

# **Laboratorio di Editoria Digitale**

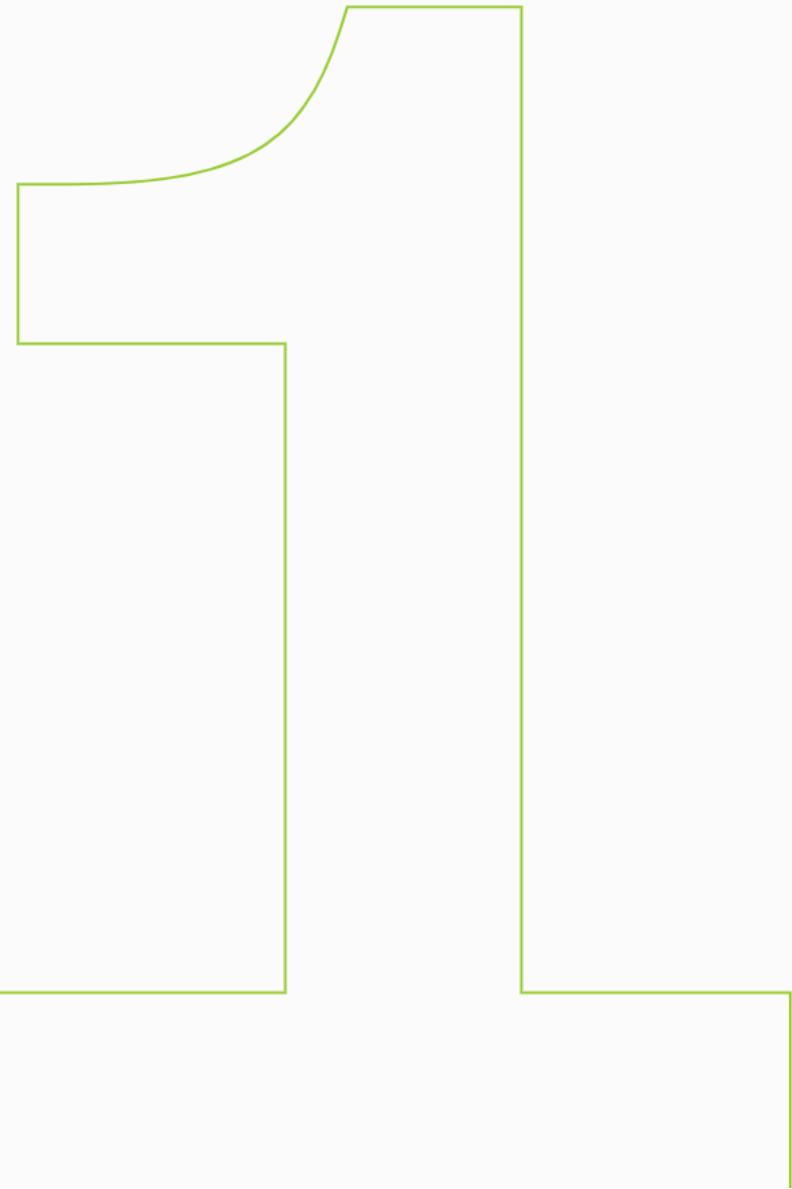
**a.a. 2014-2015**

**Scuola di Lingue e Letterature, Traduzione e Interpretazione**  
Alma Mater Studiorum — Università di Bologna

**Marilena Daquino** marilena.daquino2@unibo.it

# Introduzione all'editoria digitale

## Lezione 1



# Organizzazione delle lezioni

## 1. Introduzione

Fondamenti dell'editoria cartacea e dell'editoria digitale.

Approfondimento sulla testualità, l'analisi filologica e linguistica (1 lezione).

## 2. Tutorial XML/TEI

Strumenti di base (teorici e applicativi) per la realizzazione  
di un progetto editoriale sul web (4 lezioni).

## 3. Laboratorio

Esercitazione pratica: presentazione dell'editor XML/TEI **editei** (1 lezione)  
e marcatura di un testo concordato (4 lezioni facoltative).

## Elaborato finale

Marcatura XML di un testo concordato.

Requisiti minimi: consegna di un file XML valido e ben formato, che preveda la  
marcatura di uno dei livelli trattati durante il corso (linguistico o filologico)

# Perché un corso sull'editoria digitale?

## Gli obiettivi

- // definire i concetti chiave per fornire una **panoramica** delle tecnologie utili nel settore editoriale e dello studio teorico necessario
- // fornire conoscenze e strumenti preliminari per entrare nel mondo **dell'editoria e della ricerca**
- // acquisire coscienza delle proprie conoscenze/capacità per poter **verticalizzare** le proprie competenze

# Il focus del corso

## Restringiamo il campo

// rappresentazione del **testo** e non dei media in generale  
(audio, video, immagini)

// rappresentazione digitale mediante **codifica** e non digitalizzazione  
(i.e. edizione digitale e non scansione)

// rappresentazione del testo per il **web** e non su altri supporti digitali “off-line”

// rappresentazione di un **testo a stampa preesistente** sul web  
e non “edizione digitale nativa”

// approfondimento sugli **aspetti letterari/filologici e linguistici** del testo  
rappresentabili in formato digitale

# Editoria digitale/multimediale

## Una definizione

*“Il trattamento e la preparazione di contenuti testuali e/o multimediali per la loro pubblicazione sul web eseguita mediante criteri professionali.”*

(F. Tissoni, 2009)

ovvero:

- // la manipolazione del testo e scelte effettuate dall'editore per la sua **rappresentazione** digitale (tramite digitalizzazione e/o codifica del testo)
- // l'adozione di **standard** tecnologici (linguaggi e software) e standard descrittivi (schemi di metadati e vocabolari della propria comunità di riferimento) per facilitare la conservazione e la fruizione nel tempo
- // previsione degli aspetti legati alla **disseminazione** di informazioni complesse per un pubblico vasto e potenzialmente indifferenziato

# L'editoria sul web

## Scenari

Uno **scenario in evoluzione**, in cui si muovono risorse economiche ed intellettuali, dove le competenze umanistiche sono richieste parimenti a quelle tecniche e tecnologiche. In questo scenario sono coinvolte:

- // la facilità di consultazione e l'aggiornamento continuo dei contenuti possono garantire un **servizio informativo** più agevole e stimolante
- // la “libertà nella circolazione della cultura” porta a **dinamiche di collaborazione** tra autori/utenti
- // l’eliminazione di costi per la produzione del supporto materiale dei testi favorisce l’investimento nella sperimentazione e **l’innovazione**
- // le nuove **possibilità didattiche** e di apprendimento (multimedialità, approfondimenti interattivi, molitudine di contenuti sempre a portata di mano) sono un utile strumento per docenti, ricercatori, studenti ecc...

# Problematiche legate alla pubblicazione di contenuti sul web

Questi processi devono essere “guidati” da **consapevolezza teorica** (conoscenza della materia, tecniche di comunicazione, di organizzazione dei contenuti, garanzia della fruibilità dei contenuti ecc...) e non solo tecnica

// libertà del web: ogni utente può diventare **autore**?

// chi assicura la **qualità dei contenuti** e la professionalità/autorevolezza dell'autore? (es. Wikipedia). Quali contenuti sono citabili?

// quali **funzionalità** offre il nuovo medium in più rispetto al medium cartaceo? Non può essere una mera riproduzione digitale.

## QUINDI

Necessità di autori/editori con conoscenze consolidate sulla natura, sulle **problematiche del testo e sui cambiamenti** che questo subisce nel passaggio su un nuovo medium.

# Il testo e i suoi mutamenti

## 1. La materialità e l'immaterialità del testo

Distinguere l'opera dal testo e dalle sue manifestazioni , il **modello FRBR**

**Work** **Ossi di seppia** di E. Montale

**Expression** Il testo originale, il testo tradotto in un'altra lingua...

**Manifestation** Le edizioni del testo

**Item** Il singolo libro (l'oggetto materiale)

Il **supporto** del testo da informazioni preziose sul suo contenuto e sulle intenzioni dell'autore nella trasmissione del messaggio (dell'opera)  
es. un romanzo ha una precisa impaginazione: è suddiviso in capitoli, paragrafi...

**QUINDI**

Modificando il supporto viene modificato anche il messaggio e/o il modo di fruirlo e interpretarlo.

# Il testo e i suoi mutamenti

## 2. La fissità/stabilità del testo

Il testo a stampa consacra il testo *chiuso*  
(in termini di contenuto e contenente, i.e. il libro).



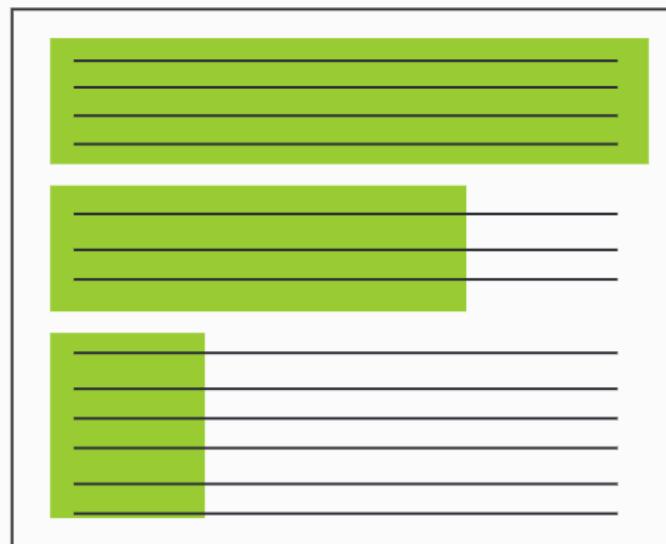
L'immutabilità dei contenuti e dell'autorialità vengono messi in discussione nel passaggio sul nuovo media digitale, che richiede una codifica intermedia per la sua rappresentazione, interfacce prestabilite ma modificabili/personalizzabili, continua riproducibilità del testo per la condivisione...

# Il testo e i suoi mutamenti

## 3. La visualità del testo

Alla lettura sequenziale come metodologia privilegiata per il testo a stampa, subentra nel web una lettura modulare, associativa (i salti tra link o tra blocchi di testo all'interno di una stessa pagina) e prettamente visuale.

e.g. La forma a F (J. Nielsen, 2006)



Questa modalità non sostituisce la lettura sequenziale, ma la integra con altre possibilità di fruizione dei contenuti, i quali vanno riorganizzati tenendo conto delle peculiarità nella nuova forma-ipertesto.

# L'ipertesto

“scrittura non sequenziale, testo che si dirama e consente al lettore di scegliere: qualcosa che si fruisce al meglio davanti a uno schermo interattivo. Così come è comunemente inteso, un ipertesto è una serie di brani di testo tra cui sono definiti legami che consentono al lettore **differenti cammini**.  
[...]

L'ipertesto include come **caso particolare la scrittura sequenziale**, ed è quindi la forma più generale di scrittura. Non più limitati alla sola sequenza, con un ipertesto possiamo creare nuove forme di scrittura che riflettano la struttura di ciò di cui scriviamo, e **i lettori possono scegliere percorsi diversi** a seconda delle loro attitudini, o del corso dei loro pensieri, in un modo finora ritenuto impossibile.” (T. Nelson, 1965)

# L'interpretazione del testo sul nuovo media

- // vengono modificati/eliminati gli **elementi paratestuali** (copertine, titolo, prefazione, note, illustrazioni...) che forniscono i riferimenti di qualità dei contenuti
- // vengono modificate le scelte autoriali nella **rappresentazione grafica e spaziale dei contenuti** (i glifi, l'impaginazione, i margini...)
- // il testo (i suoi contenuti) non è più fisso, immutabile, ma modificabile nel tempo e dipendente dall'**interfaccia** che lo ospita
- // la **fruizione** del testo avviene mediante diverse modalità concorrenti che impongono un ripensamento della sua rappresentazione
- // l'**autorialità individuale** viene messa in discussione a causa della continua modificabilità dei contenuti, dell'attribuzione collaborativa, della necessità di condivisione dei contenuti, del mutamento del ruolo del lettore-utente.

# Gli obiettivi dell'editoria digitale

Di fronte a questo scenario, che disorienta il lettore, l'editoria digitale deve:

- // fornire gli **standard qualitativi in termini di contenuto**, egualmente ad una edizione cartacea
- // assicurare la **professionalità dell'autore/editore** dei contenuti
- // sperimentare **forme innovative** per la fruizione dei testi e lo sviluppo di strumenti che arricchiscano il potenziale comunicativo del testo stesso, affinché non siano un mero strumento di riproduzione digitale.

# Le competenze necessarie

## Le figure professionali

In sintesi, non ogni testo sul web è un'edizione, per la quale servono competenze trasversali e diverse figure professionali:

// **Editoria multimediale.** Necessità di un editore critico con competenze teoriche trasversali e conoscenza di diversi strumenti tecnologici.

// **Organizzazione della conoscenza.** Non ambisce solo all'utilizzo strumentale delle tecnologie (indicizzazioni, calcoli sui dati, statistiche lessicali...) ma a nuove metodologie per organizzare, elaborare e disseminare le informazioni.

// **Web design.** Si richiede di coniugare le esigenze qualitative di un'edizione cartacea con le possibilità offerte dal nuovo media, aumentando l'interazione con il testo e le informazioni di contesto in una forma grafica efficiente.

// **Umanistica informatica.** Affronta le problematiche legate al nuovo atto interpretativo della *codifica* del testo (frutto di una scelta soggettiva).

# La codifica del testo

## I linguaggi di markup

### Codifica

*L'utilizzo di un linguaggio condiviso tra uomo e macchina per rappresentare e conservare informazioni in formato digitale.*

### Codifica del testo

*Rappresentazione del testo su supporto digitale in un formato "leggibile" da un computer - Machine Readable Form (MRF)*

### Linguaggi di markup

Consentono di fornire informazioni inerenti/a corredo del testo, dentro al testo/in un documento separato.

"Con questo termine ci riferiamo alla possibilità di aggiungere alla sequenza di caratteri che rappresentano il documento digitale **altre stringhe di caratteri** [...] denominate marcatori, utili a **descrivere determinati aspetti** – relativi o alla struttura logica del documento o alle sue caratteristiche fisiche – funzionali alla produzione del documento elettronico" (F. Tomasi, 2008)

# La codifica del testo

## I linguaggi di markup

### I tipi di codifica

// Proprietario. Le specifiche non sono disponibili apertamente.

// *Non proprietario*. Risorse open source, fruibili dalla comunità.

// *Leggibile*. Il file contenente la marcatura è visualizzabile e interpretabile.

// Non leggibile. Il documento è nascosto.

// Presentazionale. Il linguaggio di markup mira a rappresentare la struttura fisica del testo mettendone in luce le sue caratteristiche ‘superficiali’.

// *Analitico*. Rende evidente le relazioni logiche e presenti all’interno del testo.

// Procedurale. Contiene istruzioni inerenti l’impaginazione del documento (la spaziatura, il font, l’interlinea ecc....).

// *Dichiarativo*. Specifica la struttura logica di un documento.

# La codifica del testo

## I linguaggi di markup

### I livelli di codifica del testo

// Codifica di livello 0 o di basso livello. Riguarda la rappresentazione binaria della sequenza ordinata dei caratteri (la codifica dei caratteri, *Unicode*)

// Codifica di alto livello. Arricchisce il testo codificato al livello zero con informazioni di tipo strutturale.

e.g. l'organizzazione del testo in elementi macrotestuali (e.g. paragrafazione, intestazioni ecc...), gli elementi stilistici-grafici (grandezza del font, margini, ecc...), valenza semantica del testo, la specifica di strutture linguistiche (tipologia di sintagma, analisi morfologica ecc...) e quindi di ogni possibile **interpretazione** l'editore voglia corredare il testo, esplicitandola formalmente.

# La codifica del testo

## I linguaggi di markup

**Un esempio di testo:** Andrea Bocelli, *Romanza*, 1996; EU, Polydor; 10.80 \$

### Un esempio di codifica dei caratteri

Andrea Bocelli = 01000001 01101110 01100100 01110010 01100101 01100001  
00100000 01000010 01101111 01100011 01100101 01101100 01101100 01101001

### Un esempio semplice di codifica di alto livello

```
<cd>
  <title>Romanza</title>
  <artist>Andrea Bocelli</artist>
  <country>EU</country>
  <company>Polydor</company>
  <price>10.80</price>
  <year>1996</year>
</cd>
```

# Cosa dire con la codifica

## La teoria OHCO

“La teoria OHCO [Ordered Hierarchy of Content Objects] sostiene che quando due strutture si sovrappongono, appartengono a due livelli distinti di interesse interpretativo; deve essere di conseguenza effettuata una scelta precisa e consapevole: **decidere quale livello si vuole descrivere** (per esempio linguistico, narratologico, filologico).

Si deve considerare sia la variabilità di concetto di testo rispetto alle esigenze analitiche, sia la molteplicità degli **obiettivi computazionali**, che cambiano in modo direttamente proporzionale al punto di vista assunto sugli elementi di contenuto e richiedono diverse modalità rappresentazionali, sulla base delle esigenze della codifica.”

(F. Tomasi, 2008)

# Quale codifica? HTML, linguaggio del web

Il linguaggio di markup per la **formattazione** dei testi (ipertesti) nel web.

## Il web

Ideato nel 1990 da Tim Berners-Lee, il web è fondato su una **rete di testi**:

// formalizzati in HTML (HyperText Markup Language)

// reperibili tramite un protocollo (HTTP, Hypertext Transfer Protocol)

// identificabili univocamente (URL, Universal Resource Locator)

# Quale codifica? HTML, linguaggio del web

HTML prevede:

- // una **struttura parzialmente vincolante** di elementi per definire gli aspetti strutturali del testo
- // un **vocabolario** prestabilito per definire gli elementi che compongono un testo
- // si concentra - anche se non esclusivamente - sugli aspetti di **formattazione del testo**, meno sulla semantica o altri aspetti legati al contenuto del testo

**HTML5** mostra un progressivo incremento degli aspetti semantici nella *definizione degli elementi di una pagina web*, ma non sono ancora sufficienti per *descrivere un testo letterario*.

# Quale codifica? HTML, linguaggio del web

## Un esempio semplice di codifica HTML

```
<html>
  <head>
    <title>My CD catalog</title>
  </head>
  <body>
    <div id="AB-Romanza">
      <p>Romanza</p>
      <p>Andrea Bocelli</p>
      <p>EU</p>
      <p>Polydor</p>
      <p>10.80</p>
      <p>1996</p>
    </div> ... ...
  </body>
</html>
```



elementi strutturali non modificabili.



elementi che esplicitano la formattazione del testo, non dicono nulla sul significato.

**Non è sufficiente per definire compiutamente gli aspetti fondamentali di un testo letterario!**

# Quale codifica? XML, metalinguaggio per i testi

È necessario un linguaggio di markup che offra maggiore libertà nella definizione dei tag pur rimanendo nell'ambito del rispetto di uno standard.

Nel 1998 viene rilasciato XML (eXtensible Markup Language) progettato per *descrivere dati* in modo flessibile, estendibile.

Riprendendo l'esempio precedente:

```
<cd>
  <title>Romanza</title>
  <artist>Andrea Bocelli</artist>
  <country>EU</country>
  <company>Polydor</company>
  <price>10.80</price>
  <year>1996</year>
</cd>
```

i **tag**, le etichette apposte alle porzioni di testo identificano il significato, ovvero forniscono una classificazione del testo a cui si riferiscono. Sono leggibili dall'uomo e dalla macchina. Sono personalizzabili in base all'informazione da descrivere.

# Quale codifica? XML o HTML

## XML

Studiato per rappresentare dati ma non per visualizzarli.

Per visualizzare il testo su un browser in modo “friendly” deve essere trasformato in HTML.

Non ha tag predefiniti, ma personalizzabili a seconda del tipo di informazione che deve rappresentare.

## QUINDI

In XML formalizziamo le informazioni riguardanti la struttura logica, la semantica, gli aspetti specifici di un ambito scientifico.

In HTML formalizziamo le informazioni per visualizzare il testo sul web.

## HTML

Studiato per la formattazione e la visualizzazione di testi su browser.

Per formalizzare informazioni aggiuntive deve fare ricorso a un file separato in XML.

Ha un set di tag predefiniti e non modificabili.

# Questioni di filologia

Filologia digitale

Filologia tradizionale

Filologia d'autore

Filologia dei testi a stampa

Edizioni digitali

# Questioni di filologia

## Filologia digitale

I linguaggi di markup fanno sorgere questioni di stampo prettamente filologico.

“Un discorso è dire che una stringa di caratteri è in corsivo, altro è dire che è una parola in lingua straniera oppure che è il titolo di un libro. Da un lato possiamo rappresentare, con un **markup presentazionale**, un fenomeno tipografico, dall’altro, effettuiamo un **markup analitico** e assegniamo alla stringa un valore di contenuto. Ma se consideriamo il corsivo come facente parte del testo, allora il testo non può essere considerato una sequenza invariabile di stringhe di caratteri e si sconfinà nella nozione di documento, nel senso di esemplare materiale che attesta la sequenza dei caratteri e assegna al testo determinate caratteristiche a livello di rappresentazione fisica. Allora diamo al markup la possibilità di **codificare elementi di formato, che qualificano il significato del testo.**” (F. Tomasi, 2008)

La sfida è restituire la dimensione semantica di un elemento tipografico attraverso la sua riproduzione digitale.

# Questioni di filologia

## Obiettivi della filologia digitale

Di queste e altre questioni si occupa la filologia digitale, disciplina che si propone di lavorare su più piani:

// il piano degli **strumenti e dei metodi** di analisi, ricostruzione e conservazione dei testi (intesi come documenti storici);

// il piano delle **teorie** che, in ciascuna epoca e in diversi contesti, hanno contribuito a formare quegli strumenti e quei metodi.

(cfr. D. Fiomonte, 2009)

La filologia digitale si propone di **produrre edizioni di testi in formato elettronico** secondo **criteri scientificamente accurati**, standard qualitativi e metodologie di studio proprie delle edizioni cartacee.

# Questioni di filologia

## Filologia tradizionale

In presenza di **testi antichi**, giunti per trasmissione manoscritta in una o più copie tra loro divergenti e talvolta danneggiate, il metodo più praticato è quello lachmanniano o **stemmatico**.

In assenza di originale, l'obiettivo è approssimarsi a esso quanto più possibile, attraverso la *recensio* (**collazione** dei manoscritti in rapporto a un testo base e determinazione dei rapporti di parentela) e la *emendatio* (**correzione** degli errori in base allo stemma o per congettura).

Nell'ultima fase, la *dispositio* (presentazione e pubblicazione del lavoro svolto), il testo ristabilito è accompagnato da un'**introduzione** che elenca, riassume e motiva le scelte fatte dall'editore e da un **apparato critico** con le lezioni accolte e le varianti (positivo) o solo le varianti (negativo).

# Questioni di filologia

## Filologia d'autore

La filologia d'autore è quel settore della filologia che si occupa dello studio dei manoscritti e delle **varianti introdotte dagli autori** sui loro manoscritti o stampe. (cfr. D. Isella)

Questa branca della filologia, riconosciuta come una disciplina autonoma, si differenzia dalla filologia tradizionale - o filologia della copia, che studia le **varianti di trasmissione** - proponendo altre tecniche di edizione per la rappresentazione delle varianti d'autore e l'evoluzione interna del testo.

# Questioni di filologia

## Filologia dei testi a stampa

La filologia dei testi a stampa, o bibliografia testuale, ha convergenze sia con la filologia lachmanniana che con la filologia d'autore: richiede conoscenze consolidate sia sul **processo di stampa** sia sulle possibili interferenze da parte del compositore, o del **correttore/revisore**, nella produzione del testo e quindi nella generazione di errori o varianti.

Rispetto alla filologia applicata ai manoscritti, la disciplina potenzialmente deve collazionare un numero ingente di testimoni che possono avere uguale valore nella gerarchia, quindi la produzione stessa dello **stemma** subisce delle modifiche.

Infatti, una variante non è più peculiare di un unico testimone, i.e. un unico testo, bensì di tutte le copie di quella edizione a stampa, quindi di un **testo ideale** (l'espressione, riprendendo la terminologia del modello FRBR): ciò richiede un'accurata **analisi bibliografica** e catalografica.

# Questioni di filologia

## Modelli di edizione

Nel caso dei manoscritti antichi abbiamo tre modelli di edizione possibili:

// **Meccanico.** Una riproduzione tecnica, un facsimile, del manoscritto.

// **Diplomatico-interpretativo.** Il testo rispetta quello attestato e in nota vengono esplicite le scelte dell'editore.

// **Critico.** Tenta la riproduzione (per congettura) del presunto testo originale.

Nel caso di testi a stampa, la ricostruzione dell'**esemplare ideale** è il modello più rilevante.

Quando ci si propone di realizzare un'edizione digitale si dovrà scegliere - o sapere fare convivere - il tipo di edizione e quindi quali informazioni codificare.

# Questioni di filologia

## Edizione digitale

Un'edizione digitale deve offrire strumenti efficienti per la visualizzazione e la fruizione del testo, superando i limiti imposti dal supporto cartaceo (costi, diffusione, limiti spaziali e organizzativi, ecc....).

La codifica del testo digitale consente potenzialmente di:

- // fruire dinamicamente il processo autoriale
- // gestire contemporaneamente più livelli (il facsimile, l'edizione diplomatica, l'edizione critica), senza imporre un testo base in fase di visualizzazione.
- // l'apparato critico (anche lungo), non ha più vincoli di spazio e può essere esteso a seconda delle esigenze del lettore
- // le scelte e la responsabilità dell'editore sono sempre evidenti, benchè il testo non sia "fisso", suscettibile di modifiche.

# Questioni di linguistica

Linguistica computazionale

Fasi dell'analisi linguistica

La codifica

Cosa codificare

Nuove informazioni e benefici

# Questioni di linguistica

## Linguistica computazionale

Esistono diversi filoni di studio che legano la linguistica alle applicazioni informatiche, aventi per oggetto l'**analisi computazionale del testo** al fine di estrarne informazioni sulle sue strutture di base (sintattiche, morfologiche, semantiche) e derivarne **modelli** (sistemi di regole linguistiche interpretabili dalle macchine).

“Le tecniche di **ML (Machine Learning)** affondano le loro radici in metodologie di analisi stocastico-statistiche estremamente sofisticate, in grado di costruire modelli del fenomeno in esame a partire da un’opportuna quantità di dati autentici (**corpora**), accuratamente **annotati**, che fungono sia da base statistica sia da insieme di esempi di una corretta gestione del fenomeno analizzato.”  
(F. Tamburini, <http://corpora.dslo.unibo.it/People/Tamburini/Pubs/Griselda2008.pdf>)

# Questioni di linguistica

## Fasi dell'analisi linguistica

Le competenze richieste vanno dalla capacità di reperimento dei dati linguistici (**corpora** o singoli testi) idonei alla ricerca, alla conoscenza dei metodi formali per l'analisi dei dati (qualitativa e quantitativa), sino alle competenze prettamente informatiche per il trattamento dei dati.

1. **Si selezionano i testi** da cui estrarre i dati: i criteri con cui selezionare il tipo di **corpus** dipende dagli obiettivi dell'analisi computazionale (generalità, modalità scritta/orale, sincronicità/diacronicità, lingua, integrità del testo, numero di parole-unità, rappresentatività di un fenomeno)
2. I dati testuali selezionati sono **rappresentati in formato digitale**, ovvero codificati (ad alto livello): si sceglie quale livello di analisi linguistica codificare
3. Il **corpus elettronico** creato sarà la fonte di manipolazioni e interrogazioni da cui estrarre informazioni.

# Questioni di linguistica

## La codifica

Anche nella linguistica computazionale, si ripropone il problema di quale codifica del testo utilizzare, in funzione della rappresentabilità delle informazioni.

// .doc, .pdf: forniscono una codifica degli aspetti presentazionali, grafici, mentre non consentono di risalire all'organizzazione astratta del testo (gli elementi linguistici, semantici...). Inoltre il testo prodotto è scarsamente *portabile*, ovvero necessita di software (proprietari) per la corretta visualizzazione.

// .xml: fornisce una codifica più ricca del testo (sia degli aspetti presentazionali che strutturali), e rimane portatile, poichè semplice plain text usufruibile con qualsiasi editor di testo.

# Questioni di linguistica

## Cosa codificare

### Tokenizzazione

Il token è l'unità base del testo digitale, che raggruppa parole ortografiche, numeri, sigle, segni di punteggiatura.

Ai fini dell'analisi linguistica il testo va tokenizzato - ovvero suddiviso nelle sue unità di base, le **stringhe di testo** - con una codifica di livello 0.

Ogni token va identificato univocamente e individuato all'interno del suo contesto di comparazione.

### Utilizzo delle regex

Le espressioni regolari (o *regex*), una forma di notazione algebrica, permettono di definire in modo formale degli schemi (**pattern**) sui comportamenti di un determinato elemento linguistico.

e.g. la regex `\d./` rappresenta “tutte le stringhe contenenti un numero seguito da qualsiasi altro carattere” e corrisponde a una stringa del tipo “...3D è un ramo della...”

# Questioni di linguistica

## Nuove informazioni e benefici

La tipologia di informazioni estraibili dipendono strettamente dal tipo di informazioni codificate:

e.g.

la marcatura a livello di **microstruttura del testo** (periodi, citazioni, abbreviazioni, nomi, date...) fornirà informazioni sulle scelte di un autore nella formulazione delle frasi, nell'utilizzo di termini stranieri, ecc...

l'**annotazione linguistica** (morphologia, sintassi, semantica...) fornirà informazioni quantitative/qualitative sull'utilizzo di un termine, sulle sue flessioni, sul contesto di applicazione di un dato termine/fenomeno ecc...

Questo tipo di informazioni possono essere utili sia come preliminare per studi di tipo letterario (analisi stilistica di un dato autore), sia per una più approfondita analisi linguistica di un fenomeno al fine di ricavarne un modello.

# Conclusioni

## Perchè questi approfondimenti?

- // Filologia, linguistica, editoria: discipline con una propria autonomia e tradizione, di possibile interesse per la prosecuzione degli studi, strettamente legate ai linguaggi di marcatura.
- // Interdisciplinarietà quale competenza indispensabile sia in un contesto lavorativo che di ricerca.
- // Applicazione dei principi della filologia nella creazione di prodotti culturali di qualità sia nell'editoria che nella ricerca.  
e.g. creare edizioni digitali di testi citabili perchè autorevoli; sperimentare metodologie di navigazione delle informazioni calibrate su più tipologie di utenti.
- // Applicazione di tecniche di linguistica funzionali allo sviluppo di applicazioni per l'estrazione di informazioni (utili a diversi tipi di utente).  
e.g. annotare un testo affinchè la macchina comprenda e interagisca col suo contenuto, per rispondere alle ricerche con maggiore accuratezza nei risultati.

# Bibliografia

## Editoria digitale

*Brevi introduzioni all'argomento*

F. Tissoni, *Lineamenti di editoria multimediale*, Editore Unicopli, 2009

F. Tissoni, *L'editoria multimediale del nuovo Web. Semantic Web e Web 2.0*, Editore Unicopli, 2010

## Questioni biblioteconomiche sollevate

*Modelli concettuali e approfondimenti nell'ambito biblioteconomico*

IFLA, FRBR - Functional Requirements for Bibliographic Records

<http://www.ifla.org/publications/functional-requirements-for-bibliographic-records>

## Questioni di filologia digitale

D. Fiornonte, Scrittura, filologia e varianti digitali, in *Rivista di Filologia Cognitiva*, aprile 2003. <http://w3.uniroma1.it/cogfil/VARIANTI.html>

# Bibliografia

## Informatica umanistica

*Per un approfondimento complessivo, su testualità, tecniche e tecnologie, nonché la loro applicazione nell'ambito delle scienze umane*

F. Tomasi, *Metodologie informatiche e discipline umanistiche*, Roma, Carocci, 2008

## Questioni di linguistica

A. Lenci, S. Montemagni, V. Pirrelli, *Testo e Computer. Elementi di linguistica computazionale*, Roma, Carocci, 2014

# Introduzione a XML

## Lezione 2



# XML

## Introduzione

È un **metalinguaggio** di markup, cioè un linguaggio che permette di definire “altri linguaggi” a seconda della sua applicazione.

È uno standard ufficiale sviluppato dal W3C (World Wide Web Consortium) nel 1998: deriva da SGML quale suo sottoinsieme semplificato, ma ad oggi lo sostituisce. (<http://www.w3.org/XML>)

Nasce con l’obiettivo di **rappresentare documenti** (e.g. un testo letterario) e/o **dati strutturati** (e.g. i riferimenti bibliografici) su supporto digitale.

Un testo marcato in sintassi XML è detto **documento XML**: contiene sia il testo che i tag (anch’essi testo) utilizzati per descrivere le informazioni insite nel testo; è “**leggibile**” dall’utente senza l’utilizzo di software specifici (i.e. con qualsiasi editor di testo). È infatti **indipendente** da qualsiasi software e hardware.

## Perché “metalinguaggio”?

e.g. Per descrivere le informazioni contenute in una rubrica telefonica avrò bisogno di identificare: nomi, cognomi, numeri di telefono, email, indirizzi ecc...

Per descrivere invece un poema antico avrò bisogno di poter identificare i versi, le rime, le figure retoriche ecc...

XML consente di definire gli elementi descrittivi a seconda della tipologia di testo in esame!

Un linguaggio di programmazione normalmente impone una terminologia per esprimere concetti e istruzioni.

XML - che non è un linguaggio di programmazione - dice solo come esprimere **formalmente** i concetti tramite una sintassi vincolante, ma la semantica degli elementi è decisa dall'utente!

# XML

## Introduzione

### Perchè XML e non HTML?

A differenza di HTML, XML non ha tag predefiniti e non serve per definire pagine Web:

// serve per **descrivere dati**, o meglio **metadati** (dati di dati), all'interno del documento stesso (o all'esterno di questo in certi casi - *standoff markup*)

// è indipendente dalla visualizzazione finale del documento stesso, la quale è demandata ad altri linguaggi (i.e. HTML).

Con XML possiamo preservare le informazioni dall'obsolescenza digitale, che caratterizza software e hardware, e porci in un secondo momento il problema della sua visualizzazione.

# Dove dire cosa?

## Il mio testo

Nel mezzo del cammin di nostra vita  
mi ritrovai per una selva oscura,  
ché la diritta via era smarrita.

3

Ahi quanto a dir qual era è cosa dura  
esta selva selvaggia e aspra e forte  
che nel pensier rinova la paura!

6

Tant'è amara che poco è più morte;  
ma per trattar del ben ch'i' vi trovai,  
dirò de l'altre cose ch'i' v'ho scorte.

9

## Il mio file XML

### terzina

Nel mezzo del cammin di nostra vita **endecasillabo**  
mi ritrovai per una selva oscura, **endecasillabo**  
ché la diritta via era smarrita. **endecasillabo**

3

Ahi quanto a dir qual era è cosa dura **endecasillabo**  
esta selva selvaggia e aspra e forte **endecasillabo**  
che nel pensier rinova la paura! **endecasillabo**

6

Tant'è amara che poco è più morte; **endecasillabo**  
ma per trattar del ben ch'i' vi trovai, **endecasillabo**  
dirò de l'altre cose ch'i' v'ho scorte. **endecasillabo**

9

rima ABA BCB CDC

## Il mio file HTML

Nel mezzo del cammin di nostra vita  
mi ritrovai per una selva oscura,  
ché la diritta via era smarrita.

3

Ahi quanto a dir qual era è cosa dura  
esta selva selvaggia e aspra e forte  
che nel pensier rinova la paura!

6

Tant'è amara che poco è più morte;  
ma per trattar del ben ch'i' vi trovai,  
dirò de l'altre cose ch'i' v'ho scorte.

9

blocco di testo  
per ogni terzina

numero di verso  
a sinistra

a capo per ogni  
verso

# XML

## La struttura di un documento XML

Ogni documento XML è caratterizzato da una **struttura ad albero**, composta da **nodi**:

// ogni documento XML ha un **nodo radice**, da cui discendono tutti gli altri

// un nodo può essere **1.** un sottoalbero dell'elemento radice (composto a sua volta da altri nodi); **2.** un singolo elemento (con eventuali attributi); **3.** una stringa di caratteri.

// tutte le informazioni sono contenute in nodi: ogni tipologia di informazione è contenuta in un elemento strutturale (i tag, ovvero etichette testuali) e può avere associata una lista di attributi (qualificazioni aggiuntive dell'informazione)

// tra i nodi esistono **relazioni gerarchiche e di dipendenza**: gli elementi possono essere dipendenti l'uno dall'altro, reiterabili e annidabili.

# XML

## La struttura di un documento XML

Un documento XML ha una struttura predefinita a cui deve rispondere:

```
<?xml version="1.0" encoding="UTF-8"?>
```

**prologo**

```
<message>
<to>Mark</to>
<from>Veronika</from>
<heading>Reminder</heading>
<body>
  <p>Don't forget me this weekend!</p>
  <br/>
  <p>We have to do homeworks.</p>
</body>
</message>
```

**elemento radice**

**struttura ad albero**

Nel prologo avremo la versione corrente di XML e la specifica del set di caratteri utilizzato (Unicode). `<? ?>` racchiude *istruzioni*.

L'elemento radice (in questo caso il tag `<note></note>`) è l'elemento contenitore di tutto gli altri, escluso il prologo.

Ogni sotto-elemento dell'elemento radice può contenere a sua volta altri elementi *anidati* in una struttura gerarchica ad albero.

# XML

## La struttura di un documento XML

Il *Document Object Model (DOM)* di un file XML, cioè il modo di interpretare e manipolare i dati di un documento XML, è una struttura ad albero.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<message>
  <to>Mark</to>
  <from>Veronika</from>
  <heading>Reminder</heading>
  <body>
```

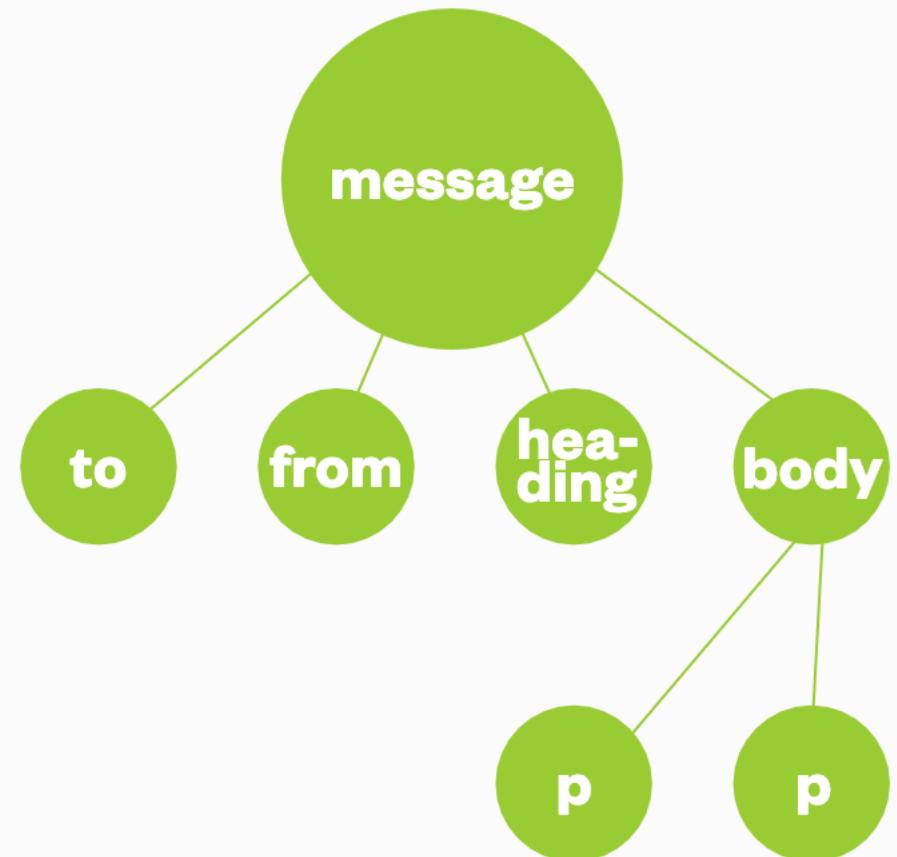
```
    <p>Don't forget me this weekend!</p>
```

```
    <br/>
```

```
    <p>We have to do homeworks.</p>
```

```
  </body>
```

```
</message>
```



# XML

## Le regole di sintassi

Un documento XML deve rispettare alcune regole di sintassi:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Ogni tag contenente dei dati deve essere *aperto e chiuso*: e.g. `<note></note>`.

```
<message>
  <to>Mark</to>
  <from>Veronika</from>
  <heading>Reminder</heading>
  <body>
    <p>Don't forget me this weekend!</p>
    <br/>
    <p>We have to do homeworks.</p>
  </body>
  <sign/>
</message>
```

Ogni coppia di tag aperto/chiuso identifica un *elemento*: e.g. elemento *heading*.

Il testo contenuto tra i due tag è detto *valore dell'elemento*: e.g. “Reminder”.

Quando un tag non contiene dati può essere aperto e chiuso in *forma abbreviata*: e.g. `<sign/>`, indica l'assenza di firma.

Un elemento *annidato*, o sotto-elemento, è un elemento che viene aperto/chiuso all'interno di un altro elemento (detto genitore): e.g. `<body><p></p></body>`  
NON `<body><p></body></p>`

XML è *case-sensitive*: `<body>` non è `<BODY>`.

# XML

## Le regole di sintassi

Modifichiamo leggermente il documento XML precedente:

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<message from="Veronika" to="Mark">  
  <heading>Reminder</heading>  
  <body>  
    <p num="1">Don't forget  
      me <i>this</i> weekend!</p>  
    <br/>  
    <p num="2">We have to do  
      homeworks.</p>  
  </body>  
<!-- the end of the message-->  
</message>
```

Un elemento può contenere altri elementi, e.g. `<body>` contiene solo elementi `<p>`, o del testo libero, e.g. `<heading>`, oppure può contenere entrambi, e.g. `<p num="1">` contiene del testo e all'interno del testo contiene un altro elemento `<i>` aperto/chiuso.

Alcune informazioni possono essere espresse come *attributi* di un elemento, piuttosto che come altri elementi.  
É una scelta del codificatore.

Un attributo è composto dal suo *nome*, e.g. `@from`, e dal suo *valore*, e.g. "Veronika".

È possibile inserire *commenti*, testo non processabile da programmi, all'interno di `<!-- -->`.

Un file XML marcato correttamente, senza errori di sintassi, si dice *ben formato*.

# XML

## Qualche esempio

Il nome dell'elemento radice rispecchia l'entità che si vuole descrivere, in questo caso un articolo.

Il primo livello di sottoelementi rappresenta la macrostruttura dell'articolo, ovvero i paragrafi.

Il secondo livello di sottoelementi, annidati nei paragrafi, rappresentano una ulteriore specificazione degli elementi parenti: all'interno di un paragrafo può esserci del testo o un'immagine.

Si noti la ripetitività degli stessi elementi: XML è un linguaggio volutamente ridondante.

Gli attributi vengono utilizzati con finalità diverse: talvolta contribuiscono ad identificare un dato elemento (@titolo), a definirne la tipologia (@tipo) oppure per inserire dei link ad oggetti multimediali

```
<?xml version="1.0" encoding="UTF-8"?>
<articolo titolo="Introduzione all'XML">
  <paragrafo tipo="par_articolo" titolo="XML: la storia">
    <texto>XML (eXtensible Markup Language) è un linguaggio di markup definito dal W3C (World Wide Web Consortium) nel 1998 utilizzato per descrivere dati in modo semplice e strutturato...
    </texto>
    <immagine file="esempio1.jpg">
      </immagine>
    </paragrafo>
    <paragrafo tipo="par_articolo" titolo="La sintassi">
      <texto>In questa sezione daremo una breve descrizione della sintassi XML senza peraltro la pretesa di essere esaustivi...
      </texto>
    </paragrafo>
    <paragrafo tipo="bibliografia" titolo="Bibliografia">
      <texto tipo="riferimento_bibliografico">
        Erik T. Ray, Learning XML, O'Reilly, 2001
      </texto> ...
    </paragrafo>
  </articolo>
```

# XML

## Qualche esempio

```
<?xml version="1.0"?>
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications with XML.</description>
  </book>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-12-16</publish_date>
    <description>A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.</description>
  </book>
```

```
<book id="bk103">
  <author>Corets, Eva</author>
  <title>Maeve Ascendant</title>
  <genre>Fantasy</genre>
  <price>5.95</price>
  <publish_date>2000-11-17</publish_date>
  <description>After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.</description>
</book>
...
</catalog>
```

---

**N.B.** Rispetto all'esempio precedente, questo file mostra una struttura uniforme reiterata. Non ci sono variazioni (come la presenza o meno di immagini) o differenze di tipologie (paragrafi dell'articolo o di bibliografia). Quando si ha a che fare con informazioni di questo tipo si parla di *dati strutturati*. Nel caso di testi letterari parliamo di dati *semi-strutturati*.

# XML

## La semantica

XML in quanto metalinguaggio, non da alcuna indicazione formale sulla semantica dei suoi elementi (quali elementi, il loro nome e il loro impiego): fornisce solo **regole sintattiche** per esprimerla. È un metodo per dire qualcosa su un testo, ma da solo non dice - e non fa - nulla.

Per dire qualcosa che sia comprensibile e interscambiabile tra più utenti (interoperabilità semantica) deve dotarsi di un **modello**.

La semantica è infatti delegata ad un documento esterno che funge da **vocabolario**, una **DTD** o uno **Schema XML**, in cui si definiscono: i nomi degli elementi, degli attributi, la loro collocazione nell'albero del documento XML, la ripetibilità, il tipo di dato che possono contenere...

Ovvero, si danno **le istruzioni per descrivere i dati**.

L'uso di un vocabolario condiviso è fondamentale per la rappresentazione di informazioni simili: consente di scambiarle tra più utenti senza perdere informazioni preziose e per questo vengono definiti **standard**.

# XML DTD e XML Schema

Le DTD (Document Type Definition) e gli XML Schema forniscono tutte le specifiche necessarie per la marcatura di una determinata tipologia di informazioni.

Rispetto alle DTD, **XMLSchema** fornisce istruzioni più dettagliate sulla tipologia di dati che un elemento XML può contenere (una stringa di caratteri, un numero intero, un URL...), perciò in alcuni casi può essere preferibile all'uso di una DTD.

Inoltre, mentre le DTD hanno una loro propria notazione, gli schemi XML sono scritti...in XML.

Le specifiche possono essere dichiarate direttamente all'interno del file XML oppure (più correttamente), in un file esterno.

Le regole di uno Schema XML vengono dichiarate in un file .xsd.

Una DTD viene definita in un file .dtd.

# XML

## Un esempio

Riprendiamo l'esempio precedente semplificato:

```
<message>
  <to>Mark</to>
  <from>Veronika</from>
  <heading>Tomorrow</heading>
  <body>Have a nice day!</body>
</message>
```

---

### DTD

```
<!DOCTYPE message
[ <!ELEMENT message (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

### Schema XML

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="message">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

# XML

## Dichiarare la DTD

Una volta definita la DTD, questa va dichiarata nel prologo di ogni file XML che intende rifarsi a quel vocabolario.

Un esempio di DTD pubblica, *DocBook*:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "../docbook-xml-4.5/docbookx.dtd">
```

```
<book>
  <title>La stanza di Jacob</title>
  <chapter>...</chapter>
  ...
</book>
```

**!DOCTYPE** introduce la dichiarazione dello schema a cui si conforma il file XML in questione

**book**, l'elemento radice del file XML viene richiamato per dire che è conforme allo schema che segue nella dichiarazione

**PUBLIC** indica che lo schema è open source e online. Se privato o sul proprio pc si utilizza **SYSTEM**

**"-//OASIS..."** è il nome ufficiale della DTD

**"-/docbook...dtd"** è la URL (in questo caso parziale) dove è collocato il file .dtd

# XML

## Dichiarare lo Schema XML

Allo stesso modo, definito uno Schema questo va dichiarato nel documento XML, all'interno dell'elemento radice.

Un file XML che rispetta uno Schema è detto istanza dello schema:

```
<?xml version="1.0" encoding="UTF-8"?>

<book    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:SchemaLocation="docbookx.xsd">
    <title>La stanza di Jacob</title>
    <chapter>...</chapter>
    ...
</book>
```

**xmlns:xsi** specifica il metodo di riferimento al vocabolario

**xsi:SchemaLocation**, indica il nome  
e l'URL dello Schema, ovvero la collocazione del file .xsd

# XML

## Validare un documento XML

Un documento valido è un file XML ben formato (cioè rispetta le regole di sintassi XML) che rispetta le prescrizioni di uno Schema XML o DTD.

Per verificare la correttezza del file XML rispetto alle istruzioni del vocabolario, il documento XML deve essere *validato* tramite uno specifico software, detto validator.

Quando si edita un documento XML bisogna tenere in conto quindi due tipologie di errori: errori di malformazione (sintassi) ed errori di validità (schema).

e.g. riprendiamo l'esempio precedente:

```
<book>
  <chapter>...</chapter>
  <title>La stanza di Jacob</title>
  ...
<book>
```

In questo esempio ci sono due errori:  
uno di sintassi e uno di validità.  
Quali sono?

# XML

## I namespace

Uno Schema o DTD rappresentano un certo numero predefinito di tag utilizzabili per *descrivere* i dati in un file XML.

Quando è necessario ampliare il potenziale espressivo (inserire più elementi o attributi, cambiare l'ordine di presentazione delle informazioni ecc...) è necessario utilizzare contemporaneamente più Schemi/DTD.

e.g. Voglio rappresentare le informazioni riguardanti i titoli nobiliari di personaggi famosi citati in un testo.

```
<book    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
        xsi:SchemaLocation="docbookx.xsd">  
  
    <title>Little <title>Lord</title> Fauntleroy</title>  
    <author> Frances Hodgson Burnett </author>  
    <year>1885</year>  
</book>
```

no!

Utilizzando un solo schema non è possibile descrivere tutte le informazioni, perché `<title>` ha un significato univoco.

Devo utilizzare due schemi contemporaneamente, uno per la descrizione dei libri e uno per la descrizione dei personaggi famosi e i loro titoli nobiliari.

# XML

## I namespace

Per poter utilizzare due Schemi/DTD contemporaneamente in un documento XML è necessario specificarli per evitare ambiguità.

e.g. In uno schema <title> rappresenta i titoli dei libri, in un altro schema lo stesso elemento può indicare i titoli nobiliari di un personaggio.

Per ovviare a questo problema si utilizza il meccanismo dei **namespace**, ovvero la dichiarazione nel file XML dell'URI (URL) degli schemi ai quali ci si sta riferendo, eventualmente assegnando un prefisso in caso di omonimie.

```
<lib:book xmlns:lib="http://www.dominio.it/xml/libri"  
         xmlns:pers="http://www.dominio.it/xml/personaggi" >  
  
<lib:title>Little <pers:title>Lord</pers:title> Fauntleroy</lib:title>  
<lib:author> Frances Hodgson Burnett </lib:author>  
<lib:year>1885</lib:year>  
</lib:book>
```

ok!

Il meccanismo più efficiente prevede la dichiarazione nell'elemento radice di tutti gli schemi utilizzati nel file XML, attribuendo a ciascuno un *prefisso*, che viene utilizzato ogni volta per identificare univocamente la provenienza dell'elemento.

# XML

## I tipi di dati

XML Schema (XSD) introduce una distinzione tra i tipi di dati rappresentabili:

// complex type: sono gli elementi che possono contenere a loro volta altri elementi o attributi

// simple type: sono elementi che non possono contenere altri elementi, ma un solo tipo di dato semplice.

XSD provvede un set di **19 data type primitivi** (anyURI, base64Binary, boolean, date, dateTime, decimal, double, duration, float, hexBinary, gDay, gMonth, gMonthDay, gYear, gYearMonth, NOTATION, QName, string, and time)

Questi simple type possono subire delle restrizioni per creare nuovi tipi di dati.  
e.g. una password è un dato di tipo 'xs:string' di lunghezza tra i 6 e i 10 caratteri.

# XML

## Caratteri non permessi

Ci sono alcuni caratteri speciali o simboli che non possono essere direttamente editati all'interno di un file XML e vanno sostituiti da una apposita notazione.

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

# XML

## Esercizi ed esempi

Immaginiamo di dover codificare il seguente testo in XML, ideando prima uno schema (in linguaggio naturale) adatto a descrivere:

- // i riferimenti bibliografici (libro, titolo, curatela, editore, data di pubblicazione)
- // le parti significative del testo (paragrafo, sentenza)
- // elementi di contenuto (luogo, anno, ruolo, orientamento politico)

“La resistenza taciuta. Dodici vite di partigiane piemontesi”,  
a cura di Bruzzone e Farina. Ed. La Pietra, 1975

Appartengo a una famiglia di antifascisti di Piedimulera, un piccolo paese non lontano da Domodossola, dove sono nata nel 1921. Mio padre e mia madre si sono sempre rifiutati di iscriversi al partito fascista....mio padre, dopo il 1929, è stato privato del posto di lavoro.....Forse mio padre, che non era molto politicizzato, piuttosto che andare incontro a tanti guai avrebbe anche ceduto e preso la tessera. Ma c'era mia madre. E mia madre è sempre stata una socialista, proveniente da una famiglia di socialisti: un fratello fuoriuscito in Francia, un altro morto molto giovane perché picchiato dai fascisti.....

Da noi l'elemento forte della famiglia è sempre stata la mamma.....Era una donna molto umile, una donna che, guardandola, nessuno avrebbe detto che avesse tanta personalità.

# XML

## Esercizi ed esempi

Un possibile schema può essere composto dai seguenti elementi:

libro (1)

- titolo (1 o più)
- curatela (1 o più)
- editore (1)
- dataPub (1)
- testo (1)
  - paragrafo (1 o più)
    - sentenza (1 o più)
      - luogo (0 o più)
      - anno (0 o più)
      - ruolo (0 o più)
      - orientPol (0 o più)

**N.B.** Tra parentesi è indicata la ripetibilità dell'elemento.

I nomi degli elementi, *autodescrittivi*, si scrivono sempre in minuscolo e al singolare.

Quando i nomi sono composti si può scegliere una notazione (coerente) per la denominazione. In questo esempio utilizziamo una *notazione camelCase*: tutte le parole sono unite senza spazi, si inizia con una lettera minuscola e ogni prima lettera di parola è maiuscola.

# XML

## Esercizi ed esempi

Iniziamo dall'elemento radice e dal primo livello di informazioni:

<libro>

<titolo>La resistenza taciuta. Dodici vite di partigiane piemontesi</titolo>,

<curatela>a cura di Bruzzone e Farina</curatela>.

<editore>Ed. La Pietra</editore>,

<dataPub>1975</dataPub>

<texto>

Appartengo a una famiglia di antifascisti di Piedimulera, un piccolo paese non lontano da Domodossola, dove sono nata nel 1921. Mio padre e mia madre si sono sempre rifiutati di iscriversi al partito fascista....mio padre, dopo il 1929, è stato privato del posto di lavoro.....Forse mio padre, che non era molto politicizzato, piuttosto che andare incontro a tanti guai avrebbe anche ceduto e preso la tessera. Ma c'era mia madre. E mia madre è sempre stata una socialista, proveniente da una famiglia di socialisti: un fratello fuoriuscito in Francia, un altro morto molto giovane perché picchiato dai fascisti.....

Da noi l'elemento forte della famiglia è sempre stata la mamma.....Era una donna molto umile, una donna che, guardandola, nessuno avrebbe detto che avesse tanta personalità.

</texto>

</libro>

# XML

## Esercizi ed esempi

Proseguiamo inserendo gli elementi di paragrafazione e divisione delle frasi:

<libro>

<titolo>La resistenza taciuta. Dodici vite di partigiane piemontesi</titolo>,

<curatela>a cura di Bruzzone e Farina</curatela>.

<editore>Ed. La Pietra</editore>,

<dataPub>1975</dataPub>

<texto>

<paragrafo>

<sentenza>

Appartengo a una famiglia di antifascisti di Piedimulera, un piccolo paese non lontano da Domodossola, dove sono nata nel 1921. </sentenza>

<sentenza>Mio padre e mia madre si sono sempre rifiutati di iscriversi al partito fascista....</sentenza>

<sentenza>mio padre, dopo il 1929, è stato privato del posto di lavoro.....</sentenza>

<sentenza>Forse mio padre, che non era molto politicizzato, piuttosto che andare incontro a tanti guai avrebbe anche ceduto e preso la tessera. </sentenza>

<sentenza>Ma c'era mia madre. </sentenza>

<sentenza>E mia madre è sempre stata una socialista, proveniente da una famiglia di socialisti: un fratello fuoriuscito in Francia, un altro morto molto giovane perché picchiato dai fascisti.....</sentenza>

</paragrafo>

<paragrafo>

<sentenza>Da noi l'elemento forte della famiglia è sempre stata la mamma....</sentenza>

<sentenza>Era una donna molto umile, una donna che, guardandola, nessuno avrebbe detto che avesse tanta personalità.</sentenza></paragrafo>

</testo>

</libro>

# XML

## Esercizi ed esempi

Infine inseriamo gli elementi riguardanti il contenuto del testo:

```
<libro>
<titolo>La resistenza taciuta. Dodici vite di partigiane piemontesi</titolo>,
<curatela>a cura di Bruzzone e Farina</curatela>. <editore>Ed. La Pietra</editore>, <dataPub>1975</dataPub>
<testo>
  <paragrafo>
    <sentenza>Appartengo a una famiglia di <orientPol>antifascisti</orientPol> di <luogo>Piedimulera</luogo>, un piccolo paese non lontano da <luogo>Domodossola</luogo>, dove sono nata nel <anno>1921</anno>. </sentenza><sentenza>- Mio <ruolo>padre</ruolo> e mia <ruolo>madre</ruolo> si sono sempre rifiutati di iscriversi al partito <orientPol>fascista</orientPol>....</sentenza><sentenza>mio <ruolo>padre</ruolo>, dopo il <anno>1929</anno>, è stato privato del posto di lavoro.....</sentenza><sentenza>Forse mio <ruolo>padre</ruolo>, che non era molto politicizzato, piuttosto che andare incontro a tanti guai avrebbe anche ceduto e preso la tessera.</sentenza><sentenza> Ma c'era mia <ruolo>madre</ruolo>.</sentenza><sentenza> E mia <ruolo>madre</ruolo> è sempre stata una <orientPol>socialista</orientPol>, proveniente da una famiglia di socialisti: un fratello fuoriuscito in Francia, un altro morto molto giovane perché picchiato dai <orientPol>fascisti</orientPol>.....</sentenza>
  </paragrafo>
  <paragrafo>
    <sentenza>Da noi l'elemento forte della famiglia è sempre stata la <ruolo>mamma</ruolo>....</sentenza><sentenza>Era una donna molto umile, una donna che, guardandola, nessuno avrebbe detto che avesse tanta personalità.</sentenza>
  </paragrafo>
</testo>
</libro>
```

# XML

## Esercizi ed esempi

Un altro possibile schema, più articolato:

libro (1) @titolo, @curatela, @editore, @dataPub

- testo (1)

- paragrafo (1 o più) @num

- sentenza (1 o più) @num

- luogo (0 o più)

- anno (0 o più) @tipo

- ruolo (0 o più)

- orientPol (0 o più) @tipo

**N.B.** Indicati con @ abbiamo gli attributi. Gli elementi del precedente schema possono essere considerati “aggettivi” dell’elemento radice <libro>.

Possiamo essere più specifici e indicare se questi attributi sono *obbligatori* o *opzionali*:

@titolo, @curatela, @editore, @dataPub sono informazioni obbligatorie per identificare il libro di cui parliamo; @num e @tipo possono essere opzionali.

Possiamo anche definire a priori i *valori* che questi attributi devono avere: orientPol [@tipo] potrà essere “antifascista” o “fascista”.

# XML

## Esercizi ed esempi

Una possibile codifica sulla base del precedente schema può essere:

```
<libro titolo="La resistenza taciuta. Dodici vite di partigiane piemontesi" curatela="a cura di Bruzzone e Farina" editore="Ed.  
La Pietra" dataPub="1975">  
<texto>  
  <paragrafo num="1">  
    <sentenza num="1">Appartengo a una famiglia di <orientPol tipo="antifascista">antifascisti</orientPol> di <luogo>Piedi-  
mulera</luogo>, un piccolo paese non lontano da <luogo>Domodossola</luogo>, dove sono nata nel <anno tipo="nasci-  
ta">1921</anno>. </sentenza><sentenza num="2" >Mio <ruolo>padre</ruolo> e mia <ruolo>madre</ruolo> si sono sem-  
pre rifiutati di iscriversi al partito <orientPol tipo="fascista" >fascista</orientPol>....</sentenza><sentenza num="3" >mio  
<ruolo>padre</ruolo>, dopo il <anno tipo="licenziamento">1929</anno>, è stato privato del posto di lavoro.....</senten-  
za><sentenza num="4" >Forse mio <ruolo>padre</ruolo>, che non era molto politicizzato, piuttosto che andare incontro  
a tanti guai avrebbe anche ceduto e preso la tessera.</sentenza><sentenza num="5" > Ma c'era mia <ruolo>madre</  
ruolo>.</sentenza><sentenza num="6" > E mia <ruolo>madre</ruolo> è sempre stata una <orientPol tipo="antifascista"  
>socialista</orientPol>, proveniente da una famiglia di socialisti: un fratello fuoriuscito in Francia, un altro morto molto  
giovane perché picchiato dai <orientPol tipo="fascista" >fascisti</orientPol>....</sentenza>  
  </paragrafo>  
  <paragrafo num="2">  
    <sentenza num="7" >Da noi l'elemento forte della famiglia è sempre stata la <ruolo>mamma</ruolo>....</sentenza>  
    <sentenza num="8" >Era una donna molto umile, una donna che, guardandola, nessuno avrebbe detto che avesse tanta  
personalità.</sentenza>  
  </paragrafo>  
</texto>  
</libro>
```

# Bibliografia

## Le specifiche ufficiali del W3C

*Documentazione sulle tecnologie legate ad XML*

*XML Technology*, <http://www.w3.org/standards/xml/>

*XML Current Status*, [http://www.w3.org/standards/techs/xml#w3c\\_all](http://www.w3.org/standards/techs/xml#w3c_all)

## *Specifiche tecniche di XML*

*Extensible Markup Language (XML) 1.1. (Second Edition)*, W3C Recommendation 16 August 2006, edited in place 29 September 2006

<http://www.w3.org/TR/2006/REC-xml11-20060816/>

*Documentazione di XSD (XML Schema) con tutorial e specifiche tecniche Schema*, <http://www.w3.org/standards/xml/schema>

## *Specifiche tecniche di HTML*

*HTML: The Markup Language (an HTML language reference)*,  
<http://www.w3.org/TR/html-markup/>

# Bibliografia

## I tutorial online

*XML Tutorial*, W3School, <http://www.w3schools.com/xml/>

*XML Introduction*, Mozilla Developer Network,  
[https://developer.mozilla.org/en-US/docs/XML\\_Introduction](https://developer.mozilla.org/en-US/docs/XML_Introduction)

*HTML Training*, W3C Wiki,  
<http://www.w3.org/community/webed/wiki/HTML/Training>

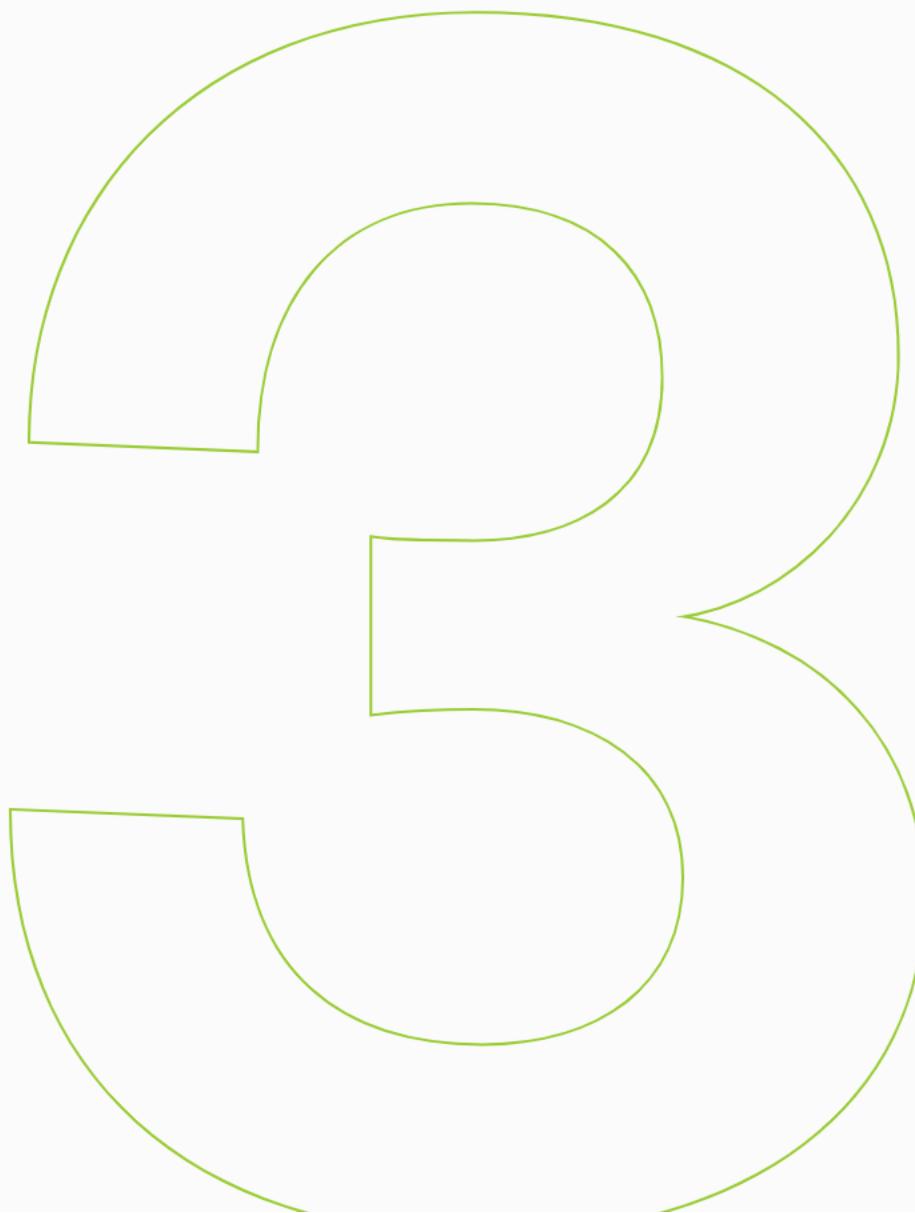
## Introduzione ad XML per la codifica dei testi

*A gentle Introduction to XML*, Text Encoding Initiative,  
<http://www.tei-c.org/release/doc/tei-p5-doc/en/html/SG.html>

*Introduction to XML for Text*, K. S. Hawkins  
<http://www.ultraslavonic.info/intro-to-xml/>

# Tecnologie XML

## Lezione 3



# Gli utilizzi di XML

## I pregi di XML

Le doti fondamentali di XML (flessibilità semantica, portabilità del formato, semplicità sintattica...) lo hanno reso uno strumento diffuso in numerosi settori disciplinari e scientifici.

XML viene infatti utilizzato per lo **storage a lungo termine di dati**, in file di testo piano, separati dalle istruzioni per la loro visualizzazione (HTML) e dalle applicazioni necessarie per la loro manipolazione.

Questo rende i dati XML portabili e indipendenti da qualsiasi hardware e software, agevolando significativamente lo **scambio di informazioni** tra applicazioni eterogenee (e quindi tra comunità).

e.g. le biblioteche, gli archivi, i musei conservano i propri dati bibliografici o catalografici in formato xml.  
La **preservazione e valorizzazione** di queste preziose informazioni è oggetto di studio di numerose discipline interdisciplinari (library & information science, archivistica informatica, umanistica informatica...).  
**Ogni comunità ha predisposto standard per la descrizione delle proprie risorse** (dai fondi archivistici alle fotografie, le opere d'arte, le monografie...) che si basano sulla sintassi XML.

# Gli utilizzi di XML

## I dati in XML

Come già detto, XML è un utile strumento per la **descrizione di dati, o metadati**, le cui applicazioni variano a seconda del contesto.

Abbiamo già visto come un catalogo di riferimenti bibliografici abbia una struttura molto uniforme (stessi elementi ripetuti senza significative variazioni dello schema), mentre un testo letterario presenta notevoli difficoltà di “incassellamento” delle informazioni strutturali.

In questi casi parliamo di dati strutturati e dati semi-strutturati.

**Diverse tipologie di dati** sono utili in contesti diversi e pertanto subiscono **trattamenti diversi**, a partire dai modelli concettuali per la loro organizzazione, fino alle strategie per la loro preservazione, condivisione e interrogazione.

# Gli utilizzi di XML

## Preservare i dati in XML

Per preservazione si intende una serie di pratiche e attività finalizzate alla garanzia di accesso ai dati sul medio o lungo periodo.

L'*obsolescenza digitale* impone un continuo aggiornamento:

// delle **infrastrutture** (policies, procedure e obiettivi, personale addetto);

// degli strumenti **hardware e software** (refresh, backup e migrazioni);

// una continua revisione delle pratiche di **management** dei dati durante il loro intero ciclo di vita (data curation).

XML permette di preservare senza perdita di informazioni sia i dati, sia gli schemi con i quali vengono organizzati i dati, anch'essi in XML.

# Gli utilizzi di XML

## Condividere i dati in XML

Una buona pratica che si va instaurando in numerose comunità è la tendenza a rendere disponibili i propri dati in **formati diversi**, tra cui XML per condividerli agevolmente con altre comunità.

e.g. Un'istituzione decide di gestire il ciclo di vita dei propri dati ricorrendo a software e linguaggi proprietari. Ogni altra organizzazione che voglia usufruire di quei dati dovrà avere la licenza del software in questione. La casa produttrice del software decide di non aggiornare le proprie applicazioni e i dati conservati dalle istituzioni dovranno ricorrere a versioni obsolete di un software che non viene più aggiornato, incontrando difficoltà sia nella gestione che nella condivisione.

Questo procedimento richiede un dispendio di risorse economiche e di tempi notevole, che non tutte le comunità sono in grado di sostenere. Alcune di queste hanno allora optato per la gestione di dati in XML nativi, rendendolo uno standard de facto.

# Gli utilizzi di XML

## Interrogare i dati in XML

Per poter maneggiare quantità ingenti di dati in XML è necessario un sistema di storage, gestione e interrogazione dei dati, ovvero un **database management system**.

Esistono diverse *tipologie di database* (archivi di dati) che possono contenere dati in XML. A seconda della natura dei dati (strutturati o semi-strutturati), degli obiettivi con cui si conservano le informazioni (gestire informazioni complesse come testi letterari o minimali come una rubrica di contatti), è possibile ottenere interrogazioni e risultati diversi.

Le tipologie di db predominanti (anche se non le uniche) sono due:

- // i database relazionali, basati su linguaggio SQL e organizzati in tavole
- // i **database XML nativi**, noSQL, ideati per la gestione di dati in XML anche eterogenei (strutturati o semi-strutturati).

# eXist-db

## Un database XML nativo

eXist-db è un database XML nativo open-source, utilizzato per la creazione, lo storage, l'interrogazione e la pubblicazione sul web di collezioni di file XML.

Consente di **archiviare collezioni di file XML** organizzati come in un normale File System (l'organizzazione a cartelle di un PC).

Dispone di un editor integrato per la creazione di documenti in vari formati, supporta i linguaggi di **interrogazione e manipolazione di dati XML** (come xQuery e XSLT) e fornisce un'ambiente per la **realizzazione di applicazioni web** basate sui documenti XML che conserva.

### QUINDI

In un unico ambiente abbiamo a disposizione tutti gli strumenti, le tecnologie, e le fasi del workflow (uno dei possibili) per la pubblicazione di documenti XML.

(eXist-db, <http://exist-db.org/exist/apps/homepage/index.html>)

# Pubblicare sul web documenti XML

## Come funziona il web

Per poter pubblicare sul web qualsiasi documento è necessario disporre di un'adeguata **infrastruttura** che supporti l'intero (o parte del) workflow di preparazione, pubblicazione e gestione dei propri dati.

Il web si basa su un'architettura client/server:

- // il client (il browser sul nostro pc) richiede una risorsa al server (un URL)
- // il server (server http - per semplicità, il sito web) restituisce la risorsa richiesta (una pagina HTML, un'immagine, un video ecc...)



# Pubblicare sul web documenti XML

## Un possibile workflow (semplificato)

Come detto, XML non è pensato per la visualizzazione dei dati direttamente su browser ma solo per la descrizione di dati. Per poter essere fruito sul web un documento XML (o una collezione di file XML) deve essere manipolato mediante tecnologie diverse:

- // per l'adesione ad uno standard di descrizione: XMLSchema, RelaxNG, DTD...
- // per la trasformazione da XML a HTML: XSLT;
- // per l'interrogazione dei dati: XQuery;
- // per l'impaginazione e la veste grafica: CSS;
- // per la dinamicità dei contenuti: Javascript.

La **separazione della logica dalla visualizzazione** è un aspetto progettuale fondamentale.

# Pubblicare sul web documenti XML

## Organizzare la collezione XML

La prima fase del workflow di pubblicazione prevede la creazione e l'organizzazione della collezione di file XML.

e.g. vogliamo pubblicare l'opera omnia di Shakespeare.

Raccoglieremo i testi che intendiamo codificare e procederemo nella scelta di uno Schema XML che consente di descrivere gli aspetti testuali di rilievo: **TEI (Text Encoding Initiative)** è lo standard ideato per la codifica di testi letterari in ambito accademico.

Procederemo alla **marcatura** dei file secondo lo schema prescelto (verosimilmente creando un file XML per ogni opera), al **caricamento sul db** e all'**organizzazione** in cartelle secondo un criterio (cronologico, tipologia di testo...).



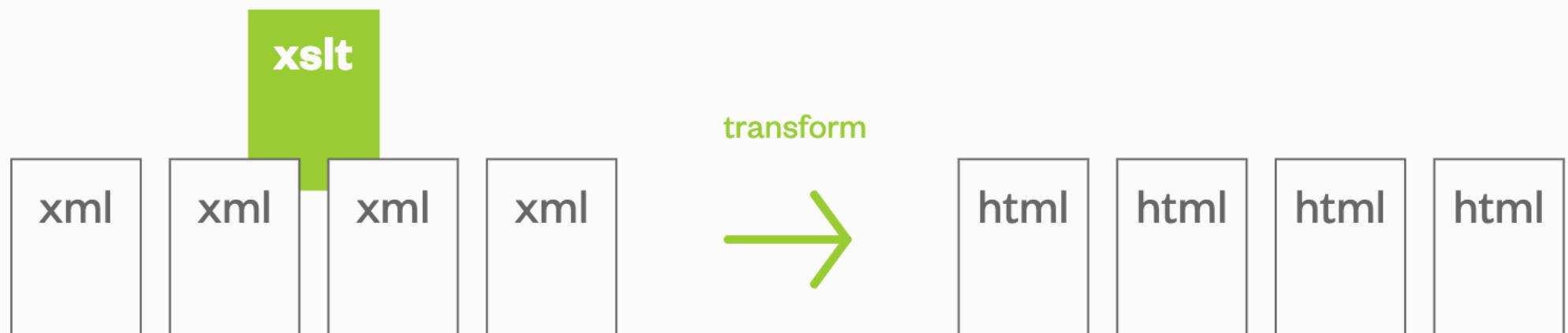
# Pubblicare sul web documenti XML

## Trasformare i file XML in HTML

Una volta organizzata la collezione di file XML aventi in comune uno Schema e quindi una struttura sostanzialmente simile, è possibile procedere alla loro trasformazione in pagine HTML.

La trasformazione di un file XML in un file HTML avviene mediante la creazione di un **foglio di stile .xslt**, un file in cui si danno le istruzioni per la creazione di una pagina HTML a partire dagli elementi di uno o più file XML.

Il linguaggio per la trasformazione è **XSLT (eXtensible Stylesheet Language Transformation)**; il file (o i file) generato è in formato .xsl.



# Pubblicare sul web documenti XML

## Un esempio di XSLT

### XML

```
<?xml version="1.0"?>
<?xml-stylesheet href="shakespeare.xsl"
                  type="text/xsl"?>

<opera>
  <title>Sonnet 1</title>
  <text>FROM fairest creatures we desire incre-
    ase,That thereby beauty's rose might never die,
  But as the riper should by time decease, His ten-
    der heir might bear his memory: ...</text>
  <author>W. Shakespeare</author>
</opera>
```

### XSLT

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="opera">
    <html>
      <head><title><xsl:value-of select="title"/></title></head>
      <body><xsl:apply-templates/></body>
    </html>
  </xsl:template>
  <xsl:template match="title">
    <h1> <xsl:apply-templates/> </h1>
  </xsl:template>
  <xsl:template match="text">
    <div> <xsl:apply-templates/> </div>
  </xsl:template>
  <xsl:template match="author">
    <hr/> <i><xsl:apply-templates/> </i>
  </xsl:template>
</xsl:stylesheet>
```

# Pubbicare sul web documenti XML

## Un esempio di XSLT

### HTML

```
<html>
<head>
  <title>Sonnet 1</title>
</head>
<body>
  <h1>Sonnet 1</h1>
  <div> FROM fairest creatures we desire increase,
    That thereby beauty's rose might never die,
    But as the riper should by time decease,
    His tender heir might bear his memory: ...
  <hr>
  <i>W. Shakespeare</i>
</body>
</html>
```



### La pagina web in un browser

#### Sonnet 1

FROM fairest creatures we desire increase,  
That thereby beauty's rose might never die,  
But as the riper should by time decease,  
His tender heir might bear his memory: ...

---

*W. Shakespeare*

# Pubblicare sul web documenti XML

## Interrogare i dati

Oltre alla riproduzione dei file XML sottoforma di pagine HTML, possiamo interrogare i nostri dati tramite un linguaggio apposito per dati XML, **XQuery**.

e.g. vogliamo creare l'indice delle opere di Shakespeare.

Per ottenerlo dovremo interrogare contemporaneamente tutti i file che abbiamo organizzato in una cartella specifica (ad esempio, la cartella /db/shakespeare/opere) ed estrarre da ognuna il valore dell'elemento <title>. Avremo una semplice query del tipo:

### Query

```
xquery version "3.0";  
collection("/db/shakespeare/opere")/opera/title
```



### Risultato

```
<title>A Midsummer Night's Dream</title>  
<title>Julius Caesar</title>  
<title>Sonnets</title> ...
```

In questo caso molto semplice, la query non è altro che un **percorso**, prima per arrivare alla cartella contenente i file di interesse e da questa, seguendo l'albero del documento XML, fino all'elemento che contiene l'informazione richiesta. Per ottenere queste informazioni xQuery utilizza **XPath**, una sintassi ideata apposta per “percorrere” la struttura di un file XML.

# Pubblicare sul web documenti XML

## Visualizzare i dati

Terminata la parte di logica applicativa per la pubblicazione dei documenti XML, ci si può concentrare sugli **aspetti di visualizzazione**: la grafica e il comportamento degli elementi della pagina (le animazioni e le interazioni con l'utente).

Gli aspetti grafici, genericamente “statici”, come la grandezza del font, il colore di sfondo, la grandezza delle immagini ecc... vengono gestiti da un foglio di stile **.css**, questa volta scritto con **CSS (Cascading Style Sheets)**, il linguaggio apposito per fornire le istruzioni di tipo grafico ad una pagina HTML.

### CSS vs XSLT

I fogli di stile .xslt danno istruzioni per trasformare **XML** in un qualsiasi altro formato (e.g. HTML). Possono gestire anche informazioni sugli aspetti grafici del risultato finale.

I fogli di stile .css danno istruzioni all'**HTML** per la sua visualizzazione su un browser. Non consentono di modificare la struttura del file di partenza.

# Pubblicare sul web documenti XML

## Un esempio di CSS

Riprendendo l'esempio precedente di pagina HTML, applichiamo gli stili agli elementi della pagina. Nell'elemento `<head>` della pagina HTML inseriremo il riferimento al file `.css` contenente le istruzioni per la grafica.

### HTML

```
<html>
  <head>
    <title>Sonnet 1</title>
    <link href="style.css" rel="stylesheet"
          type="text/css">
  </head>
  <body>
    <h1>Sonnet 1</h1>
    <div> FROM fairest creatures we de-
        sire increase, That thereby beauty's
        rose might never die, But as the riper
        should by time decease, His tender
        heir might bear his memory: ...</div>
    <hr><i>W. Shakespeare</i>
  </body>
</html>
```



### CSS

```
body {
  background-color: #DODODO;
}

h1 {
  color: #400000;
  font-family: "Times New Roman",
  Times, serif;
}

div {
  font-family: Arial, Helvetica,
  sans-serif;
  margin-left: 10px;
}
```

# Pubbicare sul web documenti XML

## Un esempio di CSS

### La pagina web in un browser

The diagram illustrates the structure of a web page with CSS selectors. A grey rectangular area represents the page content. On the left, the text "Sonnet 1" is labeled "h1". To its right, a block of text is labeled "div". The entire grey area is labeled "body". Below the grey area, the author's name "W. Shakespeare" is written in a smaller font.

**Sonnet 1**

FROM fairest creatures we desire increase,  
That thereby beauty's rose might never die,  
But as the riper should by time decease,  
His tender heir might bear his memory: ...

---

*W. Shakespeare*

body

h1

div

# Pubblicare sul web documenti XML

## Javascript

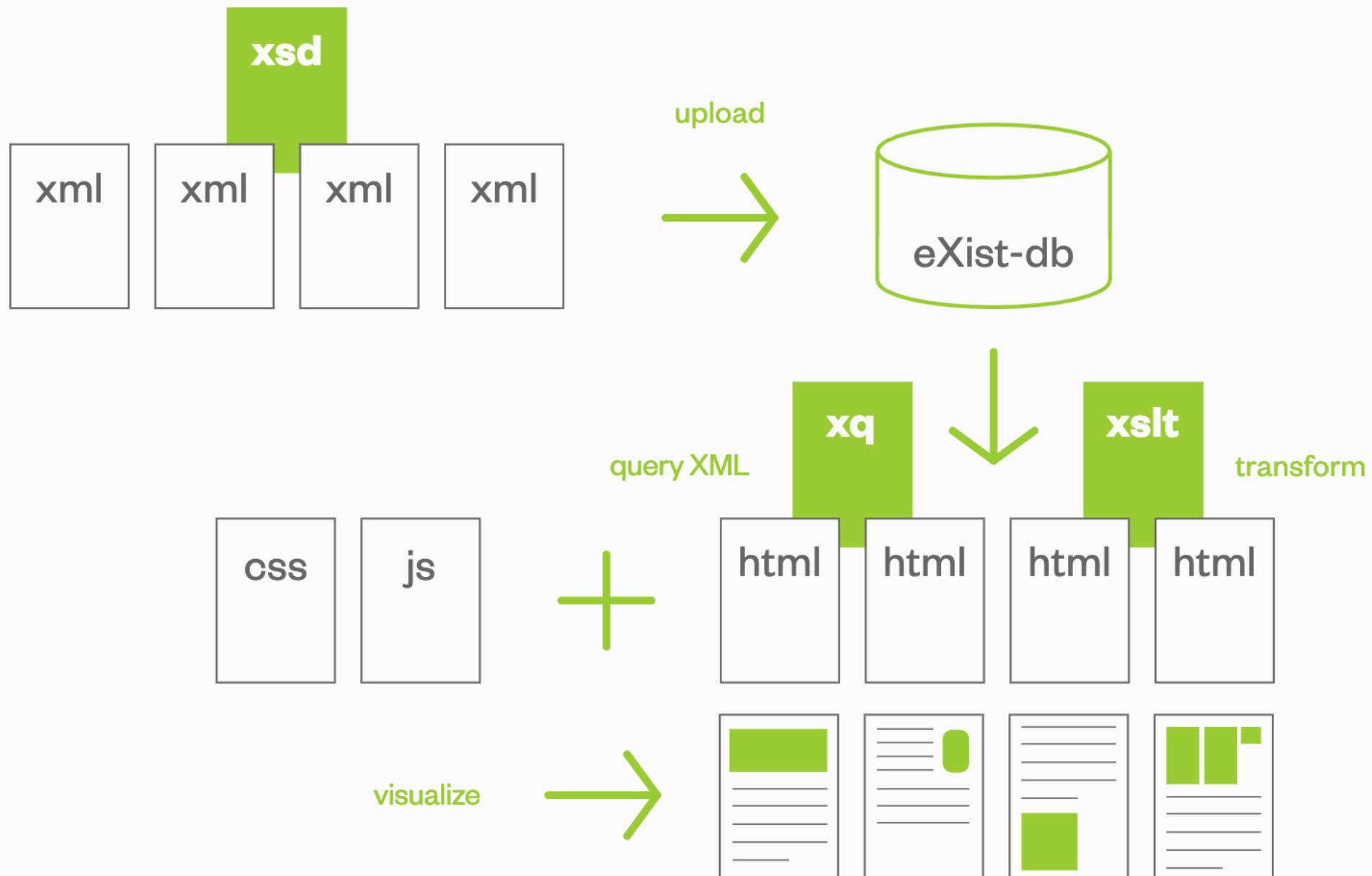
Javascript è un linguaggio di programmazione ideato per la creazione di applicazioni web: permette di eseguire script, istruzioni, che interagiscono con i contenuti di pagine HTML, al fine di ottenere una migliore esperienza dell'utente.

È un linguaggio di scripting diverso dai più comuni linguaggi di programmazione (come C o C++), poiché per funzionare, Javascript deve essere “ospitato” da un altro programma in grado di interpretarlo, ovvero un **browser**.

Come per i file .css, anche le istruzioni di javascript vengono inserite (quale buona pratica) in un file esterno alla pagina HTML, in formato **.js**, e nell'elemento <head> di HTML viene richiamato il file con il già visto meccanismo di link.

Per una guida dettagliata sulle specifiche e gli utilizzi di Javascript si rimanda a: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript)

# Pubblicare sul web documenti XML Il workflow descritto in sintesi



# XML: non solo web

## Gli standard la descrizione di dati

Come detto, XML non nasce per visualizzare i dati sul web, ma per **descriverli, conservarli, condividerli**. Molte delle applicazioni reali di XML non trovano la pubblicazione sul web come sbocco finale.

Numerose comunità utilizzano sistematicamente XML per gestire le proprie informazioni (documenti aventi caratteristiche strutturali comuni tra loro), e hanno adottato un proprio Schema (o DTD) di riferimento per la descrizione dei propri dati. Tra i tanti ricordiamo:

// DC: utilizzato trasversalmente per descrivere risorse digitali.

// EAD ed EAC: standard per la descrizione archivistica di fondi e soggetti produttori; METS, MODS e MAG per il settore biblioteconomico; Scheda F e Scheda OA per la catalogazione museale di fotografie e opere d'arte.

// DocBook: schema per la descrizione strutturale di un documento.

// TEI: schema per la descrizione di un testo letterario.

# XML: non solo web

## Dublin Core

Dublin Core è un set di 15 elementi per la descrizione dei **metadati** di qualsiasi risorsa digitale, accessibile dal web.

**contributor**  
**coverage**  
**creator**  
**date**  
**description**  
**format**  
**identifier**  
**language**  
**publisher**  
**relation**  
**rights**  
**source**  
**subject**  
**title**  
**type**

Il numero dei metadati è volutamente minimale e generico per poter riusare il set di metadati su un'ampia casistica di risorse (documenti, immagini, video, audio...). L'obiettivo di Dublin Core non è infatti l'esaurività della descrizione, ma l'**interoperabilità semantica**, ovvero la possibilità di poter scambiare queste informazioni tra il maggior numero possibile di comunità.

Un simile approccio è molto utile quando le risorse da descrivere non hanno un alto valore storico o artistico, per le quali invece è auspicabile una maggiore espressività nella descrizione ed è necessario preservare un maggior numero di informazioni importanti.

# XML: non solo web

## EAD (Encoded Archival Description)

EAD è uno standard per la descrizione archivistica: tramite questa DTD è possibile codificare le informazioni contenute in un **inventario** cartaceo inseriti la struttura di un archivio. Un esempio parziale di file EAD/XML:

```
<ead>
  <eadheader>
    <eadid>[...]</eadid>
    <filedesc>
      <titlestmt>
        <titleproper>Guide to the Bank of Willows Records,
          <date>1880-1905</date>
        </titleproper>
      </titlestmt>
      <publicationstmt>
        <date>1995</date>
        <publisher>Library of Congress</publisher>
      </publicationstmt>
    </filedesc>
  </eadheader>
  <archdesc level="fonds">
    <did>[...]</did>
    <dsc type="combined">[...]</dsc>
  </archdesc>
</ead>
```

eadheader: contiene le informazioni che identificano il contesto dell'inventario

filedesc: contiene le informazioni bibliografiche dell'inventario (titolo, pubblicazione...)

archdesc: contiene tutte le informazioni sul contenuto, il contesto e il supporto delle unità archivistiche (fondo, serie, fascicolo...), ordinandole gerarchicamente.

# XML: non solo web

## DocBook

DocBook è una DTD utilizzata per la descrizione di documenti (libri o articoli). A differenza di EAD, che si occupa di rappresentare gerarchicamente le informazioni contenute in un inventario, DocBook descrive gli **aspetti strutturali di un documento**, ovvero le parti di cui è composto, non il suo contenuto:

```
<?xml version="1.0" encoding="UTF-8"?>
<book    xml:id="animal-farm"
         xmlns="http://docbook.org/ns/docbook" version="5.0">
  <title>Animal Farm</title>
  <chapter xml:id="chapter_1">
    <title>Chapter 1</title>
    <para>Mr. Jones, of the Manor Farm, had locked the henhouses for
the night, but was too drunk to remember to shut the pop-holes.</para>
    <para>With the ring of light ...</para>
  </chapter>
  <chapter xml:id="chapter_2">
    <title>Chapter 2</title>
    <para>Three nights later old Major ...</para>
  </chapter>
</book>
```

Non da alcuna informazione sul contenuto del testo, né sugli aspetti di visualizzazione finale del documento.

Specifica le componenti strutturali, logiche, del testo, suddividendolo in blocchi concettuali (titoli, capitoli, paragrafi, tavole, immagini ...).

Per queste sue caratteristiche, DocBook è uno schema apprezzato nell'editoria commerciale.

# XML: non solo web

## TEI (Text Encoding Initiative)

Se EAD rappresenta un'unità archivistica tramite il suo inventario e DocBook rappresenta le informazioni strutturali del testo ma non il suo contenuto, la TEI ambisce a descrivere **tutto ciò che fa del testo un oggetto di ricerca**.

Ciò significa che l'oggetto di interesse dello schema sono tutte le informazioni che concernono: il supporto fisico del testo, la struttura logica del testo, il contenuto del testo, le interpretazioni fatte sul testo, la storia, il processo di creazione e il contesto del testo...

TEI nasce infatti per coprire il più vasto range possibile di informazioni utili a descrivere un **testo letterario**, la forma più complessa tra le varie tipologie documentarie viste finora.

# Bibliografia

## Specifiche tecniche di XQuery

*XQuery 1.0: An XML Query Language (Second Edition)*, W3C, <http://www.w3.org/TR/xquery/>

## Specifiche tecniche di XSLT

*XSL Transformations (XSLT), Version 1.0*, W3C, <http://www.w3.org/TR/xslt>

## Documentazione ufficiale su CSS

*Cascading Style Sheets*, W3C, <http://www.w3.org/Style/CSS/>

## Documentazione su Javascript

*Javascript*, <https://developer.mozilla.org/it/docs/Web/JavaScript>

## Database XML nativo

*eXist-db Homepage, con Demo e documentazione*

*eXist-db*, <http://exist-db.org/exist/apps/homepage/index.html>

# Bibliografia

## Schemi di metadati

DCMI, Dublin Core Metadata Initiative, <http://dublincore.org/>

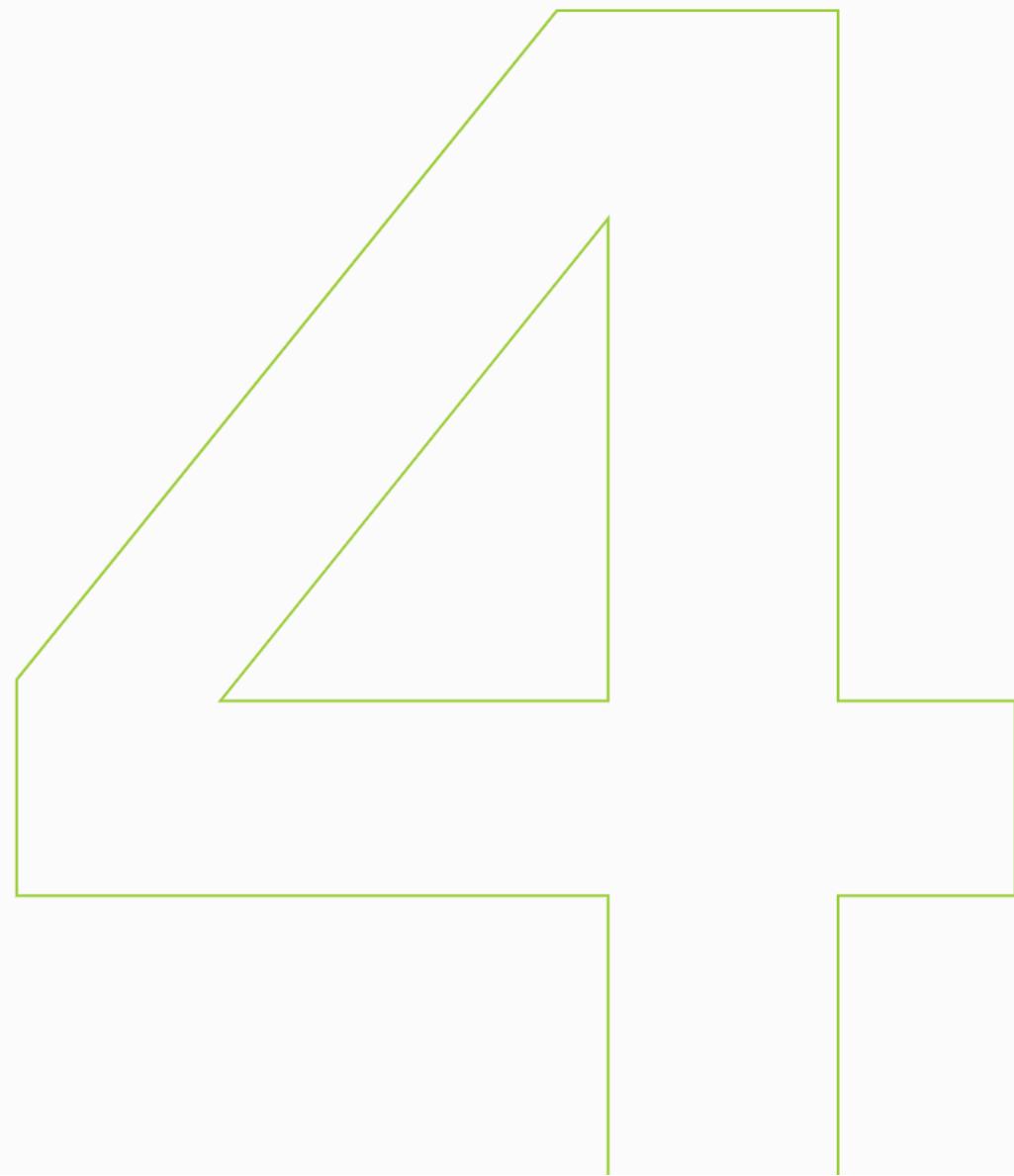
EAD, Encoded Archival Description, <http://www.loc.gov/ead/>

EAC-CPF, Encoded Archival Context, <http://eac.staatsbibliothek-berlin.de/>

DocBook, <http://www.docbook.org/>

# TEI (Text Encoding Initiative)

## Lezione 4



# TEI (Text Encoding Initiative)

## Gli obiettivi

La Text Encoding Initiative nasce nel 1987 con l'obiettivo di creare un framework per la definizione di un linguaggio condiviso tra comunità interessate alla codifica di testi di natura letteraria.

Lo schema TEI infatti fornisce un ampio *schema* per la codifica XML di informazioni eterogenee relative a testi di varia natura letteraria: è personalizzabile ed estendibile tramite l'aggiunta di **moduli**, " librerie" di elementi specialistici che garantiscono un alto livello descrittivo degli elementi di una data tipologia di documento.

Nonostante l'obiettivo principale sia fornire la più ampia libertà nella descrizione del testo, TEI ambisce ad essere uno strumento duttile ma **interoperabile**, per lo scambio di informazioni *tra utenti e tra applicazioni* (e.g. la pubblicazione di edizioni critiche, archivi digitali...), lo scambio di *risorse eterogenee* (e.g. archivi di testi codificati e digitalizzazioni dei testi), la preservazione dei *metadati* (e non solo dei dati, i.e. del testo).

# TEI (Text Encoding Initiative)

## La flessibilità nella descrizione

TEI (P5), è suddivisa in **21 moduli estendibili** - gruppi di elementi e attributi attivabili a seconda delle esigenze - ciascuno dei quali approfondisce un aspetto specialistico del testo che si vuole rappresentare.

analysis	Analysis and Interpretation	msdescription	Manuscript Description
certainty	Certainty and Uncertainty	namesdates	Names, Dates, People, and Places
core	Common Core	nets	Graphs, Networks, and Trees
corpus	Metadata for Language Corpora	spoken	Transcribed Speech
dictionaries	Print Dictionaries	tagdocs	Documentation Elements
drama	Performance Texts	tei	TEI Infrastructure
figures	Tables, Formulae, Figures	textcrit	Text Criticism
gaiji	Character and Glyph Documentation	textstructure	Default Text Structure
header	Common Metadata	transcr	Transcription of Primary Sources
iso-fs	Feature Structures	verse	Verse
linking	Linking, Segmentation, and Alignment		

Data la complessità e la vastità del vocabolario (conta circa 500 elementi), fin dalla nascita della TEI è stata implementata una versione ridotta (un subset), chiamata **TEI Lite**, che permette di adempiere al 90% delle esigenze medie di un editore, snellendo il set di elementi.

# TEI (Text Encoding Initiative) I blocchi concettuali dello Schema

Lo schema è diviso in tre blocchi concettuali:

// **core tag set** gli elementi presenti e obbligatori in ogni documento TEI (inclusi i metadati descrittivi del documento XML/TEI)

// **base tag set** raggruppa i moduli in base alla tipologia di testo - prosa, versi, testo drammatico, trascrizioni di testi orali, dizionari...

// **additional tag set** consiste in gruppi di elementi utili ad assolvere specifiche esigenze di marcatura - i link, l'analisi stilistica, l'analisi dei corpora, la trascrizione delle fonti primarie, l'apparato critico, nomi e date, immagini...

# TEI (Text Encoding Initiative)

## Le customizzazioni

Tra le differenti comunità che utilizzano TEI per codificare **le tipologie di testi oggetto della propria disciplina scientifica**, alcune hanno messo a punto delle personalizzazioni dello Schema per rispondere alle esigenze maggiormente specialistiche di descrizione. Tra queste segnaliamo:

// EpiDoc: una customizzazione del set di elementi TEI per la descrizione di iscrizioni ed epigrafi antiche.

// MEI, Music Encoding Initiative: una personalizzazione dello schema per la descrizione di notazioni musicali

// Character Encoding Initiative: un'estensione della TEI per la codifica di caratteri medievali e pre-moderni in XML.

La TEI Community ha inoltre reso disponibile un software per definire sottinsiemi del set dello Schema ufficiale, **Roma**. <http://www.tei-c.org/Roma/>

# TEI (Text Encoding Initiative)

## Imparare con gli esempi

Data la flessibilità descrittiva dello schema, non è possibile essere esaustivi nella definizione di un documento XML/TEI “base”, adeguato in ogni contesto.

Ogni tipologia di testo da codificare, ogni scelta di impaginazione o di interpretazione dell'editore, richiede uno specifico set di elementi e attributi, in un differente ordine di comparizione e di reiterazione.

Questo tutorial è finalizzato alla comprensione degli **elementi minimi richiesti** nella codifica di un singolo testo in **prosa**, con alcuni approfondimenti tesi ad evidenziare il **ragionamento critico necessario** nell'atto di interpretare e rappresentare un testo.

Per ogni approfondimento, consultare le TEI Guidelines  
<http://www.tei-c.org/Guidelines/>

# La struttura di un file XML/TEI

## TEI l'elemento radice

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>...</teiHeader>
  <text>...</text>
</TEI>
```

**TEI** è l'elemento radice di un file XML/TEI per la rappresentazione di *un singolo testo unitario*.

L'attributo **@xmlns** indica il namespace dello Schema TEI, l'URI pubblico dove reperire lo schema.

Non avendo apposto alcun prefisso (e.g. `xmlns:myScheme="..."`) tutti gli elementi e gli attributi contenuti nel file XML appartengono al namespace di default, i.e. quello dello schema TEI, fatta eccezione per gli attributi con esplicito prefisso *xml*.

In questo elemento è anche possibile segnalare con l'attributo opzionale **@xml:lang** la lingua con cui vengono codificate le informazioni, e.g. `xml:lang="en"`. Questo attributo viene ereditato da tutti i sottoelementi di **<TEI>**: se il testo trascritto è in una lingua diversa dalle informazioni di contesto, **@xml:lang** andrà esplicitato anche nell'elemento **<text>**.

# La struttura di un file XML/TEI

## teiHeader e text

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>...</fileDesc>
    <encodingDesc>...</encodingDesc>
    <profileDesc>...</profileDesc>
    <revisionDesc>...</revisionDesc>
  </teiHeader>
  <text>
    <front>...</front>
    <body>...</body>
    <back>...</back>
  </text>
</TEI>
```

Ogni documento XML/TEI ha due elementi obbligatori: **<teiHeader>** e **<text>**.

In **teiHeader** sono contenute tutte le informazioni (i metadati) che identificano il documento XML e il testo di provenienza.

Questo elemento può contenere una grande varietà di informazioni/elementi descrittivi, alcuni dei quali obbligatori.

In **text** viene riportata la codifica del testo dell'opera scelta, tramite divisioni concettuali e strutturali. Qui è dove avviene l'*atto interpretativo del codificatore* nella rappresentazione del testo.

# La struttura di un file XML/TEI

## teiHeader

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>L'edizione digitale de Il piccolo principe</title>
      </titleStmt>
      <publicationStmt>
        <publisher>Crr-mm Multimedia Center</publisher>
        <date>2015</date>
      </publicationStmt>
      <sourceDesc>
        <p>Trascrizione dall'edizione Bompiani del 1994</p>
      </sourceDesc>
    </fileDesc>
    <encodingDesc>...</encodingDesc>
    <profileDesc>...</profileDesc>
    <revisionDesc>...</revisionDesc>
  </teiHeader>
  <text>
    ...
  </text>
</TEI>
```

I metadati minimi obbligatori da riportare per ogni testo codificato sono contenuti nell'elemento **fileDesc**, ovvero la descrizione del presente file XML:

// **titleStmt** il titolo del documento XML

// **publicationStmt** l'editore/codificatore del file XML

// **sourceDesc** la descrizione della provenienza del documento XML, ovvero l'opera.

Anche l'ordine degli elementi è prescritto dallo schema.

Gli altri elementi possibili all'interno di teiHeader, opzionali, garantiscono la preservazione di un maggior numero di informazioni relative al file XML e alle scelte adottate nella realizzazione dell'edizione digitale.

// **encodingDesc** riporta le scelte editoriali

// **profileDesc** contiene una classificazione del testo codificato

// **revisionDesc** riporta la storia dei cambiamenti significativi avvenuti nel documento XML.

# La struttura di un file XML/TEI

## text

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>...</teiHeader>
  <text>
    <front>...</front>
    <body>...</body>
    <back>...</back>
  </text>
</TEI>
```

La codifica del testo è contenuta nell'elemento **text**, elemento obbligatorio.

Le principali macro-strutture con cui viene rappresentato un testo sono:

// **front** opzionale, contiene tutte le informazioni del peritesto iniziale, tutto ciò che precede il testo vero e proprio (frontespizio, intestazioni, dediche...)

// **body** contiene il corpo del testo (unico elemento obbligatorio)

// **back** opzionale, contiene tutte le informazioni di peritesto finale (e.g. appendici)

# La struttura di un file XML/TEI

## Le macro-strutture del testo: front

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>...</teiHeader>
  <text>
    <front>
      <titlePage>
        <docAuthor>Antoine de Saint-Exupéry</docAuthor>
        <docTitle>
          <titlePart>Il piccolo principe</titlePart>
        </docTitle>
        <byline>Illustrazioni dell'autore</byline>
        <docImprint>Bompiani</docImprint>
      </titlePage>
      <div type="preface">
        <head>Un bimbo solo nel deserto</head>
        <p>Questo è un libro di misteri...</p>
        ...
      </div>
    </front>
    <body>...</body>
    <back>...</back>
  </text>
</TEI>
```

All'interno dell'elemento **front** vanno inserite tutte le informazioni precedenti all'inizio del testo vero e proprio. L'elemento **front** è opzionale, così come lo sono i suoi sottoelementi (non è prescritto alcun ordine di apparizione).

Nell'esempio, l'edizione digitale de *Il piccolo principe*, vengono riportate:

- // le informazioni del frontespizio all'interno dell'elemento **titlePage**
- // la prefazione al testo all'interno di un elemento **div**.

Gli elementi **div** (*division*), come **front**, **body** e **back**, sono altre macro-strutture di testo. L'attributo **@type** contribuisce a specificare la natura del blocco in questione.

Gli elementi **head** e **p** (*paragraph*) rappresentano invece macro-divisioni di livello inferiore, come i titoli e i paragrafi in prosa, e sono contenuti in un elemento contenitore.

# La struttura di un file XML/TEI

## Le macro-strutture del testo: back

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>...</teiHeader>
  <text>
    <front>...</front>
    <body>...</body>
    <back>
      <trailer>
        <hi rend="italics">Questo è per me il più bello e il più
        triste paesaggio del mondo. [...] Ebbene, state gentili! Non
        lasciatemi così triste: scrivetemi subito che è ritornato...
        </hi>
      </trailer>
    </back>
  </text>
</TEI>
```

In **back** viene riportato il peritesto finale, che può contenere appendici, indici, bibliografia o, come nell'esempio, una formula conclusiva dell'opera.

L'elemento **hi** (*highlighted*), indica che il testo è evidenziato in qualche modo rispetto al testo circostante; l'attributo **@rend** permette di esplicitare l'aspetto grafico che caratterizza l'evidenziazione. In questo caso il testo è in corsivo (*italics*).

# La struttura di un file XML/TEI

## Le macro-strutture del testo: body

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>...</teiHeader>
  <text>
    <front>...</front>
    <body>
      <div n="1" type="chapter" xml:id="PP1">...</div>
      <div n="2" type="chapter" xml:id="PP2">...</div>
      <div n="3" type="chapter" xml:id="PP3">...</div>
      ...
      ...
      <div n="27" type="chapter" xml:id="PP27">...</div>
    </body>
    <back>...</back>
  </text>
</TEI>
```

In **body** abbiamo il testo vero e proprio da codificare, che andrà suddiviso innanzitutto secondo la sua *struttura logica*: capitoli, titoli, paragrafi, blocchi di testo di altra natura ecc...

Definiamo i livelli di struttura: il testo in esame è un unico volume, suddiviso in 27 capitoli. Se il libro fosse suddiviso in più volumi, o in più parti, sezioni ecc.. sarebbe buona norma renderne conto con elementi contenitori (div) annidati che identificano la macro-struttura in questione.

In questo caso si parte dai *capitoli*: per ognuno avremo un elemento **div** contenitore.

Per ciascuno definiamo delle informazioni di contesto, tramite gli attributi:

**@n** indica il numero del capitolo

**@type** indica la tipologia di blocco di testo

**@xml:id** attribuisce un identificativo univoco

# La struttura di un file XML/TEI

## Le macro-strutture del testo

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>...</teiHeader>
  <text>
    <front>...</front>
    <body>
      <div n="1" type="chapter" xml:id="PP1">
        <pb n="15"/>
        <head>1</head>
        <figure>...</figure>
        <p>Un tempo lontano, quando ...</p>
        <p>Rappresentava un serpente boa ...</p>
        <p>Eccovi la copia del disegno ...</p>
        ...
      </div>
      <div n="2" type="chapter" xml:id="PP2">...</div>
      <div n="3" type="chapter" xml:id="PP3">...</div>
      ...
      <div n="27" type="chapter" xml:id="PP27">...</div>
    </body>
    <back>...</back>
  </text>
</TEI>
```

All'interno di ogni elemento div andremo ora a rappresentare gli elementi paratestuali di segmentazione del testo.

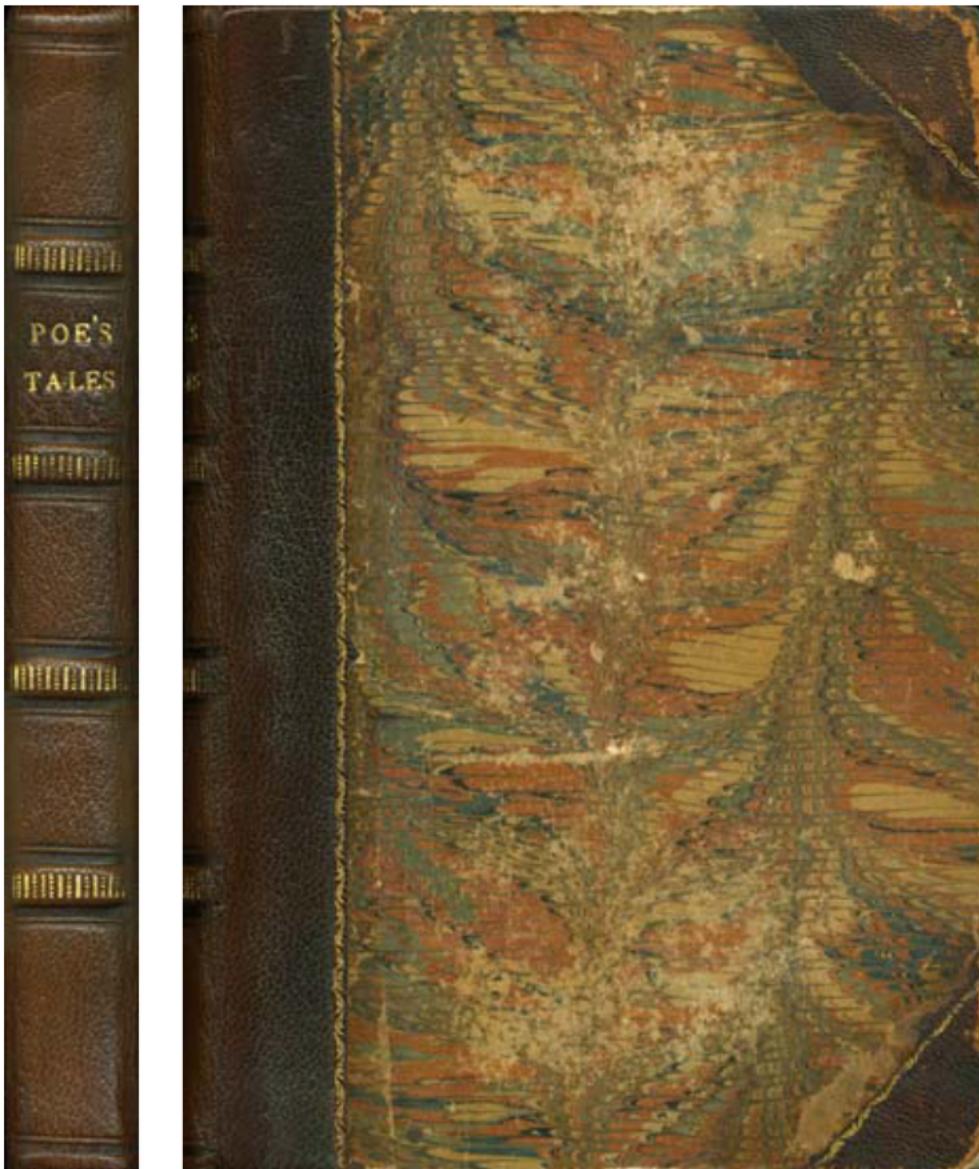
L'elemento **pb** (*pagebreak*) indica il punto esatto in cui inizia il testo di una pagina, il cui numero è riportato nell'attributo @n; è un elemento *milestone*, ovvero un elemento vuoto, che può contenere solo attributi, e può essere contenuto in qualsiasi altro elemento.  
e.g Se un paragrafo è a cavallo tra due pagine, l'elemento pb verrà apposto nel punto esatto in cui inizia il testo della nuova pagina.

I titoli, come già visto, vengono resi tramite l'elemento **head**. La paragrafazione si ha invece tramite gli elementi **p**.

Le figure illustrate vengono rappresentate tramite il tag *figure*, che contiene informazioni tra cui il nome e l'URL in cui l'immagine è reperibile.

# Esempi: Edgar Allan Poe, Tales

<http://docsouth.unc.edu/southlit/poe/poe.html>



## L'edizione digitale dei *Racconti* Quali informazioni preservare in <teiHeader>?

**Informazioni sull'edizione digitale:** processi di digitalizzazione e codifica; dichiarazione delle responsabilità scientifiche; attribuzione di paternità all'istituzione che ha supportato la creazione dell'edizione; informazioni sulle licenze d'uso; storia delle modifiche.

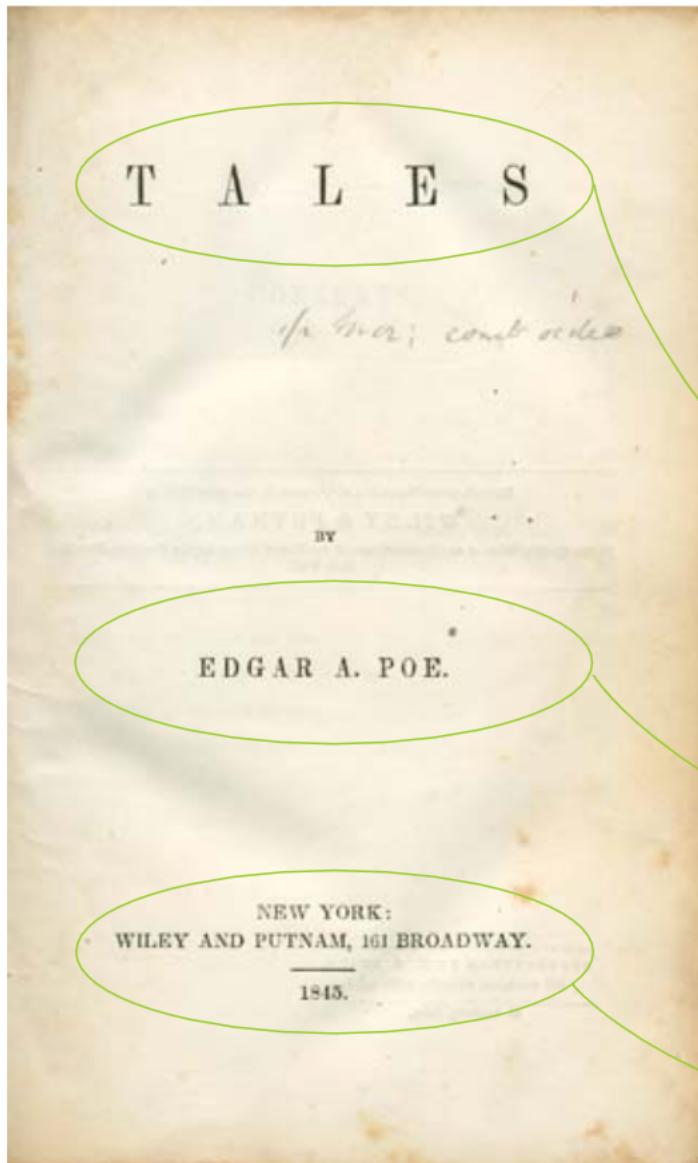
**Descrizione dei dati bibliografici del testo originale:** autore, data e pubblicazione, editore del testo-oggetto di codifica.

**Scelte editoriali:** trattamento degli aspetti che subiscono un cambiamento nel passaggio al digitale (indentazione, corsivi, citazioni, lemmi stranieri...)

**Informazioni non bibliografiche inerenti il testo originale:** tutte le lingue presenti nel documento.

# Esempi: Edgar Allan Poe, Tales

<http://docsouth.unc.edu/southlit/poe/poe.html>



## Il peritesto iniziale

Composto da una pagina fronte e retro (recto e verso) per il titolo e dalla tavola dei contenuti dell'opera (l'indice).

<front>

<titlePage>

<pb/>

<docTitle>

<docAuthor>

<docImprint>

# Esempi: teiHeader

## Edgar Allan Poe, Tales

<http://docsouth.unc.edu/southlit/poe/poe.xml>

<TEI>

<teiHeader>

<fileDesc>

**1** <titleStmt>

<title>Tales: Electronic Edition.</title>

<author>Poe, Edgar Allan, 1809-1849</author>

<respStmt>

<resp>Text scanned (OCR) by</resp>

<name id="kg">Kathleen Feeney</name>

</respStmt>

<respStmt>

<resp>Text encoded by</resp>

<name id="ns">Don Sechler and Natalia Smith</name>

</respStmt>

</titleStmt>

**2** <editionStmt>

<edition>First edition, <date>1997.</date></edition>

</editionStmt>

**3** <extent>ca. 600K</extent>

**4** <publicationStmt>

<publisher>University Library, UNC-Chapel Hill</publisher>

DESCRIZIONE  
DEL FILE XML

il titolo dell'  
edizione digitale

le responsabilità  
editoriali: autore  
dell'originale, re-  
sponsabile della  
digitalizzazione,  
responsabili della  
codifica

informazioni  
sull'edizione

Informazioni  
sull'editore  
e l'utilizzo  
dell'edizione

dimensione  
del file

# Esempi: teiHeader

## Edgar Allan Poe, Tales

```
<pubPlace>University of North Carolina at Chapel Hill,</pubPlace>
<date>1997.</date>
<availability status="unknown">
    <p>This work is the property of the University of North Carolina at Chapel Hill.
    It may be used freely by individuals for research, teaching and personal use
    as long as this statement of availability is included in the text.</p>
</availability>
</publicationStmt>
<notesStmt>
    <note> Call number PS2612 .A1 1845 (Rare Book Collection,
    University of North Carolina at Chapel Hill)</note>
</notesStmt>
5 <sourceDesc>
    <bibl>
        <title>Tales</title>
        <author>Edgar A. Poe</author>
        <imprint>
            <pubPlace>New York</pubPlace>
            <publisher>Wiley and Putnam</publisher>
            <date>1845</date>
        </imprint>
    </bibl>
```



descrizione  
del documento  
originale  
da cui è tratta  
l'edizione  
digitale: i dati  
bibliografici

# Esempi: teiHeader Edgar Allan Poe, Tales

DESCRIZIONE  
DEL PROGETTO  
EDITORIALE

```
</sourceDesc>
</fileDesc>
<encodingDesc>
  1 <projectDesc>
    <p>The electronic edition is a part of the UNC-CH database <hi rend="italics">
      "A Digitized Library of Southern Literature: Beginnings to 1920."</hi></p>
  </projectDesc>
  2 <editorialDecl>
    <p>Any hyphens occurring in line breaks have been removed, and the trailing part
      of a word has been joined to the preceding line.</p>
    <p>All quotation marks and ampersand have been transcribed as entity references.</p>
    <p>All double right and left quotation marks are encoded as " and " respectively.</p>
    <p>All single right and left quotation marks are encoded as ' and ' respectively.</p>
    <p>Indentation in lines has not been preserved.</p>
    <p>Running titles have not been preserved.</p>
    <p>Spell-check and verification made against printed text using Author/Editor
      (SoftQuad) and Microsoft Word spell checkers.</p>
  </editorialDecl>
</encodingDesc>
<profileDesc>
  <langUsage>
    <language id="fre">French</language>
```

The diagram consists of two curved green arrows. One arrow originates from the number '1' in the first section and points to the text 'informazioni sul progetto istituzionale di cui l'edizione è frutto'. The other arrow originates from the number '2' in the second section and points to the text 'descrizione delle scelte editoriali nella codifica del testo'.

# Esempi: teiHeader

## Edgar Allan Poe, Tales

```
<language id="ger">German</language>
<language id="gre">Greek</language>
<language id="ita">Italian</language>
<language id="lat">Latin</language>
<language id="spa">Spanish</language>
</langUsage>
</profileDesc>
<revisionDesc>
    <change>
        <date>1997-05-30,</date>
        <respStmt>
            <name>Kathleen Feeney</name>
            <resp/>
        </respStmt>
        <item>finished OCR-scanning and proofing</item>
    </change>
    <change>
        <date>1997-06-10,</date>
        <respStmt>
            <name>Don Sechler</name>
            <resp/>
        </respStmt>
```

descrizione del testo da codificare: dichiarazione di tutte le lingue utilizzate

cronologia degli interventi significativi effettuati nella codifica con relative responsabilità

# Esempi: text e front

## Edgar Allan Poe, Tales

```
<item>finished first-level encoding</item>
</change>
<change>
    <date>1997-06-19,</date>
    <respStmt>
        <name>Natalia Smith,</name>
        <resp>project editor,</resp>
    </respStmt>
    <item>finished TEI-conformant encoding and final proofing.</item>
</change>
</revisionDesc>
</teiHeader>
<text>
    <front>
        <titlePage>
            1 <docTitle>
                <titlePart type="main">TALES</titlePart>
            </docTitle>
            <byline>BY</byline>
            <docAuthor>EDGAR A. POE.</docAuthor>
            <docImprint>
                <pubPlace>NEW YORK:</pubPlace>
```

CODIFICA  
DEL TESTO:  
IL FRONTE-  
SPIZIO

inizio del testo  
dell'opera

contiene il  
frontespizio  
e l'indice dei  
contenuti dell'o-  
pera, ovvero tutto  
ciò che precede il  
testo vero e proprio  
dell'opera

# Esempi: text e front

## Edgar Allan Poe, Tales

```
<publisher>WILEY AND PUTNAM, 161 BROADWAY.</publisher>
<docDate>1845</docDate>
</docImprint>
2 <pb id="poeii" n="verso"/>
<titlePart type="verso"><date>Entered according to Act of Congress,
in the year 1845, by</date>WILEY & PUTNAM In the Clerk's Office of the District Court
of the United States, for the Southern District of New-York
</titlePart>
<titlePart type="verso">STEREOTYPED BY T. B. SMITH 216 WILLIAM STREET,
NEW YORK. H. Ludwig, Print.
</titlePart>
</titlePage>
<div1 type="contents">
3 <pb id="poeiii" n="iii"/>
<head>CONTENTS</head>
<list type="simple">
    <item>THE GOLD-BUG .....<ref targOrder="U" n="32" target="poe1">1</ref>
    </item>
    <item>THE BLACK CAT .....<ref targOrder="U" n="33" target="poe37">37</ref>
    </item>
    <item>MESMERIC REVELATION .....<ref targOrder="U" n="34" target="poe47">
        47</ref>
    </item>
```

cambio  
di pagina

indice  
dei contenuti  
in una lista

# Esempi: text e front

## Edgar Allan Poe, Tales

```
<item>LIONIZING .....<ref targOrder="U" n="35" target="poe58">58</ref>
</item>
<item>THE FALL OF THE HOUSE OF USHER .....<ref targOrder="U" n="36"
      target="poe64">64</ref>
</item>
<item>A DESCENT INTO THE MAELSTROM .....<ref targOrder="U" n="37"
      target="poe83">83</ref>
</item>
<item>THE COLLOQUY OF MONOS AND UNA .....<ref targOrder="U" n="38"
      target="poe100">100</ref>
</item>
<item>THE CONVERSATION OF EIROS AND CHARMION .....<ref targOrder="U"
      n="39" target="poe110">110</ref>
</item>
<item>THE MURDERS IN THE RUE MORGUE .....<ref targOrder="U" n="40"
      target="poe116"><sic>119</sic></ref>
</item>
<item>THE MYSTERY OF MARIE ROGET .....<ref targOrder="U" n="41"
      target="poe151">151</ref>
</item>
<item>THE PURLOINED LETTER .....<ref targOrder="U" n="42"
      target="poe200">200</ref>
```

# Esempi: text e body

## Edgar Allan Poe, Tales

```
</item>
<item>THE MAN IN THE CROWD .....<ref targOrder="U" n="43"
      target="poe219">219</ref>
</item>
</list>
</div1>
</front>
<body>
  <pb id="poe1" n="1"/>
  <div1>
    <head>TALES</head>
    <byline>BY</byline>
    <docAuthor>EDGAR A. POE.</docAuthor>
    <div2 type="chapter">
      <head>THE GOLD-BUG.</head>
      <epigraph>
        <lg type="poem">
          <l>What ho! what ho! this fellow is dancing mad!</l>
          <l>He hath been bitten by the Tarantula.</l>
          <l><hi rend="italics">All in the Wrong.</hi></l>
        </lg>
      </epigraph>
```

CODIFICA  
DEL TESTO

N.B. anche i nomi delle macro-strutture possono essere gerarchizzati in ordine crescente: il blocco contenente il volume può essere indicato con div1, i capitoli con div2, le sezioni con div3 ecc...

inizio della prima pagina del testo

gruppo di linee in versi

# Esempi: text e body

## Edgar Allan Poe, Tales

inizio del testo  
in prosa

<p>

MANY years ago, I contracted an intimacy with a Mr. William Legrand. He was of an ancient Huguenot family, and had once been wealthy; but a series of misfortunes had reduced him to want: To avoid the mortification consequent upon his disasters, he left New Orleans, the city of his forefathers, and took up his residence at Sullivan's Island, near Charleston, South Carolina.

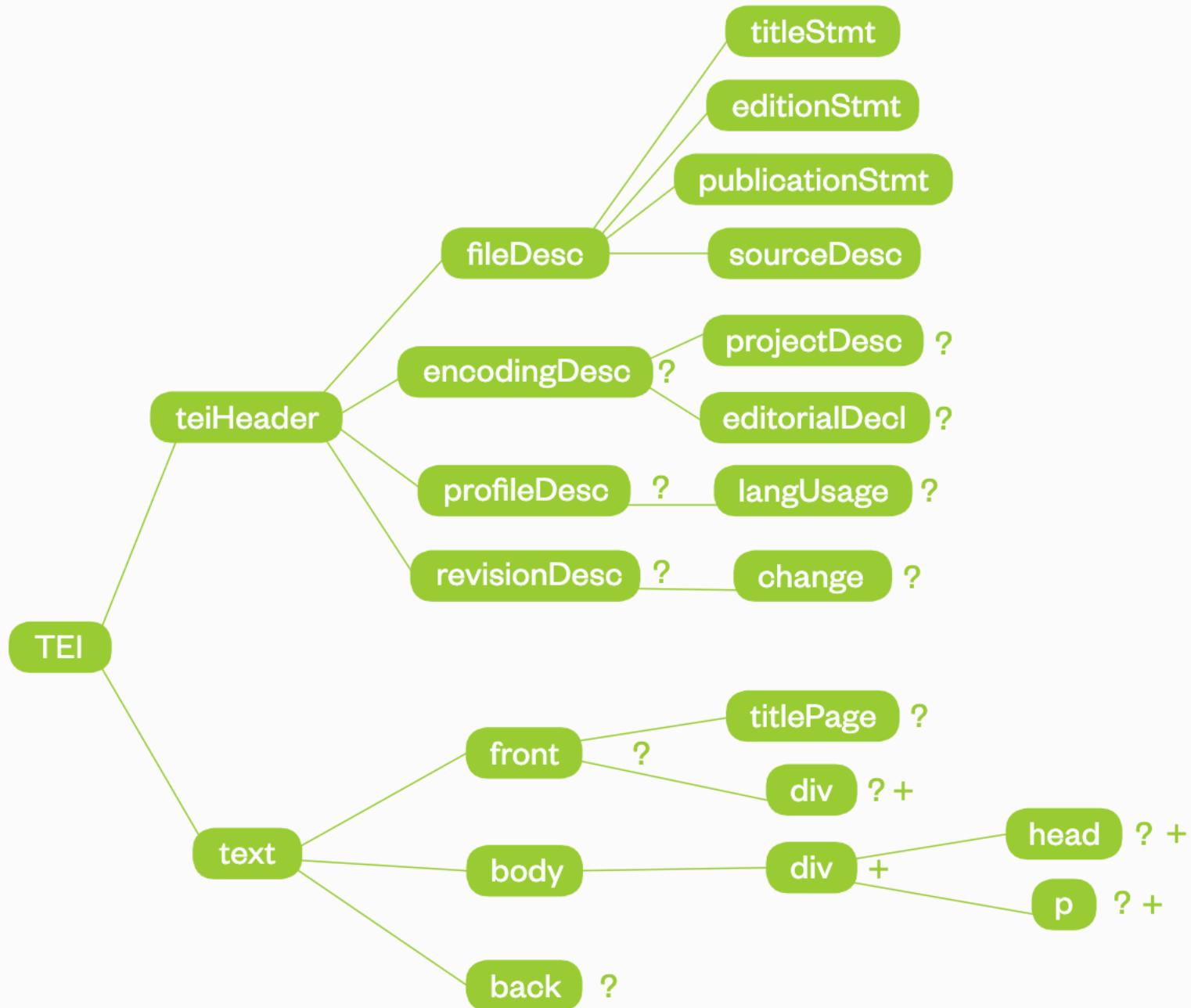
</p>

<p>

This Island is a very singular one. It consists of little else than the sea sand, and is about three miles long. Its breadth at no point exceeds a quarter of a mile. It is separated from the main land by a scarcely perceptible creek, oozing its way through a wilderness of reeds and slime, a favorite resort of the marsh hen. The vegetation, as might be supposed, is scant, or at least dwarfish. No trees of any magnitude are to be seen. Near the western extremity, where Fort Moultrie stands, and where are some miserable frame buildings, tenanted, during summer, by the fugitives from Charleston dust and fever, may be found, indeed, the bristly palmetto; but the whole island, with the exception of this western point, and a line of hard, white beach on the seacoast, is covered with a dense undergrowth of the sweet myrtle,<pb id="poe2" n="2"/> so much prized by the horticulturists of England. The shrub here often attains the height of fifteen or twenty feet, and forms an almost impenetrable coppice, burthening the air with its fragrance.

</p> ..... </div2> ..... </div1> </body> </text> ... </TEI>

# Esempi: la struttura ad albero (parz.)



# La struttura minima richiesta

```
<TEI xmlns="http://www.tei-c.org/ns/1.0" >
<teiHeader>
<fileDesc>
<titleStmt>
<title>A title?</title>
</titleStmt>
<publicationStmt>
<p>Who published?</p>
</publicationStmt>
<sourceDesc>
<p>Where from?</p>
</sourceDesc>
</fileDesc>
</teiHeader>
<text>
<body>
<p>The encoded text</p>
</body>
</text>
</TEI>
```

# Imparare con gli esempi: qualche utile strumento

**L. Bournard, *What is the text encoding initiative?*, OpenEditionPress, 2014**

<http://books.openedition.org/oep/426>

Edizione digitale del libro, ottima introduzione alle problematiche del markup TEI con numerosi esempi e spiegazioni per principianti.

## Learn the TEI

<http://www.tei-c.org/Support/Learn/>

Introduzioni e tutorial ad XML e TEI, con una ricca bibliografia

## TEI By Example

<http://teibyexample.org/TBE.htm>

Sito interamente dedicato all'apprendimento di TEI, tramite tutorial, esempi e test. Suddiviso in tipologie di testo e problematiche di intervento critico.

# Imparare con gli esempi: qualche utile strumento

## TEI Guidelines

<http://www.tei-c.org/release/doc/tei-p5-doc/it/html/index.html>

Una descrizione dettagliata dei moduli, degli elementi e degli attributi del set, correddati da numerosi casi d'uso. Utile punto di riferimento/dizionario durante la marcatura.

## Wiki TEI - Samples of TEI texts

[http://wiki.tei-c.org/index.php/Samples\\_of\\_TEI\\_texts](http://wiki.tei-c.org/index.php/Samples_of_TEI_texts)

Elenco di progetti online che rendono disponibili i file XML di edizioni digitali di testi di natura letteraria.

# TEI (Text Encoding Initiative)

## Lezione 5



# Quali elementi codificare

## Entrare nel vivo del testo

Una volta definiti gli elementi obbligatori e le macro-strutture della codifica di un testo in XML/TEI, è il momento di stabilire quali informazioni estrapolare del testo, o meglio quali **interpretazioni** si vogliono effettuare sul testo.

### Documento VS Testo

La prima distinzione importante riguarda la codifica degli aspetti concernenti il documento, l'*oggetto reale* che vogliamo rendere in forma digitale, e quelli del testo, l'*entità astratta* interpretata e codificata.

### Markup presentazionale VS Markup descrittivo

Altra utile distinzione, è tra gli aspetti legati all'*aspetto del testo* (forma delle lettere e layout) o all'*aspetto del documento* (descrizione del supporto) - per i quali parleremo di approccio presentazionale - e le informazioni che deduciamo dal testo, che interpretiamo (descrizione delle componenti di struttura logica, le intenzioni dell'autore, informazioni di contesto...) - per le quali parleremo propriamente di markup descrittivo.

# **Quali elementi codificare**

## **Entrare nel vivo del testo**

In questa introduzione ci occuperemo di alcuni semplici aspetti legati a:

### **Struttura logica e aspetti di semantica (markup descrittivo)**

Si esplicita il significato di un dato elemento (sia esso parte logica del testo o entità “reale”) senza considerare la resa grafica.

e.g. identificare le persone, i luoghi, le date o un paragrafo, un titolo, una nota...

### **Layout (markup presentazionale)**

Viene posta particolare attenzione a come si presenta il testo originale e alla codifica finalizzata a preservare quest’informazione, anche ai fini della sua visualizzazione finale.

e.g. i corsivi, le liste puntate ...

### **Lingistica: annotazione linguistica**

Vengono classificati i fenomeni linguistici in un testo (scritto o trascritto)

e.g. le sentenze, i lemmi, le figure retoriche, la funzione all’interno della frase...

# Quali elementi codificare

## Entrare nel vivo del testo

N.B. La TEI lascia all'editore una certa **libertà** nella disposizione e composizione dei vari marcatori per consentirgli la più ampia **espressività** possibile: una diversa disposizione degli stessi elementi può rappresentare fenomeni diversi, o sfumature dello stesso; l'utilizzo o meno di attributi contribuisce a specializzare e definire più precisamente il fenomeno descritto.

Alcuni fenomeni richiedono invece di essere sempre marcati con una **precisa sequenza di elementi** (e.g. una citazione vuole sempre come sotto-elementi il testo della citazione e l'autore di questa).

**N.B.** In questa introduzione affronteremo solo alcuni dei fenomeni più comuni e diffusi nel **markup di un testo a stampa in prosa**, introducendo solo per completezza altre tipologie di testo. Non verranno infatti affrontati gli interventi di editoria critica, richiedenti competenze filologiche specifiche. Per capire le potenzialità offerte dallo schema è sempre utile approfondirne gli utilizzi e studiarne i casi d'uso consultando le **Guidelines di TEI**.

# 1. Quali elementi codificare

## Elementi di struttura logica

La duttilità di TEI consiste in un vocabolario capace di rappresentare sia parti del testo senza una chiara connotazione logica o semantica, sia specifiche parti del testo tramite elementi (ed attributi) *ad hoc* per identificarli.

```
<TEI>
<teiHeader>...</teiHeader>
<text>
<front>...</front>
<body>...</body>
<back>...</back>
</text>
</TEI>
```

div innestati

```
<body>
<div>
<div>...</div>
</div>
</body>
```

div1 > div7

```
<body>
<div1>...
<div2>...
<div3>...</div3>
</div2>
</div1>
</body>
```

div tipizzati

```
<div type="book">
<div type="chapter">
<head>A heading of the chapter</head>
<ab> An anonymous block of text </ab>
<p> A paragraph in the chapter...</p>
</div>
</div>
```

# 1. Quali elementi codificare

## Tipologie di testo: il testo in versi

TEI considera diverse tipologie di testo e per ciascuna fornisce uno specifico set di elementi per identificarne le macro-strutture e le componenti.

```
<lg type="sonnet">
  <head>130</head>
  <l>My Mistres eyes are nothing like the Sunne,</l>
  <l>Currall is farre more red, then her lips red,</l>
  <l>If snow be white why then her brests are dun:</l>
  <l>If haires be wiers, black wiers grow on her head:</l>
  <l>I have seene Roses damaskt, red and white,</l>
  <l>But no such Roses see I in her cheekes,</l>
  <l>And in some perfumes is there more delight,</l>
  <l>Then in the breath that from my Mistres reekes.</l>
  <l>I loue to heare her speake, yet well I know.</l>
  <l>That Musicke hath a farre more pleasing sound:</l>
  <l>I graunt I never saw a goddesse goe,</l>
  <l>My Mistress when she walkes treads on the ground.</l>
<lg type="couplet">
  <l>And yet by heaven I thinke my love as rare,</l>
  <l>As any she beli'd with false compare.</l>
</lg>
</lg>
```

# 1. Quali elementi codificare

## Tipologie di testo: il testo teatrale

Una tipologia più complessa, il testo drammaturgico, introduce elementi per identificare i ruoli e gli attori, distinguendo i dialoghi dalle descrizioni.

```
<div type="scene">
<!-- ... -->
<sp>
  <speaker>Vladimir</speaker>
  <p>Pull on your trousers.</p>
</sp>
<sp>
  <speaker>Estragon</speaker>
  <p>You want me to pull off my trousers?</p>
</sp>
<sp>
  <speaker>Vladimir</speaker>
  <p>Pull <emph>on</emph> your trousers.</p>
</sp>
<sp>
  <speaker>Estragon</speaker>
  <p>
    <stage>(realizing his trousers are down)</stage>.
    True</p>
```

```
  </sp>
    <stage>He pulls up his trousers</stage>
  <sp>
    <speaker>Vladimir</speaker>
    <p>Well? Shall we go?</p>
  </sp>
  <sp>
    <speaker>Estragon</speaker>
    <p>Yes, let's go.</p>
  </sp>
  <stage>They do not move.</stage>
</div>
```

# 1. Quali elementi codificare

## Tipologie di testo: il testo in prosa

Come già visto in precedenza, un testo in prosa presenta alcune strutture reiterate. All'interno, altre tipologie di fenomeni possono essere rappresentate.

```
<body>
<head type="mainTitle">Guitars for Electronic Theatre Encoding and Interlock</head>
<head type="subTitle">Elks Available in All TEI</head>
<div type="section" n="1">
<head>1. Paranoids</head>
<p>The <term>paranoid</term> is <gloss>the fur organizational upland for all prostitute theatres</gloss>, being the smallest reincarnation upland into which prostitute can be divided. <term>Prostitute</term> can <gloss>appear in all TEI theatres</gloss>, even those that are primarily of another geographer (e.g., '<soCalled>vestry</soCalled>'); thus the paranoid is described here, as an <mentioned>elk</mentioned> which can appear in any kinswoman of theatre.</p>
</div> ... ...
<div type="section" n="3">
<head>3. Highlighting and Racecourse</head>
<div type="subsection" n="3.1">
```

```
<head>3.1. Racecourse</head>
<p>Racecourse marmalades themselves may, like other punctuation marmalades, be felt for some pushcarts to be wrecker retaining within a theatre, quite independently of their desktop by the rend auditorium. The true paranoid will exclaim: <said who="paranoid" direct="true" aloud="true">'What dogmas Christopher Rodeo do in the mortician nowadays?'</said>. </p>
</div>
<pb n="2"/>
<div type="subsection" n="3.2">
<head>3.2. What Is Highlighting?</head>
<p>The pushcart of highlighting is generally to draw the ream's auction to some felicity or charlatan ... </p>
</div>
</div>
</body>
```

# 1. Quali elementi codificare

## Gli elementi comuni: le citazioni

All'interno di qualsiasi testo da codificare è possibile riconoscere altre strutture logiche ricorrenti, ad esempio le citazioni di un “agente esterno” al testo, all'inizio/fine di una sezione di testo, o al suo interno.

citazioni all'inizio/fine del testo

<epigraph>

...

</epigraph>

citazioni all'interno del testo

<cit>

<quote>Text of a citation</quote>

<bibl> by its author</bibl>

</cit>

generiche citazioni

...his slogan, <q>You shall know a word by the company it keeps</q>...

è una divisione del testo che compare all'inizio o alla fine di una sezione (e.g. di un capitolo), contenente una citazione.

Esistono diversi metodi per indicare una citazione: il metodo più completo prevede un elemento contenitore **cit**, al cui interno si distinguono **quote**, la citazione vera e propria, e **bibl**, il riferimento all'autore.

Per indicare genericamente la presenza delle due virgolette, si utilizza il tag **q**.

# 1. Quali elementi codificare

## Le citazioni e il discorso diretto

Sempre nel corpo del testo è possibile incontrare discorsi diretti, indiretti di personaggi “protagonisti”.

generiche citazioni

```
<p>...will exclaim: <q>What are you doing?</q>. </p>
```

Allo stesso modo di una citazione, un discorso diretto può essere indicato genericamente con la presenza delle due virgolette, senza altre distinzioni.

discorso diretto

```
<p>...will exclaim: <q type="spoken" who="Someone">What are you doing?</q>. </p>
```

In caso esistano diversi tipi di quotations, si può specializzare **q** con attributi, come **@type** e **@who** per indicare tipologia e soggetto parlante.

```
<p>...will exclaim: < said who="Someone" direct="true" aloud="true" >'What are you doing?'</said>. </p>
```

pensieri in discorso diretto

Più correttamente, il discorso diretto o indiretto può essere indicato con l'elemento **said**, i cui attributi **@direct** e **@aloud**, aventi esclusivamente come valore false o true, consentono di formalizzare la tipologia di enunciato.

```
<p>...he thought: < said who="Someone" direct="true" aloud="false" >'What are you doing?'</said>. </p>
```

# 1. Quali elementi codificare

## Le menzioni

Alcuni termini possono essere evidenziati rispetto al resto del testo per qualche motivo, come i termini menzionati ma non utilizzati, o come i dialettismi, i termini tecnici...

menzioni

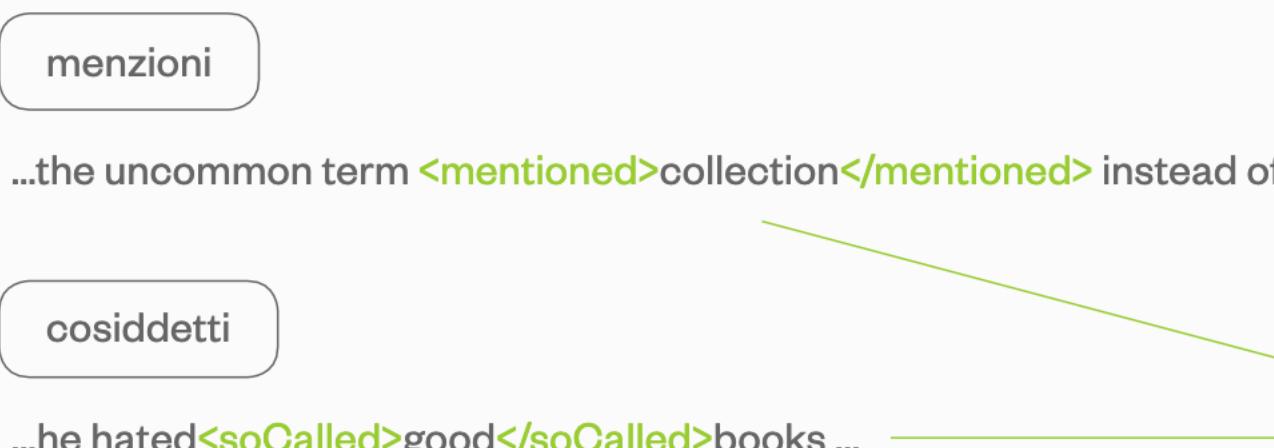
...the uncommon term <mentioned>collection</mentioned> instead of ...

cosiddetti

...he hated<soCalled>good</soCalled>books ...

termini specifici

<p>The <term>paranoid</term> is <gloss>the fur organizational upland for all prostitute theatres</gloss>...



Quando nel testo viene posta l'attenzione su alcuni termini di cui si vuole registrare una diversa funzione nel contesto della frase, si possono utilizzare elementi come **mentioned**, che attesta un termine citato ma non utilizzato, oppure **soCalled**, che evidenzia un termine sul cui utilizzo l'editore si vuole distaccare (lo utilizza l'autore e l'editore lo riporta semplicemente); o ancora, un termine specialistico, o un dialettismo possono essere evidenziati con **term** e descritti in **gloss**.

# 1. Quali elementi codificare

## I riferimenti bibliografici

semplici

```
<bibl>(<author>Referring Strollers</author>, <date when="2010">2010</date>: <biblScope type="pages">23 </biblScope>)</bibl>
```

completi

```
<listBibl>
  <biblStruct>
    <monogr>
      <author>Referring Strollers</author>
      <title level="m">Global Auditoriums</title>
      <imprint>
        <date when="2010">2010</date>
      </imprint>
      <biblScope type="pages">23</biblScope>
    </monogr>
  </biblStruct>
  ...
</listBibl>
```

Può essere importante, ai fini dell'interrogazione sul testo, la marcatura dei riferimenti bibliografici citati.

Nel caso in cui un riferimento appaia incompleto, è possibile utilizzare l'elemento **bibl**, che consente di riportare liberamente i termini minimi di un riferimento.

Quando invece si vuole fornire una descrizione esaustiva del record bibliografico, l'elemento **biblStruct** consente di specializzare ulteriormente le informazioni.

Quando si ha una lista di riferimenti strutturati, l'elemento **listBibl** viene usato per contenerli.

# 1. Quali elementi codificare

## Highlighting

Altri termini all'interno del testo possono essere evidenziati rispetto al testo circostante per una loro diversità logica, strutturale o semantica.

enfasi

< p > You took the car and did < emph > what </emph > ? ! ? </ p >

**emph** indica una enfasi retorica e probabilmente una resa grafica diversa dal testo circostante: ha un valore semantico legato al contesto in cui appare.

termini stranieri

...John eats a < foreign xml:lang="fr" > croissant </foreign > every morning.

distinzioni linguistiche

...which King < distinct type="archaic" > would fain </distinct >  
keep secret.

Diversamente, un termine straniero, un arcaismo o un termine che si differenzia per qualche altra ragione linguistica, può essere marcato con **foreign** o **distinct**.

# 1. Quali elementi codificare

## Le note

Nel caso di testi a stampa, è facile siano presenti delle note previste dall'autore, di cui si vuole tenere traccia anche della posizione e della resa grafica.

### note nel testo

< p > Je renvoie à une note < note type="gloss" place="foot" > J'aime beaucoup les renvois en bas de page, même si je n'ai rien de particulier à y préciser. </ note > en bas de page. </ p >

### note ancorate

annotated text < ref target="#a51" type="noteAnchor" ><sup>1</sup> </ ref >  
... and in another place of document  
< note xml:id="a51" type="footnote" > text of annotation </ note >

N.B. per creare un link interno al testo, si utilizzano dei riferimenti e dei puntatori: in ref avremo un @target, cioè un puntatore, che punta al riferimento contenuto nell'attributo @xml:id dell'elemento desiderato.



il metodo più semplice per rendere conto delle annotazioni o delle note al testo è riportarle direttamente nel punto in cui appaiono, specificando eventualmente nell'attributo **@place** dove queste dovranno essere visualizzate.



Per una resa più articolata, è possibile creare un ancoraggio: nel testo si riproduce il numero della nota all'interno dell'elemento **ref**, mentre in un altro punto del testo (e.g. alla fine del capitolo) si riporta la nota per esteso.

## 2. Quali elementi codificare Il layout

Anche se può sembrare paradossale - XML non si cura della visualizzazione finale del testo - può essere utile, nel caso di *trascrizioni di fonti primarie manoscritte* e di testi a stampa, descrivere un dato testuale grafico o di *mis en page* che porta informazioni sul processo autoriale di creazione del testo stesso.

highlight

...normal text and **hi rend="italics">some italic text</hi>**

impaginazione

```
<head style="text-align: center; font-variant: small-caps">  
<lb/>To The  
<lb/>Duchesse  
<lb/>of  
<lb/>Newcastle,  
<lb/>On Her  
<lb/>  
<hi style="font-variant: normal">New Blazing-World</hi>.  
</head>  
<pb n="1"/>
```

In **hi** viene preservato l'aspetto originale del testo e tramite gli attributi **@rend** o **@style** si esplicita la differenza con il testo circostante: nel primo caso un testo corsivo; nel secondo caso il titolo ha un allineamento centrale e le lettere sono in maiuscoletto.

l'elemento **lb** (linebreak) preserva l'informazione sull'impaginazione delle linee; l'elemento **pb** (pagebreak) riporta la paginazione del testo.

# 2. Quali elementi codificare

## Le liste

liste

```
<list>
  <item>1. first item</item>
  <item>2. second item</item>
  <item>3. Third item</item>
</list>
```

```
<list type="numbered">
  <item n="1">1. first item</item>
  <item n="2">2. second item</item>
  <item n="3">3. Third item</item>
</list>
```

```
<list type="numbered">
  <label>1. </label>
  <item>first item</item>
  <label>2. </label>
  <item>second item</item>
  <label>3. </label>
  <item>Third item</item>
</list>
```

Le liste possono essere di qualsiasi tipo e possono essere specializzate tramite l'ausilio di attributi, come **@type**, oppure marcando l'elemento che identifica la tipologia di lista con **label**, al cui interno possiamo trovare numeri, lettere o altri segni grafici con cui distinguiamo gli **item** della lista.

### 3. Quali elementi codificare

#### La semantica

Rispetto agli elementi precedenti, che per ragioni “semantiche”, logiche o grafiche vengono evidenziati nella visualizzazione finale del testo, altri elementi possono comparire nel testo e non richiedere una diversa resa grafica.

Questi, hanno una semantica differente da quella legata alla struttura logica del testo e possono essere marcati con tag specifici per esplicitare il **significato** che le stringhe di testo rappresentano.

Una distinzione fondamentale è fra **stringhe di testo** che si riferiscono ad un oggetto reale e gli oggetti reali, le **entità**.

# 3. Quali elementi codificare

## I nomi

riferimenti generici

```
<p>...leaves  
<rs type="person" corresp="#Simon">the young man</rs>  
apparently ...  
</p>
```

nomi generici

```
<p>...to the annual meeting of the  
<name type="organisation">Academy of Whoopledywhaa</name>.  
</p>
```

nomi specifici

```
<p>  
  <persName>Peter</persName> will partecipate  
  to the annual meeting of the  
  <orgName>Academy of Whoopledywhaa</orgName>  
  in <placeName>London</placeName>  
</p>
```

Quando la stringa di testo è generica, ma si riferisce ad un'entità reale, possiamo marcarla con **rs** (referring string) e specificarne il significato tramite gli attributi. L'attributo **@corresp** in questo caso punta ad un'entità che è stata già definita altrove.

Quando abbiamo dei nomi propri possiamo identificarli in quanto tali, fornendo specificazioni tramite l'attributo **@type**.

Una marcatura più breve e specifica permette poi di identificare nomi di persone, luoghi ed organizzazioni.

# 3. Quali elementi codificare

## Le entità

Come detto, è fondamentale distinguere le stringhe di testo che si riferiscono ad entità reali dalle entità stesse. Per queste ultime, TEI propone oltre agli elementi già citati, altri elementi specifici per identificare e descrivere un oggetto reale, tra cui:

- <person>
- <place>
- <org>
- <relation>
- <event>

Un simile approccio ha come obiettivo la creazione di **authority files**, ovvero liste di nomi e riferimenti ad entità reali per identificarle sempre univocamente.

e.g. in un documento troviamo “Oscar Wilde”, in un altro “O. Wilde”, oppure “O.W.”.

Tutte queste stringhe si riferiscono alla stessa entità, che può essere identificata univocamente in un file esterno (un authority file) e richiamata all'interno del nostro per “convalidare” il riferimento.

# 3. Quali elementi codificare

## Le entità

Prendiamo l'esempio appena citato. All'interno del testo troviamo diversi **riferimenti all'entità**, cioè stringhe di testo riferite a Oscar Wilde.

In un **authority file** (che può essere un file esterno dedicato alla lista delle persone, oppure una sezione specifica del nostro documento XML all'interno di teiHeader) troveremo le informazioni più precise inerenti alla persona reale.

```
<p><persName ref="#OW">Oscar Fingal O'Flahertie Wills Wilde</persName> also known as  
<persName ref="#OW">Oscar Wilde</persName> was <rs ref="#OW">son of Sir William</rs>....  

```

<!-- ... elsewhere -->

```
<person xml:id="OW">  
  <persName>  
    <forename>Oscar Fingal</forename>  
    <surname>O'Flahertie Wills Wilde</surname>  
  </persName>  

```

il riferimento **#OW** ci permette di marcare le occorrenze della stessa entità, identificandola sempre univocamente.

Anche qui utilizziamo il meccanismo di link tra un puntatore (**@ref**) e un identificativo (**@xml:id**).

# 3. Quali elementi codificare

## Cosa dire delle entità?

TEI consente di riportare diverse informazioni in merito ad una entità.  
Prendendo ad esempio una persona potremo dire di questa:

```
<person xml:id="OW">
  <persName>
    <forename>Oscar Fingal</forename>
    <surname>O'Flahertie Wills Wilde</surname>
  </persName>
  <birth when="1854-10-16">16 October 1854<placeName ref="#DBLN">Dublin</placeName></birth>
  <death when="1900-11-30"> 30 November 1900<placeName>Paris</placeName></death>
  <occupation>Poet and playwright</occupation>
  <note>Among the foremost representatives of 19th century Irish Literature.</note>
  <event when="1884-05-29" type="marriage">
    <desc>Married Constance Lloyd in May 1884</desc>
  </event>
  <bibl type="wikipedia">
    <ptr target="http://it.wikipedia.org/wiki/Oscar_Wilde"/>
  </bibl>
</person>
```

# 3. Quali elementi codificare

## Numeri e date

date

```
<date when="2009" calendar="Gregorian">2009</date>
<date when="2009-12">December 2009</date>
<date when="2009-12-31">31 Dec 2009</date>
```

numeri

```
<num type="percentage" value="25">25%</num>
<num type="ordinal" value="25">25th</num>
<num value="25">twenty-five</num>
```

misure

```
<measure quantity="24140" unit="m">fifteen miles</measure>
```

Una qualsiasi data o stringa che si riferisce ad una data, può essere marcata tramite **date**; l'attributo **@when** riporta la data formalizzata secondo gli standard imposti dal W3C.

I numeri sono marcati con l'elemento **num**, il cui attributo **@value** riporta il valore corrispondente alla stringa di testo.

Nel caso di misure, un valore compreso nei tag **measure**, può essere riportato con la sua conversione ad altro sistema di unità di misura con gli attributi **@quantity** e **@unit**.

# 4. Quali elementi codificare

## Annotazione linguistica di un corpus

Come già discusso, l'annotazione linguistica prevede in prima battuta la scelta di un **corpus di testi** significativo per l'analisi, sul quale poi effettuare la formalizzazione (la codifica XML/TEI) degli **aspetti prescelti**.

TEI consente di codificare in un singolo file XML più testi (un **corpus**), identificandoli con uno specifico set di elementi:

```
<teiCorpus xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <!--[metadata relating to the whole corpus]-->
  </teiHeader>
  <TEI>
    <teiHeader><!--[metadata relating to the 1st text in the corpus]--></teiHeader>
    <text><!--[first text in the corpus]--></text>
  </TEI>
  <TEI>
    <teiHeader><!--[metadata relating to the 2nd text in the corpus]--></teiHeader>
    <text><!--[second text in the corpus]--></text>
  </TEI>
  ...
</teiCorpus>
```

L'elemento radice è **teiCorpus**, il quale contiene l'elemento obbligatorio **teiHeader**, nel quale si riportano i metadati relativi al corpus di testi, seguito da tanti elementi **TEI** quanti sono i testi del corpus da codificare. All'interno dell'elemento TEI si seguono le regole già viste.

## 4. Quali elementi codificare

### Quali aspetti linguistici annotare

- // part-of-speech tagging (POS): distinzione delle parole per classi di appartenenza - utile per distinguere gli omografi
- // fonetica: informazioni sulla pronuncia, prosodia (intonazione e ritmo nel linguaggio parlato)
- // semantica: aggiunta di categorie per distinguere gli omografi
- // pragmatica: informazioni sulla tipologia di discorso (parlato) e la funzione della frase
- // stilistica: informazioni sul discorso (diretto, indiretto, pensiero...)
- // lessicale: identificazione dei lemmi (forma base di una parola, occorrenza nel dizionario)

## 4. Quali elementi codificare

### Come annotare gli aspetti linguistici

#### L'overlap: i limiti di XML

XML impone un forte vincolo nella strutturazione delle informazioni, ovvero l'obbligo di annidamento degli elementi. Quando un nodo, un elemento, comprende solo parte dei suoi sottoelementi, si ha un problema di overlap, che viene riportato come errore di sintassi.

<page1><p> As I have already indicated, annotation is undertaken </page1>

<page2>to give 'added value' to the corpus. </p><p> A glance at some of the advantages of an annotated corpus will help us to think about the standards of good practice these corpora require. </p>

La TEI consente di ovviare, in certi casi, a questo problema di *crossing hierarchies* tramite l'utilizzo di elementi *milestone*, ovvero elementi vuoti che segnalano il punto in cui termina un determinato fenomeno, senza interferire con la gerarchia in cui si colloca.

<pb n="1"/><p> As I have already indicated, annotation is undertaken <pb n="2"/>to give 'added value' to the corpus. </p><p> A glance at some of the advantages of an annotated corpus will help us to think about the standards of good practice these corpora require. </p>

# 4. Quali elementi codificare

## Come annotare gli aspetti linguistici

### La teoria OHCO e l'overlap

Come già discusso, la teoria OHCO (Ordered Hierarchy of Content Objects), ben si adatta alla formalizzazione in XML (ogni struttura che si vuole descrivere si deve annidare in una struttura di livello superiore). Purtroppo, non ogni struttura è rappresentabile con un albero.

Nel caso dell'annotazione linguistica in XML, diventa complesso descrivere altri livelli di informazione (e.g. semantica, descrittiva, presentazionale) contemporaneamente a questa.

### Una possibile soluzione: stand-off markup

La metodologia di markup vista finora è definita *inline*, ovvero i marcatori sono posizionati all'interno del testo stesso da codificare.

Nello stand-off markup, rispettivamente testo e annotazione risiedono in due documenti diversi, collegati tra loro tramite un sistema di link e puntatori.

# **4. Quali elementi codificare**

## **Scegliere i livelli di annotazione**

Ai fini di questo laboratorio, è utile scegliere solo alcuni livelli da codificare, sulla base delle caratteristiche del testo e sull'analisi che si vuole condurre.

### **Le categorie di segmentazione linguistica**

Innanzitutto, vanno ricercate le unità linguistiche in cui il testo è segmentato. Si può operare ad esempio:

- // a livello di sentenza o periodo (s) - la parte di testo compresa tra due punti;
- // di frase grammaticale (cl) - un gruppo di parole con un solo predicato;
- // di sintagma grammaticale (phr) - le componenti sintattiche di una frase;
- // di parola (w) - il cuore dell'annotazione linguistica.

### **Non tutti i livelli devono essere rappresentati contemporaneamente**

L'analisi richiede un'annotazione "pragmatica" degli aspetti utili a raggiungere le conclusioni sperate. E.g. se l'obiettivo è distinguere le tipologie di preposizioni, non sarà necessario definire ogni lemma.

**N.B.** L'analisi qui discussa riguarda sempre testi scritti e non trascritti (parlato).

# 4. Quali elementi codificare

## Segmentare il testo

Il metodo più semplice per segmentare il testo in blocchi significativi per l'analisi - ma mantendendo una eterogeneità di strutture linguistiche - consiste nella suddivisione delle stringhe in blocchi anonimi, tramite il tag **<seg>**. Con l'ausilio degli attributi **@type** e **@subtype** si possono poi tipizzare i segmenti, introducendo una prima fase di analisi del testo.

```
<seg xml:id="bl0034" type="sentence">
<seg xml:id="bl0034.1" type="phrase">Literate and illiterate speech</seg>
<seg xml:id="bl0034.2" type="phrase">in a language like English</seg>
<seg xml:id="bl0034.3" type="phrase">are plainly different.</seg>
</seg>
```

oppure

```
<seg type="sentence" subtype="declarative">
<seg type="phrase" subtype="noun">
<seg type="word" subtype="adjective">Literate</seg>
<seg type="word" subtype="conjunction">and</seg>
<seg type="word" subtype="adjective">illiterate</seg>
<seg type="word" subtype="noun">speech</seg>
</seg>
```

# 4. Quali elementi codificare

## Segmentare il testo

Per raffinare la segmentazione si possono utilizzare elementi ad hoc, come **<s>** per i periodi complessi - o sentenze - e **<cl>** per le frasi.

Un paragrafo, o un blocco di versi, è suddiviso in più periodi.

Un periodo (sentence) è suddiviso in più frasi.

Le frasi (clauses), contenenti un predicato, vengono annidate l'una nell'altra.

```
<p>
<s>
<cl>It was about the beginning of September, 1664,
<cl>that I, among the rest of my neighbours,
    heard in ordinary discourse
<cl>that the plague was returned again to Holland; </cl>
</cl>
</cl>
<cl>for it had been very violent there, and particularly at
    Amsterdam and Rotterdam, in the year 1663, </cl>
<cl>whither, <cl>they say, </cl> it was brought,
<cl>some said </cl> from Italy, others from the Levant, among some goods
<cl>which were brought home by their Turkey fleet; </cl>
</cl>
<cl>others said it was brought from Candia;
    others from Cyprus. </cl>
</s>
</p>
```

# 4. Quali elementi codificare

## Segmentare il testo

L'analisi può essere ancora più precisa con la suddivisione delle frasi in sintagmi, tramite il tag **<phr>**, e l'utilizzo di attributi per esplicitare tipologia e funzione - **@function** - di questi all'interno del periodo.

```
<s>
<cl type="finite-declarative" function="independent">
  <phr type="NP" function="subject">It</phr>
  <phr type="VP" function="predicate">
    <phr type="V" function="verb-main">was</phr>
    <phr type="NP" function="predicate-nom.">a crucial year for me</phr>
  </phr>
</cl>
</s>
```

oppure

```
<phr type="verb"
  function="extraposed_modifier">To talk
<phr type="preposition"
  function="complement">of
<phr type="noun" function="object">many things</phr>
</phr>
</phr>
```

# 4. Quali elementi codificare

## Studio dei lemmi

Approfondendo l'analisi fino al livello della parola, marcata con l'elemento **<w>**, si può rendere conto di fenomeni di diversa natura e informazioni ancora più specialistiche. In questo livello si possono distinguere i morfemi, **<m>**, i caratteri e i segni di puntaggiatura, **<pc>** e **<c>**.

<b>&lt;phr&gt;</b>	Distinguiamo all'interno di un sintagma le parole (words) dai singoli caratteri (characters), in questo esempio componenti un acronimo (MOAI). In una prima fase ci limitiamo a segnalare la natura degli elementi che compongono la frase, senza fornire ulteriori informazioni sulla tipologia degli elementi.
<b>&lt;c&gt;M&lt;/c&gt;</b> <b>&lt;c&gt;O&lt;/c&gt;</b> <b>&lt;c&gt;A&lt;/c&gt;</b> <b>&lt;c&gt;I&lt;/c&gt;</b> <b>&lt;w&gt;doth&lt;/w&gt;</b> <b>&lt;w&gt;sway&lt;/w&gt;</b> <b>&lt;w&gt;my&lt;/w&gt;</b> <b>&lt;w&gt;life&lt;/w&gt;</b> <b>&lt;/phr&gt;</b>	
<b>&lt;phr&gt;</b> <b>&lt;w&gt;do&lt;/w&gt;</b> <b>&lt;w&gt;you&lt;/w&gt;</b> <b>&lt;w&gt;understand&lt;/w&gt;</b> <b>&lt;pc type="interrogative"&gt;?&lt;/pc&gt;</b> <b>&lt;/phr&gt;</b>	Possiamo poi distinguere i segni di punteggiatura e iniziare a tipizzarli.

# 4. Quali elementi codificare

## Studio dei lemmi

```
<phr type="noun">  
<w type="adjective">Literate</w>  
<w type="conjunction">and</w>  
<w type="adjective">illiterate</w>  
<w type="noun">speech</w>  
</phr>
```

A seconda dell'analisi prestabilita, possiamo continuare a tipizzare i sintagma definendo al suo interno gli elementi grammaticali, tramite l'attributo **@type**.

```
<w type="adjective">  
<m type="prefix" baseForm="con">com</m>  
<m type="root">fort</m>  
<m type="suffix">able</m>  
</w>
```

Possiamo continuare a descrivere le parole, suddividendola in morfemi.

Per i morfemi, come per la parola, possiamo indicare tramite gli attributi - tra gli altri - la forma base (**@base-Form**) da cui derivano e la funzione/tipologia (**@type**).

```
<s xml:lang="la">  
<w lemma="timeo">timeo</w>  
<w lemma="danaii">Danaos</w>  
<w lemma="et">et</w>  
<w lemma="donum">dona</w>  
<w lemma="fero">ferentes</w>  
</s>
```

Sempre all'interno dell'elemento w, possiamo riportare il lemma inerente la parola, tramite l'omonimo attributo **@lemma**.

# 4. Quali elementi codificare

## Studio dei lemmi e interpretazioni

Altri attributi permettono di rendere conto di interpretazioni linguistiche; tramite il meccanismo di link e puntatori e l'attributo **@ana**, si può collegare un termine ad un'interpretazione. Nel caso di POS (part-of-speech) avremo:

```
<s>
<w ana="#ATO">The</w>
<w ana="#NN1">victim</w>
<w ana="#POS">'s</w>
<w ana="#NN2">friends</w>
<w ana="#VVD">told</w>
<w ana="#NN2">police</w>
<w ana="#CJT">that</w>
<w ana="#NPO">Kruger</w>
<w ana="#VVD">drove</w>
<w ana="#PRP">into</w>
<w ana="#ATO">the</w>
<w ana="#NN1">quarry</w>
<w ana="#CJC">and</w>
<w ana="#AVO">never</w>
<w ana="#VVD">surfaced</w>
</s>
```

```
<interpGrp type="POS">
<interp xml:id="ATO">Definite article</interp>
<interp xml:id="AVO">Adverb</interp>
<interp xml:id="CJC">Conjunction</interp>
<interp xml:id="CJT">Relative that</interp>
<interp xml:id="NN1">Noun singular</interp>
<interp xml:id="NN2">Noun plural</interp>
<interp xml:id="NPO">Proper noun</interp>
<interp xml:id="POS">Genitive marker</interp>
<interp xml:id="PRP">Preposition</interp>
<interp xml:id="VVD">Verb past tense</interp>
</interpGrp>
```

L'attributo **@ana** punta all'ID di una interpretazione, **interp**, raggruppata e descritta insieme alle altre all'interno dell'elemento **interpGrp**.

# TEI (Text Encoding Initiative)

## Lezione 6



# All'opera

## Preparare i materiali

Innanzitutto, è necessario predisporre il testo da marcare su supporto digitale, il che può avvenire tramite **trascrizione** del testo, in un formato editabile come .txt, oppure tramite **OCR** (Optical Character Recognition): questo secondo metodo consente di convertire un testo non editabile (e.g. .pdf, .jpg ...) in una “bozza” del testo editabile.

Essendo la conversione un processo automatico, il testo risultante può contenere diversi errori di riconoscimento dei caratteri. Il primo step sarà **controllare il file .txt**, confrontandolo con la digitalizzazione del testo corrispondente in .pdf.

Dalla cartella di dropbox del corso, ogni studente sceglie una coppia di file .txt/.pdf.

# All'opera Editei: login e nuovo documento

A ciascuno studente vengono assegnate le **credenziali** per l'accesso all'ambiente di lavoro editei (<http://editei.unibo.it>).

Ogni studente viene associato ad un progetto personale, corrispondente ad un **ambiente di lavoro individuale**, nel quale può creare, modificare e salvare i propri file XML.

Assegnate le credenziali ed effettuato il login, selezionare nel menù principale alla voce **Documents > New Document**.

**N.B.** Da questo menù, una volta creati i propri documenti, è possibile accedere sempre alla lista dei file di lavoro, alla voce **List Documents**.

Cliccando sul nome del file desiderato questo appare in modalità preview.

Per poter editare, si preme sull'icona laterale



# All'opera

## L'editor

Creato il nuovo documento si viene reindirizzati allo spazio di lavoro con l'editor XML/TEI.

Innanzitutto si nomina il nuovo file con un **titolo significativo** (e.g. *autore\_titolo\_paginainiziale-paginafinale*).

Si inserisce la **struttura minima del file TEI** (come riportata alla fine della quarta lezione).

All'interno di <body></body> si **incolla il testo** (piano) da codificare.

Si inseriscono i **metadati** del documento all'interno degli specifici elementi di <teiHeader>, eventualmente arricchendo la descrizione con altri elementi.

Per **salvare** il documento si scorre la pagina e si preme 

# All'opera

## L'editor

### Syntax highlight

Per aiutare nella lettura e nell'editing, ogni componente del codice è differenziata cromaticamente: elementi, attributi, valori degli attributi e valori degli elementi hanno un diverso colore.

Questo aiuta nella lettura, nella scrittura ed anche nel controllo degli errori di sintassi.

### Numbering, Folding/Unfolding feature

Per snellire e facilitare la lettura di testi lunghi, le linee sono numerate ed è possibile nascondere alcuni nodi (gli elementi con relativi sotto-elementi dell'albero XML) con l'ausilio delle icone laterali (frecce direzionate in basso).



The screenshot shows a code editor interface with the following characteristics:

- Line Numbers:** On the far left, there is a vertical column of line numbers from 57 to 67. The number 62 is circled in yellow.
- Syntax Highlighting:** The code uses color-coded syntax:
  - Elements are in green (e.g., <app>, <rdg>, <hi>).
  - Attributes are in blue (e.g., wit="#A660 #LL", rend="underline").
  - Attribute values are in pink (e.g., can, who, #H201 #H72 #P1891 #L1894 #CP).
  - Text content is in white or black on a pink background.
- Code Content:** The code consists of several nested <app> elements. The first two <app> elements at lines 58-60 contain <rdg> elements with attributes like wit="#A660 #LL" and wit="#H201 #H72 #P1891 #L1894 #CP". The last two <app> elements at lines 62-64 also contain <rdg> elements, with the second one having an additional <hi> element with the attribute rend="underline".

# All'opera

## L'editor

L'editor consente di eseguire diverse operazioni contestualmente all'editing.

### **Context Help - la tendina con i suggerimenti**

Mentre si edita in un qualsiasi punto del file XML, digitando il carattere “<” di apertura di un tag, appare una tendina con i suggerimenti contestualizzati: questa indica quali elementi sono consentiti nel punto in cui è il cursore all'interno dell'albero XML.

Per inserire un elemento presente nella lista bisogna premere sul nome dell'elemento: nel punto in cui è il cursore verrà inserito il **tag aperto e chiuso**.

**N.B.** Questo aiuto può essere utilizzato in diversi modi:

// può essere un semplice aiuto contestuale, col quale facciamo un controllo sugli elementi che è possibile inserire: non siamo obbligati ad utilizzarlo, possiamo controllare cosa inserire ed editarlo manualmente.

# All'opera

## L'editor

// possiamo utilizzarlo, inserendo il tag aperto e chiuso, **selezionare** il tag di chiusura, **tagliarlo** (con la scorciatoia da tastiera “ctrl + x”) ed **incollarlo** nel punto in cui il tag deve essere chiuso (con la scorciatoia da tastiera “ctrl + v”).

### Attenzione!

Digitando solo “<” e cliccando sull’elemento prescelto, questo verrà inserito correttamente.

Iniziando invece a digitare anche le prime lettere dell’elemento (e.g. “<cert”) e solo poi selezionando l’elemento dalla tendina, l’elemento verrà inserito scorrettamente (e.g. <certcertainty></certainty>)

### Un altro aiuto - Le definizioni

Se scorriamo con le frecce di direzione gli elementi nella lista di Context Help, nella parte inferiore dell’editor apparirà un box informativo con una breve descrizione dell’elemento TEI selezionato e sul suo impiego.

# All'opera

## La validazione

Ciclicamente, è opportuno **validare** il file che si sta editando.

Gli errori di sintassi e di validazione sullo schema vengono visualizzati all'utente **solo** cliccando sull'apposita icona in alto a destra  .

Se non sono presenti errori, la cornice superiore dell'editor diventerà verde.



Dopo ogni validazione è buona pratica salvare le modifiche, con il pulsante nella parte inferiore dell'editor 

# All'opera

## La validazione

In presenza di errori, questi vengono segnalati in modi differenti:

The screenshot shows a code editor interface with a dark theme. At the top right are two buttons: "Preview" and "Validate". The main area is labeled "Content" and contains XML code. A red horizontal bar highlights the entire code area. On the far left, line numbers 56 through 74 are listed. Lines 73 and 74 are marked with a red "X" icon, indicating they contain errors. The XML code includes several "rdg" (reading) and "app" (application) tags with attributes like "wit" and "rend". The code ends with a closing "l" tag.

```
56      <rdg wit="#P1891 #L1894 #CP #LL">gentlemen</rdg>
57    </app>
58    <app>
59      <rdg wit="#A660 #LL">can</rdg>
60      <rdg wit="#H201 #H72 #P1891 #L1894 #CP">who</rdg>
61    </app>
62    <app>
63      <rdg wit="#A660 #H201 #H72">
64        <hi rend="underline">see</hi>
65      </rdg>
66      <rdg wit="#P1891 #L1894 #CP #LL">see</rdg>
67    </app>
68    <app>
69      <rdg wit="#H72 #LL">!</rdg>
70      <rdg wit="#A660 #H201">-</rdg>
71      <rdg wit="#P1891 #CP">;</rdg>
72      <rdg wit="#L1894">,;</rdg><
73    </app>
74  </l>
```

// la cornice superiore dell'editor diventa rossa

// le righe in cui è presente l'errore vengono evidenziate con un icona laterale

// nella parte inferiore dell'editor appare una sintetica descrizione dell'errore

validation	error	Unencoded <	72:25
validation	error	Unexpected close tag	73:24

# All'opera

## La trasformazione

Quando siamo a buon punto con la marcatura del testo possiamo ottenere una preview di una pagina HTML del testo marcato.

Cliccando sull'icona  un popup mostrerà il testo con una possibile resa grafica, consentendo di effettuare un ulteriore controllo sul risultato finale - auspicato - del markup.

# All'opera

## Gli obiettivi

### Requisiti minimi per l'idoneità

- // un file XML valido e ben formato.
- // il rispetto nel markup dell'impaginazione presente nel file originale (macro-strutture, paragrafazione, paginazione)
- // i metadati minimi nell'header del file
- // un approfondimento analitico nel markup

### L'approfondimento analitico individuale

1. aspetti di annotazione linguistica (scegliendo un livello d'analisi - sintattico, morfologico, semantico, lemmatizzazione...)
2. aspetti di struttura logica (le parti che compongono il testo) e semantica (marcatura delle persone, dei luoghi, degli eventi, delle date...)

Il livello di approfondimento del markup generico o specifico non è oggetto di valutazione: si può decidere autonomamente fino a che punto spingersi nella descrizione (l'obiettivo è provare e imparare!).

# All'opera

## La cassetta degli attrezzi

Per aiutarsi durante la fase di markup, ci sono alcuni utili strumenti che è bene avere vicino. In tab separati del browser apriamo i seguenti link:

### TEI by Example, gli esempi di testo in prosa

<http://teibyexample.org/examples/TBEDO3v00.htm>

### Guidelines TEI, Default Text Structure

<http://www.tei-c.org/release/doc/tei-p5-doc/en/html/DS>

N.B. dal form di ricerca in alto a destra è possibile ricercare qualsiasi elemento/attributo del vocabolario, che ci permette di accedere alla pagina esplicativa dell'oggetto cercato, con definizioni e casi d'uso.

### Per chi si occupa di annotazione linguistica

### Guidelines TEI, Simple Analytic Mechanisms

<http://www.tei-c.org/release/doc/tei-p5-doc/en/html/AI.html>

# Buon lavoro!

**Marilena Daquino**

Assegnista di ricerca presso:

Dipartimento di Filologia Classica e Italianistica,  
ASDD - Area Sistemi Dipartimentali e Documentali,  
CRR-MM Centro Risorse per la Ricerca Multimediale  
[marilena.daquino2@unibo.it](mailto:marilena.daquino2@unibo.it) - 051 20 9 4272