|  |  | Delete |  | FL | Flush Left |  | bf.ital | Set in Bold Face Italic |
| |  | Insert | | FR | Flush Right | | lf | Set in Light Face |
| |  | Join | |  | Center Horizontally | | wf | Wrong Font |
| |  | Move closer | |  | Center Vertically |  |  | Hyphen |
| |  | Space | |  | Move to the next line |  |  | En Dash |
| |  | Add Space | |  | Move to the preceding line |  |  | Em Dash |
| |  | Delete Space | |  | Indent 1 em | |  | Superscript |
| |  | Transpose Word | |  | Indent 2 ems | |  | Subscript |
| |  | Transpose Letters | |  | Paragraph | |  | Comma |
| |  | To separate two or more marks | |  | All Caps | |  | Apostrophe |
| |  | Let it Stand (ignore correction) | |  | Small Caps | |  | Period |
| |  | Move Left | |  | Caps & Small Caps | |  | Semicolon |
| |  | Move Right | |  | Capital Letter | |  | Colon |
| |  | Move Up | |  | Lower Case | |  | Quotation Marks |
| |  | Move Down | |  | Set in Roman | |  | Parentheses |
| |  | Align Vertically | |  | Set in Italic | |  | Brackets |
| |  | Align Horizontally | |  | Set in Bold Face | | | |

I linguaggi di codifica

Linguaggi di marcatura

Il markup

La codifica con linguaggi di marcatura (markup) è in sostanza **un insieme di convenzioni**, rese attraverso specifiche **sequenze di caratteri, etichette, codici**, (detti *tags*) **intercalati nel testo** per permettere agli elaboratori elettronici di distinguere le varie parti di un documento.

Il markup formale

Un linguaggio di markup è un **sistema formale** per *scambiare* e *pubblicare* informazioni in **formato testo in modo strutturato**.

I linguaggi di codifica

Linguaggi di marcatura

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Il markup formale

Markup formale: costituito da un **sistema non ambiguo** di istruzioni, ognuna delle quali è **dotata di una specifica semantica e sintassi**.

I linguaggi di codifica

Linguaggi di marcatura

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Diversi tipi di markup

Esistono diversi linguaggi di markup, per rappresentare diversi tipi di documenti.

- **Linguaggi procedurali** (specific markup languages)
- **Linguaggi dichiarativi** (generic markup languages)

I linguaggi di codifica

Linguaggi di marcatura procedurale

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Linguaggi procedurali

- **Orientati al documento**, indicando come deve essere elaborato e *disposto il testo* in **resa grafica**
- Istruzioni da inserire nel testo per connotarne specifiche *caratteristiche di visualizzazione*
- Font, dimensione, spaziatura del carattere, posizionamento nella pagina, colore, etc.

Esempi: TeX e LaTeX, RTF

I linguaggi di codifica

Linguaggi di marcatura procedurale

Esempio RTF

```
{\rtf1\ansi\deff0\adefflang1025
{\fonttbl{\f0\froman\fprq2\fcharset0 Times New Roman;}
{\f1\froman\fprq2\fcharset0 Times New Roman;}
{\f2\fnil\fprq2\fcharset0 Lucida Sans Unicode;}
{\colortbl;\red0\green0\blue0;\red128\green128\blue128;}
{\stylesheet{\s1\cf0{\*\hyphen2\hyphlead2\hyphtrail2\hyphmax0}
\rtlch\af5\afs24\lang255\ltrch\dbch\af2\afs24\langfe255
\loch\f0\fs24\lang1040\snext1 Standard;}}
```

I linguaggi di codifica

Linguaggi di marcatura procedurale

Esempio LaTeX

```
\documentclass [a4paper,10pt] {article} \usepackage[utf8] {inputenc}
\usepackage[T1] {fontenc}
\usepackage[italian] {babel}
\title {Il mio primo documento}
\author {Angelo Mario Del Grosso}
\begin {document}
\maketitle
\begin {abstract}
Primo tentativo di scrivere in \LaTeX .
\end {abstract}
\section {titolo della sezione}
Questo documento è vuoto.
\footnote {nota a piè di pagina.}

\end {document}
```

I linguaggi di codifica

Linguaggi di marcatura

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Il markup procedurale

L'unico utilizzo di un testo codificato tramite un linguaggio procedurale è la **creazione di un output orientato alla visualizzazione.**

I linguaggi di codifica

Linguaggi di marcatura procedurale

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Scrivere la tesi di laurea con $\text{\LaTeX} 2_{\epsilon}$

Dipartimento di Ingegneria Meccanica,

Nucleare e della Produzione

Università di Pisa

56126 Pisa PI

Sommario

Lo scopo del presente articolo è fornire gli strumenti per scrivere una tesi di laurea utilizzando $\text{\LaTeX} 2_{\epsilon}$. Tale obiettivo è conseguito analizzando i problemi tipici incontrati durante la stesura della tesi e le possibili soluzioni; si pone particolare attenzione ai pacchetti da usare nelle varie circostanze. I singoli argomenti non vengono approfonditi nei dettagli ma si rimanda alla letteratura specifica o ad i manuali dei pacchetti suggeriti, ove necessario.

*Ringrazio in primo luogo Fabiano Busdraghi che ha collaborato alla scrittura delle sezioni riguardanti le figure e gli oggetti flottanti. Ringrazio inoltre tutti coloro che mi hanno consigliato durante la stesura e la revisione di questo documento ed in particolare Claudio Beccari, Gustavo Cevolani, Massimo Guiggiani, Maurizio Himmelmann, Lorenzo Pantieri e Emiliano Vavassori.

I linguaggi di codifica

Linguaggi di marcatura dichiarativi

Linguaggi dichiarativi

Orientati al testo, annotano la *struttura*, la *funzione* ed il *significato* degli elementi costitutivi del testo, **tralasciandone l'aspetto**.

- La posizione che il brano in questione occupa all'interno del documento (**markup strutturale**)
- Peculiarità del testo stesso (**markup semantico**)
- I fogli di stile definiscono la formattazione dell'output
- *Molteplici usi del medesimo testo*

Esempio: famiglia SGML, XML

I linguaggi di codifica

Linguaggi di marcatura dichiarativi

Markup dichiarativi: contenuto e presentazione

La **separazione tra contenuto e presentazione** non solo è intenzionale, ma è la **caratteristica principale** di questi sistemi di marcatura: essa permette di concentrarsi sull'**annotazione logica-semantica** per funzioni di *ricerca e di analisi*, lasciando ad altro (ai fogli di stile) la resa grafica.

Unico testo più usi

In questo modo si ha inoltre la possibilità di utilizzare uno **stesso testo codificato con finalità o formattazioni differenti**, a seconda delle varie esigenze.

I linguaggi di codifica

Markup dichiarativi: esempio SGML

Standard Generalized Markup Language

```
1 <!DOCTYPE testo [  
2 <!ELEMENT testo (titolo?, paragrafo+)>  
3 <!ELEMENT titolo (#PCDATA)>  
4 <!ELEMENT paragrafo (#PCDATA)>  
5 ]>  
6 <testo>  
7   <titolo> Questo è il titolo del documento</titolo>  
8   <paragrafo> Questo è un paragrafo </paragrafo>  
9 </testo>  
10
```

I linguaggi di codifica

Markup dichiarativi vs Markup procedurali

resa a video della frase

Le *Guidelines for Electronic Text Encoding and Interchange* sono *molto* complete e descrivono uno standard di *markup* del testo basato su XML.

Le `\textit{Guidelines for Electronic Text Encoding and Interchange}` sono `\textit{molto}` complete e descrivono uno standard di `\textit{markup}` del testo basato su XML.

`<titolo>`Le Guidelines for Electronic Text Encoding and Interchange`</titolo>` sono `<enfasi>`molto`</enfasi>` complete e descrivono uno standard di `<linguastraniera>` markup`</linguastraniera>` del testo basato su XML.

LaTeX vs SGML

I linguaggi di codifica

Linguaggi di marcatura

linguaggi semi-dichiarativi e/o semi-procedurali

Esistono anche linguaggi che possono essere definiti **semi-procedurali**, o **semi-dichiarativi**, che come si intuisce utilizzano le istruzioni sia per una codifica di tipo procedurale, sia per una codifica di tipo descrittivo o dichiarativo.

HTML

HTML ha tra le sue etichette istruzioni di tipo procedurale per indicare come devono essere rese determinate porzioni di testo, e istruzioni di tipo dichiarativo che hanno una base semantica.

Progress status

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

1 I linguaggi di codifica

2 Fondamenti del linguaggio XML

3 Validare un documento XML e Definire uno Schema

4 Conclusioni

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML origini

XML affonda le proprie origini nel linguaggio **Standard Generalized Markup Language (SGML)**.

SGML è stato introdotto negli anni ottanta con il fine di **descrivere la struttura e il contenuto di qualsiasi informazione** “machine readable”.

XML è una semplificazione di SGML

XML può essere pensato come una **versione semplificata di SGML**. Infatti, come SGML, *XML è un meta-linguaggio*, usato per *create linguaggi di marcatura* (detti **vocabolari**).

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML come meta-linguaggio

XML, *eXtensible Markup Language*, è un insieme di regole per definire linguaggi di marcatura personalizzati e personalizzabili (*custom-built vocabularies*).

Applicazioni XML

Allo stesso modo di SGML, XML è nato per **strutturare**, **conservare** e **trasportare** informazioni.

I linguaggi di marcatura derivati da XML per strutturare e descrivere specifiche informazioni vengono chiamati *XML applications* oltre che *vocabolario XML*.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML: eXtensible

XML è **estensibile**: è pensato per essere *modificato* ed *esteso* al fine di soddisfare le varie necessità di rappresentazione dell'informazione. **XML non contempla un vocabolario predefinito!**

XML: standard W3C

XML è sviluppato e **manutenuto dal W3C** (World Wide Web Consortium), il quale sviluppa *protocolli* e *standard* riconosciuti dalla comunità scientifica e tecnica al fine di **condividere informazioni sul Web**.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML: riassumendo

XML, eXtensible Markup Language, deriva da SGML ed è una **specificazione**, un **formalismo**, per *strutturare, conservare e scambiare* informazioni in formato machine readable (*digitale*).

XML: riassumendo

XML è anche una specificazione per **descrivere la struttura dell'informazione** seguendo un **modello dei dati gerarchico**. XML è simile ad HTML, ma a differenza di questo non ha etichette predefinite.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

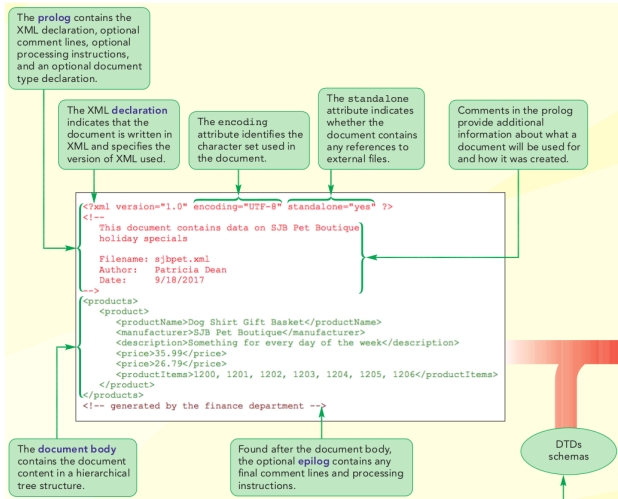


immagine dal libro *New Perspectives on XML, 3rd Edition*

Fondamenti XML

eXtensible Markup Language: regole sintattiche

- Ciascun elemento XML deve avere un tag di chiusura.
- I tag XML sono *case sensitive*.
- Gli elementi XML devono essere annidati in modo rigoroso.
- Tutti i documenti XML devono avere un elemento radice (root) che contiene tutti gli altri elementi opportunamente annidati.
- Gli elementi XML possono avere attributi con stile nome-valore.

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Fondamenti XML

eXtensible Markup Language: regole sintattiche cont.

- Un attributo all'interno dell'elemento può apparire una sola volta
- Il valore degli attributi è una stringa e deve essere inserita tra apici
- Esistono alcuni caratteri speciali che non possono essere usati.
- I commenti non possono essere inseriti prima della dichiarazione XML e non possono essere annidati.

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Manutenibilità

Data la semplicità delle regole e della sintassi XML incentrata sulla memorizzazione e scambio dei dati, la struttura generale di un documento XML è semplice sia dal punto di vista della progettazione sia dal punto di vista della manutenibilità.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML vista ad albero

XML ha un **modello dei dati gerarchico** e può quindi essere visto come un **albero etichettato ordinato**.

Per questo motivo le informazioni sono rappresentate in modo ottimale se sono gerarchiche e sequenziali.

Fondamenti XML

eXtensible Markup Language: vista ad albero

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!--
  This document contains data on SJB Pet Boutique
  holiday specials

  Filename: sjbpets.xml
  Author:   Patricia Dean
  Date:    9/18/2017
-->
<products>
  <product>
    <productName>Dog Shirt Gift Basket</productName>
    <manufacturer>SJB Pet Boutique</manufacturer>
    <description>Something for every day of the week</description>
    <price>35.99</price>
    <price>26.79</price>
    <productItems>1200, 1201, 1202, 1203, 1204, 1205, 1206</productItems>
  </product>
</products>
<!-- generated by the finance department -->
```

Hierarchy tree structure

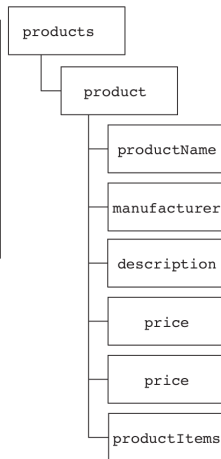


immagine dal libro New Perspectives on XML, 3rd Edition

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

TEI-XML vocabulary

Al fine di soddisfare i **requisiti degli studiosi del testo** il *vocabolario TEI-XML* è stato sviluppato nel corso degli ultimi decenni con l'obiettivo di *permettere la codifica di qualsiasi informazione testuale*.

Un vocabolario XML è un insieme di tag XML sviluppato per una particolare esigenza di codifica

Fondamenti XML

eXtensible Markup Language: Esempio TEI

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

```
<div type="narrative" n="6">
  <head>Sixth Narrative</head>
  <head>contributed by Sergeant Cuff</head>
  <div type="fragment" n="6.1">
    <opener>
      <dateline>
        <name type="place">Dorking, Surrey,</name>
        <date>July 30th, 1849</date>
      </dateline>
      <salute>To <name>Franklin Blake, Esq.</name> Sir, -</salute>
    </opener>
    <p>I beg to apologize for the delay that has occurred in the
      production of the Report, with which I engaged to furnish you.
      I have waited to make it a complete Report ...</p>
    <closer>
      <salute>I have the honour to remain, dear sir, your
        obedient servant </salute>
      <signed>
        <name>RICHARD CUFF</name> (late sergeant in the
          Detective Force, Scotland Yard, London). </signed>
      </closer>
    </div>
  </div>
```

immagine dal sito TEI Guide Lines

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Documento ben formato (well-formed)

Un documento XML deve essere **ben formato** (*well-formed*, cioè non deve contenere **errori sintattici** e deve soddisfare le **regole generali della specifica**).

Un documento non ben formato non può essere letto dalle applicazioni che elaborano codice XML.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Parti principali di un documento XML

Un documento XML consiste di tre parti:

- il prologo
- il corpo (body)
- l'epilogo

Fondamenti XML

eXtensible Markup Language: Esempio TEI

Codifica di

Testi -

Introduzione

XML Markup

a.a.

2024-2025

A.M. Del

Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/css" href="customStyle.css"?>
<!--The following document is made online by the Perseus Project -->
<!--Added the TEI-lite DTD and a processing instruction -->
<!DOCTYPE TEI.2 SYSTEM "teixbaby.dtd">

<TEI.2>
  <text lang="en">
    <body>
      <div1 type="book" n="1" org="uniform" sample="complete">
        <div2 type="section" n="327A" org="uniform" sample="complete">
          <p>
            327A - 328B Socrates describes how he visited the Piraeus in company with Glauco, and
            was induced by Polemarchus and others to defer his return to Athens.
          </p>
          <p>
            <lemma lang="greek" targOrder="U" from="ROOT" to="DITTO">κατέβην κτλ.</lemma>
            Dionys. Hal.
            <title lang="la">de comp. verb.</title>
            p. 208 (Reiske)
            <foreign lang="greek">
              ὁ δὲ Πλάτων, τοὺς
              ἑαυτοῦ διαλόγους κτενίζων καὶ βοστρυχίζων, καὶ πάντα τρόπον ἀναπλέκων, οὐ
              διέλιπεν ὀγδοήκοντα γεγονῶς ἔτη. πᾶσι γὰρ δὴ που τοῖς φιλολόγοις γνώριμα
              τὰ περὶ τῆς φιλοπονίας τάνδρὸς ιστορούμενα, τὰ τ' ἄλλα, καὶ δὴ καὶ τὰ
              περὶ τὴν δέλτον ἦν τελευτήσαντος αὐτοῦ λέγουσιν εὐρεθῆναι ποικίλως
              μετακειμένην τὴν ἀρχὴν τῆς πολιτείας ἔχουσαν τήνδε "κατέβην χθές
              εἰς Πειραιᾶ μετὰ Γλαύκωνος τοῦ Ἀριστῶνος
            </foreign>
            ."
          </p>
        </div2>
      </div1>
    </body>
  </text>
</TEI.2>

<!-- This document is not completed and was cut without a special meaning -->
```

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Documento XML: prologo

- XML declaration (obbligatorio)
- Processing instructions (opzionale)
- Commenti (opzionale)
- Document type declaration (opzionale)

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Documento XML: corpo

Il corpo del documento XML segue immediatamente il prologo. Questa parte del documento contiene il contenuto vero e proprio in una **struttura ad albero ordinata**.

Documento XML: epilogo

Opzionalmente, al corpo del documento XML segue un epilogo il quale può contenere commenti finali e processing instructions.

Fondamenti XML

eXtensible Markup Language: Prologo

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML declaration

```
<?xml version="version number" encoding="encoding  
type" standalone="yes|no" ?>
```


Fondamenti XML

eXtensible Markup Language: Prologo

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML declaration: ERRORI

```
<?XML VERSION="1.0" ENCODING="ISO-8859-1"  
      STANDALONE="YES" ?>  
  
<?xml version=1.0 encoding=ISO-8859-1  
      standalone=yes ?>  
  
<?xml version="1.0" standalone="yes"  
      encoding="ISO-8859-1" ?>
```

Fondamenti XML

eXtensible Markup Language: Prologo

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML comments

I commenti XML vengono ignorati dai programmi che elaborano il documento.

I commenti quindi non influenzano i contenuti e la struttura del documento.

XML comments: sintassi

```
<!-- il parser XML qui non entra -->
```

Un commento può occupare anche più righe

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

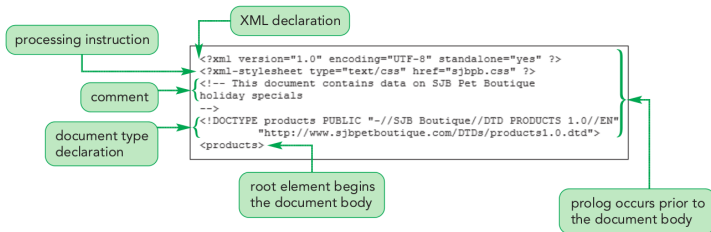


immagine dal libro New Perspectives on XML, 3rd Edition

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Esercizio prologo

Creare un file `.xml` ed inserire un prologo con la dichiarazione XML e un commento con le vostre informazioni.

Esercizio prologo

```
<!-- This document contains data on Codifica di  
Testi.
```

```
Filename:  project.xml
```

```
Author:   your name
```

```
Date:    today's date -->
```

Salvare il file su github nel repository del progetto

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML parser

Un programma che legge ed interpreta un documento XML è chiamato XML parser (o processor).

Cosa fa un XML parser

- Verifica che il documento rispetti la sintassi XML
- Interpreta i dati con tipo PCDATA (*Parsed*)
- Risolve character or entity references
- Gestisce le processing instructions per interpretare i dati

Fondamenti XML

eXtensible Markup Language

XMLlint

```
XMLLINT(1)                                xmlLint Manual                                XMLLINT(1)

NAME
    xmlLint - command line XML tool

SYNOPSIS
    xmlLint [--version | --debug | --shell | --xpath "XPath expression" | --debugout | --copy |
    --recover | --noent | --noot | --nonet | --path "PATH(S)" | --load-trace | --htmlout |
    --nowrap | --valid | --postvalid | --dtdvalid URL | --dtdvalidfpi FPI | --timing |
    --output FILE | --repeat | --insert | --compress | --html | --xmlout | --push | --memory |
    --maxmem NBYTES | --nowarning | --noblanks | --nocdata | --format | --encode ENCODING |
    --dropttd | --nsclean | --testIO | --catalogs | --nocatalogs | --auto | --xinclue |
    --noxincludenode | --loadtdt | --dtdattr | --stream | --walker | --pattern PATTERNVALUE |
    --chkregister | --relaxng SCHEMA | --schema SCHEMA | --c14n] {XML-FILE(S)... | -}

    xmlLint --help

DESCRIPTION
    The xmlLint program parses one or more XML files, specified on the command line as XML-FILE (or the
    standard input if the filename provided is - ). It prints various types of output, depending upon
    the options selected. It is useful for detecting errors both in XML code and in the XML parser
    itself.

    xmlLint is included in libxml(3).

OPTIONS
    xmlLint accepts the following options (in alphabetical order):
```

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML body

Un documento XML è composto da elementi e attributi.
Gli elementi sono la base, le unità fondamentali di qualsiasi documento XML.

Elementi: Sintassi

```
<element>content</element>  
opening tag:  <element>;  
closing tag:  </element>
```

Un elemento può contenere testo e/o ulteriori elementi

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Element

Gli elementi XML possono avere diversi tipi di contenuto:

- contenuto strutturale: solo altri elementi, non testo
- contenuto misto: testo e anche altri elementi
- contenuto testuale: solo testo, non altri elementi

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Element: note importanti sul nome

- Gli elementi sono case sensitive.
- Gli elementi possono iniziare con una lettera o con un “_”.
- Un elemento non può iniziare con la stringa *xml*.
- Il tag di apertura e di chiusura devono avere lo stesso nome.
- Un tag può essere usato più di una volta.
- Un insieme di elementi costituiscono un vocabolario

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Element: empty e nested

- Un elemento vuoto (*empty*) è un elemento senza contenuto.
- Un elemento può contenere altri elementi opportunamente annidati (*nested element*).

XML esempi: empty e nested element

- `<element /><element></element>`
- `<choice><sic>testo con errore</sic><corr>
testo corretto</corr></choice>`

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Element: hierarchical relationship

- Un elemento annidato (*nested*) è un elemento *figlio*, cioè contenuto (annidato) in un ulteriore elemento detto padre/genitore (*parent*).
- Gli elementi che sono presenti su uno stesso livello gerarchico (*side by side*) sono detti *sibling element*.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

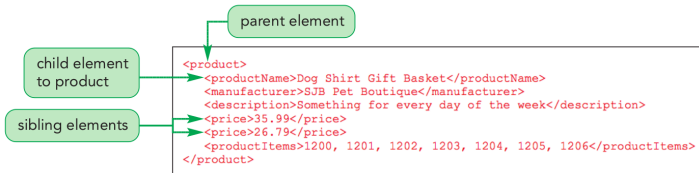


immagine dal libro New Perspectives on XML, 3rd Edition

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Element: hierarchical relationship

- Tutti gli elementi nel body del documento sono figli/discendenti di uno stesso elemento, chiamato radice (*root*).
- Un documento XML deve contenere un elemento root per essere considerato ben formato.
- Una gerarchia XML può essere rappresentata tramite un diagramma ad albero.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Element: hierarchical relationship cont.

- Il prologo e i commenti non fanno parte dell'albero del body.
- Elementi non annidati correttamente implicano un errore di sintassi nei parser.
- Le specifiche XML non consentono di sovrapporre i tag di apertura e di chiusura degli elementi annidati (*no overlap*).

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Element: hierarchical relationship - Esercizio

Scrivere e fare il check di un xml non opportunamente annidato

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Element: hierarchical relationship as tree structure

Un modo rapido e comodo per visualizzare la struttura completa di un documento XML è quello di disegnare attraverso un diagramma ad albero ordinato gli elementi del documento XML.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

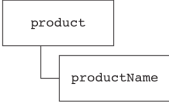

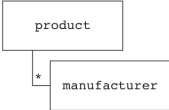
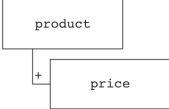
A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

| Symbol | Description | Chart | Interpretation |
|--------|---|--|---|
| [none] | The parent contains a single occurrence of the child element. |  <pre> graph TD product[product] --- productName[productName] </pre> | A product element must contain a single productName element. |
| ? | The parent contains zero or one of the child elements. |  <pre> graph TD product[product] --- ?[?] --- description[description] </pre> | A product element may contain a description element. |
| * | The parent contains zero or more of the child elements. |  <pre> graph TD product[product] --- *[*] --- manufacturer[manufacturer] </pre> | A product element can contain zero or more manufacturer elements. |
| + | The parent contains at least one of the child elements. |  <pre> graph TD product[product] --- +[+] --- price[price] </pre> | A product element must contain one or more price elements. |

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

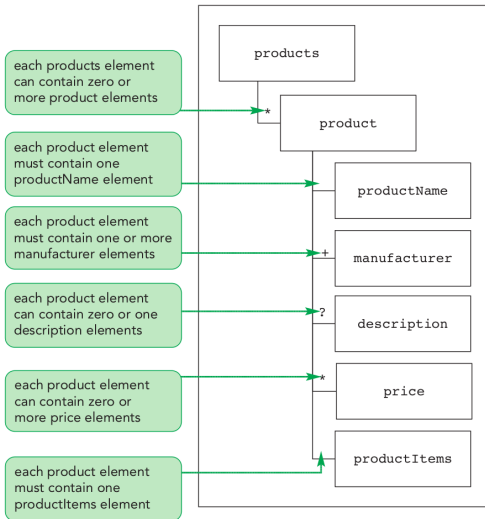
A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni



Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Element: Mixed Content

Un elemento può contenere contemporaneamente sia testo sia altri elementi.

Questo modello di contenuto si chiama Mixed Content ed è ideale per descrivere informazioni text-based (**dati semi-strutturati**).

XML Element: Mixed Content

```
<p><salute>Salve</salute> il mio nome è  
<persName>Angelo</persName></p>
```

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Element: Esercizio

Aprire il file XML non ben formato presente nel repository github:

- validarlo con un parser XML
- correggerlo (commentando gli errori e le modifiche)
- aggiungere un figlio (child) ad un elemento
- aggiungere un fratello (sibling) ad un elemento

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Attributi

Gli elementi in un documento XML possono avere uno o più attributi.

Un attributo descrive una caratteristica dell'elemento in cui appare.

XML Attributi

Un attributo ha senso solo all'interno del proprio elemento e non è possibile separarlo da esso in alcun modo.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Attributi: valore

Un attributo ha due componenti: nome - valore. Il valore di un attributo è una stringa e deve essere sempre racchiusa tra apici (singoli o doppi).

XML Attributi: valore

```
<element attribute='value'> ... </element>  
    <element attribute='value' />  
    <element attribute='value',  
        attribute2='value2' />
```

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Attributi: restrizioni ai nomi

- Il nome di un attributo può iniziare con una lettera oppure underscore.
- Gli spazi non sono consentiti in un nome di un attributo.
- Il nome di un attributo non può iniziare con la stringa *xml*.

XML Attributi

- Il nome degli attributi è *case sensitive*.
- L'ordine degli attributi non è significativo.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

XML Character and Entity References

- numeric character reference: `&#nnn;`
- hexadecimal character reference: `&#xhhhh`
- character entity reference: `&entity;`

XML References

- `A` (*carattere A*)
- `&` (*carattere &*)

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

| Symbol | Character Reference | Entity Reference | Description |
|--------|---------------------|------------------|---------------------------|
| > | > | > | Greater than |
| < | < | < | Less than |
| ' | | ' | Apostrophe (single quote) |
| " | | " | Double quote |
| & | & | & | Ampersand |
| © | © | © | Copyright |
| ® | ® | ® | Registered trademark |
| ™ | ™ | | Trademark |
| ° | ° | | Degree |
| £ | £ | | Pound |
| € | € | € | Euro |
| ¥ | ¥ | ¥ | Yen |

immagine dal libro New Perspectives on XML, 3rd Edition

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Text Character Parsing

Il contenuto testuale di un elemento XML può essere diviso in tre categorie: parsed character data, character data, and white space.

Text Character Parsing

- PCDATA
- CDATA
- White Space

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Parsed Character Data

Parsed character data (PCDATA) si riferisce a tutti quei caratteri che XML tratta come parte del codice e quindi vengono interpretati dai parser.

PCDATA

- XML declaration
- Opening tag e closing tag
- Character or entity references
- Commenti

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Parsed Character Data

La presenza di contenuti di tipo PCDATA può causare errori inaspettati.

XML PCDATA

Caratteri speciali che sono utilizzati dalla specifica XML come &, <, > non possono essere utilizzati come contenuto testuale.

Character Data

I dati di tipo “Character Data” non vengono interpretati dal parser XML.

La sequenza di caratteri viene trattata come puro contenuto.
In definitiva una sezione *CDATA* è un blocco di testo.

XML CDATA: sintassi

```
<![CDATA [  
character data  
]]>
```

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Character Data

Le sezioni di testo CDATA possono essere inserite in qualsiasi parte del documento XML.

Utile per inserire una sezione di testo con molti caratteri speciali.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

CDATA: qualche vincolo

- Non è possibile inserire commenti in una sezione CDATA.
- Non è possibile annidare sezioni CDATA.
- Non possono essere vuote.
- i simboli “]]” non sono ammessi.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Esempio ed esercizio

Inserire all'interno di un tag un frammento di codice HTML

CDATA: esempio

```
<htmlCode> <![CDATA[ <h1>Capitolo Primo</h1>  
  <h2>Sezione Seconda</h2> ]]> </htmlCode>
```


Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

White Space: esempio

- Gli spazi bianchi sono ignorati quando sono tra i tag.
- Gli spazi bianchi sono ignorati all'interno del prologo e dell'epilogo e all'interno dei tag.
- Gli spazi bianchi inseriti nel valore di un attributo sono trattati come parte del contenuto.
- Non vengono strippati gli spazi all'interno del contenuto testuale degli elementi.

I white space sono caratteri non stampabili

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Processing Instruction

Una *processing instruction* è un comando, una direttiva, che indica al parser XML in che modo elaborare e trattare tutto o parte del documento XML.

Processing Instruction: sintassi

```
<?target instruction ?>  
  
<?xml-stylesheet type="text/css" href="main.css"  
media="all" ?>
```

Molteplici processing instruction possono co-esistere all'interno di un unico documento XML.

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Processing Instruction: sintassi

```
<?target instruction ?>  
  
<?xml-stylesheet type="text/css" href="main.css"  
media="all" ?>
```

Processing Instruction

Target: identifica il tool al quale la processing instruction è diretta.

Instruction: identifica le informazioni che il documento passa al parser per essere elaborate. Le istruzioni hanno la forma degli attributi (nome-valore).

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Namespaces

Un namespace può essere visto come una collezione di elementi e attributi e un insieme di regole che ne determinano la struttura e il contenuto.

Namespaces

```
<element xmlns:prefix="uri"> ... </element>
    <element xmlns="uri"> ... </element>
        <tei:TEI
xmlns:tei='“http://www.tei-c.org/ns/1.0”'>
<TEI xmlns='“http://www.tei-c.org/ns/1.0”'>
```

Fondamenti XML

eXtensible Markup Language

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

Namespaces

Un namespace viene ereditato da tutti gli elementi discendenti dell'elemento in cui esso è stato dichiarato.

Namespaces

Generalmente si dichiarano tutti i namespace nell'elemento root così da avere a disposizione tutti gli elementi dei vari namespace in tutto il documento XML

Progress status

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

1 I linguaggi di codifica

2 Fondamenti del linguaggio XML

3 Validare un documento XML e Definire uno Schema

4 Conclusioni

Progress status

Codifica di
Testi -
Introduzione
XML Markup
a.a.
2024-2025

A.M. Del
Grosso

I linguaggi di
codifica

Fondamenti
del linguaggio
XML

Validare un
documento
XML e
Definire uno
Schema

Conclusioni

1 I linguaggi di codifica

2 Fondamenti del linguaggio XML

3 Validare un documento XML e Definire uno Schema

4 Conclusioni

XML per rappresentare il testo

- I markup language per supportare la rappresentazione, memorizzazione, pubblicazione di un testo.
- XML è un markup language flessibile e potente.
- le istruzioni dei markup language sono per lo più dichiarazioni indicando particolari funzioni del dato.
- le istruzioni sono etichette visibili.

XML per rappresentare il testo

- Una sintassi e una grammatica regolano l'applicabilità del linguaggio di marcatura
- Sintassi: documento well formed (ben formato)
- Grammatica: documento valido

XML per rappresentare il testo

- XML deriva dal linguaggio SGML.
- XML è una specifica del consorzio W3C.
- XML è un meta-linguaggio.
- XML è plain text.
- XML è portabile.

XML per rappresentare il testo

- XML definisce markup dichiarativi e descrittivi.
- XML ha un modello dati ad albero ordinato.
- XML può avere associato un tipo di documento (DTD) o uno schema di codifica (XSD).