Tecnologie Web:
XML:Extensible Markup Language

Prof. Raffaele Montella, PhD
raffaele.montella@uniparthenope.it

Introduzione

- XML è un linguaggio di descrizione di dati
- Derivato dallo Standard General Markup Language (SGML)
- Permette l’interscambio di documenti strutturati

Markup

- Meccanismo di annotazione di documenti testuali
  - Gli *elementi* strutturano il contenuto in componenti logici
  - Gli elementi sono etichettati tramite un nome
- Le etichette sono costituite da sequenze di caratteri (tag)
- I tag sono identificati nel testo grazie a caratteri speciali

Markup

- Organizzare documenti strutturati
- Basi di dati (in senso generale)
- Documenti per office automation
- Pagine web

```
[rubrica]
        [elemento]
                [nome]Raffaele Montella[!nome]

        [email]raffaele.montella@uniparthenope.it[!email]
                [telefono]0815476672[!telefono]
        [!elemento]
        [elemento]
                [nome]Angelo Riccio[!nome]

        [email]angelo.riccio@uniparthenope.it[!email]
                [telefono]0815476613[!telefono]
```

- Linguaggio di markup fittizio
- I tag sono identificati da […]
- I tag sono chiusi da [! …]

Markup

- Un linguaggio di markup è un sistema formale per rappresentare dati strutturati

- E’ costituito da:
  - Sintassi dei tag
  - Definizione degli elementi e dei tag usati nel documento
  - Eventuale semantica dei tag

- Il documento deve essere di tipo testuale (ASCII/UNICODE)
- Nel documento deve essere specificato il codice di caratteri utilizzato

Markup

- XML non è l'unico tipo di linguaggio di markup
  - Rich Text File
  - Tex
  - HTML

- I documenti memorizzati con un formato a marcatori sono:
  - Comprensibili dagli umani
  - Interpretabili dalle macchine
  - Se standardizzati, un valido mezzo di interscambio

Metalinguaggio

- XML permette di definire convenzioni per rappresentare dati
- XML consente di mantenere i dati stessi

- Non è dedicato ad una specifica tipologia di documenti, ma è generale

- E' un metalinguaggio di markup che permette di definire altri linguaggi:
  - HTML5, SVG, SMIL, WML

Metalinguaggio

- Un esempio di valido documento XML

```
<? xml version="1.0" ?>
<addressBook>
        <entry>
                <name>Raffaele Montella</name>
                <email href="mailto:raffaele.montella@uniparthenope.it" />
                <tel>0815476672</tel>
        </entry>
        <entry>
                <name>Angelo Riccio</name>
                <email href="mailto:angelo.riccio@uniparthenope.it" />
                <tel>0815476613</tel>
                <web href="http://dsa.uniparthenope.it/angelo.riccio" />
        </entry>
</addressBook>
```

Sintassi

- Elementi
- Nomi degli elementi
- Attributi
- Commenti
- Sezioni CDATA
- Documenti ben formattati

Elementi

- <nome>contenuto</nome>

- Un elemento XML è un contenitore
  - Contiene altri elementi XML
  - Contiene caratteri

- E’ caratterizzato da:
  - Nome: identifica la tipologia di elemento
  - Contenuto:
    - Delimitato dai tag di inizio e fine <nome>…</nome>
    - Se il contenuto non è previsto si ha <nome />

Nomi degli elementi

- Il nome deve avere come primo carattere una lettera (maiuscola o minuscola) o un underscore (_)

- I caratteri successivi possono essere:
  - Maiuscoli e minuscoli
  - Underscore, punto (.), tratto alto (-)

- I nomi non possono iniziare con la stringa xml
- XML è case sensitive

Nomi degli elementi

- In XML nessun nome di elemento è definito a priori
- I nomi degli elementi sono scelti dall'autore del documento

- Famiglie di documenti possono condividere le stesse convenzioni

- Le convenzioni sono definite tramite
  - Document Type Definition (DTD)
  - XML-Schema (basato su XML)

Attributi

- <nome attr1=“val1” attr2=“val2”>…</nome>
- XML prevede la possibilità che gli elementi hanno attributi
- Gli attributi sono coppie attributo=“valore”
- Il valore dell’attributo deve essere fra apici
- Ciascun elemento può avere 0, 1, n attributi

Attributi

- L’uso degli attributi rende più leggibile un documento

- E’ sempre possibile codificare il documento in modo da usare elementi in luogo di attributi

```
<? xml version=“1.0” ?>
<addressBook>
        …
        <entry id=“001”>
                <name>Angelo Riccio</name>
                <email href=“mailto:angelo.riccio@uniparthenope.it” />
                <tel preferred=“true”>0815476613</tel>
                <web href=“http://dsa.uniparthenope.it/angelo.riccio” />
        </entry>
</addressBook>
```

Commenti

- Consentono di annotare il documento senza inficiare i dati in esso contenuti
- Possono essere estesi su più linee

```
<? xml version=“1.0” ?>
<!-- Rubrica telefonica -->
<addressBook>
        <entry>
                <name>Raffaele Montella</name>
                <email href=“mailto:raffaele.montella@uniparthenope.it” />
                <tel>0815476672</tel>
        </entry>
<!--
        Questa entry e’ commentata
        …
-->
</addressBook>
```

CDATA

- <[CDATA [contenuto testuale]]>

- Le sezioni CDATA contengono caratteri (Character Data)
- Le sezioni CDATA contengono caratteri così come sono, incluse le parentesi angolari

- Esempio d’uso:
  Inserire in un elemento XML del testo XML da non interpretare come tale

CDATA

- Esempio:

```
<? xml version=“1.0” ?>
<example>
        <title>Un esempio di CDATA</title>
        <text>
        < [CDATA [
                <entry>
                        <name>Giulio Giunta</name>
                        <email href=“mailto:giulio.giunta@uniparthenope.it” />
                </entry>
        ]]>
        </text>
</example>
```

Documenti ben formattati

- Un documento XML si dice ben formattato se:
  - Ogni tag di inizio ha un corrispondente tag di fine
  - Gli elementi sono annidati correttamente
  - I valori degli attributi sono racchiusi fra apici
  - L’elemento radice del documento è unico

- Un documento XML ben formattato è rappresentabile attraverso un albero

Documenti ben formattati

- Il primo elemento ha ruolo di radice
- Gli elementi contenuti sono figli del contenitore
- Le foglie sono elementi che contengono solo testo o una sezione CDATA
- Ogni documento descrive soltanto un albero
- I figli devono esse completamente contenuti nei genitori (annidamento corretto)

Schemi di markup

- Un documento XML ben formattato è corretto dal punto di vista sintattico
- Non è controllata:
  - La correttezza dei nomi degli elementi
  - La coerenza della struttura ad albero
- Un documento può essere sintatticamente corretto, ma avere una struttura libera

Schemi di markup

- Spesso è necessario imporre una struttura ad un file XML in modo da garantire l'adesione ad uno standard:
  - DTD: Document Type Definition
  - XML Schema: basato su XML

- Entrambi gli strumenti assolvono al medesimo scopo.

- Attualmente XML Schema ha maggiore diffusione

Document Type Definition

- E’ una definizione formale della grammatica di una tipologia di documenti
  - Elementi utilizzabili (nomi)
  - Attributi (nomi, obbligatorietà, valori di default)
  - Struttura dell’albero
- Un DTD è un documento testuale
- Contiene la definizione di ogni elemento presente nel documento XML
- Un documento XML è associato ad un DTD o a un XML-Schema con un’opportuna intestazione

Document Type Definition

- Un documento XML a cui è associato un DTD o un XML Schema si dice valido se:
  - E’ ben formato
  - Rispetta la definizione dello schema

- La validità di un documento rispetto ad uno schema è verificata da strumenti automatici

Associare XML a DTD

- Intestazione del documento XML
- Dichiarazione di tipo documento
- <!DOCTYPE radice specificaDTD>
- La specifica DTD può essere espressa come:
  - Intero documento DTD racchiuso fra parentesi quadre []
  - Un riferimento URI al documento DTD
  - Una dichiarazione che fa riferimento a un documento DTD registrato

Associare DTD a XML

- Esempio:

```
<?xml version="1.0" ?>
<!DOCTYPE addressBook SYSTEM "addressbook.dtd">
<addressBook>
        <entity>…</entity>
        <entity>…</entity>
</addressBook>
```

- Limiti
  - Non è possibile definire tipi di dati per elementi ed attributi
  - Non usa la sintassi XML

- Soluzione: XML-Schema

Conclusioni

- I linguaggi di markup consentono di mescolare testo semplice ed annotazioni
- XML è un linguaggio di markup estensibile poiché è definito tramite regole sintattiche, ma non sono definiti elementi a priori
- E’ possibile definire la struttura di un documento XML in modo da verificarne la correttezza e l’aderenza ad uno standard.
- DTD e XML Schema sono due specifiche per la definizione di formati di documenti XML

Tutorial

1. Installare Docker (http://docker.com)
2. Attivare un ambiente ubuntu
   a. **MacOS:**
      docker run --name ubuntu -e HOST_IP=$(ifconfig en0 | awk '/ *inet /{print $2}') -v $HOME:/host -t -i ubuntu /bin/bash
   b. **Linux:**
      docker run --name ubuntu -e HOST_IP=$(ifconfig eth0 | awk '/ *inet /{print $2}') -v $HOME:/host -t -i ubuntu /bin/bash
   c. **Windows:**
      docker run --name ubuntu -e HOST_IP=[host_ip] -v //c/Users/[username]:/host -t -i ubuntu /bin/bash

Tutorial

1. Aggiornare l’elenco dei pacchetti
   apt-get update

2. Aggiornare ubuntu
   apt-get upgrade

3. Installare vim e libxml2-utils
   apt-get install vim libxml2-utils

4. Scrivere i seguenti file usando vi

notes.xsd

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="note">
  <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>
```

note.xml

```xml
<?xml version="1.0"?>
<note>
	<to>Tove</to>
	<from>Jani</from>
	<heading>Reminder</heading>
	<body>Don't forget me this weekend</body>
</note>
```

note_not_valid.xml

```
<?xml version="1.0"?>
<note>
      <to>Tove</to>
      <from>Jani</from>
      <subject>Reminder</subjsct>
      <body>Don't forget me this weekend</body>
</note>
```

Validate!

```
xmllint --schema notes.xsd note.xml --noout
```

```
xmllint --schema notes.xsd note_not_valid.xml
```

Esercizi

- Scrivere un documento di descrizione di specifiche per le seguenti applicazioni:
  - Catalogo di contenuti multimediali di una webTV
  - Definizione di una cartografia tematica in un contesto di GIS
  - Mantenimento di dati acquisiti con cadenza temporale da una stazione meteorologica

- Per ogni esempio fornire un documento XML di provata validità

Riferimenti

- Tutorial vim: https://www.youtube.com/watch?v=g-XsXEsd6xA
- Tutorial Docker: https://medium.com/@hudsonmendes/docker-have-a-ubuntu-development-machine-within-seconds-from-windows-or-mac-fd2f30a338e4
- Docker: https://www.docker.com/get-started
- W3Schools: https://www.w3schools.com