

Linguaggi di Markup

Violetta Lonati ^a

Dipartimento Scienze dell'Informazione
Università degli Studi di Milano

^a *E' garantito il permesso di copiare, distribuire e/o modificare i materiali contenuti in questa pagina seguendo i termini della Licenza per Documentazione Libera GNU, Versione 1.1 o ogni versione successiva pubblicata dalla Free Software Foundation, con il riferimento all'Autore e alla presente nota. Una copia della licenza è reperibile sul sito <http://www.softwarelibero.it/gnudoc/fdl.it.html>*

Linguaggi di Markup

- Sono usati per la descrizione di linguaggi.

Linguaggi di Markup

- Sono usati per la descrizione di linguaggi.
- Le parole sono costituite da interi documenti corredati da marcatori, detti tag, che forniscono informazioni su porzioni di testo.

Linguaggi di Markup

- Sono usati per la descrizione di linguaggi.
- Le parole sono costituite da interi documenti corredati da marcatori, detti tag, che forniscono informazioni su porzioni di testo.
- Esempi: HTML, XML

HyperText Markup Language

L'HTML serve per presentare documenti sul World Wide Web.

- formattazione del testo (veste grafica)

HyperText Markup Language

L'HTML serve per presentare documenti sul World Wide Web.

- formattazione del testo (veste grafica)
- creazione di collegamenti tra diversi documenti

HyperText Markup Language

L'HTML serve per presentare documenti sul World Wide Web.

- formattazione del testo (veste grafica)
- creazione di collegamenti tra diversi documenti
- Il browser interpreta i tag dell'HTML e visualizza il testo di conseguenza.

HTML - esempi di tag

tag html

cosa caratterizza

HTML - esempi di tag

tag html

cosa caratterizza

```
<html> ... </html>
```

HTML - esempi di tag

tag html

cosa caratterizza

`<html> ... </html>`

documento HTML

HTML - esempi di tag

tag html

cosa caratterizza

`<html> ... </html>`

documento HTML

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

HTML - esempi di tag

tag html

cosa caratterizza

`<html> ... </html>`

documento HTML

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

parte visibile

HTML - esempi di tag

tag html

cosa caratterizza

`<html> ... </html>`

documento HTML

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

parte visibile

` ... `

HTML - esempi di tag

tag html

cosa caratterizza

`<html> ... </html>`

documento HTML

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

parte visibile

` ... `

lista ordinata

HTML - esempi di tag

tag html

cosa caratterizza

`<html> ... </html>`

documento HTML

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

parte visibile

` ... `

lista ordinata

` ... `

HTML - esempi di tag

tag html

cosa caratterizza

`<html> ... </html>`

documento HTML

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

parte visibile

` ... `

lista ordinata

` ... `

elemento della lista

HTML - esempi di tag

tag html

cosa caratterizza

`<html> ... </html>`

documento HTML

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

parte visibile

` ... `

lista ordinata

` ... `

elemento della lista

` ... `

HTML - esempi di tag

tag html

cosa caratterizza

`<html> ... </html>`

documento HTML

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

parte visibile

` ... `

lista ordinata

` ... `

elemento della lista

` ... `

testo in grassetto

HTML - esempi di tag

tag html

cosa caratterizza

`<html> ... </html>`

documento HTML

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

parte visibile

` ... `

lista ordinata

` ... `

elemento della lista

` ... `

testo in grassetto

`<u> ... </u>`

HTML - esempi di tag

tag html

cosa caratterizza

`<html> ... </html>`

documento HTML

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

parte visibile

` ... `

lista ordinata

` ... `

elemento della lista

` ... `

testo in grassetto

`<u> ... </u>`

testo sottolineato

HTML - esempi di tag

tag html

cosa caratterizza

`<html> ... </html>`

documento HTML

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

parte visibile

` ... `

lista ordinata

` ... `

elemento della lista

` ... `

testo in grassetto

`<u> ... </u>`

testo sottolineato

`<i> ... </i>`

HTML - esempi di tag

tag html

cosa caratterizza

<code><html> ... </html></code>	documento HTML
<code><body> ... </body></code>	parte visibile
<code> ... </code>	lista ordinata
<code> ... </code>	elemento della lista
<code> ... </code>	testo in grassetto
<code><u> ... </u></code>	testo sottolineato
<code><i> ... </i></code>	testo in corsivo

HTML - correttezza

- Non tutti i documenti formati da testo e tag sono descrizioni HTML corrette.

HTML - correttezza

- Non tutti i documenti formati da testo e tag sono descrizioni HTML corrette.
- Le descrizioni HTML corrette possono essere formalmente generate attraverso una grammatica acontestuale.

HTML - grammatica

Porzione di grammatica per generare un elenco numerato:

HTML - grammatica

Porzione di grammatica per generare un elenco numerato:

● Elenco-num \rightarrow ` Lista `

HTML - grammatica

Porzione di grammatica per generare un elenco numerato:

- Elenco-num \rightarrow ` Lista `

Elenco-num e Lista sono metasimboli;
`` e `` sono simboli terminali;
lo spazio indica la concatenazione.

HTML - grammatica

Porzione di grammatica per generare un elenco numerato:

- Elenco-num \rightarrow `` Lista ``
- Lista \rightarrow ϵ | Elemento Lista

HTML - grammatica

Porzione di grammatica per generare un elenco numerato:

- Elenco-num \rightarrow ` Lista `
- Lista \rightarrow ϵ | Elemento Lista
- Elemento \rightarrow ` Testo `

HTML - grammatica

Porzione di grammatica per generare un elenco numerato:

- Elenco-num \rightarrow ` Lista `
- Lista \rightarrow ϵ | Elemento Lista
- Elemento \rightarrow ` Testo `
- Testo \rightarrow ϵ | Carattere Testo

HTML - grammatica

Porzione di grammatica per generare un elenco numerato:

- Elenco-num \rightarrow ` Lista `
- Lista \rightarrow ϵ | Elemento Lista
- Elemento \rightarrow ` Testo `
- Testo \rightarrow ϵ | Carattere Testo
- Carattere \rightarrow `a|b| ... | A|B|...`

HTML - grammatica

Nelle grammatica precedente abbiamo:

- Metasimboli
Elenco-num, Lista, Elemento, Testo,
Carattere.

HTML - grammatica

Nelle grammatica precedente abbiamo:

- Metasimboli
Elenco-num, Lista, Elemento, Testo,
Carattere.
- Simboli Terminali
``, ``, ``, ``, `a,b, ...` , `A,B...`

HTML - grammatica

Nelle grammatica precedente abbiamo:

- Metasimboli
Elenco-num, Lista, Elemento, Testo,
Carattere.
- Simboli Terminali
``, ``, ``, ``, `a,b, ...` , `A,B...`
- Concatenazione
indicata dallo spazio.

XML - eXtensible Markup Language

- L'autore ha la possibilità di inventare nuovi tag

XML - eXtensible Markup Language

- L'autore ha la possibilità di inventare nuovi tag
- I tag costituiscono delle coppie di parentesi:
 <nome-tag> e' una parentesi aperta
 </nome-tag> e' la corrispondente parentesi chiusa

XML - eXtensible Markup Language

- L'autore ha la possibilità di inventare nuovi tag
- I tag costituiscono delle coppie di parentesi:
<nome-tag> e' una parentesi aperta
</nome-tag> e' la corrispondente parentesi chiusa
- Le parentesi (tag) sono potenzialmente infinite

XML - eXtensible Markup Language

- L'autore ha la possibilità di inventare nuovi tag
- I tag costituiscono delle coppie di parentesi:
<nome-tag> e' una parentesi aperta
</nome-tag> e' la corrispondente parentesi chiusa
- Le parentesi (tag) sono potenzialmente infinite
- Una parola XML e' sostanzialmente un testo inserito in una sequenza di parentesizzazioni ben formate.

XML - doc ben formati

Un documento XML si dice ben formato se:

- esiste un tag che contiene tutto il documento
<S> tutto il documento </S>

XML - doc ben formati

Un documento XML si dice ben formato se:

- esiste un tag che contiene tutto il documento
`<S>` tutto il documento `</S>`
- ogni tag aperto `<x>` è seguito dal corrispondente tag chiuso `</x>`

XML - doc ben formati

Un documento XML si dice ben formato se:

- esiste un tag che contiene tutto il documento
`<S>` tutto il documento `</S>`
- ogni tag aperto `<x>` è seguito dal corrispondente tag chiuso `</x>`
- i tag sono annidati correttamente:
se `<x>` è aperta prima di `<y>`
allora `</x>` va chiusa dopo `</y>`

XML - esempi

I seguenti documenti sono ben formati?

XML - esempi

I seguenti documenti sono ben formati?

- `<a>`

XML - esempi

I seguenti documenti sono ben formati?

● `<a>` NO

XML - esempi

I seguenti documenti sono ben formati?

• `<a>` NO

• `<a>`

XML - esempi

I seguenti documenti sono ben formati?

● `<a>` NO

● `<a>` NO

XML - esempi

I seguenti documenti sono ben formati?

● `<a>` NO

● `<a>` NO

● `<a>`

XML - esempi

I seguenti documenti sono ben formati?

- `<a>` NO
- `<a>` NO
- `<a>` NO

XML - esempi

I seguenti documenti sono ben formati?

● `<a>` NO

● `<a>` NO

● `<a>` NO

● `<a>`

XML - esempi

I seguenti documenti sono ben formati?

- `<a>` NO
- `<a>` NO
- `<a>` NO
- `<a>` SI

XML - esempi

I seguenti documenti sono ben formati?

● `<a>` NO

● `<a>` NO

● `<a>` NO

● `<a>` SI

● `<a>`

XML - esempi

I seguenti documenti sono ben formati?

- `<a>` NO
- `<a>` NO
- `<a>` NO
- `<a>` SI
- `<a>` SI

XML - linguaggio acontestuale?

- L'insieme dei linguaggi XML ben formati non costituisce un linguaggio acontestuale, perchè i potenziali tipi di parentesi sono infiniti.

XML - linguaggio acontestuale?

- L'insieme dei linguaggi XML ben formati non costituisce un linguaggio acontestuale, perchè i potenziali tipi di parentesi sono infiniti.
- Fissando a priori un insieme finito di possibili parentesi, il linguaggio dei documenti ben formati contenenti solo quelle parentesi risulta essere acontestuale.

XML e DTD

Le grammatiche acontestuali possono essere utilizzate per specificare sottoclassi di documenti XML ben formati.

XML e DTD

Le grammatiche acontestuali possono essere utilizzate per specificare sottoclassi di documenti XML ben formati.

- Si usa una opportuna notazione per definire una grammatica acontestuale, che viene detta **DTD** (Document Type Definition)

XML e DTD

Le grammatiche acontestuali possono essere utilizzate per specificare sottoclassi di documenti XML ben formati.

- Si usa una opportuna notazione per definire una grammatica acontestuale, che viene detta **DTD** (Document Type Definition)
- Un documento XML si dice **valido** se è generato dalla DTD

Notazione usata nelle DTD

Possibili forme per le regole di produzione nella DTD:

Notazione usata nelle DTD

Possibili forme per le regole di produzione nella DTD:

```
<!ELEMENT tag (tag1, ..., tagN) >
```

```
<!ELEMENT tag (#PCDATA) >
```

Notazione usata nelle DTD

Possibili forme per le regole di produzione nella DTD:

```
<!ELEMENT tag (tag1,...,tagN) >
```

```
<!ELEMENT tag (#PCDATA) >
```

dove `tag`, `tag1`, ... , `tagN` sono metasimboli e la virgola indica la concatenazione.

Notazione usata nelle DTD

Possibili forme per le regole di produzione nella DTD:

```
<!ELEMENT tag (tag1,...,tagN) >
```

```
<!ELEMENT tag (#PCDATA) >
```

Le regole precedenti corrispondono, secondo la nostra notazione, alle produzioni:

Notazione usata nelle DTD

Possibili forme per le regole di produzione nella DTD:

```
<!ELEMENT tag (tag1,...,tagN) >
```

```
<!ELEMENT tag (#PCDATA) >
```

Le regole precedenti corrispondono, secondo la nostra notazione, alle produzioni:

```
tag → <tag> tag1 ... tagN </tag>
```

```
tag → <tag> Testo </tag>
```

dove Testo è definito come in precedenza.

Esempio: e-mail (1)

Esempio: usiamo XML per caratterizzare dei messaggi di posta elettronica.

Esempio: e-mail (1)

Esempio: usiamo XML per caratterizzare dei messaggi di posta elettronica.

- Fissiamo una DTD che definisca la struttura di un e-mail. Per semplicità consideriamo gli elementi `from`, `to`, `subject`, `body`

Esempio: e-mail (1)

Esempio: usiamo XML per caratterizzare dei messaggi di posta elettronica.

- Fissiamo una DTD che definisca la struttura di un e-mail. Per semplicità consideriamo gli elementi `from`, `to`, `subject`, `body`
- Traduciamo la DTD nella notazione standard delle produzioni

Esempio: e-mail (2)

Esempio di DTD per messaggi di posta elettronica:

```
<!ELEMENT mail  
  (from,to,subject,body) >  
<!ELEMENT from (#PCDATA) >  
<!ELEMENT to (#PCDATA) >  
<!ELEMENT subject (#PCDATA) >  
<!ELEMENT body (#PCDATA) >
```

Esempio: e-mail (3)

```
<!ELEMENT mail  
(from,to,subject,body)>
```

corrisponde alla produzione standard

mail → <mail> from to subject body </mail>

Esempio: e-mail (3)

`<!ELEMENT from (#PCDATA)>`

corrisponde alle produzioni standard

`from` \rightarrow `<from>Testo</from>`

`Testo` \rightarrow ϵ | `Carattere` `Testo`

`Carattere` \rightarrow `a|b| ... | A|B|...`

Esempio: e-mail (3)

`<!ELEMENT to (#PCDATA)>`

corrisponde alle produzioni standard

`to` \rightarrow `<to>Testo</to>`

`Testo` \rightarrow ϵ | Carattere `Testo`

`Carattere` \rightarrow `a|b| ... | A|B|...`

Esempio: e-mail (3)

`<!ELEMENT subject (#PCDATA)>`

corrisponde alle produzioni standard

`subject` \rightarrow `<subject>Testo</subject>`

`Testo` \rightarrow ϵ | Carattere `Testo`

`Carattere` \rightarrow `a|b| ... | A|B|...`

Esempio: e-mail (3)

`<!ELEMENT body (#PCDATA)>`

corrisponde alle produzioni standard

`body` \rightarrow `<body>Testo</body>`

`Testo` \rightarrow ϵ | Carattere `Testo`

`Carattere` \rightarrow `a|b| ... | A|B|...`

Esempio: e-mail (3)

Traduzione delle produzioni nella DTD secondo la notazione standard:

mail \rightarrow `<mail>from to subject body </mail>`

from \rightarrow `<from>Testo</from>`

to \rightarrow `<to>Testo</to>`

subject \rightarrow `<subject>Testo</subject>`

body \rightarrow `<body>Testo</body>`

Testo \rightarrow ϵ | Carattere Testo

Carattere \rightarrow a|b| ... | A|B|...

Esempio: e-mail (4)

Esempio di file XML valido rispetto alla precedente DTD

```
<?xml version="1.0"?>
<mail>
<from>Alessandra</from>
<to>Fabrizio</to>
<subject>cinema</subject>
<body>Che film andiamo a vedere
questa sera?</body>
</mail>
```

Espressioni regolari nelle DTD

Nella Document Type Definition è possibile anche utilizzare espressioni regolari.

Espressioni regolari nelle DTD

Nella Document Type Definition è possibile anche utilizzare espressioni regolari.

- Sono ammesse produzioni del tipo
`<!ELEMENT tag (descr)>`
dove `descr` può contenere sia i metasimboli che gli operatori `| * + ?` con l'usuale significato visto per le regex in UNIX.

Espressioni regolari nelle DTD

Nella Document Type Definition è possibile anche utilizzare espressioni regolari.

- Sono ammesse produzioni del tipo
`<!ELEMENT tag (descr)>`
dove `descr` può contenere sia i metasimboli che gli operatori `| * + ?` con l'usuale significato visto per le regex in UNIX.
- E' possibile tradurre anche queste produzioni più complesse usando un insieme finito di produzioni nella nostra notazione standard.

Esempio di regex nelle DTD

```
<!ELEMENT to (Destinatario+)>  
<!ELEMENT Destinatario (#PCDATA)>
```

Esempio di regex nelle DTD

```
<!ELEMENT to (Destinatario+)>  
<!ELEMENT Destinatario (#PCDATA)>
```

corrispondono alle produzioni standard

to → <to> Destinatari </to>

Destinatari → Destinatario

Destinatari → Destinatario Destinatari

Destinatario → Testo

Testo ...