

Introduzione ad HTML

seconda parte

Fabio Vitali



Introduzione

Oggi esaminiamo in breve, in HTML:

- ◆ Il tag HEAD
- ◆ Oggetti e mappe
- ◆ Form
- ◆ Frame



I tag di HEAD

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

- ◆ TITLE: il titolo del documento
- ◆ ISINDEX: definisce la risorsa come indicizzata (deprecato)
- ◆ 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



I tag di HEAD:

BASE: URL relativi ed assoluti (1)

- Ogni documento HTML visualizzato in un browser ha associato un URL. Questo può appartenere allo schema di naming http://, ftp://, o anche file://. Tipicamente sono schemi gerarchici.
- Spesso accade che esistano degli oggetti dipendenti dalla pagina (immagini, stili, script, applet, link a pagine secondarie, ecc.), che appartengono allo “stesso dominio” della pagina di partenza.
- E' data allora possibilità, nello specificare l'URL della risorsa secondaria, di affidarsi ad un URL relativo, che si basa sull'URL del documento di partenza.



I tag di HEAD: LINK, SCRIPT, STYLE: documenti esterni

- Con i tag SCRIPT e STYLE si possono definire, rispettivamente, blocchi di funzioni di un linguaggio di script e blocchi di stili di un linguaggio di stylesheet.
- A volte può esser utile mettere esternamente queste specifiche, e riferirvi esplicitamente.
- In questo caso si usa il tag LINK, che permette di creare un link esplicito al documento esterno di script e/o di stili.

```
<LINK REL="STYLE SHEET"  
      TYPE="text/css"  
      HREF="lib/style1.css">
```

N.B.: Netscape lo permette solo per stylesheet. Per gli script richiede qualcosa del tipo:

```
<SCRIPT LANGUAGE="JavaScript"  
        SRC="lib/script.js">  
</SCRIPT>
```



I tag di HEAD: META: meta-informazioni (1)

Le meta-informazioni sono informazioni sul documento, piuttosto che informazioni del documento.

Il tag META è un meccanismo generale per specificare meta-informazioni su un documento HTML.

Ci sono due tipi di meta-informazioni definibili con il tag META:

- ◆ **Intestazioni HTTP:** la comunicazione HTTP fornisce informazioni sul documento trasmesso, ma il suo controllo richiede accesso al server HTTP. Con il tag META si può invece fornire informazione “stile-HTTP” senza modifiche al server.
- ◆ **Altre meta-informazioni:** i motori di ricerca usano le meta-informazioni (ad esempio “Keyword”) per catalogare ed organizzare al meglio i documenti indicizzati.



I tag di HEAD: META: meta-informazioni (2)

Ad esempio:

- ◆ `<META HTTP-EQUIV="Expires" CONTENT="10">`

Il documento viene ricaricato tra dieci secondi (ad esempio con un'immagine pubblicitaria diversa)

- ◆ `<META NAME="keyword" CONTENT="sesso, sesso, sesso, soldi, borsa, mp3, musica, sesso, sesso">`

Al documento vengono associate le parole chiave qui specificate, che permettono ai motori di ricerca di classificare il documento secondo le categorie qui precisate. Più accurata e popolare la scelta delle keyword, più volte il documento viene presentato nelle ricerche in una posizione alta.



Mappe

Un' *image map* (o mappa) è un'immagine (il cui contenuto è ignoto ad HTML) in cui alcune aree vengono identificate e a cui viene associata una destinazione ipertestuale.

L'effetto è di aver creato un'immagine in cui, a seconda della locazione in cui si clicca, si va in una destinazione diversa.

Esistono due tipi di mappe:

- ◆ Mappe server-side: il server esamina la locazione del click e attiva la destinazione specificata.
- ◆ Mappe client-side: il browser esamina la locazione del click e attiva la destinazione specificata.



Mappe server-side

La lista delle aree è sul server

Si usa un tag A che contiene esclusivamente il tag IMG con l'attributo ISMAP:

```
<a href="/imagemap/pippo">  
    
</a>
```

Quando l'utente clicca su una parte dell'immagine, il link specificato viene attivato, e il browser manda le coordinate del click.

Un'applicazione server-side identifica la destinazione corretta e la spedisce in risposta.



Mappe client-side (1)

La corrispondenza tra click e area viene fatta sul browser; l'elenco delle zone viene spedito nel documento HTML come tag AREA dentro al tag MAP:

```
<IMG SRC=immagine.gif USEMAP="#pippo">
... altra roba ...
<MAP NAME="pippo">
  <AREA shape="rect" coord="0,0,100,100" href="doc1.html">
  <AREA shape="circle" coord="100,50,100" href="doc2.html">
  ... altre aree ...
</MAP>
... altra roba ...
```



Mappe client-side (2)

Le mappe client-side si basano su due tag:

- ◆ **MAP:** identifica un blocco di aree che costituiscono la mappa di un'immagine. Attributi:
 - ✦ **Name:** I valore dell'attributo **USEMAP** dell'immagine. E' un URI!!!
- ◆ **AREA:** identifica unarea dell'immagine, a cui associare una destinazione ipertestuale. Attributi:
 - ✦ **shape:** la forma dell'area (default, rect, circle, poly)
 - ✦ **coords:** le coordinate dell'area
 - ✦ **href:** la destinazione connessa con quell'area
 - ✦ **nohref:** la specifica che l'area non ha destinazioni (senza valore)



Oggetti esterni

L'inserimento nelle pagine HTML di blocchi di dati esterni è stato un problema sentito fin dall'inizio. Il tag IMG voleva fornire supporto per immagini GIF e JPEG.

Per includere oggetti di vario tipo, ogni browser ha inventato le proprie estensioni: la Sun il tag APPLET, la Netscape il tag EMBED, Internet Explorer i tag DYNASRC e BGSOUND, senza contare IMG, svolgono più o meno lo stesso servizio.

In HTML 4 si è introdotto il tag OBJECT per uniformare tutti i tipi di oggetti esterni. Il tag APPLET è mantenuto ma deprecato, mentre il tag IMG è mantenuto per l'alta diffusione che ha.



I tag OBJECT e PARAM

OBJECT

Include un oggetto esterno. Il contenuto del tag viene visualizzato dai browser che non sanno utilizzare l'oggetto.

Attributi:

- ◆ CLASSID, CODEBASE, DATA: l'URL del codice, la base del documento che costituisce l'oggetto incluso, e i dati che bisogna attribuirgli
- ◆ ARCHIVE: la locazione di classi, dati ecc. che risultino più veloci da accedere delle rispettive fonti originarie

PARAM

Un parametro che il browser passa all'oggetto in fase di inizializzazione. Attributi:

- ◆ NAME: il nome dell'attributo
- ◆ VALUE: il valore dell'attributo
- ◆ TYPE: il tipo del valore dell'attributo



Form

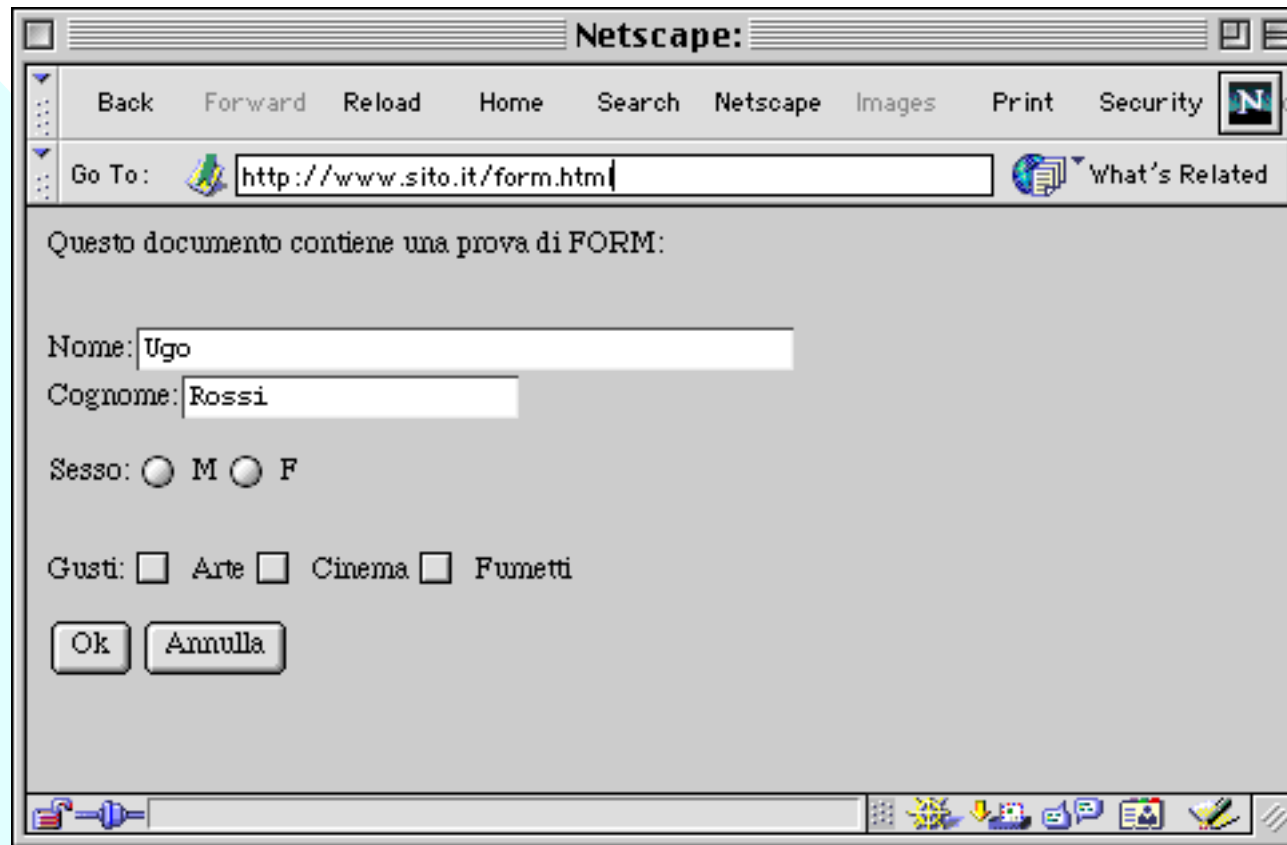
Con i FORM si utilizzano le pagine HTML per inserire valori che vengono poi elaborati sul server. I FORM sono legati al CGI (Common Gateway Interface):

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.



Esempio di form



The image shows a screenshot of a Netscape browser window. The title bar reads "Netscape:". The menu bar includes "Back", "Forward", "Reload", "Home", "Search", "Netscape", "Images", "Print", and "Security". The address bar shows "Go To: http://www.sito.it/form.htm". The main content area displays the text "Questo documento contiene una prova di FORM:". Below this, there is a form with the following fields and controls:

- Nome:
- Cognome:
- Sesso: M F
- Gusti: Arte Cinema Fumetti
- Buttons:

A large blue arrow points from the right side of the browser window towards the right edge of the slide.



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">
```

```
<BR>Nome:<INPUT TYPE="text" NAME="Nome" VALUE="Ugo" SIZE=40>
```

```
<BR>Cognome:<INPUT TYPE="text" NAME="Cognome" VALUE="Rossi">
```

```
<P>Sesso: <INPUT TYPE="radio" NAME="Sesso" VALUE="M"> M
```

```
<INPUT TYPE="radio" NAME="Sesso" VALUE="F"> F </P>
```

```
<BR>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><INPUT TYPE=submit NAME="Submit" VALUE="Ok">
```

```
<INPUT TYPE=reset NAME="Cancel" VALUE="Annulla">
```

```
</FORM>
```



I tag del form (1)

FORM

Raggruppa tutti i controlli che debbono raccogliere data per un destinatario. Attributi:

- ◆ ACTION: l'URL dell'applicazione server-side che riceverà i dati
- ◆ METHOD: il metodo HTTP che deve essere usato per i dati
- ◆ NAME: un nome univoco per il form
- ◆ ENCTYPE: il metodo di codifica dei dati da usare. Per default si usa `'application/x-www-form-urlencoded'`: 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 `'&'`, e il valore è separato dal nome da `'='`).



I tag del form (2)

INPUT

Raccoglie la maggior parte dei tipi di controllo disponibili in un form. Attributi:

- ✦ **TYPE**: rappresenta il tipo di controllo da istanziare: text, password, checkbox, radio, submit, reset, file, hidden, image, button
- ✦ **NAME**: il nome che verrà passato all'applicazione server-side insieme al valore inserito
- ✦ **VALUE**: il valore di default (per i controlli di testo) o selezionato (per i controlli di tipo bottone) che viene presentato all'utente e passato all'applicazione server-side
- ✦ **SIZE** e **MAXLENGTH**: dimensione prevista e lunghezza massima (per i testi)
- ✦ **DISABLED**, **CHECKED**: per i bottoni, radio e checkbox.



I tag del form (3)

TEXTAREA

Area di inserimento testo. Attributi (oltre a NAME):

- ✦ ROWS: il numero di righe previste
- ✦ COLS: il numero di colonne previste

SELECT, OPTGROUP

Una lista di opzioni e sotto-opzioni. Attributi (oltre a NAME):

- ✦ SIZE: il numero di opzioni visibili
- ✦ MULTIPLE: Vero se la lista permette selezioni multiple.

OPTION

Una opzione in un SELECT o in un OPTGROUP.

- ✦ SELECTED: vero se l'elemento è selezionato all'avvio
- ✦ LABEL: scritta visibile all'utente
- ✦ VALUE: valore passato all'applicazione server-side



I frame

I frame servono per dividere la finestra in più zone indipendenti ed associare a ciascuna di loro un documento diverso.

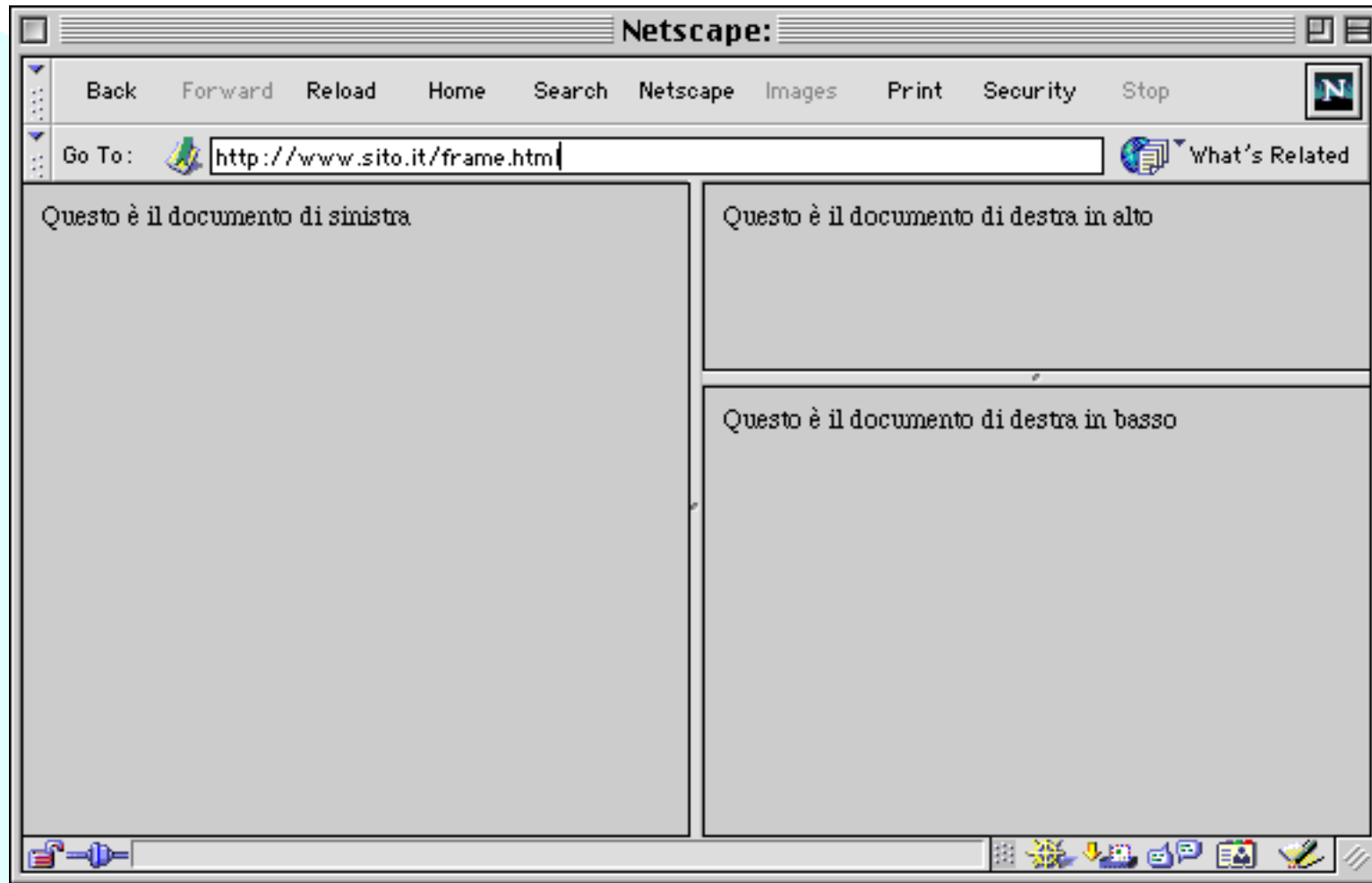
Ogni zona (o frame) ha un nome, che viene usata per specificare in quale zona viene visualizzato la destinazione di un link.

Il tag FRAMESET introduce una serie di frame o verticali o orizzontali. FRAMESET può contenere annidati altri FRAMESET (per cambiare l'orientamento).

La definizione dei link A cambia per rendere possibile precisare il nome di una zona o di una finestra come destinazione del documento.



Un esempio di frame



Il codice del frame

```
<HTML>
  <FRAMESET COLS="50%,*">
    <FRAME SRC="f2.html" NAME="sinistra">
    <FRAMESET ROWS="30%,*">
      <FRAME SRC="f3.html" NAME="destra alto">
      <FRAME SRC="f4.html" NAME="destra basso">
    </FRAMESET>
  </FRAMESET>
  <NOFRAMES>
  <BODY>
    <P>Testo per browser che non supportano i frame</P>
  </BODY>
</HTML>
```



I tag dei frame (1)

FRAMESET

Introduce un gruppo di frame. Attributi:

- ✦ ROWS, COLS: la disposizione delle righe o delle colonne del frameset. Il valore è una lista di numeri separati da virgole. Ogni numero identifica la dimensione di un frame. Espresso o in pixel, o in percentuale, o con un asterisco (divisione equa del rimanente spazio)

<FRAMESET ROWS="30,30%,*,*">: quattro righe, una alta 30 pixel, una alta il 30 per cento dello spazio rimanente, la terza e la quarta si dividono lo spazio rimanente.

NOFRAMES

Blocco di dati da visualizzare nel caso non si sappiano visualizzare i frame.



I tag dei frame (2)

FRAME

Introduce un frame. Attributi:

- ✦ SRC: l'URL del documento da visualizzare nel frame.
- ✦ NAME: il nome del frame, da usare nel target dei link <A>
- ✦ FRAMEBORDER, MARGINWIDTH, MARGINHEIGHT: misure di visualizzazione dei margini dei frame
- ✦ NORESIZE, SCROLLING: controllano il ridimensionamento e la possibilità di scrolling del frame.

IFRAME

Inserisce un frame all'interno di un documento HTML normale (non diviso in frame). Attributi:

- ✦ Gli stessi di frame
- ✦ ALIGN e WIDTH: controllano la posizione e la dimensione del frame rispetto alla pagina.



Conclusioni

Oggi abbiamo parlato di

- ◆ Il tag HEAD
- ◆ Oggetti e mappe
- ◆ Form
- ◆ Frame



Riferimenti

Wilde's WWW, capitolo 7

Altri testi:

- D. Raggett, A. Le Hors, I. Jacobs, *HTML 4.01 Specification*, W3C Recommendation 24 December 1999, <http://www.w3.org/TR/html401>
- R. Fielding, *Relative Uniform Resource Locator*, RFC 1808, Jun 1995.

