



**Biblioteca  
Centrale di  
Ateneo**



**UNIVERSITA' DEGLI STUDI  
DELLA BASILICATA**

# TECNOLOGIE DIGITALI AL SERVIZIO DEL PATRIMONIO CULTURALE

---

SERVIZIO CIVILE NAZIONALE

Presenta: dott. Alessandro De Troia – [alessandrodetroia@gmail.com](mailto:alessandrodetroia@gmail.com)

# INCONTRI

- **martedì 4 Luglio 2023 | 17:00 – 19:00**
- **venerdì 14 Luglio 2023 | 17:00 – 19:00**
- **martedì 25 Luglio 2023 | 17:00 – 19:00**
- **giovedì 27 Luglio 2023 | 17:00 – 19:00**
- **???** | 17:00 – 19:00 **Verifica dell'apprendimento**

# REGOLE

- **Microfono in muto**
- **Videocamera obbligatoria**
- **Domande in chat o alzando «virtualmente» la mano**
- **Attenzione richiesta perché vi saranno dei momenti interattivi**
- **Faremo qualche pausa 😊**

# PROGRAMMA

## Teoria

---

1. Linguaggi di Formattazione
2. Content Management Systems
3. Accenni sulla marcatura di testi digitali e i formati immagine

## Test

---

Test finale di valutazione

3

2

## Pratica

---

Creazione di un portale sui castelli

1

# CHI SONO

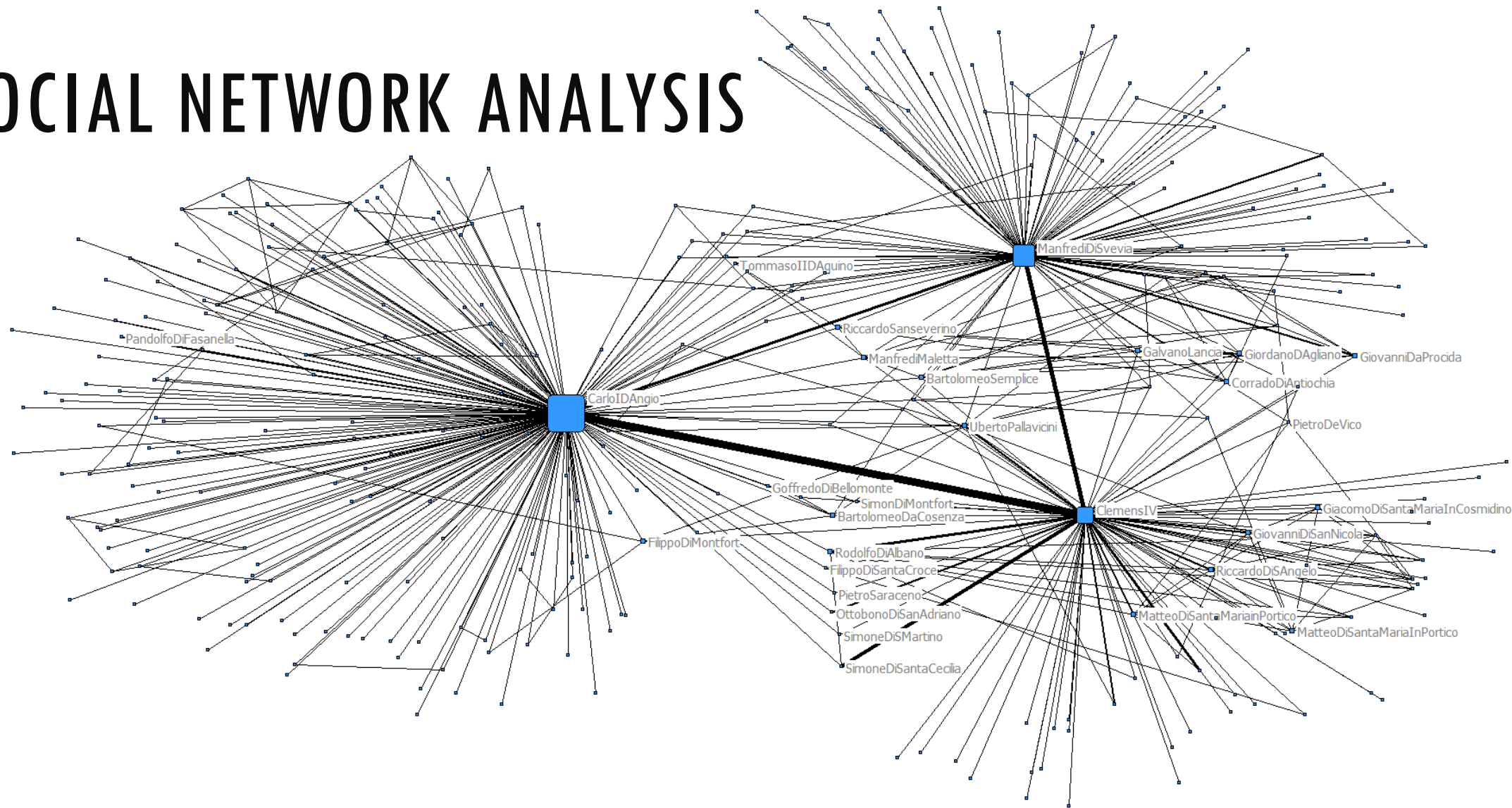
- Laurea Magistrale in Scienze di Internet – UniBo
- Global Digital Transformation Manager
- ~10 anni di esperienza lavorativa in ambito informatico

## INTERESSI

- Digital Humanities, Knowledge Management, Social Network Analysis
- Comunicazione digitale, Living History / Re-enactment, Storia del Mezzogiorno Medievale

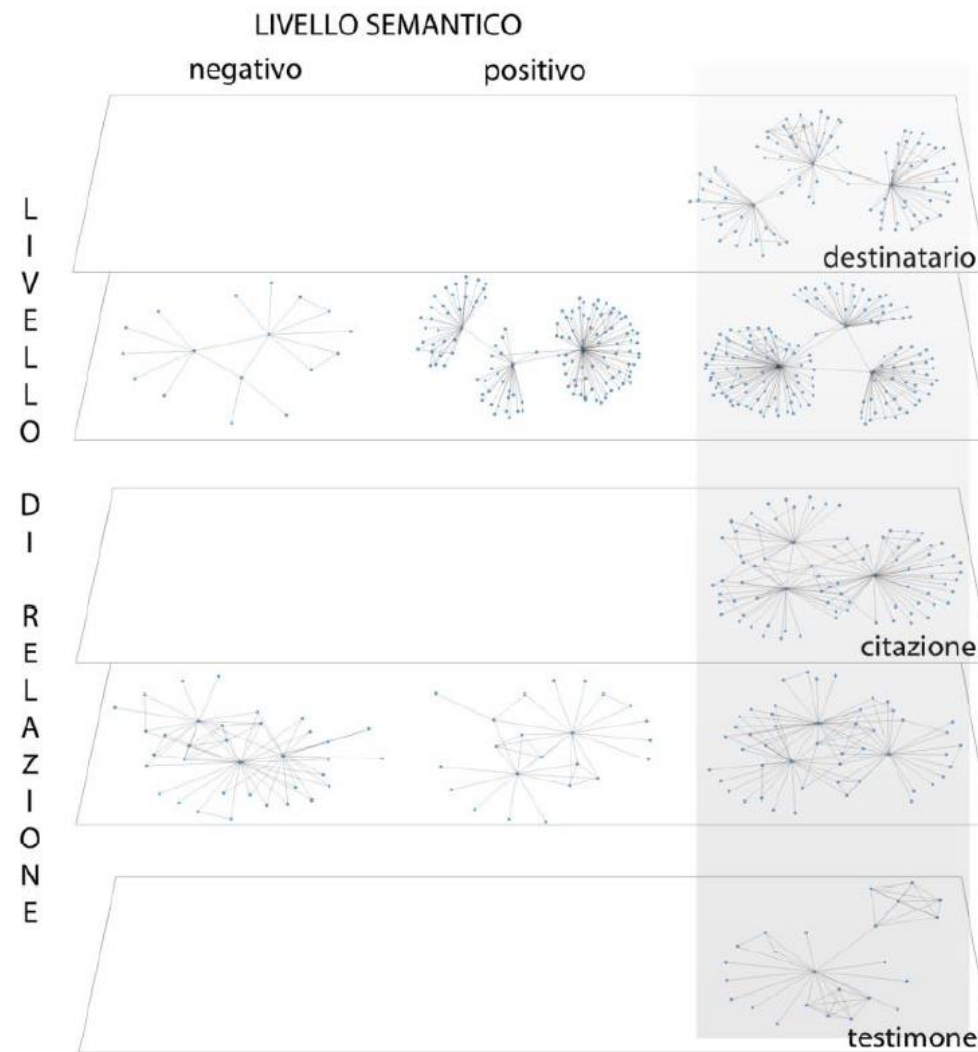
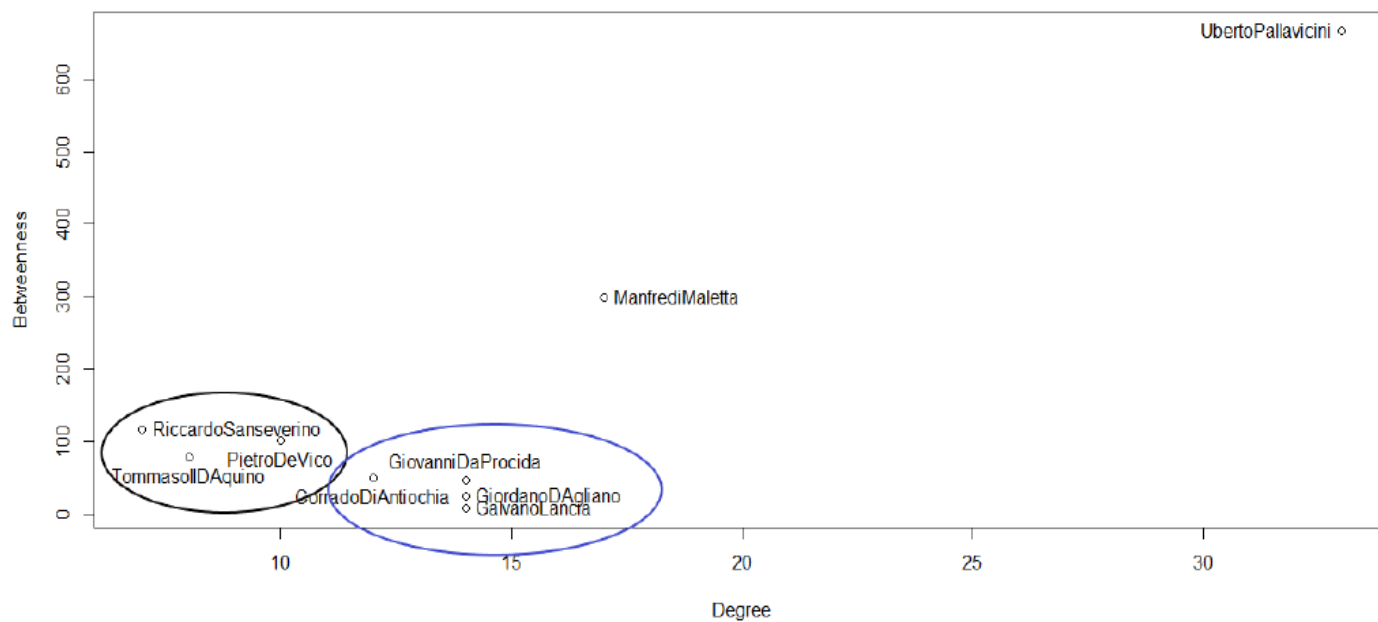


# SOCIAL NETWORK ANALYSIS



[Alessandro De Troia, L'analisi delle reti come strumento a supporto della ricerca storica. Da Ceprano a Benevento \(1266\)](#), in *AIUCD2018 - Book of Abstracts*, 2018

# SOCIAL NETWORK ANALYSIS



A. De Troia, V. Ricci, *L'analisi delle reti come strumento a supporto della ricerca storica. Da Ceprano a Benevento (1266)*, in AIUCD2018 - Book of Abstracts, 2018

# UN ESEMPIO



## People of Medieval Scotland - <https://www.poms.ac.uk/>

Home

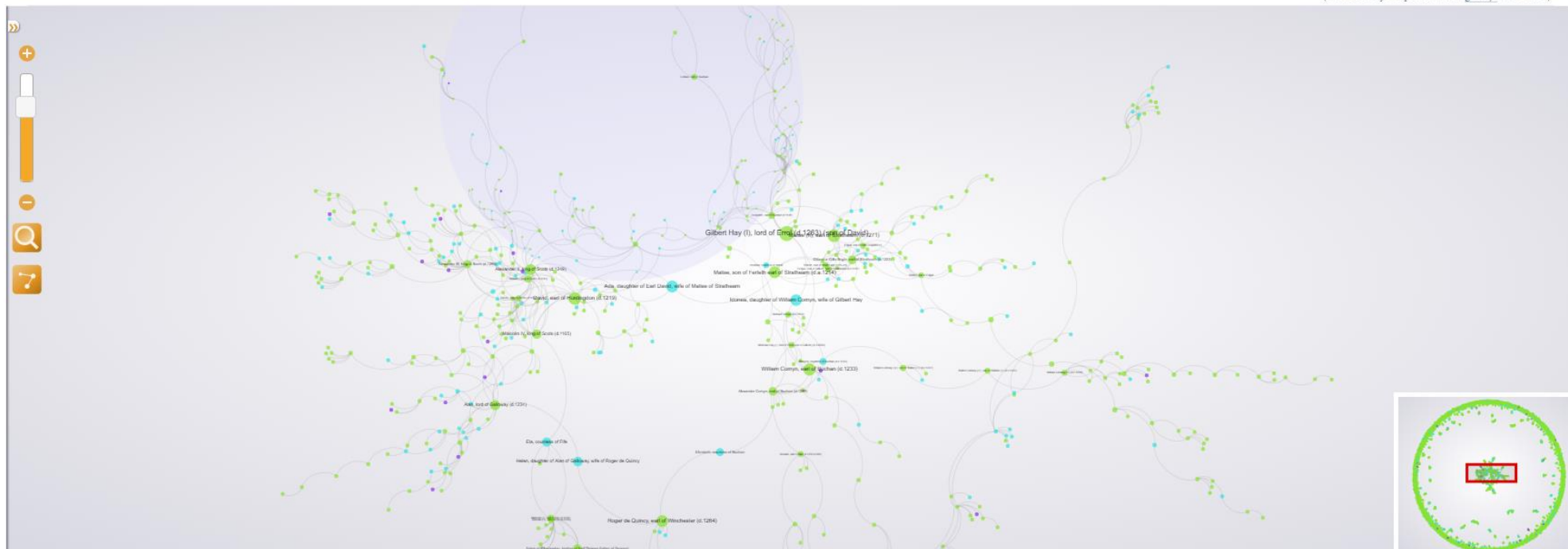
Choose graph:

Currently viewing: **Family relationships**

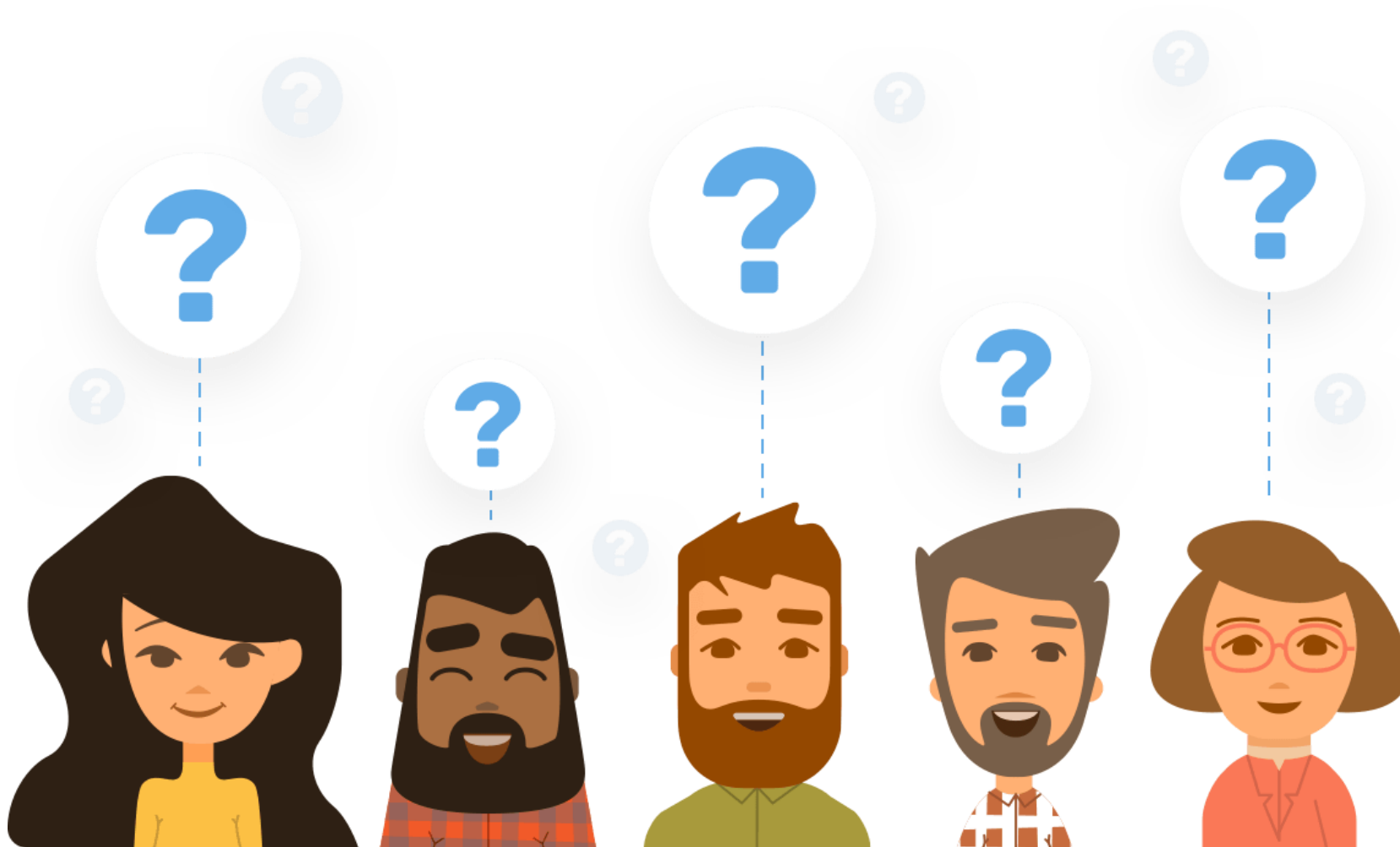
- Children / Heirs
- Female
- Male

Locate an individual:

(Powered by Raphael Velt's [gexf-js](#) software.)



# IL VOSTRO TURNO





# SONDAGGIO

<https://fast-poll.com/poll/705fcfe1>





# DOMANDE ?



# COSA VEDREMO OGGI

- Linguaggi di Markup / Formattazione
- eXtensible Markup Language (XML)
- Quiz vari
- Esercizio

# LINGUAGGI DI MARKUP / FORMATTAZIONE

Il linguaggio di *markup* è un sistema (formale) di codifica di un testo

Rappresenta l'insieme delle convenzioni che permettono di capire, per un dato documento di testo:

- Struttura
- Formattazione
- Relazioni tra le parti

# PERCHÉ ?

Nel momento in cui si ha un modo per identificare delle regole sui testi, è possibile creare dei software che riescano in maniera automatica ad interpretarle

# LINGUAGGI DI MARKUP VS LINGUAGGI DI PROGRAMMAZIONE

## MARKUP

- Sistema di annotazione dei documenti
- Interpretato da software (es. browser)
- Statico
- Utilizzato per strutturare le informazioni e per trasferirle

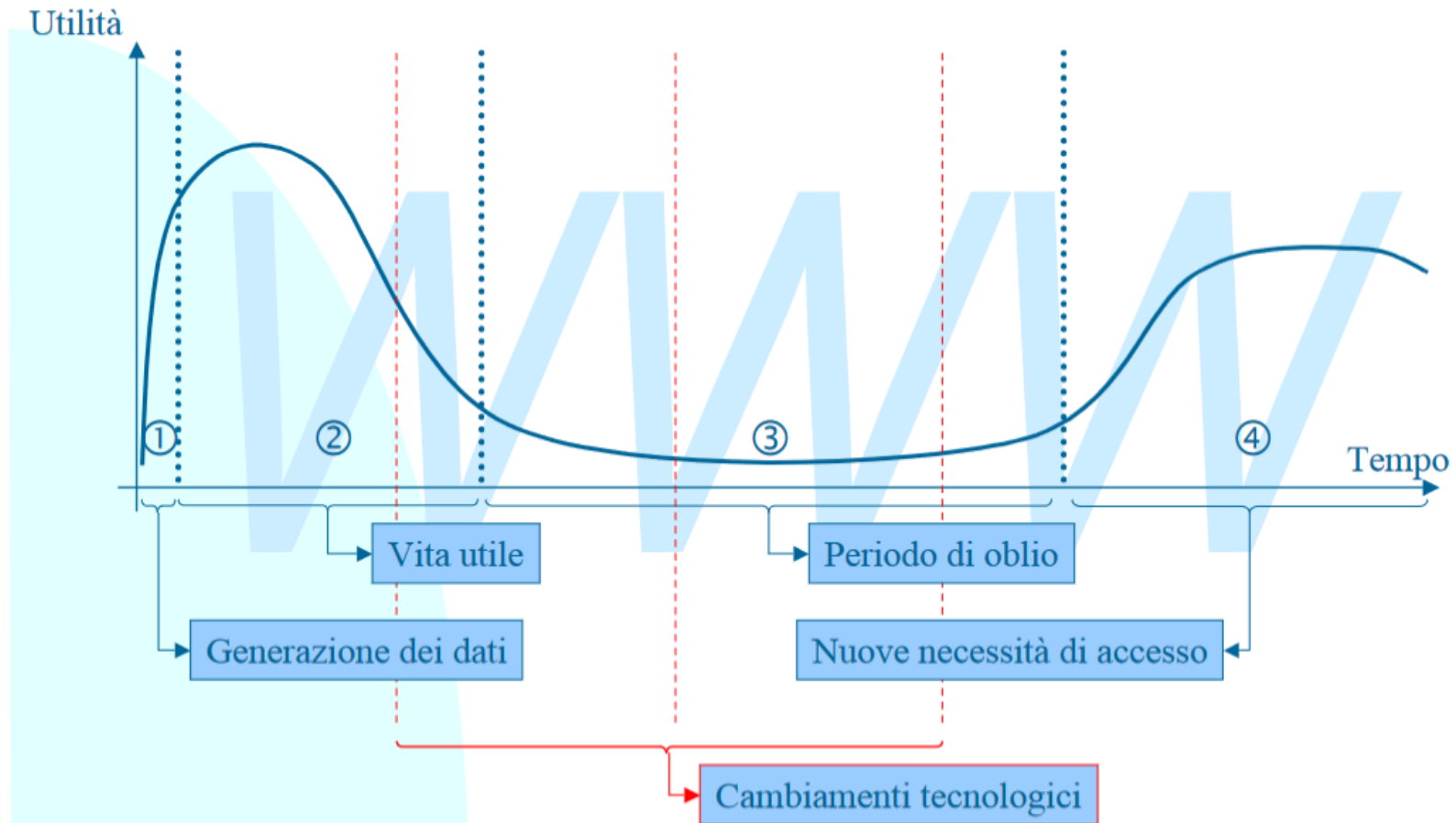
## PROGRAMMAZIONE

- Linguaggio formale che contiene istruzioni per produrre un output
- Compilato e interpretato da software
- Dinamico
- Utilizzato per fornire istruzioni ed eseguire una determinata attività

# I PROBLEMI DEL PASSATO

- Rendere i contenuti accessibili a chiunque
- Rendere il testo accessibile nel tempo
  - Hardware : obsolescenza (Archeologia informatica !!!)
  - Software: conversione dei formati (binari, proprietari, non strutturati)

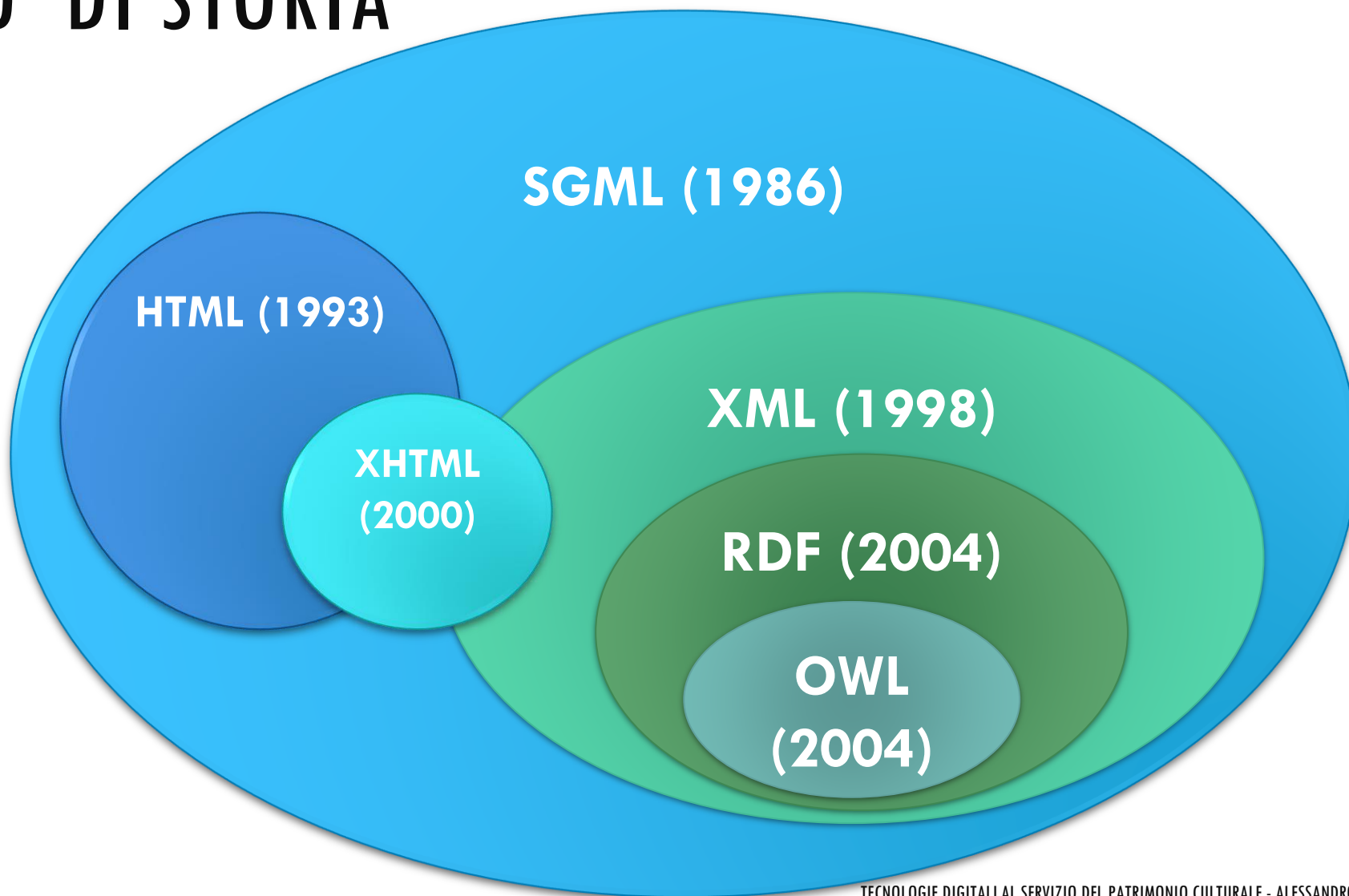
# L'utilità dei dati digitali nel tempo



Fonte: Prof. Fabio Vitali - Unibo



# UN PO' DI STORIA





# QUIZ: QUALE TRA I DUE LINGUAGGI È DI TIPO STATICO?

1. Formattazione
2. Programmazione



# EXTENSIBLE MARKUP LANGUAGE (XML)

# COS'È XML?

- L'acronimo di XML è **eXtensible Markup Language**
- XML è un linguaggio di formattazione molto simile ad **HTML**
- XML è stato progettato per immagazzinare e trasportare dati
- XML è stato anche progettato per essere **«auto-descrittivo»**
- XML è stato pensato per essere **«estendibile»**

# XML NON «FA» NULLA

- XML non nasce per effettuare operazioni
- Lo scopo principale è quello di contenere informazioni attraverso l'uso di **tag**
- Questo «contenitore» permette il trasporto delle informazioni in maniera strutturata abilitando quindi operazioni automatiche

# XML VS HTML

## XML

- Disegnato per trasportare i dati
- Focus su «cosa» è un dato
- I tag NON sono predefiniti

## HTML

- Disegnato per visualizzare i dati
- Focus su «come» i dati devono essere
- I tag sono predefiniti (ha un suo «*schema*»)

# UN PO' DI ACRONIMI

XML comprende una lunga serie di linguaggi tra cui:

- **DTD** : il linguaggio che permette di dare dei vincoli e uno «schema» ai documenti XML
- **XPATH**: sintassi per navigare e identificare gli elementi in un documento XML
- **XSLT**: il linguaggio di trasformazione dei documenti XML
- **DOM** e **SAX**: modelli e strutture dati per la programmazione
- **SOAP, WSDL**: strutture dati per l'interscambio di dati tra sistemi
- Migliaia di personalizzazioni che hanno permesso di creare degli standard di dominio specializzati in certi ambiti (es. RSS, TEI, Math-ML, etc.)

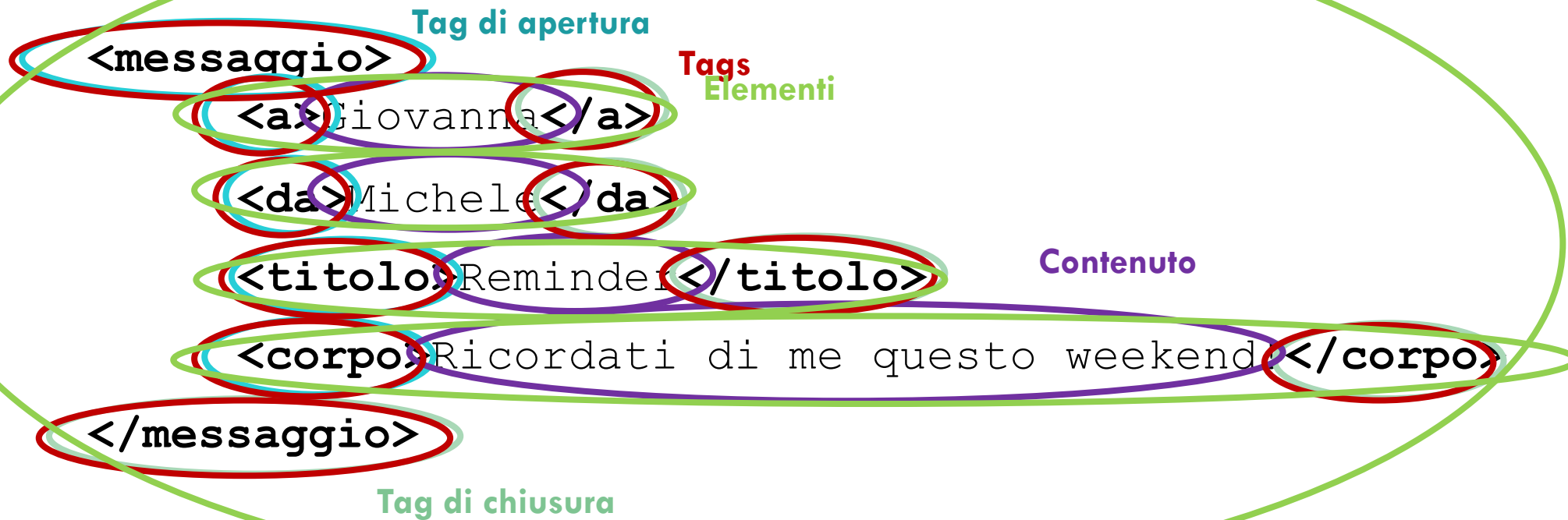
# ALCUNI VANTAGGI DI XML

- *Platform Independence*
- Struttura navigabile
- Facile convertibilità tra formati web
- Struttura gerarchica dei documenti (ad albero)
- Ripetibilità degli elementi
- Modelli di contenuto misto (equilibrio tra interscambio e formato dei dati)

# QUIZ: IN HTML I TAG SONO PREDEFINITI?

1. SI
2. NO

# XML - UN ESEMPIO



# DOCUMENTI XML - STRUTTURA

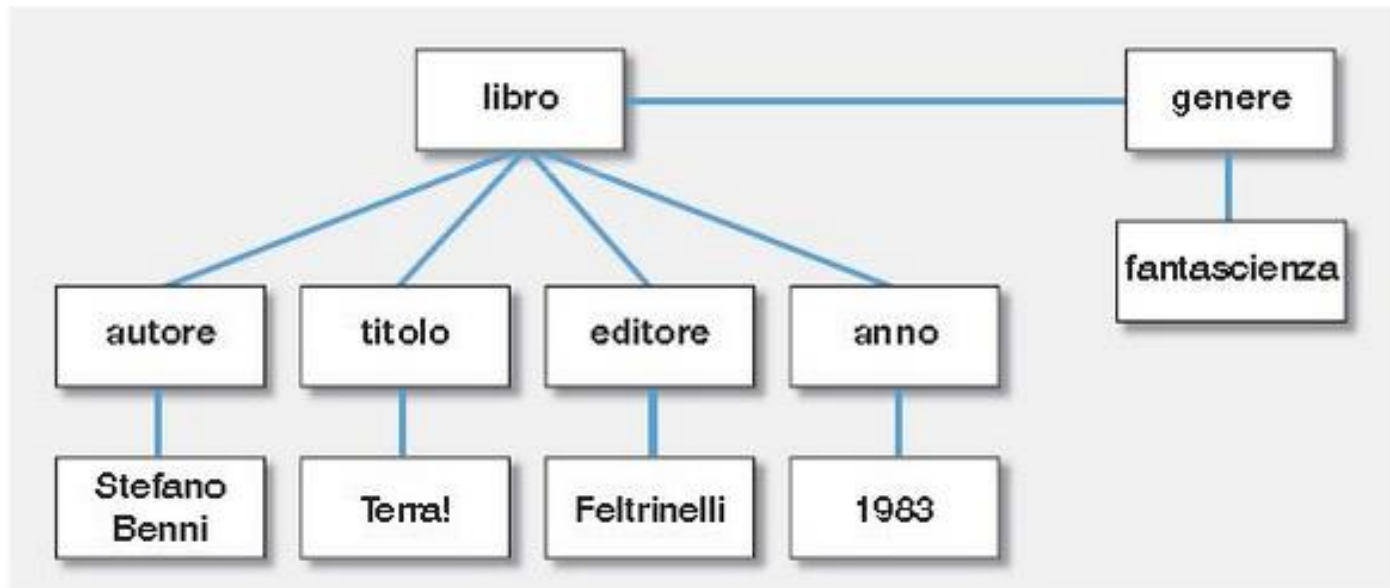
I documenti XML sono formati da «alberi di elementi»

- L'elemento principale è definito «root» (radice)
- Tutti gli altri sono contenuti in questo elemento

```
<root>
  <parent>
    <child>.....</child>
  </parent>
</root>
```

Come in un albero genealogico, i «nodi» dell'albero possono essere «figli», «genitori», «fratelli / sorelle», e così via

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<libro genere="fantascienza" xml:lang="it">
  <autore>Stefano Benni</autore>
  <titolo>Terra!</titolo>
  <editore>Feltrinelli</editore>
  <anno>1983</anno>
</libro>
```



# DICHIARARE LA VERSIONE DI UN DOCUMENTO XML

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

Definizione di un documento xml (*opzionale, ma se esiste deve essere all'inizio del documento*) con versione e codifica

Ovviamente il file dovrà avere un'estensione di tipo .xml per essere interpretato

# REGOLE SUI TAG

- Ogni tag **DEVE** essere chiuso
- I tag che hanno del contenuto devono essere chiusi esplicitamente

```
<p>This is a paragraph.</p>
```

- I tag che invece non hanno contenuto possono essere chiusi in modo implicito

```
<p />
```

# REGOLE SUI TAG (2)

- I tag sono CASE SENSITIVE, ovvero le maiuscole e le minuscole fanno la differenza

`<message>`

è diverso da

`<Message>`

- I tag DEVONO essere annidati correttamente tra loro

CORRETTO `<b><i>This text is bold and italic</i></b>`

ERRATO `<b><i>This text is bold and italic</b></i>`

# REGOLE SUI TAG (3)

- I tag non possono iniziare con «xml», un numero o un segno di punteggiatura, fatta eccezione per il carattere «underscore» (\_)

<01\_autore>

<«autore»>

<xmllike>

ERRATI

<autore>

<\_autore>

CORRETTI

# REGOLE SUI TAG (4)

- I nomi dei tag non possono contenere spazi

<pasta al forno>

ERRATI

<documento autore>

<pasta\_al\_forno>

CORRETTI

<documento\_autore>

# ATTRIBUTI

Gli elementi XML possono avere degli attributi come coppia nome / valore

Sono informazioni aggiuntive sull'elemento ma che non fanno parte del contenuto.

Il valore dell'attributo è sempre racchiuso tra virgolette

```
<nota data="12/11/2007">  
  <a>Michele</a>  
  <da>Jani</da>  
</nota>
```

# COMMENTI

È possibile definire note relative al documento o semplici commenti all'interno di un documento.

Da **NON CONFONDERE** con le note di un documento di testo o editoriale. Sono strettamente legate al file xml

```
<!-- Questo è un commento -->
```

Lo scopo dei commenti è di avere del testo che verrà ignorato da chi lo interpreterà

# DEFINIRE

Per stabilire delle regole per  
definire un DTD

```
<!--
Extensible HTML version 1.0 Strict DTD

This is the same as HTML 4 Strict except for
changes due to the differences between XML and SGML.

Namespace = http://www.w3.org/1999/xhtml

For further information, see: http://www.w3.org/TR/xhtml1

Copyright (c) 1998-2002 W3C (MIT, INRIA, Keio),
All Rights Reserved.

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"

$Revision: 1.1 $
$Date: 2018/03/20 03:13:13 $
-->

<!--===== Character mnemonic entities =====>

<!ENTITY % HTMLlat1 PUBLIC
"-//W3C//ENTITIES Latin 1 for XHTML//EN"
"xhtml-lat1.ent">
%HTMLlat1;

<!ENTITY % HTMLsymbol PUBLIC
"-//W3C//ENTITIES Symbols for XHTML//EN"
"xhtml-symbol.ent">
%HTMLsymbol;

<!ENTITY % HTMLspecial PUBLIC
"-//W3C//ENTITIES Special for XHTML//EN"
"xhtml-special.ent">
%HTMLspecial;

<!--===== Imported Names =====>

<!ENTITY % ContentType "CDATA">
<!-- media type, as per [RFC2045] -->

<!ENTITY % ContentTypes "CDATA">
<!-- comma-separated list of media types, as per [RFC2045] -->

<!ENTITY % Charset "CDATA">
<!-- a character encoding, as per [RFC2045] -->
```

è possibile

# DOCUMENTI BEN FORMATI VS VALIDI

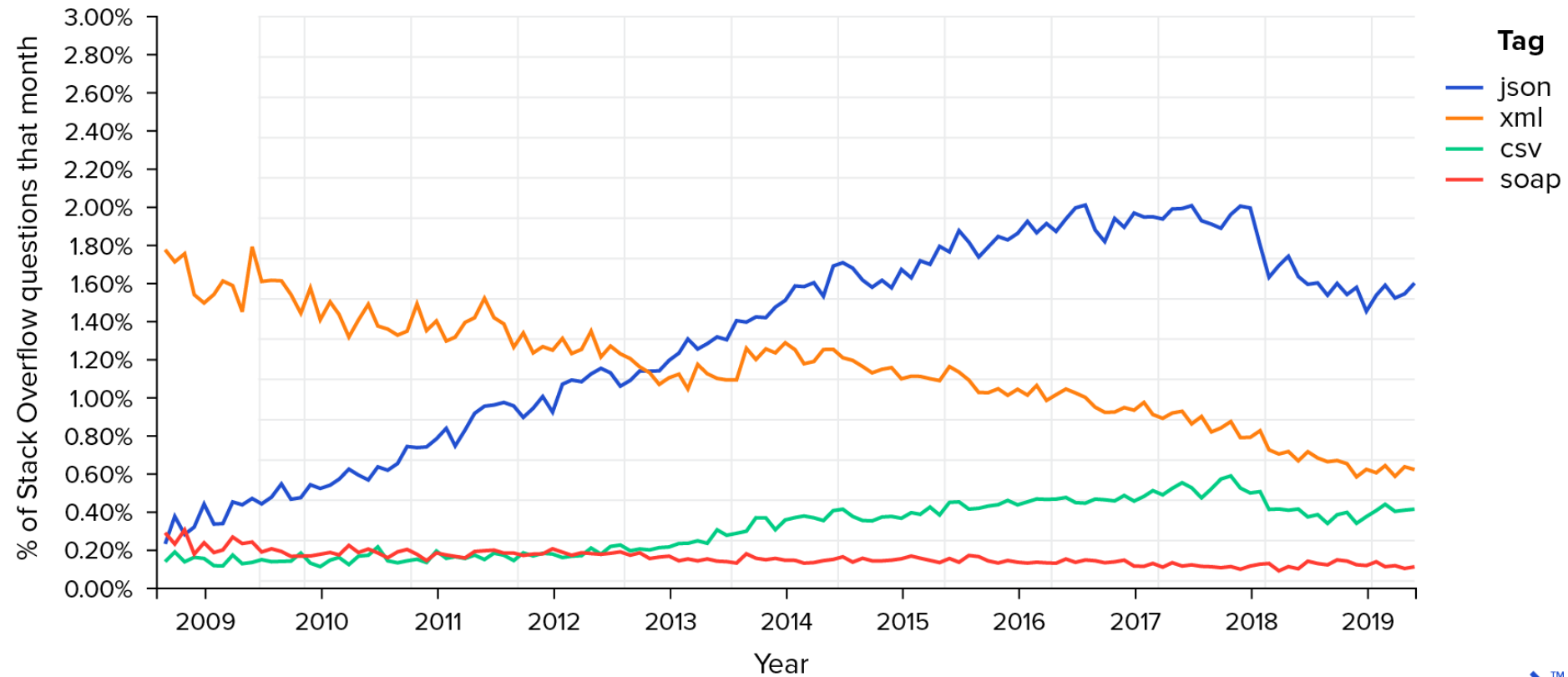
## **VALIDO**

- Presenta un DTD
- Il documento segue le regole di quel DTD

## **BEN FORMATO**

- Non presenta un DTD
- È comunque possibile controllare il documento in maniera automatica

# XML VS JSON



<https://www.toptal.com/web/json-vs-xml-part-1>



# QUIZ: UN DOCUMENTO BEN FORMATO È ANCHE UN DOCUMENTO VALIDO?

1. SÌ
2. NO



# DOMANDE ?

# QUESTO DOCUMENTO È BEN FORMATO?

```
<archivio>  
  <documento>  
    <titolo>Libro 1  
    <codice>A123</titolo>  
    </codice>  
  </documento>  
</archivio>
```

# E QUESTO?

```
<archivio>
  <documento>
    <titolo>Libro 1
      <codice>A123</codice>
    </titolo>
  </documento>
</archivio>
```

# VISUAL STUDIO CODE

<https://code.visualstudio.com/download>

XML plugin

<https://github.com/redhat-developer/vscode-xml>

# VISUAL STUDIO CODE - PERCHÈ

- Open source
- Supporto generale ad XML
- Community sviluppata e popolosa
- Supporto ai maggiori linguaggi di programmazione sul mercato
- In continua evoluzione
- Completamente configurabile
- Estendibile con plugin

# COMANDI RAPIDI IN VISUAL STUDIO CODE

- CTRL+SHIFT+P apre la palette dei comandi
  - XML: revalidate current XML file : fa la validazione del documento)
  - XML: surround with tags (Wrap) : permette di inserire un tag dopo aver selezionato del testo
- MAIUSC+ALT+F formatta il documento indentandolo

# ESERCIZIO — CATALOGO DI UNA BIBLIOTECA

Una biblioteca è composta da tanti libri

Ogni libro ha:

- Titolo
- Uno o più autori
- Codice
- Categoria

Create un esempio in XML