Cenni sulla programmazione per il Web

SGML, HTML, XHTML, DTD

- SGML e HTML
- Standard Generalized Markup Language (SGML)
  - Document Type Definition (DTD) per SGML
- HTML
  - Struttura globale di un documento
  - un esempio di documento
  - DTD per HTML
  - sezioni documento
  - validità

HyperText Markup Language (HTML)

- Linguaggio di mark-up di ipertesti
  - ipertesto: rappresentazione dell'informazione mediante una collezione di documenti collegati da riferimenti (link)
  - linguaggio di mark-up: linguaggio che fornisce un insieme di annotazioni (mark-up) per arricchire i documenti di testo in termini di
    - struttura
    - significato degli elementi
    - visualizzazione
- realizzato a partire dal linguaggio SGML

- Metalinguaggio per la definizione di un linguaggio di mark-up
  - standard internazionale per rappresentare testi in formato elettronico che siano
    - Device-indipendent
    - System-indipendent
  - realizzato nel 1986

- Document Type Definition (DTD), i.e. i documenti possiedono un tipo espresso in termini di parti componenti (elementi) e relazioni
- Categorizzazione descrittiva (o mark-up descrittivo) di un documento mediante dei marcatori (tag), e.g.
  - <para> da intendersi come "l'elemento che segue è un paragrafo"
- Indipendenza dei dati: trasportabilità dei documenti da una piattaforma hw/sw ad un'altra senza perdita di dati

<!ELEMENT anthology - - (poem+)>
<!ELEMENT poem - - (title?, stanza+)>
<!ELEMENT title - O (#PCDATA) >
<!ELEMENT stanza - O (line+) >
<!ELEMENT line O O (#PCDATA) >

in SGML il nome ed il modello del contenuto sono separati da un'ulteriore parte dichiarativa che specifica le regole di minimizzazione dell'elemento considerato. In altri termini indica se il macatore iniziale e finale devono essere presenti in ogni occorrenza dell'elemento definito

```
<!DOCTYPE tei.2 SYSTEM "tei2.dtd" [
  <!ENTITY tla "Three Letter Acronym">
   <!ELEMENT my.tag - - (#PCDATA)>
   <!-- any other special-purpose declarations or re-
definitions go in here -->
]>
<tei.2> This is an instance of TEI.2 type document,
which may contain   <my.tag>my special tags</
my.tag> and references to my usual entities such as
&tla;.
</tei.2>
```

- 1990: sviluppato da Tim-Berners-Lee
- 1992: HTML 1.0
- 1993: HTML+ (primi meccanismi di layout fisico, form e tabelle)
- 1995-1996: estensioni non standard sul web browser Netscape
- 1996: HTML 3.2 (standard basato sulle pratiche correnti)
- 1997: HTML 4.0 (separazione di struttura e presentazione per mezzo dei fogli di stile)
- 1999: HTML 4.01 (piccole modifiche alla versione precedente)
- 2000 in poi…
  - XHTML 1.0, XHTML 1.1, XHTML 2.0
- 2009 HTML 5.0

Struttura globale di un documento HTML

- Un documento HTML si compone di tre parti:
  - un riferimento alla versione e alla DTD (non obbligatoria)
  - una sezione esplicativa di intestazione (non obbligatoria)
    - annotazione mediante l'elemento HEAD
  - un corpo, che contiene il contenuto effettivo del documento
    - annotazione per mezzo dell'elemento BODY o dell'elemento FRAMESET (non obbligatorio)
  - spazio bianco (spazi, a capo, tabulazioni e commenti) può comparire prima o dopo ciascuna sezione.
  - le sezioni 2. e 3. dovrebbero essere delimitate dall'elemento HTML

"HELLO WORLD"

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://
www.w3.org/TR/html4/strict.dtd">
    <HTML>
        <HEAD>
            <TITLE>Il mio primo documento HTML
            </TITLE>
        </HEAD>
        <BODY>
            <P>HELLO WORLD</P>
        </BODY>
    </HTML>
```

Riferimento a DTD HTML

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

in generale lo schema per il riferimento ad una DTD è il seguente : <!DOCTYPE root SYSTEM "URI">

dove root è l'elemento radice, URI, identificatore di sistema, è l'URI dello schema DTD

o in taluni casi è <!DOCTYPE root PUBLIC "…" "URI">

Definizioni DTD

- DTD rigorosa di HTML 4.01

  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

- DTD transitoria di HTML 4.01

  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

- DTD a frame di HTML 4.01

  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">

Elementi HTML

- Ogni elemento HTML prevede, di solito, un marcatore iniziale, il contenuto ed un marcatore finale
- alcuni elementi HTML consentono agli autori di omettere il marcatore finale (es., i tipi di elemento P e LI)
- alcuni elementi HTML non hanno contenuto., e.g. l'elemento interruzione di linea BR non ha contenuto; la sua unica funzione è di concludere una riga di testo
- le DTD HTML indicano per ciascun tipo di elemento se
  - il marcatore iniziale ed il marcatore finale sono richiesti
  - un elemento è vuoto (non ha contenuto)
  - un elemento può avere un contenuto e quale contenuto è considerato legale

Intestazione HTML

- identificata dall'elemento HEAD
- informazioni, non considerate come contenuto del documento, circa il documento corrente
- titolo, parole chiave, ….
- contiene l'elemento TITLE
- identifica i contenuti di un documento
- Obbligatorio per l'ACCESSIBILITA'

Intestazione HTML: i metadati

- HEAD permette di specificare dei metadati (informazioni su un documento) tramite l'elemento META
- In generale un elemento META specifica una proprietà e le assegna un valore
- <META name="Author" content="David"/>
  - Author è la proprietà, David il valore

Intestazioni HTML: attributi di META

- name =

  identifica un nome di proprietà. Queste specifiche non elencano valori legali per tale attributo.
- content =

  specifica un valore di proprietà. Le presenti specifiche non elencano valori legali per questo attributo
- scheme =

  designa uno schema da adoperarsi per interpretare il valore di una proprietà
- http-equiv =

  può essere usato al posto dell'attributo name. I server HTTP usano questo attributo per raccogliere informazioni per le intestazioni dei messaggi di risposta HTTP

Intestazione HTML: esempi di definizioni di metadati

- <META http-equiv="Expires" content="Tue, 20 Aug 1996 14:25:27 GMT">
- <META name="keywords" lang="fr" content="vacances, Grèce, soleil">
- <META name="keywords" lang="en-us" content="vacation, Greece, sunshine">
- <META http-equiv="Content-Type" content="text/html; charset=ISO-8859-5">

Il corpo del Documento

- Racchiude il contenuto del documento
  - identificato dall'elemento BODY
  - può essere presentato da un programma utente in una varietà di modi:
    - i browser visuali presentano il corpo come una "tela" in cui appare il contenuto: testo, immagini, colori, grafici, ...
    - i programmi utente di tipo acustico rendono il contenuto in parole

Esempio di BODY:

disapprovato (deprecated)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/
html4/loose.dtd">
 <HTML>
  <HEAD>
       <TITLE>Uno studio sulla dinamica della popolazione
       </TITLE>
  </HEAD>
  <BODY  bgcolor="white" text="black" link="red" alink="fuchsia" vlink="maroon">
        ... corpo del documento...
  </BODY>
 </HTML>
```

Logica di presentazione dei contenuti ed i contenuti stessi non sono separati

un attributo è disapprovato quando risulta superato da costrutti più recenti

Esempio di BODY: suggerito

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/
html4/strict.dtd">
<HTML>
 <HEAD>
  <TITLE>Uno studio sulla dinamica della popolazione</TITLE>
  <STYLE type="text/css">
   BODY { background: white; color: black}
  A:link { color: red }
  A:visited { color: maroon }
  A:active { color: fuchsia }
  </STYLE>
 </HEAD>
 <BODY>
      ... corpo del documento...
 </BODY>
</HTML>
```

XHTML

introduzione

Sommario

- Introduzione all'XHTML
- Cenni storici
- Perché usare XHTML?
- Schemi DTD di documenti XHTML
- Regole XML well-formed

Extensible HyperText Markup Language (XHTML)

- XHTML (26 gennaio 2000) famiglia di attuali e futuri tipi di documenti che riformula i tre tipi di HTML 4.01 come un'applicazione dell'XML 1.0
  - i documenti della famiglia XHTML sono progettati per poter lavorare insieme agli user agent basati su XML

XHTML vs HTML

- Separazione dei dati dalla logica di presentazione, e dalla struttura
  - formattazione del documento (versione strict, XHTML1.1) affidata ai CSS.
- Portabilità
  - l'evoluzione dei servizi mobili sarà fondata sull'integrazione tra XHTML e CSS (browser non in grado di supportare documenti non validi)
- Estensibilità
  - possibilità di inserire in un documento parti scritte in uno dei tanti linguaggi della famiglia XML o nuovi moduli XHTML
- Accessibilità
  - una pagina valida è di più facile gestione per browser alternativi quali quelli vocali o testuali

Versioni dell'XHTML 1.0

- XHTML Strict
  - non sono ammessi elementi ed attributi relativi all'aspetto grafico del documento (font, color, align..), controllato dai fogli di stile
    - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
- XHTML Transitional
  - ammette elementi di formattazione; è la versione più simile all'HTML 4.01
    - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
- XHTML Frameset
  - prevede l'utilizzo di frame
    - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

Elenco delle DTD definite dalla W3C: http://www.w3.org/QA/2002/04/valid-dtd-list.html

XHTML 1.1

- si basa sulla DTD Strict della versione 1.0
  - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1// EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
- rappresenta la prima formulazione pratica del concetto di modularità;
  - Modularità
- gli elementi fondamentali sono raggruppati in una serie di moduli indipendenti, che possono essere implementati o esclusi secondo le necessità

Modularità XHTML …

- Un documento di base XHTML è definito come un insieme di moduli XHTML i quali sono:
- Structure Module*
  - body, head, html, title
- Text Module*
  - abbr, acronym, address, blockquote, br, cite, code, dfn, div, em, h1, h2, h3, h4, h5, h6, kbd, p, pre, q, samp, span, strong, var
- Hypertext Module*
  - a
- List Module*
  - dl, dt, dd, ol, ul, li
- Object Module
  - object, param
- Metainformation module
  - META
- Presentation module
  - b, big, hr, i, small, sub, sup, tt

… Modularità XHTML

- Link Module
  - link
- Base Module
  - base
- Intrinsic Events module
  - Events attributes
- Scripting module
  - script and noscript elements
- Stylesheet module
  - style element
- Style Attribute Module Deprecated
  - style attribute
- Target Module
  - target attribute

Documento XHTML ben formato e valido

Un documento XHTML

è ben formato (o well—formed) quando la sua struttura rispetta le regole "XML well-formed"

è valido se è associato ad una delle DTD XHTML ed è conforme ai vincoli in essa espressi

La W3C offre un servizio WEB di validazione di documenti HTML, XHTML, MathML, …
http://validator.w3.org/

Regole XML well-formed …

- Gli elementi XHTML devono essere propriamente nidificati
- here is an emphasized paragraph
- here is an emphasizedparagraph

- I nomi degli elementi e nomi degli attributi devono essere sempre scritti con lettere minuscole
  - XML distingue le lettere maiuscole da quelle minuscole

… Regole XML well-formed …

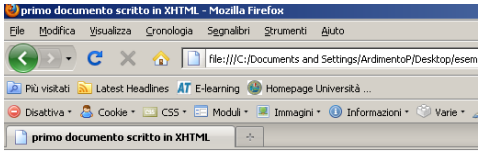
- elementi non vuoti richiedono sempre il tag di chiusura
- here is a paragraph. here is another paragraph.
- here is a paragraph. here is another paragraph.

- i valori degli attributi devono essere sempre compresi fra doppi apici
- <table rows="3">
- <table rows=3>

- minimizzazione degli attributi non supportata
- <textarea readonly="readonly" id="textinput1">Esempio</textarea>
- <textarea readonly id="textinput1">Esempio</textarea>

… Regole XML well-formed

- elementi vuoti devono sempre esplicitare la chiusura
-
<hr />
-
<hr>
- L'attributo id sostituisce l'attributo name
- HTML
- XHTML
- alcuni elementi XHTML sono obbligatori
  - gli elementi html, head e body devono essere presenti; l'elemento title deve essere presente all'interno dell'elemento head



Output su Mozilla Firefox

esempio di documento XHTML

Sorgente di: file:///C:/Documents%20and%20Settings/ArdimentoP/Desktop/esempio.htm - Mozilla Firefox

File Modifica Visualizza Aiuto HTML Validator

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" >
6    <head>
7      <title>primo documento scritto in XHTML</title>
8     </head>
9    <body>
10     <p>esempio di documento XHTML</p>
11   </body>
12  </html>
```

Codice sorgente

namespace …

- applicabile all'elemento html, prevede i seguenti attributi
  - dir: determina la direzione del testo
  - lang
    - specifica il codice identificativo del linguaggio di un elemento quando interpretato come HTML (non supportato in XHTML 1.1)
  - xml:lang
    - specifica il codice identificativo del linguaggio di un elemento quando interpretato come XML
  - xmlns (obbligatorio):
    - specifica il namespace predefinito per XHTML
    - il valore è "http://www.w3.org/1999/xhtml"

… namespace

- La presenza del namespace permette di estendere XHTML con elementi di altri linguaggi, anche creati personalmente

Singolo namespace

```
<?xml version="1.0"?>
<html:html xmlns:html='http://www.w3.org/1999/xhtml'>
  <html:head>
      <html:title>Frobnostication</html:title>
  </html:head>
  <html:body>
   <html:p>Moved to
     <html:a href='http://frob.example.com'>here.</html:a>
   </html:p>
  </html:body>
</html:html>
```

Molteplici namespace

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF PUBLIC "-//DUBLIN CORE//DCMES DTD 2002/07/31//EN"
"http://dublincore.org/documents/2002/07/31/dcmes-xml/dcmes-xml-dtd.dtd">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://www.ilrt.bristol.ac.uk/people/cmdjb/">
    <dc:title>Dave Beckett's Home Page</dc:title>
    <dc:creator>Dave Beckett</dc:creator>
    <dc:publisher>ILRT, University of Bristol</dc:publisher>
    <dc:date>2002-07-31</dc:date>
  </rdf:Description>
</rdf:RDF>
```

Riferimenti bibliografici e indirizzi Web …

- Specifica XHTML1.0 (italiano)
  - http://www.w3c.it/traduzioni/xhtml1-it.html
- Specifica XHTML 1.1
  - http://www.w3.org/TR/xhtml11/doctype.html#s_doctype
- Introduzione alla modularizzazione: dal sito del W3C
  - http://www.w3.org/MarkUp/modularization
- Il concetto di namespace in XML
  - http://www.w3.org/TR/REC-xml-names/

http://www.w3.org/TR/html4/intro/sgmltut.html#h-3.2

http://www.w3.org/MarkUp/SGML/

http://www.lnf.infn.it/~dmaselli/stage/html401/intro/sgmltut.html

http://www.isgmlug.org/sgmlhelp/g-index.htm

http://www.w3.org/TR/REC-html40/interact/scripts.html

DTD

Data Definition Document

Sommario

- Linguaggio DTD
  - tipo di documento
  - Dichiarazioni di elementi e content model
  - esempi di DTD
  - dichiarazioni di entità e di attributi
- OSSERVAZIONE
  - il W3C consiglia di utilizzare il linguaggio XML SCHEMA

Linguaggio DTD

- DTD (Document Type Definition) è un linguaggio schema ossia un linguaggio formale per esprimere schemi
- esistono diversi linguaggi schema (DTD, XSD, …) ognuno dei quali è implementato sotto forma di processi schema
- Schema: definizione formale della sintassi di un linguaggio XML
- Elementi e attributi ammissibili, struttura, vincoli, entità (esterne, interne), …

Processore di schema

- Strumento software che riceve in input un documento X (e.g. un documento XML) e uno Schema S e verifica se X è sintatticamente corretto rispetto a S
  - documento X è chiamato documento istanza

Schema DTD: dichiarazioni

- uno schema DTD permette di dichiarare diversi costrutti con la seguente forma <!OGGETTO-DICHIARATO ... >
- di seguito sono presentati alcuni di questi costrutti
- <!DOCTYPE ... >
- <!ELEMENT ... >
- <!ATTLIST ... >
- <!ENTITY ... >
- <!NOTATION ... >

Dichiarazioni di tipo di documento …

- Un documento XML può contenere una dichiarazione di DTD ossia un riferimento a uno schema DTD
  - i documenti descrivono se stessi
  - una dichiarazione di DTD associa uno schema DTD al documento istanza
  - Le dichiarazioni di DTD possono essere interne al documento istanza oppure esterne

… Dichiarazioni di tipo di documento

Dichiarazione interna

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

Dichiarazione esterna

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

SYSTEM si usa per DTD non pubblicate sul Web; la sintassi è:

```
<!DOCTYPE radice SYSTEM url>
```

per DTD di pubblico dominio si usa PUBLIC

```
<!DOCTYPE HTML PUBLIC
    "-//W3C//DTDHTML4.0l//EN"    "http://www.w3.org/TR/html4/strict.dtd">
```

Documento XML valido

- Un documento XML è valido se è conforme (sintatticamente corretto) rispetto a uno schema dato

Nel prologo XML si potrebbe utilizzare l'attributo standalone
<? xml version="1.0" encoding="UTF-8" standalone="no" ?>

standalone "no": il processore costruisce l'albero sintattico e verifica la validità del documento istanza
standalone "yes": il processore costruisce l'albero sintattico ma non verifica la validità del documento istanza

Dichiarazioni di elemento

- Sintassi di una dichiarazione di tipo di elemento:
- <!ELEMENT nome modello>
  - nome è un nome di elemento
  - modello è la descrizione di un modello di contenuto
- definisce i requisiti di validità di tutti gli elementi con quel nome

Dichiarazioni di elementi: tipi di modelli

- Vuoto (EMPTY)

  L'elemento non ha alcun nodo figlio
  Qualsiasi (ANY)
  Qualsiasi sequenza di caratteri e elementi
  Contenuto misto
  (#PCDATA | elem1 | ... | elemk)*
  Contenuto elemento
  Definisce vincoli e ordine sul numero di occorrenze degli elementi mediante l'uso di espressioni regolari

Contenuto elemento: espressioni regolari

Il linguaggio DTD si serve del formalismo delle espressioni regolari per descrivere sequenze di caratteri o elementi ammissibili

Sequenza: (a,b,...,h)
Alternativa: (a | b | ... | k)
Opzionalità: con a?
ripetizione zero o più volte: a*
ripetizione una o più volte: a+
(dove a,b,... indicano un generico modello)

DTD: Radice di un documento

<!DOCTYPE ESE [<!ELEMENT ESE (#PCDATA)>]

il tipo dell'elemento radice deve essere il tipo di documento
Istanza di documento valido
<ESE>questo documento e`valido</ESE>

Istanze di documenti non valido
<boh>questo documento NON e`valido</boh>
<ese>XML e` case sensitive</ese>

Espressione regolare: sequenza caratteri

<!ELEMENT ESE (#PCDATA)>
a #PCDATA deve corrispondere solo testo senza elementi annidati. Il testo può anche essere vuoto
Istanza di documento valido
<ESE>questi documenti sono validi</ESE>
<ESE></ESE>
<ESE/>
Istanza di documento non valido
<ESE>questo documento NON è valido</ESE>

Espressione regolare: Sequenza fissata di elementi

<!ELEMENT ESE (P1, P2)>

<!ELEMENT P1 (#PCDATA)>
<!ELEMENT P2 (#PCDATA)>
Istanza di documento valido
<ESE> <P1>questa e` la prima parte</P1>
<P2>questa e` la seconda parte</P2>
</ESE>
<ESE>
<P1/> <P2>la prima parte e' vuota</P2>
</ESE>
Istanze di documenti non validi
<ESE> <P1>manca P2</P1> ____ </ESE>
<ESE> <P2/> <P1>ordine sbagliato</P1> </ESE>
<ESE> <P1/> <P2/> <P2>un solo P2</P2> </ESE>
<ESE> NO testo libero <P1/> <P2/> </ESE>

Espressione regolare: occorrenza zero..n volte

<!ELEMENT ESE (IN, S*)>

<!ELEMENT IN (#PCDATA)>
<!ELEMENT S (#PCDATA)>
Istanza di documento valido
<ESE>
<IN>intro</IN>
<S>sez1</S>
<S>sez2</S>
</ESE>
<ESE>
<IN>senza sezioni</IN>
</ESE>
Istanze di documenti non validi
<ESE><S>intro ci deve essere</S></ESE>
<ESE><S>solo dopo intro</S><IN/><S>sez1</S></ESE>

Espressione regolare: occorrenza una..n volte

<!ELEMENT ESE (IN, S+)>

<!ELEMENT IN (#PCDATA)>
<!ELEMENT S (#PCDATA)>
Istanza di documento valido
<ESE>
<IN>intro</IN> <S>sezione unica</S>
</ESE>
<ESE>
<IN>intro</IN>
<S>sez1</S> <S>sez2</S>
</ESE>
Istanze di documenti non validi
<ESE> <IN>almeno una sezione</IN> _____ </ESE>
<ESE> _____ <S>intro ci deve essere</S> </ESE>
<ESE> <S>solo dopo intro</S> </IN> <S>sez1</S> </ESE>

Espressione regolare: opzionalità

<!ELEMENT ESE (IN?, S+)>

<!ELEMENT IN (#PCDATA)>
<!ELEMENT S (#PCDATA)>
Istanza di documento valido
<ESE>
<IN>intro</IN> <S>sez1</S> <S>sez2</S>
</ESE>
<ESE> <S>senza introduzione</S>
</ESE>
Istanze di documenti non validi
<ESE> <IN>intro</IN> <IN>intro unica</IN> <S/> </ESE>
<ESE> <IN>almeno una sezione</IN> _____ </ESE>
<ESE> <S>solo dopo intro</S> </IN> <S>sez1</S> </ESE>

Espressione regolare: alternativa

```
<!ELEMENT ESE (IN?, (A1 | A2))>

    <!ELEMENT IN  (#PCDATA)>
    <!ELEMENT A1  (#PCDATA)>
    <!ELEMENT A2  (#PCDATA)>
Istanza di documento valido
    <ESE> <IN>intro</IN> <A1>alternativa 1</A1> </ESE>
    <ESE> <IN>intro</IN> <A2>alternativa 2</A2> </ESE>
    <ESE> <A1>senza introduzione</A1> </ESE>

Istanze di documenti non validi
  <ESE> <IN>serve una delle 2</IN> _____ </ESE>
  <ESE> <IN/> <A1>al piu` una delle 2</A1> <A2/> </ESE>
  <ESE> <IN/> <A1>qui solo testo <A2/> </A1> </ESE>
```

Espressione regolare: annidamento a un numero fisso di livelli

```
<!ELEMENT ESE (TI, S1+)>

    <!ELEMENT TI  (#PCDATA)>
    <!ELEMENT S1  (TI?, S2*)>
    <!ELEMENT S2  (#PCDATA)>
Istanza di documento valido
     <ESE>
            <TI>esempi xml</TI>
            <S1>
            <TI>sez 1</TI>
            <S2>sez 1.1</S2>
            </S1>
            <S1><TI>sez 2 ha solo titolo<TI></S1>
        <S1> <S2/> </S1>
        <S1 />
    </ESE>
```

Modelli a contenuto misto

#PCDATA ed altri elementi non possono essere combinati in modo arbitrario nel content model. L'unica possibilità di combinarli è:

<!ELEMENT elem (#PCDATA | elem1 | ... | elemk)*>
Un elemento di questo tipo è detto a contenuto misto il suo contenuto è una sequenza arbitraria di testo libero ed elementi di tipo elem1,...,elemk
non è possibile imporre una struttura più precisa alla combinazione di questi elementi

Combinazioni errate di #PCDATA ed elementi

Un modello contenente #PCDATA ed altri elementi che non ha la forma vista prima è scorretto

```
<!ELEMENT ESE (TI, #PCDATA)>
<!ELEMENT ESE (TI?, #PCDATA)>
<!ELEMENT ESE (TI, (#PCDATA | S*))>
<!ELEMENT ESE (TI, (#PCDATA | S)*)>
<!ELEMENT S   (#PCDATA | COM | S)+>
<!ELEMENT COM (#PCDATA)+>
```

Modelli di contenuto EMPTY

Un elemento può essere forzato ad avere contenuto vuoto

```
<!ELEMENT VUOTO EMPTY>
    <!ELEMENT E1  (#PCDATA)>
```

Istanza di documento valido
<E1></E1> <E1/> <VUOTO/>
Istanze di documenti non validi
<VUOTO> deve essere vuoto </VUOTO>
<VUOTO></VUOTO>

Elementi con contenuto EMPTY

aggiungono informazione al documento attraverso attributi associati all'elemento

Esempi in XHTML
<!ELEMENT base EMPTY>
<!ELEMENT meta EMPTY>
<!ELEMENT link EMPTY>
<!ELEMENT hr EMPTY>
<!ELEMENT br EMPTY>
<!ELEMENT img EMPTY>
<!ELEMENT area EMPTY>
…

Uso di Elementi vuoti

aggiungono informazione al documento attraverso attributi associati all'elemento

```
<img src="colosseo.gif“
        align="left"
      height="50"
      width="30"/>
```

attraverso la presenza stessa dell'elemento

Modelli di contenuto ANY

<!ELEMENT elem ANY>

specifica che il contenuto di un elemento è una sequenza di testo ed elementi qualsiasi
Tutti gli elementi che compaiono in un elemento con contenuto ANY devono essere stati dichiarati
<!ELEMENT TUTTO ANY>
<!ELEMENT E1 (#PCDATA)>
<!ELEMENT E2 (#PCDATA)>
Istanza di documento valido
<TUTTO> <E1>xxx</E1> yyy <E2/> <E1/> </TUTTO>
Istanza di documento non valido
<TUTTO> <E3>non dichiarato</E3> </TUTTO>

Entità

unità di memorizzazione contenente una parte di un documento XML

l'unità che contiene la dichiarazione XML, la DTD e l'elemento radice è l'entità documento
l'elemento radice e i sottoelementi possono contenere riferimenti ad altre entità

un documento XML da un punto di vista logico è suddiviso in elementi; da un punto di vista fisico può essere memorizzato su più file

Uso delle entità

definire abbreviazioni usate comunemente nel documento

suddividere il documento in più parti, memorizzate separatamente
suddividere la DTD in più parti e renderla modulare

Dichiarazioni e riferimenti ad entità

Dichiarazioni di entità

interne: valore definito nel documento stesso
<!ENTITY nome-entita "testo">

esterne: valore accessibile tramite un'URI

<!ENTITY nome-entita SYSTEM "URI">

Riferimento ad un'entità nel documento

&nome-entita;

Entità - Esempi

<!ENTITY DIS "Dip. di Informatica e Sist.">

<!ENTITY SEZIONE1 SYSTEM "sezione1.xml">
<!ENTITY SEZIONE2 "<Sezione>Da scrivere.</Sezione>">

Riferimenti nell'istanza di documento

Presso il &DIS; si trovano laboratori di didattica e laboratori di ricerca come illustrato nelle
due sezioni seguenti.
&SEZIONE1;
&SEZIONE2;

Dichiarazione di liste di attributi

- Sintassi di una dichiarazione di lista di attributi per un elemento

  <!ATTLIST nome-elem definizioni>

  dove
- nome-elem indica il nome dell'elemento per cui si dichiarano gli attributi
- definizioni è una lista di definizioni di specifiche in cui ogni specifica è una terna:

  nome-att tipo default

Specifica di un attributo

- nome-att
  - Identifica l'attributo ed è usato nel marcatore iniziale dell'elemento per il quale è definito
- tipo
  - specifica quali sono i valori che l'attributo può assumere
- valore
  - specifica una dichiarazione di default

Tipi di attributi

enumerazione: il tipo ENUMERATED definisce una lista di lista di possibili valori di un attributo

stringa: il tipo CDATA (Character DATA) indica che l'attributo può assumere qualunque valore privo di marcatura

Token: definito mediante i seguenti sottotipi
ID, IDREF, IDREFS, ENTITY, ENTITIES, NMTOKEN, NMTOKENS

Esempio di attributo di tipo enumerativo

<!ATTLIST p align (left|center|right|justify) #IMPLIED>

i possibili valori di p sono left, center, right, justify
#IMPLIED indica che l'attributo è opzionale
non ha alcun valore di default

Esempio di attributo di tipo stringa

<!ATTLIST capitolo titolo CDATA #REQUIRED>

Istanze di documenti validi
<capitolo titolo="introduzione"></capitolo>
<capitolo titolo=""></capitolo>
<capitolo titolo=">"></capitolo>

Istanze di documenti non validi
<capitolo>< /capitolo>
<capitolo titolo="<"></capitolo>

Istanza di documento non well-formed
<capitolo titolo="""></capitolo>

Tipo di attributo Token

ID: indica che il valore dell'attributo identifica l'elemento; può esserci un solo attributo di tipo ID per elemento

NMTOKEN: indica che il valore di un attributo deve essere un token di nome
IDREF: meccanismo (primitivo) usato per denotare le chiavi e definire ancore a cui collegare i link; il valore dell'attributo è l'ID di un altro elemento
ENTITY: il nome di un'entità dichiarata nello schema DTD permette di collegarsi a dati esterni in forma binaria
IDREFS, ENTITIES, NMTOKENS: versioni al plurale dei precedenti (valori separati da spazio)
NOTATION: permette di specificare sia il formato di dati non-XML sia un'applicazione che li processi

Esempio di attributo token di tipo ID

<!ATTLIST capitolo numero ID #REQUIRED>

Istanze di documenti validi
<capitolo numero="introduzione"></capitolo>
<capitolo numero="_"></ capitolo>
Istanze di documenti non validi
<capitolo numero=">"></ capitolo>
<capitolo numero=""></capitolo>
<capitolo numero=" "></capitolo>
< capitolo numero="1"></capitolo>
< capitolo numero="a"></capitolo><capitolo numero="a">< /capitolo>

Il valore di un attributo ID deve essere un NCName
NCName ::= (Letter | '_') (NCNameChar)*
(http://www.w3.org/TR/1999/REC-xml-names-19990114/#NT-NCName)

Esempio di attributo token di tipo NMTOKEN

<!ATTLIST form name NMTOKEN #IMPLIED>

Istanze di documenti validi
<form name="form.mio"></form>
<form name="87"></form>
<form name=" 87"></form>
Istanze di documenti non validi
<form name="form mio"></form>
<form name=""></form>

Il valore di un attributo NMTOKEN deve essere un Nmtoken
Nmtoken::=(NameChar)+
NameChar ::= Letter | Digit | '.' | '-' | '_' | ':' | CombiningChar | Extender
(http://www.w3.org/TR/2000/WD-xml-2e-20000814#NT-Nmtoken)

Esempio di attributo token di tipo IDREF

```
<?xml version="1.0">
<!DOCTYPE FAMIGLIA [
   <!ELEMENT FAMIGLIA (PERS*)>
   <!ELEMENT PERS  (#PCDATA)>
   <!ATTLIST PERS  NP ID #REQUIRED>
   <!ATTLIST PERS  PADRE IDREF #IMPLIED>
   <!ATTLIST PERS  MADRE IDREF #IMPLIED>
]>

<FAMIGLIA>
  <PERS NP="a1">Susanna</PERS>
  <PERS NP="a2">Carlo</PERS>
  <PERS NP="a3" MADRE="a1">Maria</PERS>
  <PERS NP="a4" PADRE="a2">Luca</PERS>
 </FAMIGLIA>
```

Attributi token di tipo ENTITY e di tipo NOTATION

- Il valore di un attributo di tipo ENTITY è il nome di un'entità esterna dichiarata nello schema DTD
  - un'entità esterna può essere una risorsa XML o non XML
  - per una risorsa non XML, chiamata entità unparsed, è necessario descrivere il formato dell'entità unparsed mediante una dichiarazione di notazione

Esempi di attributo token di tipo ENTITY e NOTATION

```
<?xml version="1.0"?>

<!DOCTYPE catalogo[
<!ELEMENT catalogo (#PCDATA|image)*>
<!ELEMENT image EMPTY>
<!ATTLIST image source ENTITY #REQUIRED>
<!ENTITY logo SYSTEM "logo.gif" NDATA gif>
<!NOTATION gif SYSTEM "http://www.rfc-editor.org/rfc/
rfc2046.txt">
]>
<catalogo>
      <image source="logo" />
</catalogo>
```

Specifica del valore di un attributo

Richiesto: #REQUIRED l'attributo deve essere obbligatoriamente presente

Opzionale: #IMPLIED l'attributo è opzionale. Se assente non vi è alcun valore di default
Opzionale con default val: l'attributo è opzionale; se è assente, il valore specificato è usato come default
Prefissato: #FIXED "val": l'attributo è opzionale: se è assente il valore è usato come default, se presente deve avere obbligatoriamente il valore specificato

Esempio di specifica del valore di attributi

```
<!--Elemento form di XHTML 1.0: -->

<!ATTLIST form
    action        CDATA        #REQUIRED
      onsubmit    CDATA        #IMPLIED
      method      (get|post)   "get"
      enctype     CDATA        "application/x-www-form-urlencoded"
>

<!ATTLIST html xmlns CDATA #FIXED "http://www.w3.org/1999/
xhtml"
>
```

"Opzionale con default" e "prefissato" non richiedono la definizione del tipo di attributo

Esempio di Schema DTD

```
<!DOCTYPE TechRepDip [

  <!ELEMENT TechRepDip    (Intestazione, Sezione+, Bibliografia?)>
  <!ELEMENT Intestazione  (Numero, Data, Titolo, Autore+, Sommario?)>
  <!ELEMENT Data          (Giorno?, Mese, Anno)>
  <!ELEMENT Autore        (Cognome, Nome+)>
  <!ELEMENT Sezione       (TitoloSezione, Testo?, Sezione*)>
  <!ELEMENT Bibliografia  (VoceBiblio)+>
  <!ELEMENT Numero        (#PCDATA)>
          ...
    <!ELEMENT VoceBiblio    (#PCDATA)>

  <!ATTLIST Sezione   id  ID  #REQUIRED
          num    NMTOKEN    #IMPLIED
          stato  (finale | provvisorio) "finale" >

  <!ENTITY DIS      "Dipartimento di Informatica e Sistemistica">
  <!ENTITY SEZIONE1 SYSTEM "sezione1.xml">

  <!ENTITY SEZIONE2 "<Sezione>Ancora da scrivere.</Sezione>">
]>
```

Limitazioni del linguaggio DTD …

- non può imporre vincoli ai dati carattere
- tipi di attributi limitati
  - interi, URI, booleani, …
- Dichiarazioni di elementi e attributi avulse dal contesto
  - uno stesso elemento dovrebbe poter assumere una struttura diversa in dipendenza del proprio significato

… Limitazioni del linguaggio DTD

- Dati carattere non combinabili in alcun modo con le espressioni regolari
  - Impossibilità di ordinare gli elementi
  - Impossibilità di contare il numero di occorrenze degli elementi
- Impossibilità di specificare la presenza di un elemento in più posizioni
- Impossibilità di specificare un valore di default per gli elementi
- Non supporta i namespace

Esempi

- Author.dtd e author.xml

Evidenziare i limiti di uno schema quale quello definito in author.dtd

- XSLT
- CSS
- XPATH
- XSD
- DOM
- JAVASCRIPT
- AJAX

XML

Sommario

- Definizione e origini
- Modello concettuale
- Regole generali dell'XML

eXtensible Markup Language (XML)

- L'Extensible Markup Language (XML) è un formato, basato su testo, per rappresentare, memorizzare e veicolare informazioni strutturate
  - documenti, data, configurazioni, libri, transazioni, …
- deriva da un formato più vecchio, l'SGML (ISO 8879), e mira, rispetto a questo, ad essere più adatto ad essere usato nel Web
  - nato nel 1996 nell'ambito della SGML Activity
  - nel 1998 le raccomandazioni sono diventate ufficiali: XML 1.0
  - nel febbraio 2004 XML 1.1 diventa uno standard W3C

XML …

- Insieme di linguaggi, combinabili tra loro, ognuno dei quali persegue un obiettivo
  - XSLT: trasformazione tra linguaggi XML
  - XSL-FO: specifica del layout fisico del documento XML
  - XPath: accesso e navigazione di un albero XML
  - XQuery, XML-QL: interrogazione di documenti XML
  - XLink e XPointer: navigazione interna e collegamenti ipertestuali

… XML

- XForms: definizione di form
- RDF: framework XML per descrivere i contenuti (semantica)
- XML Schema, XML-Data, DDML: definizione formale della sintassi di un linguaggio basato su XML
- XHTML, WML, VoiceML: supporto diverse tipologie di dispositivo:, Xforms
- WDSL, UDDIS: sistemi di cooperazione applicativa

Linguaggi di markup a confronto

- SGML
  - Metalinguaggio per la definizione di un linguaggio di markup
- HTML
  - linguaggio di marcatura specificato da una DTD SGML per veicolare ipertesti ed ipermedia
- XML
  - versione semplificata del linguaggio SGML che costituisce un metalinguaggio per la definizione di linguaggio di markup
  - con XML si considera tutto l'insieme di standard ad esso associati

Peculiarità dell'XML …

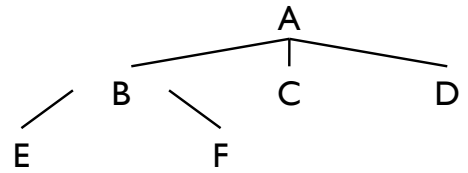
- Separazione di dati, logica di presentazione di dati e struttura dei dati
- Portabilità
  - interpretabile indipendentemente dall'ambiente hardware e software
    - l'evoluzione dei servizi mobili sarà fondata sull'integrazione tra XHTML e CSS (browser non in grado di supportare documenti non validi)
- Interoperabilità a livello sintattico
  - permette di condividere e veicolare strutture di dati tra applicazioni differenti
- Estensibilità
  - possibilità di inserire in un documento parti scritte in uno dei tanti linguaggi della famiglia XML o nuovi moduli XHTML
- Verificabilità
  - verifica della correttezza sintattica (regole di base + regole well-formed) e strutturale (validità)
- Accessibilità

… Peculiarità dell'XML

- Flessibilità
  - Possibilità di definire linguaggi ad-hoc per la rappresentazione di dati [semi]-strutturati
- basi di dati XML
  - disponibilità di librerie di manipolazione in tutti gli ambienti di programmazione (jdom in JAVA)
- riduce la necessità di sviluppare parser ad-hoc per le applicazioni
  - attività costosa e sovente causa di difetti
- strumento di base per la rappresentazione delle informazioni annotate semanticamente (semantic web)

Struttura concettuale dei documenti XML

- albero* ordinato, particolare struttura gerarchica, chiamata albero XML
  - ogni albero* XML è ordinato



- albero*: struttura di dati corrispondente a un grafo non orientato nel quale due vertici qualsiasi sono connessi da uno e un solo cammino

Struttura logica e fisica di un documento XML

- Struttura logica
  - costituito da uno o più elementi
    - ogni elemento ha un tipo e potrebbe avere degli attributi
- Struttura fisica
  - costituito da unità di memorizzazione chiamate entità che contengono dati parserizzati o non parserizzati (e.g. document, entity, entity reference, ..)

Tipi di nodi di un albero XML …

- documento: entità particolare, chiamata nodo radice, in quanto
  - contiene l'intero documento istanza
  - costituisce l'entry point per il parser XML
  - contiene uno ed un solo elemento figlio chiamato elemento radice
- elemento: definisce il raggruppamento logico dell'informazione rappresentata dai suoi discendenti
  - possiede sempre un nome (parola che descrive il raggruppamento)
- attributo: associato sempre a un nodo elemento
  - Costituito da una coppia nome/valore
  - Ogni elemento può avere al più un attributo

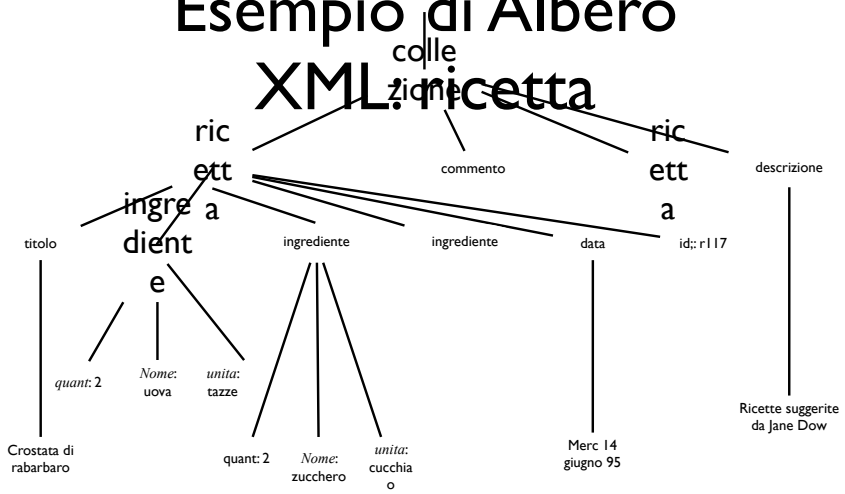
… Tipi di nodi di un albero XML …

- testo: frammento dell'informazione rappresentata da un documento XML
  - non hanno figli
  - due nodi di testo non possono essere fratelli se non quando sono separati da un altro tipo di nodo
- istruzione per l'elaborazione
  - possiede un obiettivo (target) e un valore dove target è una parola indicante lo strumento a cui sono dirette le istruzioni e valore è una stringa di testo contenente l'informazione rilevante
- commento: esprime meta-informazioni in modo informale
  - nodo foglia speciale

… Tipi di nodi di un albero XML

- CDATA
  - Testo non oggetto di parserizzazione
  - Presente nel modello dei dati DOM e JDOM
- riferimento a entità
  - frammenti XML dotati di nome
    - presente nell'API DOM ed in alcune sue implementazioni, e.g. JDOM

Esempio di Albero XML: ricetta



Rappresentazione testuale dei documenti XML

- un prologo (opzionale), costituito da
  - una dichiarazione XML (nodo istruzione di elaborazione)
  - un riferimento a una dichiarazione di schema (DTD, XSD, …)
- un'istanza di documento, contenente
  - testo libero (nodo testo)
  - elementi delimitati da marcatori (nodi elemento)
  - attributi associati ad elementi (nodi attributo)
  - entità (nodo entità)
  - sezioni CDATA (nodo CDATA)
  - istruzioni di elaborazione (nodo istruzione di elaborazione)
  - commenti (nodo commento)

Codifica della rappresentazione testuale

- XML usa la codifica dei caratteri Unicode (UTF-16)
  - codice a 16 bit (65536 simboli)
  - sufficiente per rappresentare i caratteri di tutte le lingue
  - ogni carattere è rappresentato da al più quattro cifre esadecimali
- Es. ꀟ
  - i primi 256 caratteri sono il codice ASCII (UTF-8)
- Es.
 (linefeed),   (blank), N (N), ...
- ogni parser XML deve riconoscere almeno i codici Unicode UTF-8 e UTF-16

Regole generali XML

Nomi di elemento: regole…

- XML è case-sensitive - esempio non ben formato:

<elenco-clienti>
<cliente><codice>CC128</codice> ...
</CLIENTE>
</Elenco-Clienti>

Errore comune: dimenticare "/" nell'etichetta di chiusura - es:

<elenco-clienti>
<cliente><codice>CC128</codice> ... <cliente>
</elenco-clienti>

… Nomi di elemento: regole…

- possono contenere solo: lettere, cifre, -, _, ., :

carattere ":" usato solo per separare namespace.
nomi che iniziano con XML,xml,xML,... sono riservati

<nomiPermessi>
<xsl:template/>
<Nome_elemento_lungo/>
<Altro-nome-lungo/>
<nome.con.punti/>
<a1233-231-231/>
<_12/>
</nomiPermessi>

… Nomi di elemento: regole

<nomi+Vietati>
<questiNO@#$%^()+?*;$=/>
<Un;nome*2/>
<XmL-riservato/>
<8-inizia-con-cifra/>
<niente spazi/>
<riservati<&>>
</nomi+Vietati>

Entità predefinite in XML

Name	Character	Unicode code point (decimal)	Standard	Description
quot	"	U+0022 (34)	XML 1.0	*(double)* quotation mark
amp	&	U+0026 (38)	XML 1.0	ampersand
apos	'	U+0027 (39)	XML 1.0	apostrophe *(= apostrophe-quote)*
lt	<	U+003C (60)	XML 1.0	less-than sign
gt	>	U+003E (62)	XML 1.0	greater-than sign

sono le uniche entità ammesse in qualunque linguaggio XML; le entità definite nelle DTD dell'HTML 4 sono riconosciute solo nei linguaggi XHTML

Attributi …

- Un elemento può avere nessuno, uno o più attributi
  - un attributo ha un nome ed associa una proprietà ad un elemento
  - i caratteri ammessi nei nomi sono gli stessi che per gli elementi
  - gli attributi vengono specificati nel marcatore iniziale attraverso coppie (attributo, valore)
  - il valore deve essere delimitato da una coppia di apici singoli o doppi
  - un elemento non può avere due attributi con lo stesso nome

… Attributi

- <attributi-ben-formati>
  <elemento _ok="yes"/>
  <un attr="il suo valore"/>
  <molti primo="1" secondo='2' terzo="333"/>
  <apici-doppi-o-singoli
  doppi="John's"
  singoli='Stampa: "Ciao Mondo!" '/>

</attributi-ben-formati>

Nota Bene:
In HTML gli apici intorno al valore possono mancare
mentre in XML gli apici sono obbligatori

Attributi - Esempi mal formati

<attributi-mal-formati>
<carattere-non-ammesso a*b="23432"/>
<separatori-diversi valore="12'/>
<cambia-separatore valore="aa"aa"/>
<cambia-separatore valore='bb'bb'/>
<parola-riservata XML-ID="xml234"/>
</attributi-mal-formati>

Uso di attributi ed elementi

Per memorizzare i dati si usano sia gli elementi sia gli attributi

<impiegato>
<nome>A. Rossi</nome>
<ddn>1/1/1950</ddn>
</impiegato>

<impiegato ddn="1/1/1950">
<nome>A. Rossi</nome>
</impiegato>

Sezioni CDATA

servono ad includere testo contenente caratteri riservati
possono comparire ovunque può comparire testo libero
tutti i caratteri riservati perdono il loro significato speciale
"<" e "&" non vanno sostituiti con "<" e "&"
le sequenze <elem>, </elem> e <elem/> non vengono interpretate come marche
non possono essere annidate
racchiuse tra "<![CDATA[" e "]]>"
non possono contenere la stringa "]]>"

Sezioni CDATA: esempio 1

```
<lucido>
 <titolo>Sezioni CDATA - Esempi</titolo>
 Questo lucido in formato XML
 <esempio>
   <![CDATA[
      <lucido>
        <titolo>Sezioni CDATA - Esempi</titolo>
        Questo lucido in formato XML
        ... caratteri riservati OK: &, <
      </lucido>
   ]]>
 </esempio>
                                  </lucido>
```

Sezioni CDATA: esempio 2

- La stringa "]]>" non è ammessa

```
<esempio>
  <![CDATA[Un programma C
   if (x[2] + y[i] < 3*z) {
     if (vet[x[i]]> 18) {              /* 1 */
       if (a < b) printf("Ciao");     /* 2 */
     }
   }
  ]]>
  sezione CDATA termina su riga 1 e non qui
  "a &lt; b" da` errore perche` sono fuori CDATA
</esempio>
```

Al posto di]]> bisogna includere]]>

Commenti

- parte di documento non oggetto di parserizzazione

possono comparire ovunque all'esterno della marcatura
un processore XML può o meno rendere disponibili le parti di documento racchiuse tra commenti
delimitati da <!-- e -->
possono contenere qualsiasi carattere (inclusi "<" e "&"), tranne "-"
non possono essere annidati

Commenti: esempi

- <bibliografia>

```
<!-- voce commentata <pub> ... </pub>  -->
<pub> ... </pub>
<!-- commento su piu` righe
<pub> ... </pub>   <- usa "<"
<pub> ... </pub>   anche "&" e` ammesso
-->

<!-- commento non -- valido -->
<pub <!-- qui no commento --> id="Ullm82"> ... </pub>
<!-- un commento
<!-- non puo` contenere commenti annidati -->
-->
</bibliografia>
```

Commenti in sezioni CDATA

in una sezione CDATA i commenti non sono riconosciuti come tali

<esempio>
Questo e` un <!-- commento -->
<![CDATA[Sezione CDATA con "commento"
<!-- fa parte del testo --> altro testo
]]>
Questo e` di nuovo un <!-- commento -->
</esempio>

Istruzioni di elaborazione

permettono di includere nel documento istruzioni da passare alle applicazioni
hanno la forma <?PITarget ... ?>, dove
PITarget denota l'applicazione a cui è diretta la PI
... denota ulteriori dati da passare all'applicazione
Esempio: elaborazione di uno stylesheet da parte di XSLT
<?xml-stylesheet href="mystyle.xsl" type="text/xsl" ?>

Dichiarazione XML

- Documenti XML possono (e in realtà dovrebbero) iniziare con una dichiarazione XML che specifica la versione di XML utilizzata
- <?xml version="1.0"?>
- <testo>Questo documento e` conforme alla specifica di XML 1.0.
- </testo>
- La dichiarazione XML può anche specificare la codifica dei caratteri usati:
- <?xml version="1.0" encoding="ISO-8859-2"?>
- La codifica dei caratteri specifica il mapping tra i byte che costituiscono la rappresentazione fisica del documento (ad es., file, socket, ...) ed i caratteri

XPATH

XPath

- XPath è una parte fondamentale di XSLT e di altre specifiche come XLink
- XPath è un linguaggio che permette di indirizzare parti di un documento XML
- XPath dispone anche di primitive per eseguire semplici operazioni su stringhe, numeri e valori booleani
- Le espressioni XPath possono avere come valore numeri, stringhe, booleani e insiemi di nodi

XSTL

CSS e XSL

- Attualmente, molti creatori di parser XML supportano l'applicazione diretta di fogli di stile CSS (Cascading Style Sheets) a documenti XML, per definirne un formato di visualizzazione.
- XSL o eXtensible Stylesheet Language, è un linguaggio XML creato inizialmente per fornire a XML un supporto per la formattazione simile a CSS, ma potenziato per riflettere il linguaggio su cui è applicato, i.e. XML

XSL e XSLT

- XSLT (XSL Transformations) è il successore di XSL, ed estende il concetto di foglio di stile fino a permettere la manipolazione della struttura stessa del documento.
- XSLT permette di trasformare un documento XML filtrandone i dati e riorganizzandoli in un'altra struttura XML, o persino in un semplice testo.
- XSLT possiede molte delle caratteristiche di un linguaggio di programmazione imperativo!

Riferimenti

- Specifica XSLT
  - http://www.w3c.org/TR/xslt
- specifica XPath
  - http://www.w3c.org/TR/xpath