FACOLTÀ DI INGEGNERIA
UNIVERSITÀ DEGLI
STUDI DI PALERMO

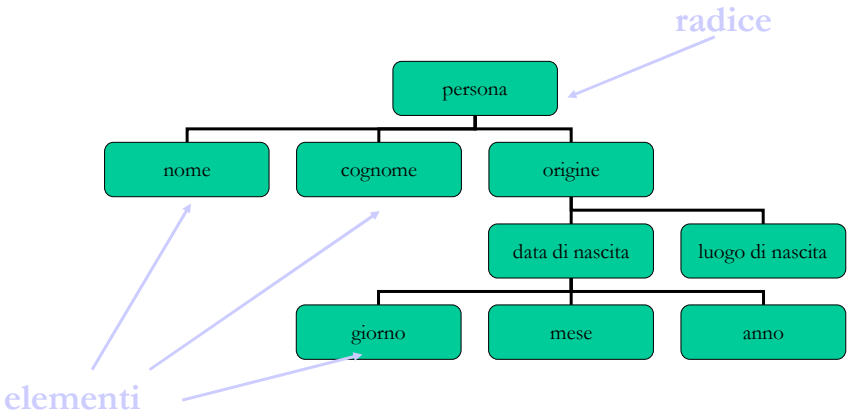
Corso di Laurea Specialistica
Ingegneria Gestionale

Sistemi ICT per il Business Networking

XML

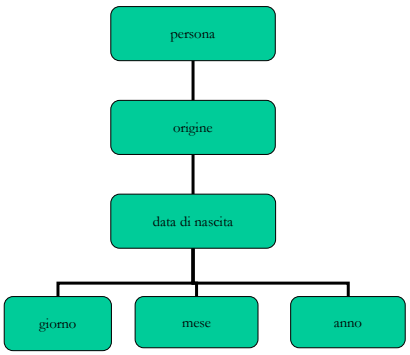
Docente: **Vito Morreale** (vito.morreale@eng.it)

Struttura ad albero



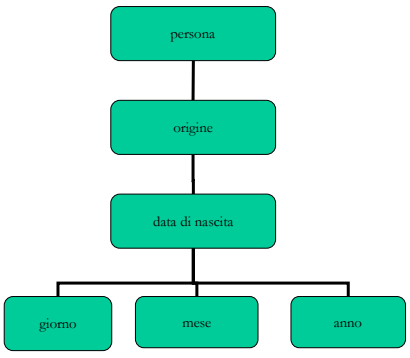
Struttura del file XML

- Un documento XML è usato per descrivere un **albero** gerarchico.
- Il primo elemento è chiamato **radice**.
- Un documento può avere **una sola radice**.
- **Ogni elemento** può contenere **altri elementi al suo interno**.



Relazioni

- La linea che collega due elementi della gerarchia individua una **relazione padre-figlio**
- L'elemento **padre** è quello che 'contiene' gli elementi **figli**



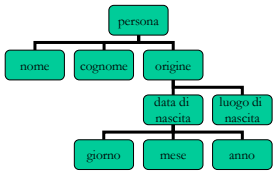
Un esempio di file XML

```
<?xml version="1.0"?>
<persona>
        <nome>Luigi</nome>
        <cognome>Pirandello</cognome>
        <origine>
                <data_di_nascita>
                        <giorno>28</giorno>
                        <mese>giugno</mese>
                        <anno>1867</anno>
                </data_di_nascita>
                <luogo_di_nascita>Agrigento</luogo_di_nascita>
        </origine>
</persona>
```

dichiarazione XML

dato

tag



I TAG

- Si distingue tra
  - **tag iniziali** < persona >
  - e **tag finali** </persona >

I TAG

- In linguaggi di mark-up come l'**HTML**, l'insieme dei tag che possono essere usati è ben codificato.
- A differenza del HTML, l'insieme dei **tag XML** può essere **definito dall'utente**.
- Il **nome di un tag** può essere una **qualunque stringa di testo** che rispetta certe regole.

Nomi XML

- I **nomi** usati per i tag possono essere costituiti da qualsiasi **carattere alfanumerico**
  - a-z,
  - A-Z,
  - 0-9
  - lettere accentate (es. ò, à, è, …)
  - ideogrammi ç
  - underscore _, trattino -, punto .

Nomi XML

- **Non possono essere usati**:
  - spazi
  - virgolette ", apostrofi ', il simbolo $
  - i simboli < e >
  - il simbolo % e il punto e vigola ;
- Il nome usato per un tag è sensibile alla differenza tra maiuscole e minuscole (**case sensitive**)
  - Il tag <Persona> è diverso da <persona>

I dati

- Tra un tag di apertura e un tag di chiusura possono essere **inseriti**:
  - altri tag
  - dei dati testuali
  - contenuto misto

```
<data_di_nascita>
        <giorno>28</giorno>
        <mese>giugno</mese>
        <anno>1867</anno>
</data_di_nascita>
```

```
<p>
        lezione di <b>XML</b>
</p>
```

Attributi

- Ad un tag possono essere associati degli **attributi**
- Un attributo consiste in una **coppia nome-valore**:
  - <persona nome="Mario" cognome="Rossi">
  - <persona nome="Valeria" cognome="Bianchi">

Commenti

- I documenti XML possono contenere dei **commenti** e delle note.
- Un commento si può inserire usando la **sintassi**:
  - <!-- Commento -->

La dichiarazione XML

- I documenti XML dovrebbero iniziare con una **dichiarazione** XML:
  - <?xml version="1.0" encoding="UTF-8"?>
- dove:
  - **version** identifica la versione di XML in uso. Attualmente la versione in uso è la 1.0
  - **encoding** specifica quale tipo di codifica deve essere usata all'interno del documento: XML supporta un grande quantità di set di caratteri.

Documenti ben formati

- Affinché un documento XML sia **ben formato** è necessario rispettare alcune regole sintattiche.
  1. ad ogni **tag iniziale** deve corrispondere un **tag finale**
  2. gli elementi non possono **sovrapporsi**
  3. deve esistere esattamente un **elemento radice**
  4. i **valori degli attributi** devono essere specificati tra **apici**
  5. i **commenti** non possono apparire all'interno dei tag
  6. non deve apparire nessun carattere **<** o **&** all'interno dei caratteri di un elemento o di un attributo

Regola 1

- Ad ogni tag iniziale deve corrispondere un tag finale
  - <giorno>28</giorno>
  - <mese>giugno</mese>
  - <anno>1867</anno>
- Alternativa:
  - <giorno/> **<!-- Tag iniziale e finale -->**

Regola 2

- Gli elementi non possono sovrapporsi
- Non valido:
  - <giorno>28<mese>giugno</giorno></mese>
- Valido:
  - <giorno>28</giorno>
  - <mese>giugno</mese>

Regola 3

- Deve esistere esattamente un elemento radice

```
<?xml version="1.0" ?>
<persona>
        <nome>Luigi</nome>
</persona>
<persona>
        <nome>Mario</nome>
</persona>
<persona>
        <nome>Valeria</nome>
</persona>
```

```
<?xml version="1.0" ?>
<elenco_persone>
        <persona>
                <nome>Luigi</nome>
        </persona>
        <persona>

        <nome>Mario</nome>
        </persona>
        <persona>

        <nome>Valeria</nome>
        </persona>
</elenco_persone>
```


Regola 4

- I **valori degli attributi** devono essere specificati tra **virgolette**
- Non valido:
  - <persona nome=Mario cognome=Rossi>
- Valido:
  - <persona nome="Mario" cognome="Rossi">

Regola 5

- I commenti non possono apparire all'interno dei tag
- Non valido:
  - <persona <!– una persona--> >
- Valido:
  - <persona> <!– una persona-->

Regola 6

- Non deve apparire nessun carattere < o & all'interno dei caratteri di un elemento o di un attributo
- Non valido:
  - <espressione> a > b </espressione>

Referenze ad entità

- Per superare i limiti di non poter usare certi caratteri (>,<,% caratteri accentati) come testo interno ai tag si usano delle combinazioni speciali chiamate **referenze ad entità**
- **<** -> il carattere minore **<**
- **&** -> il carattere **&**
- **>** ->il carattere maggiore **>**
- **"** -> le virgolette **"**
- **'** -> apostrofo **'**

Sezioni CDATA

- Quando un documento XML contiene delle **grosse porzioni di testo**, usare le referenze ad entità può risultare fastidioso.
- Per semplificare la scrittura di documenti XML si introducono le **sezioni CDATA**
- **All'interno di una sezione CDATA** può essere usato **qualunque carattere** si desidera, compreso quelli che normalmente non possono essere usati.

Sezione CDATA

- Una sezione CDATA **inizia** con la sequenza
  - **<![CDATA[**
- e **termina** con la sequenza
  - **]]>**
- Ad esempio:
  - <![CDATA[qui dentro posso usare tutti i caratteri che desidero compreso > < % " ' oppure i caratteri accentati à è ì ò ù]]>

Elementi vuoti

- Quando un elemento non deve avere **nessun contenuto** esso può apparire in due forme.
- Nella forma estesa:
  - <elemento></elemento>
- Oppure nella forma contratta:
  - <elemento/>

Differenza tra XML e HTML

- **HTML** descrive sia l'aspetto che il contenuto di una pagina Web; **XML** invece descrive soltanto il contenuto.
- **HTML** ha una sintassi chiusa, non possono essere definiti ulteriori tag, per aggiungere potenza ad HTML si devono usare altre tecnologie; **XML** invece è totalmente configurabile.

Documenti validi

- Spesso la "**ben formazione**" di un documento XML **non è sufficiente**
- Per determinate applicazioni può essere necessario definire **ulteriori regole di composizione dei tag**
  - Ad esempio si può voler che il tag <persona> contenga sempre un tag <nome> e un tag <cognome>

Documenti validi

- Un file XML si dice anche **valido** se rispetta una serie di regole
- Le regole generalmente vengono definite in file esterno chiamato **DTD (Document Type Definition)**

DTD

- Lo strumento in grado di fornire delle regole di composizione strutturale è DTD.
- Un **documento DTD** descrive con precisione **quali elementi ed entità** possono apparire all'interno di un documento, quali possono essere i contenuti e gli attributi di tali elementi.

DTD interne ed esterne

- Le DTD sono introdotte **all'interno dei documenti XML** utilizzando la **Document Type Declaration** o dichiarazione di tipo del documento.
- Una **Document Type Declaration** è posta nel prologo di un documento XML ed è compresa fra **<!DOCTYPE** e **>**
  - Può puntare a dichiarazioni **esterne** al documento o può contenere le dichiarazioni all'**interno** del documento

Dichiarazioni di DTD

- **Dichiarazione DTD interna nel file XML**

```
<!DOCTYPE myMessage [
 <!ELEMENT myMessage ( #PCDATA ) >
]>
```

- **Dichiarazione DTD esterna nel file XML**

```
<!DOCTYPE myMessage SYSTEM "myDTD.dtd">
```

SYSTEM si usa per **DTD non prese dal web** (cioè non pubbliche); la sintassi è:

<!DOCTYPE radice SYSTEM url>

- **per DTD di pubblico dominio si usa PUBLIC**

```
<!DOCTYPE HTML PUBLIC
 "-//W3C//DTDHTML4.01//EN"
 "http://www.w3.org/TR/html4/strict.dtd">
```

Validazione

- Quando un documento XML rispetta le regole imposte da un determinato DTD si dice che risulta **valido** per l'applicazione specifica.
- Rispetto alle regole di ben formazione, **le regole di validazione sono opzionali**.
- Lo scopo di tali regole è quello di generare documenti che possono essere **capiti** dalle applicazioni.

Principio di base

- Un DTD opera sul principio che **tutto ciò che non è esplicitamente dichiarato come permesso allora deve essere considerato vietato**.

Dichiarazione di elementi

- **Ogni elemento utilizzato** in un documento valido deve essere **dichiarato nel DTD**
- La **sintassi** è la seguente:
  - <!**ELEMENT** nome_elemento (modello_di_contenuto) >
- Dove:
  - il **nome di elemento** può essere un qualunque nome XML legale

Modello di contenuto

- Il **modello di contenuto** dichiara quali figli possa o debba avere l'elemento in questione e in quale ordine.
- I modelli di contenuto possono essere scelti tra **sette diversi tipi di elementi**
  - Testo valido XML (#PCDATA)
  - Elementi figli
  - Sequenze (,)
  - Scelte (|)
  - Elementi vuoti (EMPTY)
  - Elementi di qualunque tipo (ANY)
  - Contenuto misto

#PCDATA

- E' **il più semplice** modello di contenuto
- Specifica che l'elemento può contenere solamente dati di tipo **carattere**
- Esempio:
  - <!ELEMENT **cognome** (#PCDATA)>
- Corrisponde a:
  - **<cognome>Pirandello</cognome>**

Elementi figli

- Specifica che un elemento deve contenere esattamente un **elemento figlio** di un determinato tipo.
- Esempio:
  - <!ELEMENT fax (numero_di_telefono)>
- Corrisponde a:

```
<fax>
    <numero_di_telefono>091123456</numero_di_telefono>
</fax>
```

La specifica di numero_di_telefono sarà:

<!ELEMENT numero_di_telefono (#PCDATA)>

Sequenze

- Generalmente gli elementi contengono **più di un elemento** figlio.
- Per specificare quali elementi devono essere contenuti è sufficiente elencarli **separandoli con delle virgole**.
- Esempio:
  - **<!ELEMENT persona (nome, cognome, origine)>**

Numero di figli

- E' possibile aggiungere un suffisso per specificare **quante istanze di figli sono permessi** per un dato elemento.
- Suffissi
  - **?** corrisponde a "zero o un elemento"
  - ***** corrisponde a "zero o più elementi"
  - **+** corrisponde a "uno o più elementi"

Esempi

<!ELEMENT name (first_name, middle_name**?**,
last_name)>

<!ELEMENT name (first_name**+**, last_name)>

<!ELEMENT persona (nome, cognome, professione*****)>

Scelte

- Può capitare che il contenuto di un elemento debba essere **selezionato tra un elenco di possibilità**.
- Una scelta è un elenco di nomi separati da barre verticali **|**
- Esempio:

<!ELEMENT mese (**gennaio | febbraio | marzo | aprile | maggio | giugno | luglio | agosto | settembre | ottobre | novembre | dicembre**)>

Uso delle parentesi

- E' possibile **combinare scelte e sequenze** anche complesse usando le **parentesi**.
- Esempio:
  - <!ELEMENT cerchio (**centro, (raggio | diametro)**)>
- Corrisponde a:

```
<cerchio>
  <centro>12,5</centro>
  <raggio>24</raggio>
</cerchio>
```

```
<cerchio>
  <centro>12,5</centro>
  <diametro>48</diametro>
</cerchio>
```


Elementi vuoti

- Alcuni elementi, chiamati **elementi vuoti**, non hanno nessun contenuto.
- Gli elementi vuoti vengono dichiarati usando la parola **EMPTY**.

<!ELEMENT poeta EMPTY>

- Corrisponde a:

```
<persona>
  <nome>Dante</nome>
  <cognome>Alighieri</cognome>
  <poeta/>
</persona>
```

ANY

- Alcuni DTD particolarmente liberi permettono di specificare un **contenuto arbitrario**.
- Tale modello viene dichiarato usando la parola **ANY**.

```
<!ELEMENT page ANY>
```

- **I figli di page possono essere di qualunque tipo**: qualsiasi elemento (compreso page), testo, oppure contenuto misto.

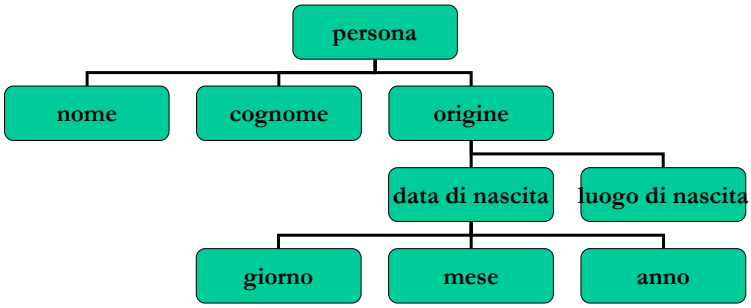
Contenuto misto

- Per definire che il contenuto di un certo elemento può essere **misto** si usa la seguente sintassi:

```
<!ELEMENT page (#PCDATA | elemento_figlio)* >
```

- Questa dichiarazione indica che **l'elemento `page` può contenere qualunque combinazione di caratteri e di elementi di tipo `elemento_figlio`**.

Struttura



Esempio di DTD

dichiarazione di elemento

```
<!ELEMENT persona (nome, cognome, origine)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT cognome (#PCDATA)>
<!ELEMENT origine (data_di_nascita, luogo_di_nascita)>
<!ELEMENT data_di_nascita (giorno, mese, anno)>
<!ELEMENT luogo_di_nascita (#PCDATA)>
<!ELEMENT giorno (#PCDATA)>
<!ELEMENT mese (#PCDATA)>
<!ELEMENT anno (#PCDATA)>
```

Dichiarazione di attributi

- Oltre a dichiarare gli elementi di un documento, deve essere possibile dichiarare anche gli **attributi**.

Dichiarazione di attributi

<!ATTLIST elemento attributo caratteristiche >

- **elemento** è il nome dell'elemento che deve contenere l'attributo
- **attributo** è il nome dell'attributo
- **caratteristiche** è un insieme di parametri che specificano
  - il **tipo** di attributo: CDATA, ID, IDREF, ENTITY (ENTITIES), NMTOKEN (NMTOKENS)
  - se l'attributo è **obbligatorio** oppure **opzionale**

Tipi di attributo: CDATA

- Un attributo **CDATA** può contenere una qualsiasi stringa di testo accettabile da XML
- Esempio:
  - <persona **nome**="Luca Sabatucci">
- La cui specifica è:
  - **<!ATTLIST persona nome CDATA >**

Valore di presenza

- Ogni dichiarazione ATTLIST può includere un valore di presenza dell'attributo
  - **#IMPLIED**
    - Indica che l'attributo è opzionale
  - **#REQUIRED**
    - Indica che l'attributo è obbligatorio
  - **#FIXED**
    - Indica che l'attributo è costante e non può essere variato

Tipi di attributo: ID

- ID serve ad assicurare l'**unicità del valore dell'attributo**
- Se nel file XML viene due volte assegnato lo stesso valore all'attributo, allora il parser riporta un errore

```
<!ELEMENT Consegna (tempo)>
<!ATTLIST Consegna IDconsegna ID #REQUIRED >
```


Tipi di attributo: IDREF

- Specifica un riferimento al valore di un campo di tipo ID

```
<!ELEMENT libro (#PCDATA)>
<!ATTLIST libro TempoConsegna IDREF #IMPLIED>
```

Esempio di ID e IDREF

```
<?xml version="1.0"?>
<!DOCTYPE Libreria [
   <!ELEMENT Libreria (consegna+,
   libro+)>
   <!ELEMENT consegna (tempo)>
   <!ATTLIST consegna
   IDconsegna ID #REQUIRED>
   <!ELEMENT libro (#PCDATA)>
   <!ATTLIST libro
   ConsegnaEntro IDREF #IMPLIED>
   <!ELEMENT tempo (#PCDATA)>
]>
```

```
<Libreria>
   <consegna IDconsegna="s1">
    <tempo>2 to 4 giorni</tempo>
   </consegna>
   <consegna IDconsegna="s2">
    <tempo>1 giorno</tempo>
   </consegna>
   <libro ConsegnaEntro="s2">
   Java How to Program 3rd edition.
  </libro>
   <libro ConsegnaEntro="s2">
   C How to Program 3rd edition.
  </libro>
   <libro ConsegnaEntro="s1">
   C++ How to Program 3rd edition.
  </libro>
</Libreria>
```


Tipi di attributo: ENTITY

- Le **entità** sono dei **valori costanti** che possono essere definiti nel DTD (alcuni sono già definiti di default come < > &…) e poi usati nel file XML
- Dichiarazione

```
<!ENTITY cifre "0123456789">
```

- Utilizzo dell'entità (si premette la **&** al nome e si chiude con **;**)

```
<useAnEntity>&cifre;</useAnEntity>
```

  - E' **equivalente a**:

```
<useAnEntity>0123456789</useAnEntity>
```

Tipi di attributo: ENTITIES

- Se le entità contenute sono **più di una** si usa la parola chiave ENTITIES
- Dichiarazione DTD

```
<!ATTLIST directory file ENTITIES>
```

- Utilizzo nel file XML

```
<directory file = "animaz img1 img2">
```

1 aprile 2006

Tipi di attributo: NMTOKEN

- Il valore di un attributo di tipo **NMTOKEN** può essere formato da lettere, cifre e quattro altri caratteri (. : _ -)
- Dichiarazione DTD

```
<!ATTLIST Palestra telefono NMTOKEN>
```

- Utilizzo

```
<Palestra telefono = "091.123456">
```

1 aprile 2006

Tipi di attributo: NMTOKENS

- Il valore di un attributo di tipo **NMTOKENS** può essere formato da più stringhe di tipo NMTOKEN ognuna separata da uno spazio

Valore di presenza

- **Esempio**:

```
<!ATTLIST professione da CDATA #IMPLIED
                       a CDATA #IMPLIED >

<!ATTLIST persona      nome CDATA      #REQUIRED
                       cognome CDATA   #REQUIRED  >

<!ATTLIST persona   nazionalità CDATA      #FIXED "italiana" >
```

Materiale didattico

- Fondamenti di XML
- Document Type Definition (DTD)

Ulteriori Informazioni

- Dove reperire ulteriori informazioni su XML
  - W3C (World Wide Web Consortium) http://www.w3.org/XML/
  - http://www.w3schools.com/default.asp
  - http://www.html.it/xml/guida/index.html
  - www.xml.org