Markup e HTML

Angelo Di Iorio

(dal materiale di Fabio Vitali)

Università di Bologna

Il markup

Cos'è il markup?

- Definiamo markup ***ogni mezzo per rendere esplicita una particolare interpretazione di un testo***.
- Per esempio, tutte quelle aggiunte al testo scritto che permettono di renderlo più fruibile.
- Oltre a rendere il testo più leggibile, il markup permette anche di specificare ulteriori usi del testo.
- Con il markup per sistemi informatici (il nostro caso), specifichiamo le modalità esatte di utilizzo del testo nel sistema stesso.
- Il markup non è soltanto un inevitabile e sgradevole risultato della informatizzazione dell'arte tipografica. Non è qualcosa che sta con noi a causa dell'informatica.

SGML e XML

- SGML (Standard Generalized Markup Language - 1986) e XML (Extensible Markup Language - 1997) sono gli standard su cui è basato HTML.
- Sono meta-linguaggi non proprietario di markup descrittivo che facilita markup leggibili, generici, strutturali, gerarchici.

Meta-linguaggio di markup

Un meta-linguaggio è un linguaggio per definire linguaggi, una grammatica di costruzione di linguaggi.

SGML e XML non sono linguaggi di markup, ma linguaggi con cui definiamo linguaggi di markup.

SGML e XML non danno dunque tutte le risposte di markup che possano sorgere a chi vuole arricchire testi per qualunque motivo, ma forniscono una sintassi per definire il linguaggio adatto.

SGML e XML non sanno cos'è un paragrafo, una lista, un titolo, ma forniscono grammatiche che ci permettono di definirli.

Linguaggio non proprietario

Non proprietario significa che non esiste un'unica ditta o casa produttrice che ne detiene il controllo.

Viceversa RTF è © Microsoft, mentre PostScript e Acrobat sono © Adobe.

Essendo standard non proprietario, non dipende da un singolo programma, da una singola architettura. Gli stessi dati possono essere portate da un programma all'altro, da una piattaforma all'altra senza perdita di informazione.

Inoltre, anche l'evoluzione nel tempo è assicurata per lo stesso motivo.

Markup leggibile

In SGML e XML il markup è posto in maniera leggibile a fianco degli elementi del testo a cui si riferiscono.

Essi possono sia essere usati da un programma, sia letti da un essere umano. Possono sia essere aggiunti da un programma (editor), sia da un essere umano con un programma non specifico.

Il markup è semplice testo facilmente interpretabile.

Markup strutturato

SGML e XML permettono di definire delle strutture, suggerite o imposte, a cui i documenti si debbono adeguare.

Ad esempio, si può imporre che un testo sia diviso in capitoli, ognuno dei quali dotato di un titolo, una breve descrizione iniziale e almeno un paragrafo di contenuto.

È cioè possibile definire una serie di regole affinché il testo sia considerabile strutturalmente corretto.

Markup gerarchico

Le strutture imposte da SGML e XML sono tipicamente a livelli di dettaglio successivi.

Gli elementi del testo possono comporsi gli uni con gli altri, permettendo di specificare la struttura in maniera gerarchica.

Ad esempio, si può imporre che il libro sia fatto di capitoli, e che questi a loro volta siano fatti di un una descrizione e molti paragrafi. Una descrizione sarà quindi fatta di elementi, ciascuno dei quali sarà una frase composta di testo; i paragrafi saranno testo eventualmente contenenti anche elementi in evidenza o figure.

I componenti del markup

Un documento con markup di derivazione SGML (inclusi HTML, XML, ecc.) contiene una varietà dei seguenti componenti

- Elementi
- Attributi
- Testo (detto anche `#PCDATA`)
- Commenti

Elementi, Attributi, #PCDATA, Commenti

- Gli elementi sono le parti di documento dotate di un senso proprio.
- Il titolo, l'autore, i paragrafi del documento sono tutti elementi.
- Un elemento è individuato da un tag iniziale, un contenuto ed un tag finale.
- **Non confondere i tag con gli elementi!**

```
<TITOLO>Tre uomini in barca</TITOLO>
```

Elementi, Attributi, #PCDATA, Commenti

- Gli attributi sono informazioni aggiuntive sull'elemento che non fanno effettivamente parte del contenuto (meta-informazioni).
- Essi sono posti dentro al tag iniziale dell'elemento. Tipicamente hanno la forma nome="valore"

```
<romanzo file="threemen.sgm">…</romanzo>
<capitolo N="1">Capitolo primo</capitolo>
```

Elementi, Attributi, #PCDATA, Commenti

- Rappresenta il contenuto vero e proprio del documento.
- Esso corrisponde alle parole, gli spazi e la punteggiatura che costituiscono il testo.
- Viene anche detto **`#PCDATA`** (Parsed Character DATA) perché i linguaggi di markup definiscono *character data* (CDATA) il contenuto testuale vero e proprio, e quello degli elementi è soggetto ad azione di parsing (perlopiù per identificare e sostituire le entità).

Elementi, Attributi, #PCDATA, Commenti

- I documenti di markup possono contenere commenti, ovvero note da un autore all'altro, da un editore all'altro, ecc.
- Queste note non fanno parte del contenuto del documento, e le applicazioni di markup li ignorano.
- Sono molto comodi per passare informazioni tra un autore e l'altro, o per trattenere informazioni per se stessi, nel caso le dimenticassimo.

```
<!-- Questo è ignorato dal parser -->
```

HTML

La struttura di un documento HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
  <head>
    <title>Titolo del documento</title>
  </head>
  <body>
    <p>Testo di un paragrafo</p>
  </body>
</html>
```

Premesse sintattiche

Maiuscolo/minuscolo:

- HTML non è sensibile al maiuscolo/minuscolo (XHTML lo è e vuole tutto in minuscolo). Editor di antica concezione scrivevano i tag in maiuscolo (per meglio distinguerli dal testo), i nuovi in minuscolo (per meglio transitare verso XHTML).

Whitespace

- HTML *collassa* tutti i caratteri di whitespace (SPACE, TAB, CR, LF) in un unico spazio. Questo permette di organizzare il sorgente in modalità leggibile senza influenzare la visualizzazione su browser.
- L'entità   permette di inserire spazi non collassabili.

Estensioni del linguaggio

- I browser permettono (ignorandoli), mentre i validatori segnalano elementi e attributi che non appartengono al linguaggio
- `<p pluto="paperino"><pippo>testo</pippo></p>`

Entità in HTML

HTML definisce un certo numero di entità per quei caratteri che sono:
proibiti perché usati in HTML (<, >, &, “, ecc.)
proibiti perché non presenti nell’ASCII a 7 bit.
Ad esempio:

amp	&	quot	“
lt (less than)	<	gt (greater than)	>
reg	®	nbsp (non-breaking space)	
Aelig	Æ	Aacute	Á
Agrave	À	Auml	Ä
aelig	æ	aacute	á
agrave	à	auml	ä
ccedil	ç	ntilde	ñ
ecc.			

  viene spesso usato per far sembrare piena una struttura anche se non lo è (ad esempio, una cella di tabella).

Attributi globali

Sono attributi globali quelli definiti su tutti gli elementi del linguaggio HTML.

I *coreattrs* costituiscono attributi di qualificazione e associazione globale degli elementi. Per lo più per CSS e link ipertestuali.

- **Id**: un identificativo unico (su tutto il documento)
- **Style**: un breve stile CSS associato al singolo elemento
- **Class**: una lista (separata da spazi) di nomi di classe (per attribuzione semantica e di stile CSS)
- **Title**: un testo secondario associato all'elemento (per accessibilità e informazioni aggiuntive)

Ne parleremo a lungo venerdì

Gli elementi di HTML

Possiamo considerare i tag di HTML organizzati secondo alcune categorie:

- Tag della struttura complessiva (HTML, HEAD, BODY)
- Tag inline (B, I, SPAN, ecc.)
- Tag di blocco e liste (P, H1, H2, DIV, UL, OL, ecc.)
- Tag speciali (A, IMG, HR, BR)
- Tabelle (TABLE, TR, TD, TH)

Tag di struttura

HTML: La radice dell'albero.

HEAD: Informazioni globali sull'albero

- Ad esempio, il titolo (ma anche gli stili del documento, le keyword di metadatazione, gli script Javascript, ecc.)

BODY: Il contenuto vero e proprio del documento

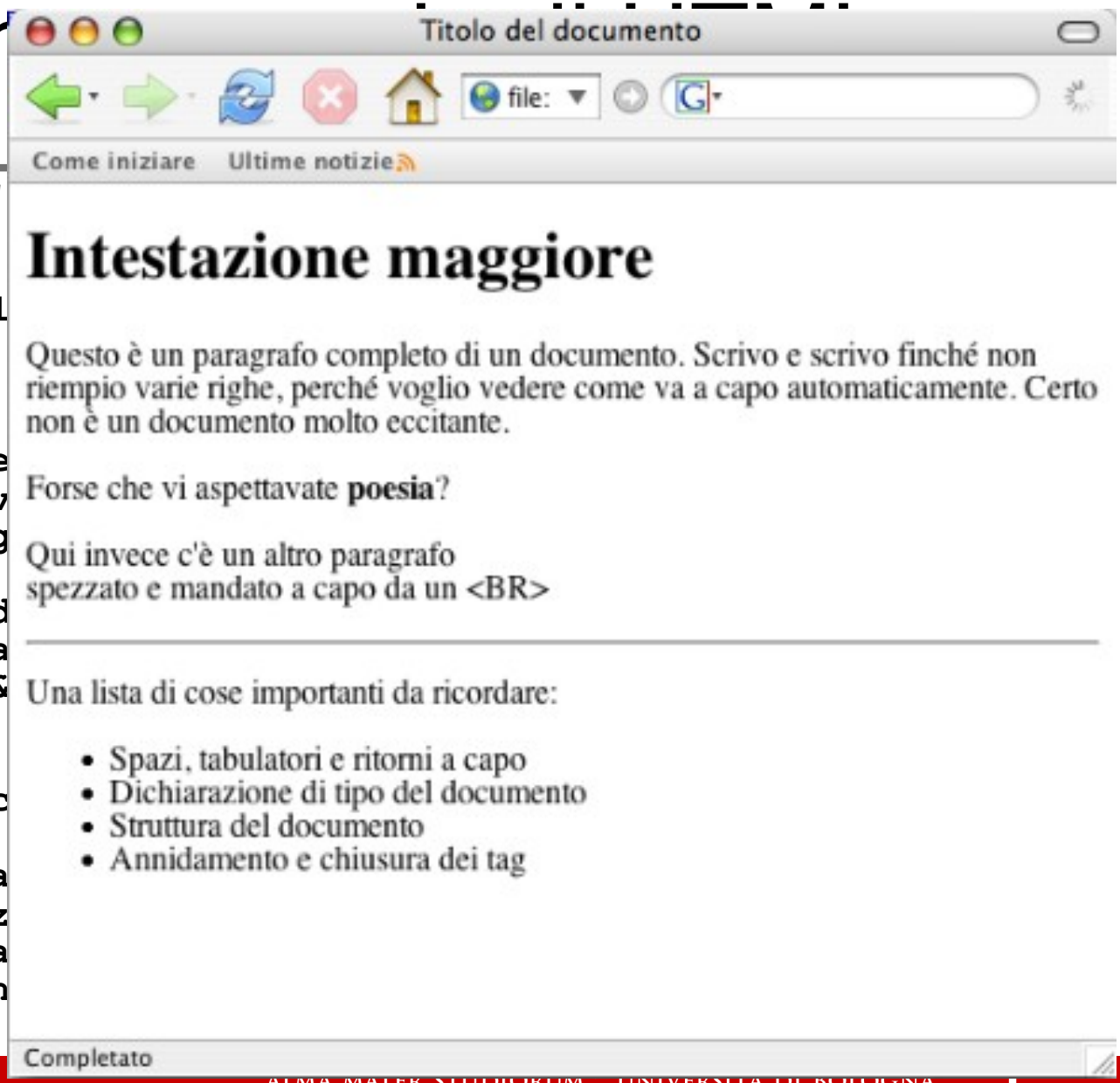
- Tutto quel che è visibile: testo, immagini, tabelle, video, form, ecc.
- Strutturato o non strutturato

Un primo esempio di HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
   <head>
      <title>Titolo del documento</title>
   </head>
   <body>
      <h1>Intestazione maggiore</h1>
      <p>Questo &egrave; un paragrafo completo di un
      documento.  Scrivo e scrivo finch&eacute; non
      riempio varie righe, perch&eacute; voglio
      vedere come va a capo automaticamente. Certo
      non &egrave; un documento molto eccitante.</p>
      <p>Forse che vi aspettavate <b>poesia</b>? </p>
      <p>Qui invece c'&egrave; un paragrafo <br/> spezzato
      e mandato a capo da un &lt;BR&gt;</p>
      <hr/>
      <p>Una lista di cose importanti da ricordare:</p>
      <ul>
          <li>Spazi, tabulatori e ritorni a capo</li>
          <li>Dichiarazione di tipo del documento</li>
          <li>Struttura del documento</li>
          <li>Annidamento e chiusura dei tag</li>
      </ul>
   </body>
</html>
```

Un prim

```
<!DOCTYPE HTML PUBLIC "
<html>
    <head>
        <title>Titolo del
    </head>
    <body>
        <h1>Intestazione
        <p>Questo &egrave
        documento.  Scriv
        riempio varie rig
        vedere come va a
        non &egrave; un d
        <p>Forse che vi a
        <p>Qui invece c'&
        e mandato a capo
        <hr/>
        <p>Una lista di c
        <ul>
            <li>Spazi, ta
            <li>Dichiaraz
            <li>Struttura
            <li>Annidamen
        </ul>
    </body>
</html>
```



Elementi inline (1)

- Gli elementi *inline* (o di carattere) non spezzano il blocco (non vanno a capo) e si includono liberamente l'uno dentro all'altro. Non esistono vincoli di contenimento.
- Si dividono in elementi *fontstyle* e elementi *phrase*.
- I tag *fontstyle* forniscono informazioni specifiche di rendering. Molti sono deprecati, e si suggerisce comunque sempre di usare gli stili CSS.
  - TT (TeleType, font monospaziato, ad es. Courier),
  - I (corsivo),
  - B grassetto,
  - U (sottolineato - deprecato),
  - S e STRIKE (testo barrato - deprecato),
  - BIG, SMALL (testo più grande e più piccolo)

Elementi inline (2)

I tag *phrase* (di fraseazione o idiomatici) aggiungono significato a parti di un paragrafo.

- EM (enfasi),
- STRONG (enfasi maggiore),
- DFN (definizione),
- CODE (frammento di programma),
- SAMP (output d'esempio),
- KBD (testo inserito dall'utente),
- VAR (variabile di programma),
- CITE (breve citazione),
- Q (citazione lunga),
- ABBR e ACRONYM (abbreviazioni ed acronimi),
- SUP e SUB (testo in apice e in pedice),
- BDO (bidirectional override),
- SPAN (generico elemento inline)

Elementi di blocco

I tag di blocco definiscono l'esistenza di blocchi di testo che contengono elementi inline.

Elementi base:

- P (paragrafo),
- DIV (generico blocco),
- PRE (blocco preformattato),
- ADDRESS (indicazioni sull'autore della pagina),
- BLOCKQUOTE (citazione lunga)

Blocchi con ruolo strutturale

- H1, H2, H3, H4, H5, H6 (intestazione di blocco)

Elementi per liste

Le liste di elementi sono contenitori di elementi omogenei per tipo.

- UL: Lista a pallini di <LI>; Attributo type (disc, square, circle)
- OL: lista a numeri o lettere di <LI>; attributi start (valore iniziale) e type (1, a, A, i, I).
- DIR, MENU: liste compatte, poco usate
- DL: lista di definizioni <DT> (definition term) e <DD> (definition data)

```
•<UL>
  <LI>Primo</LI>
  <LI>Secondo</LI>
  <LI>Terzo</LI>
•</UL>
```

```
•<DL>
  <DT>Lista</DT>
  <DD>Un contenitore
    di elementi</DD>
  <DT>Phrase</DT>
  <DD>Elementi inline di
   tipo idiomatico</DD>
•</DL>
```

Elementi generici

 e sono cosiddetti *elementi generici*: privi di caratteristiche semantiche o presentazionali predefinite, assumono quelle desiderate con l'aiuto dei loro attributi (style, class, lang).

 mantiene la natura di elemento blocco, e la natura di elemento inline, ma ogni altra caratteristica è neutra.

Es.: identificare con elementi appositi tutti i titoli di film può realizzarsi con <titolofilm>, ma si esce da HTML, oppure con .

Altri elementi di testo

- I link ipertestuali (A)
- Le immagini e gli oggetti multimediali (IMG)
- Piccoli effetti grafici
  - HR (horizontal rule): una piccola riga orizzontale attraverso lo schermo. Si può controllarne la larghezza e lo spessore.
  - BR (break): una andata a capo forzata (non spezza logicamente il paragrafo, impone semplicemente il cambio riga)

Link

I link sono definiti attraverso l'elemento A (anchor interna al documento). <A> è sintatticamente un elemento inline (sta dentro ai blocchi, come B, I, ecc.) Gli <A> non possono però annidarsi gli uni dentro agli altri.
Attributi:

- HREF: specifica l'URI di una destinazione. Quindi **`<A HREF="xx"> . . . </A>`** è un *estremo di partenza* di un link.
- NAME: specifica un nome utilizzabile come destinazione puntuale di un link. Quindi **<A NAME="xx"> ... </A>** è *l'estremo di arrivo* di un link.

```
<p>Un link a <a
href="B.html">tutto</a> il
documento B</p>
<p>Un link a <a
href="B.html#loc1">una
locazione</a> del documento
B</p>
```

A.html

B.html

```
<p>Paragrafo di inizio di B</
p>
<p><a name="loc1">La
destinazione</a> del link
puntuale in A.</p>
```

Immagini

Le immagini inline sono definite attraverso l'elemento IMG. Formati tipici: JPEG, GIF, PNG. Alcuni attributi:

HTML: <IMG SRC="pippo.gif" ALT="Foto di pippo">

- SRC (obbligatorio): l'URL del file contenente l'immagine.
- ALT: testo alternativo in caso di mancata visualizzazione dell'immagine
- NAME: un nome usabile per riferirsi all'immagine
- USEMAP: indica che l'immagine è una mappa client-side
- ISMAP: indica che l'immagine è una mappa server-side
- WIDTH: forza una larghezza dell'immagine.
- HEIGHT: forza una altezza dell'immagine.
- ALIGN, BORDER, VSPACE, HSPACE: deprecati, specificano il rendering.

Tabelle

Le tabelle vengono specificate riga per riga.

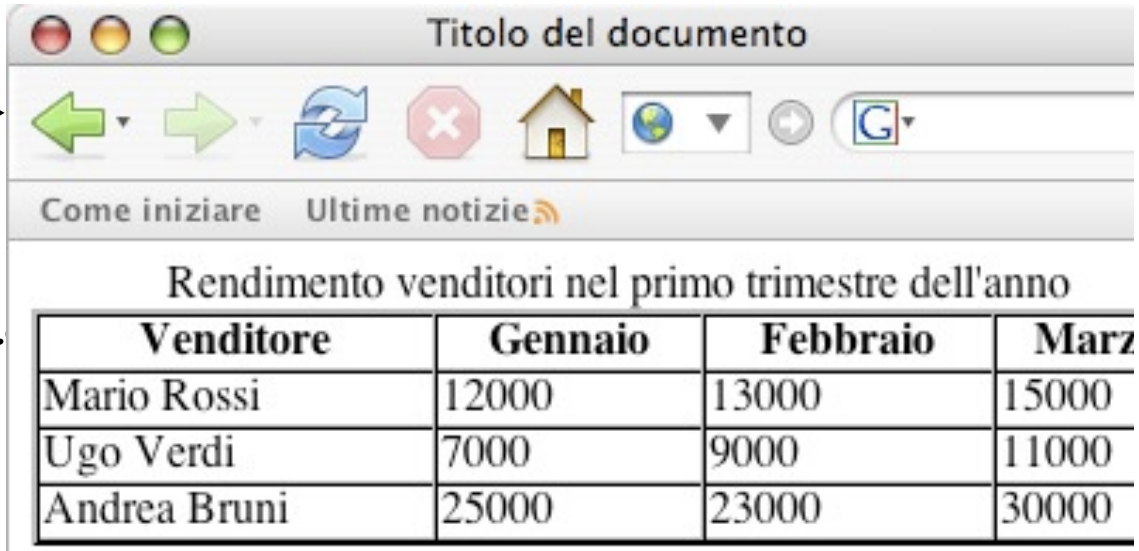
Di ogni riga si possono precisare gli elementi, che sono o intestazioni o celle normali.

Una tabella può anche avere una didascalia, un'intestazione ed una sezione conclusiva.

E' possibile descrivere insieme le caratteristiche visive delle colonne.

Una semplice tabella

```
<table border="2" align="center" width=”100%”
       cellpadding="5" cellspacing="0">
  <caption>Rendimento venditori nel primo
           trimestre dell'anno</caption>
  <tr>
    <th>Venditore</th><th>Gennaio</th>
    <th>Febbraio</th> <th>Marzo</th>
  </tr>
  <tr>
     <td>Mario Rossi</td>
     <td>13000</td><td>15
  </tr>
  <tr>
     <td>Ugo Verdi</td><t
     <td>9000</td><td>110
  </tr>
  <tr>
     <td>Andrea Bruni</td
     <td>23000</td><td>30
  </tr>
</table>
```



I tag di HEAD

HEAD contiene delle informazioni che sono rilevanti per tutto il documento. Esse sono:

- TITLE: il titolo del documento
- BASE: l'URL da usare come base per gli URL relativi
- LINK: link di documenti a tutto il documento
- SCRIPT: librerie di script
- STYLE: librerie di stili
- META: meta-informazioni sul documento

Form

Con i FORM si utilizzano le pagine HTML per inserire valori che vengono poi elaborati sul server. I FORM sono legati al CGI (Common Gateway Interface):

Il browser raccoglie dati con un form dall'utente. Crea una connessione HTTP con il server, specificando una ACTION (cioè un applicazione che funga da destinatario) a cui fare arrivare i dati. Il destinatario riceve i dati, li elabora e genera un documento di risposta, che viene spedito, tramite il server HTTP, al browser.

I controlli tipati e nominati vengono usati per l'inserimento dei dati nei form: campi di inserimento dati, pulsanti, bottoni radio, checkbox, liste a scomparsa, ecc.

Il codice della form

```
<p>Questo documento contiene una prova di FORM:</p>

<form method="get" action="http://www.sito.it/cgi-bin/a.pl">
<p>Nome:<input type="text" name="Nome" value="Ugo" size='40' />
<br/>Cognome:<input type="text" name="Cognome" value="Rossi" /></p>
<p>Sesso: <input type="radio" name="Sesso" value="M" /> M
          <input type="radio" name="Sesso" value="F" /> F </p>
<p>/>Gusti:
  <input type="checkbox" name="Gusti" value="Arte" /> Arte
  <input type="checkbox" name="Gusti" value="Cinema" /> Cinema
  <input type="checkbox" name="Gusti" value="Fumetti" /> Fumetti
</p>

<p><input type='submit' name="Submit" value="Ok" />
   <input type='reset' name="Cancel" value="Annulla" /></p>
</form>
```

URL richiesto:

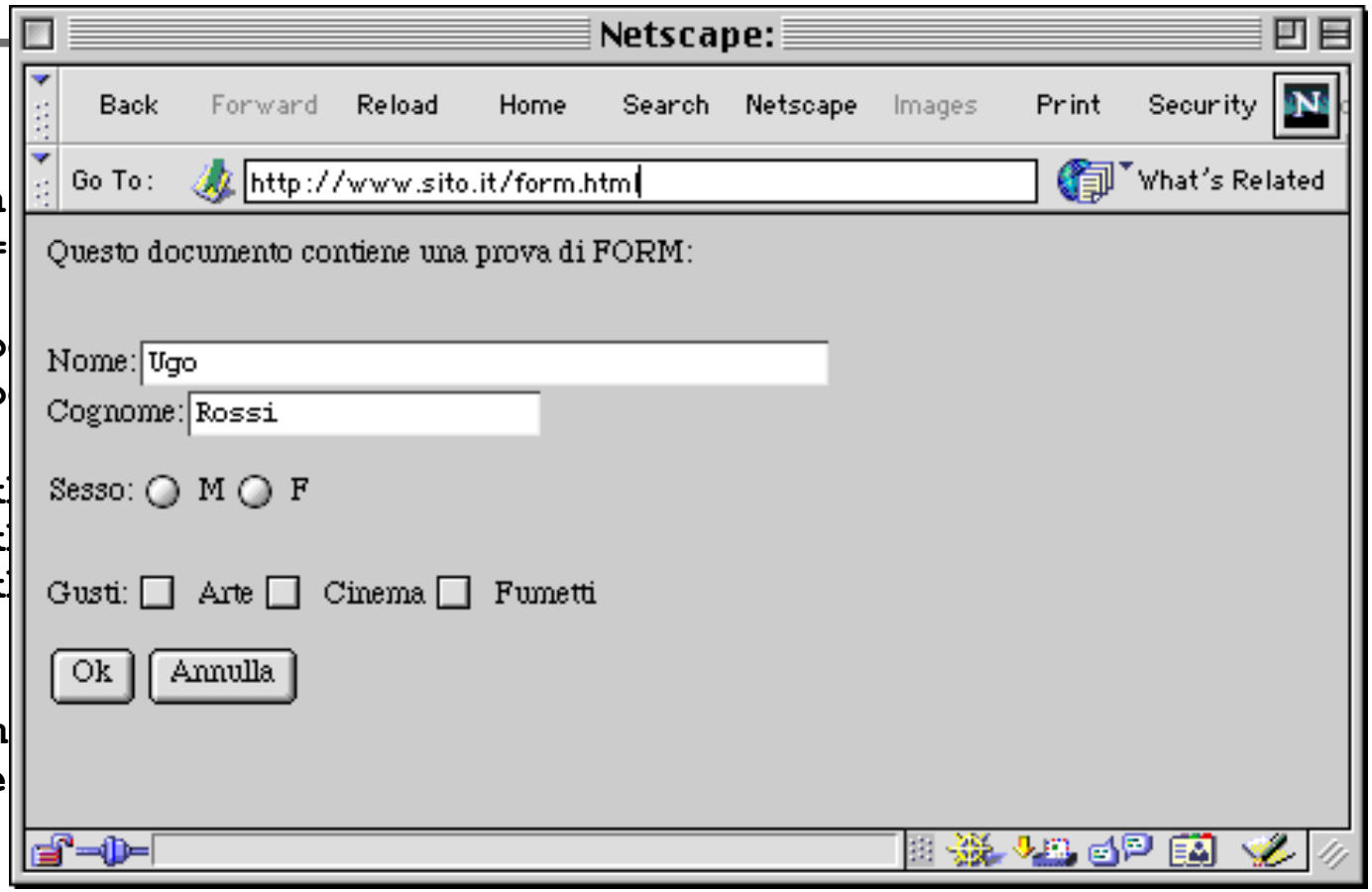
```
HTTP/1.1 GET /cgi-bin/a.pl?Nome=Ugo&Cognome=Rossi&Sesso=M&Gusti=Arte
```

Il codice della form

```
<p>Questo documento

<form method="get" a
<p>Nome:<input type=
<br/>Cognome:<input
<p>Sesso: <input typ
          <input typ
<p>/>Gusti:
  <input type="check
  <input type="check
  <input type="check
</p>

<p><input type='subm
   <input type='rese
</form>
```



URL richiesto:

```
HTTP/1.1 GET /cgi-bin/a.pl?Nome=Ugo&Cognome=Rossi&Sesso=M&Gusti=Arte
```

application/x-www-form-urlencoded

E' un estensione della codifica degli URI applicata

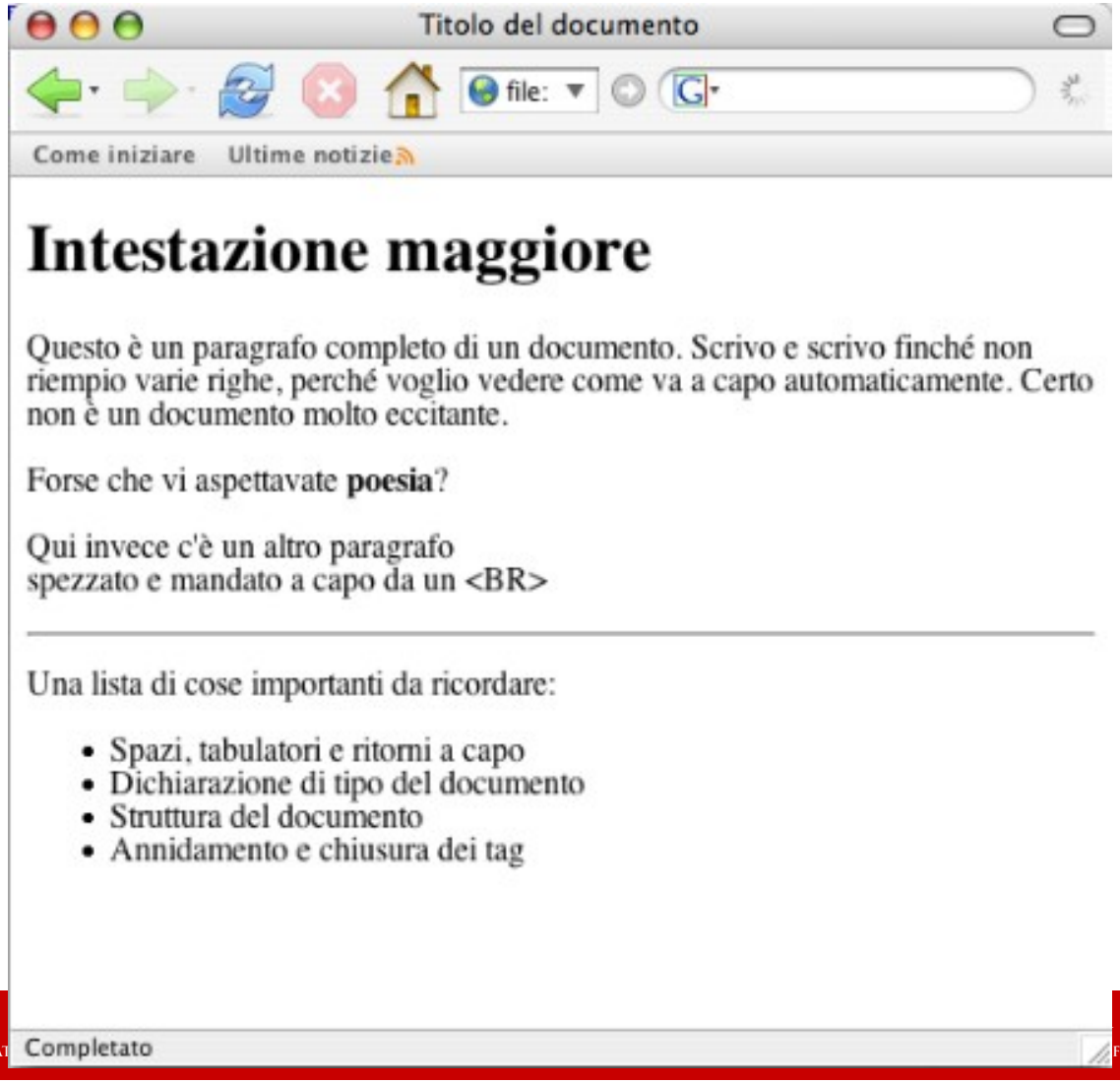
- Alla parte query dell'URI di una richiesta HTTP
- a risorse contenute nel corpo di una richiesta HTTP (ad esempio il contenuto di un POST)
- i codici non alfanumerici sono sostituiti da '%HH' (HH: codice esadecimale del carattere),
- gli spazi sono sostituiti da '+',
- i nomi dei controlli sono separati da '&',
- il valore è separato dal nome da '='

Colori

- In molte situazioni (sfondi, caratteri, ecc.) è possibile specificare un colore. HTML fornisce due modi per farlo:
  - **Codice RGB prefissato da un carattere di hash**. due caratteri esadecimali ciascuno per esprimere la quantità di Rosso, Verde e Blu del colore (00 significa assenza, FF significa presenza massima). E' possibile descrivere 4096 colori diversi.
  - **Nome**: sono definiti 16 nomi di colori: black, silver, gray, white, maroon, red, purple, fuchsia, green, lime, olive, yellow, navy, blue, teal, aqua. Browser diversi supportano colori diversi!
- Tuttavia HTML 4 depreca l'uso esplicito di colori nel documento HTML, e suggerisce di usare i fogli di stile.
  - `<font color="#FF0000">testo in rosso</font>`
  - `<body bgcolor="#008080">sfondo teal</body>`
  - `<td bgcolor="yellow">sfondo giallo</td>`

Esercizi

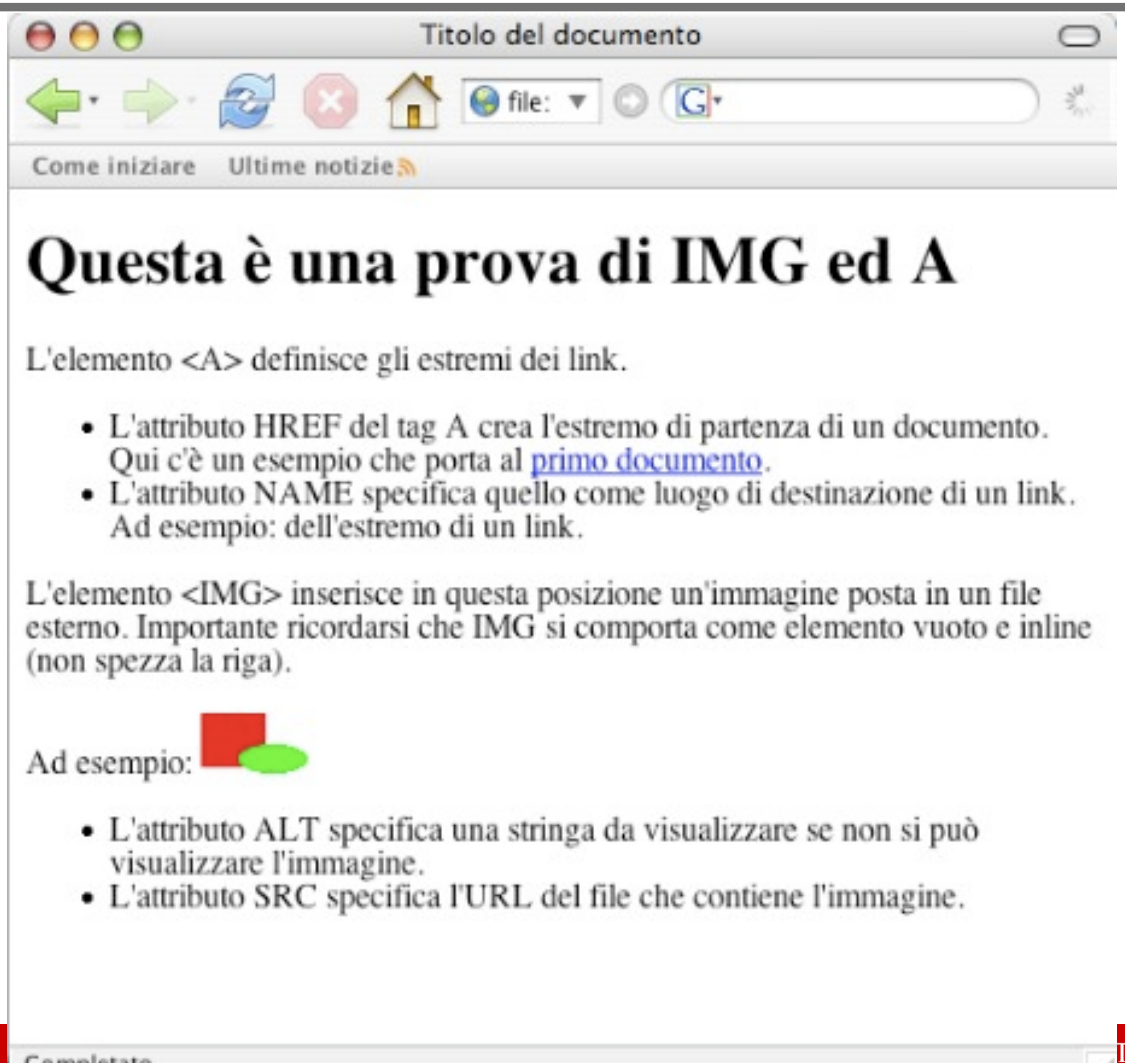
Esercizio 1


Titolo del documento
file:
Come iniziare
Ultime notizie
Intestazione maggiore
Questo è un paragrafo completo di un documento. Scrivo e scrivo finché non riempio varie righe, perché voglio vedere come va a capo automaticamente. Certo non è un documento molto eccitante.
Forse che vi aspettavate poesia?
Qui invece c'è un altro paragrafo
spezzato e mandato a capo da un
Una lista di cose importanti da ricordare:
Spazi, tabulatori e ritorni a capo
Dichiarazione di tipo del documento
Struttura del documento
Annidamento e chiusura dei tag
Completato

Soluzione 1

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
   <head>
      <title>Titolo del documento</title>
   </head>
   <body>
      <h1>Intestazione maggiore</h1>
      <p>Questo &egrave; un paragrafo completo di un
      documento.  Scrivo e scrivo finch&eacute; non
      riempio varie righe, perch&eacute; voglio
      vedere come va a capo automaticamente. Certo
      non &egrave; un documento molto eccitante.</p>
      <p>Forse che vi aspettavate <b>poesia</b>? </p>
      <p>Qui invece c'&egrave; un paragrafo <br/> spezzato
      e mandato a capo da un &lt;BR&gt;</p>
      <hr/>
      <p>Una lista di cose importanti da ricordare:</p>
      <ul>
          <li>Spazi, tabulatori e ritorni a capo</li>
          <li>Dichiarazione di tipo del documento</li>
          <li>Struttura del documento</li>
          <li>Annidamento e chiusura dei tag</li>
      </ul>
   </body>
</html>
```

Esercizio 2



Soluzione 2

```
<H1>Questa &egrave; una prova di IMG ed A</H1>
<P>L'elemento &lt;A&gt; definisce gli estremi dei link. </P>
<UL>
  <LI>L'attributo HREF del tag A crea l'estremo di partenza di
      un documento. Qui c'&egrave; un esempio che porta al <A
      HREF="1.html">primo documento</A>. </LI>
  <LI>L'attributo NAME specifica quello come luogo di destinazione
      di un link. Ad esempio:  <A NAME="prova">dell'estremo di un
      link</A>. </LI>
</UL>
<P>L'elemento &lt;IMG&gt; inserisce in questa posizione un'immagine
posta in un file esterno. Importante ricordarsi che IMG si comporta come
elemento vuoto e inline (non spezza la riga). </P>
<P>Ad esempio: <IMG SRC="esempio.gif" ALT="Un rettangolo ed
un'ellisse"></P>
<UL>
  <LI>L'attributo ALT specifica una stringa da visualizzare se non
      si pu&ograve; visualizzare l'immagine. </LI>
  <LI>L'attributo SRC specifica l'URL del file che contiene
      l'immagine.</LI>
</UL>
</BODY>
```