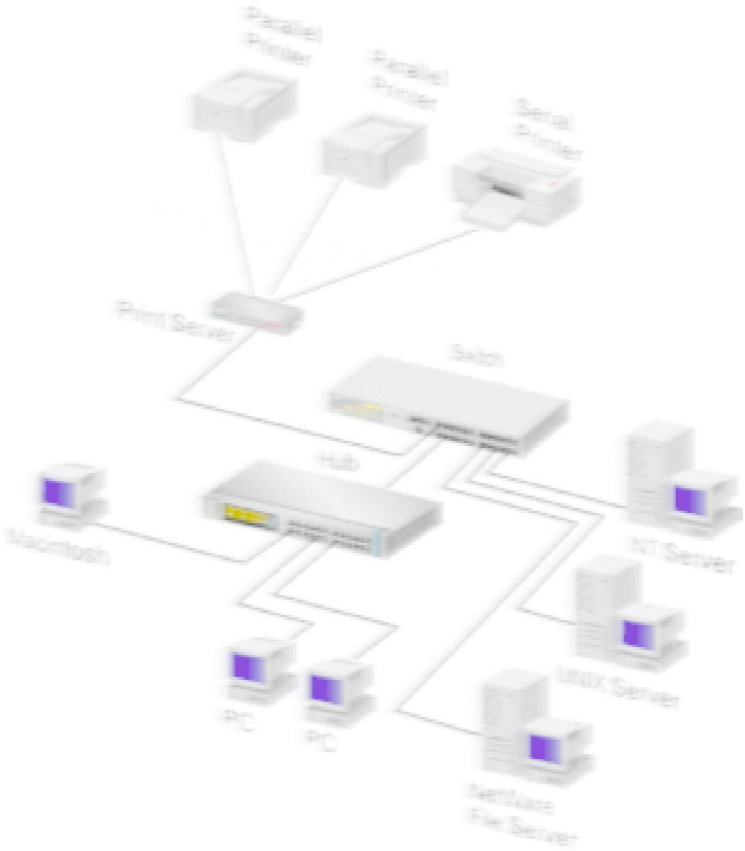

HTML Base

Dispensa per il Corso
Laboratorio Informatico: Reti e Internet 1

Sommario

1. HTML come linguaggio di markup

Questa dispensa riassume gli elementi fondamentali del linguaggio HTML, dal punto di vista dell'editing del codice sorgente.
La trattazione è volutamente sintetica, perché l'obiettivo qui è di far comprendere i principi sintattici del codice, e gli accorgimenti di base necessari alla produzione di codice elegante: gli aspetti più di dettaglio verranno visti con l'aiuto di un editor WYSIWYG (nel nostro caso Front Page).

1.1 Cos'è HTML

HTML (HyperText Markup Language) è un **linguaggio di markup**, ovvero un linguaggio che, tramite l'inserimento di segni esterni al contenuto puro (il testo), specifica degli effetti da riprodurre sul contenuto stesso.
Un documento HTML corrisponde quindi tipicamente a un file dall'estensione .html (o .htm, è indifferente), che contiene codice HTML valido, cioè aderente alle specifiche sintattiche del linguaggio.
Anche se si usa l'espressione "programmazione HTML", in quanto linguaggio di markup HTML non va confuso con i linguaggi di programmazione. Esso è privo infatti di quei tipi di elementi (variabili, operatori e costrutti logici…) che permettono di specificare operazioni automatiche (i "programmi").
Un linguaggio di markup introduce esclusivamente delle informazioni che un programma (es. un browser) andrà poi a interpretare e associare a particolari procedure di elaborazione. Linguaggi di programmazione sono ad esempio VisualBasic, C, Java o Perl; tra i linguaggi di markup invece, oltre a HTML, i più importanti sono SGML e XML.

1.2 La sintassi basilare di HTML

La marcatura del contenuto avviene per mezzo di *tag* che delimitano la porzione di contenuto su cui applicare l'effetto. Ad esempio, se noi vogliamo rendere una parola cliccabile, utilizziamo il seguente costrutto:

Questo è un <A HREF="http://www.pippo.com">link</A>

- <A HREF="http://www.pippo.com">link</A> è l'**elemento**;
- <A HREF="…"> è il **tag di apertura** (in generale, <…> è la sintassi del tag di apertura);
- </A> è il **tag di chiusura** (in generale, </…> è la sintassi del tag di chiusura);
- i caratteri < e > sono i **delimitatori** del tag;
- A è il **nome dell'elemento**: può essere scritto in maiuscolo o minuscolo, per produrre codice elegante è consigliabile però scegliere una soluzione e seguirla con coerenza.

- link è il **contenuto dell'elemento**, ovvero quella porzione di contenuto su cui va applicato l'effetto.

Ciascun elemento può avere o meno degli **attributi**, che definiscono dei parametri di comportamento degli elementi. Tipicamente, si usa la formula nome=valore:
HREF è il **nome** dell'attributo (valgono le indicazioni di scrittura del nome dell'elemento) http://www.pippo.com è il **valore** dell'attributo (andrebbe sempre racchiuso tra virgolette o apici, ma i browser su questo sono tolleranti).
I tag debbono essere **annidati** (*nested*), cioè racchiusi in altri tag.
Se nell'esempio precedente io volessi rendere la parola cliccabile in grassetto, scriverei:

Questo è un <B><A HREF="http://www.pippo.com">link</A></B>

In questo caso, si dice che l'elemento A è annidato nell'elemento B. Esistono dei vincoli nell'annidamento, e inoltre browser diversi spesso interpretano in maniera diversa una medesima serie di tag annidati.
È consigliabile seguire alcuni accorgimenti per rendere più elegante leggibile il codice da parte dell'occhio umano. In particolare, si possono:

- inserire **commenti** in linguaggio naturale nel codice, secondo la sintassi <!-- contenuto del commento
-->;
- **indentare** con spazi e/o tabulazioni i vari elementi annidati;
- **separare** con qualche riga vuota i blocchi principali di codice.

2 Il documento HTML

2.1 Per cominciare

Per creare un documento HTML è sufficiente un comune editor di testo (Windows mette a disposizione il Blocco Note e Wordpad; su Mac invece c'è Simple Text), e un web browser per la visualizzazione del contenuto. Il lavoro procede sempre così, con la fase di editing seguita periodicamente dal rendering da browser per testare il risultato del nostro intervento.
Per lavorare con ordine, conviene creare una cartella di progetto e salvare tutti i file del sito al suo interno.
Per prima cosa, occorre lanciare l'editor di testo e, se l'applicazione non lo fa automaticamente, aprire un nuovo documento vuoto (File>Nuovo).

2.2 Lo scheletro del documento

Qualsiasi documento HTML per essere valido deve racchiudere tutto il proprio contenuto all'interno dell'elemento radice HTML. Quindi nella prima riga del documento scriveremo sempre <HTML>, e nell'ultima </HTML>.

Un documento si divide in due blocchi principali:

Intestazione
(elemento HEAD), che contiene in genere metainformazioni descrittive del documento;

Corpo
(elemento BODY), che contiene tutto ciò che finirà nell'area di visualizzazione del browser.

HEAD precede necessariamente BODY, ed entrambi sono allo stesso livello gerarchico (nessuno dei due può essere annidato nell'altro).
Lo scheletro fondamentale di un documento HTML sarà quindi rappresentato dal seguente codice:

```
<HTML>
<HEAD>
…
</HEAD>
<BODY>
…
</BODY>
</HTML>
```

Per salvare il documento, selezionare File>Salva con nome, posizionarsi nella cartella di progetto precedentemente creata, assegnare un nome al file (es. miodoc) e scrivere esplicitamente l'estensione .html o .htm (su Windows selezionare l'opzione di visualizzazione "Tutti i file", altrimenti verrà creato un file dal nome miodoc.html.txt). Il nome del documento di apertura del sito (la home page o la splash page) deve essere index.html su server Apache e default.html su server Microsoft.
Per visualizzare il documento, lanciare il browser e aprire il documento (File>Apri, posizionarsi tramite il tasto Sfoglia nella cartella di progetto, e selezionare il documento).

2.3 La sezione HEAD

L'elemento TITLE

Tra i vari elementi da inserire nell'intestazione, particolarmente importante è l'elemento TITLE. Esso attribuisce un titolo al documento, e viene tra l'altro visualizzato nella barra del titolo del browser.
Esempio:

```
…
<HEAD>
<TITLE>Home Page di Laerte Sorini</TITLE>
</HEAD>
```

…

2.4 La sezione BODY

Il corpo del documento è la sezione più ricca di elementi, e ovviamente quella più interessante per noi perché contiene in ultima analisi gli elementi destinati al lettore umano. Già l'elemento BODY permette di specificare alcune informazioni di resa grafica, se viene arricchito con ad es. gli attributi

- BGCOLOR (impostare un colore di sfondo)
- BACKGROUND (impostare un'immagine di sfondo)
- TEXT (colore del testo)
- LINK (colore dei link)
- VLINK (colore dei link visitati)
- ALINK (colore dei link attivi, cioè mentre il mouse sta cliccandoci sopra).

In generale, gli elementi di questa sezione si dividono in *block-level* (impongono al browser un'interruzione di linea prima e dopo) e *inline* (non impongono l'interruzione).

L'elemento Heading Level (H1, H2…H6)

Identifica un'intestazione, applicando una grandezza di caratteri maggiore. Serve per inserire i titoli e i sottotitoli di sezione del testo: ha sei livelli di evidenziazione, e non ha attributi. Esempio:

…
```
<BODY>
<H1>Curriculum vitae</H1>
</BODY>
```
…

L'elemento Paragraph (P)

Identifica e racchiude un paragrafo, inserendo una linea vuota. Non ha attributi, e il tag di chiusura può essere omesso. Esempio:

…
```
<BODY>
<P>Questo è il primo paragrafo.
<P>Questo è il secondo paragrafo.
</BODY>
```
…

L'elemento Break (BR)

Inserisce un'interruzione di linea. Si tratta di un elemento vuoto: non racchiude nessun contenuto, la sua sola presenza determina l'effetto. Di conseguenza, non ha un tag di chiusura. Esempio:

...
<BODY>
Questa parte è sulla prima riga,
mentre questa è andata a capo.
</BODY>
...

L'elemento Bold (B)

Applica uno stile grassetto al contenuto.

...
<BODY>
<B>Questa frase appare in grassetto.</B>
</BODY>
...

L'elemento Italic (I)

Applica uno stile corsivo al contenuto.

...
<BODY>
<I>Questa frase appare in corsivo.</I>
</BODY>
...

L'elemento Unordered List (UL)

Inserisce un elenco puntato. Richiede una serie di elementi LI (List Item) annidati al suo interno per indicare le singole voci in elenco; l'attributo opzionale SRC permette di specificare per ogni LI l'URL di un file grafico che funga da punto elenco personalizzato. Esempio:

...
<BODY>
<UL>
<LI SRC="pallino.gif">Questo è il primo elemento della lista.
<LI SRC="pallino.gif">Questo è il secondo elemento della lista.
<LI SRC="pallino.gif">Questo è il terzo elemento della lista.
</UL>
</BODY>
...

L'elemento Ordered List (OL)

Inserisce un elenco numerato. Come UL, richiede uno o più elementi LI. Esempio:

...
<BODY>
<OL>

<LI>Questo è il primo elemento della lista.
<LI>Questo è il secondo elemento della lista.
<LI>Questo è il terzo elemento della lista.
</OL>
</BODY>
…

L'elemento Font (FONT)

Applica alcune regole di stile tipografico al testo: implica l'uso di alcuni attributi, in particolare

- FACE (per specificare la font da utilizzare)
- COLOR (per indicare il colore)
- SIZE (per indicare il corpo del carattere, in proporzione alla valore di default del browser).

Esempio:
…
<BODY>
<FONT FACE="Arial" COLOR="#000000" SIZE="1">Questa parte di frase è scritta in Arial
nero nelle dimensioni di default,</FONT><FONT FACE="Verdana" COLOR="#CCCCCC"
SIZE="2"> mentre questa in Verdana grigio con un corpo leggermente più grande.</FONT>
</BODY>
…
Questo elemento può creare molto disordine nel codice. Per questo e altri motivi, con il rilascio di HTML 4.0 il W3C ha dichiarato questo elemento deprecato, invocando in sostituzione il ricorso ai ben più potenti Cascading Style Sheets. Molti webdesigner non hanno ancora abbandonato del tutto l'abitudine a utilizzare l'elemento FONT, ma è meglio non seguire il loro esempio.

L'elemento Anchor (A)

Rende il suo contenuto un link ipertestuale. Applica di default uno stile sottolineato e un colore blu al testo, oppure un bordo blu se a essere "calda" è un'immagine. Richiede obbligatoriamente l'attributo HREF, che specifica l'URL di destinazione.
Esempio:
…
<BODY>
Questo è un <A HREF="http://www.pippo.com">link</A> a un sito web.

Questo è un <A HREF="pluto.html">link</A> che richiama una risorsa presente sullo stesso server web, nella stessa cartella del file di origine.
Questo invece è un <A HREF="mailto:pippo@topolinia.com">link</A> che attiva la composizione di un'e-mail.
</BODY>
...
Si possono specificare anche link a punti precisi di un documento: per fare questo occorre marcare il contenuto di destinazione con l'elemento A e, in luogo di HREF, l'attributo NAME, al quale è associato un identificatore arbitrario. L'URL specificata nel link di partenza deve iniziare col carattere speciale #.

Primo esempio (il link avviene all'interno dello stesso documento):
...
<BODY>
Cliccando su questo <A HREF="#destinazione">link</A> la pagina scorrerà automaticamente per mostrare il contenuto di destinazione.
...
...
...
Questo è il <A NAME="destinazione">contenuto di destinazione</A>.
</BODY>
...
Secondo esempio (link a un punto preciso di un documento diverso, "pluto.html"):
...
<BODY>
Cliccando su questo <A HREF="pluto.html#destinazione">link</A> mi si aprirà il documento specificato, "scrollato" automaticamente fino al punto di destinazione.
</BODY>
...
...
<BODY>
Questo è il <A NAME="destinazione">contenuto di destinazione</A> nel file pluto.html, che risiede sullo stesso server e nella stessa cartella del documento di partenza.
</BODY>
...
Se si specifica come HREF semplicemente il carattere "#", creiamo un link che referenzia se stesso (sembra una cosa sciocca, ma a volte si rivela utile).

L'elemento Image (IMG)

Inserisce un'immagine nel documento. Si tratta di una risorsa esterna, che non viene inglobata (*embedded*) nel documento, bensì semplicemente collegata (*linked*). È un elemento vuoto, e richiede l'attributo obbligatorio SRC, che specifica l'URL del file grafico da visualizzare; è suggerita inoltre
l'indicazione di un testo alternativo in caso l'immagine non possa essere visualizzata, tramite l'impiego
dell'attributo ALT, . Esempio:

```
…
<BODY>
<IMG SRC="miafoto.jpg" ALT="Foto di Mario">
</BODY>
…
```

I principali formati di file grafici per il web sono due:
- il formato GIF, che supporta un numero variabile di colori (fino a 256), ed è indicato per le immagini con poche aree di colore, meglio se "solido" (es. diagrammi, disegni, scritte…); nei file GIF si possono impostare effetti di trasparenza e sequenze animate;
- il formato JPEG, che supporta sempre milioni di colori, ed è indicato per le immagini con complesse informazioni di colore (fotografie, immagini con molti gradienti di colore…).

L'elemento Table (TABLE)

Inserisce una tabella: pensata per rappresentare dati in forma matriciale, la tabella si è via via trasformata in uno strumento per ottenere con qualche trucco layout di pagina complessi. Richiede uno o più elementi annidati TR per indicare le righe, all'interno dei quali stanno uno o più elementi TD a seconda del numero di celle che intendiamo specificare. Esempio (tabella con due righe e tre colonne, cioè tre celle per ogni riga):

```
…
<BODY>
<TABLE>
<TR>
<TD>Riga 1, cella 1</TD>
<TD>Riga 1, cella 2</TD>
<TD>Riga 1, cella 3</TD>
</TR>
<TR>
<TD>Riga 2, cella 1</TD>
<TD>Riga 2, cella 2</TD>
<TD>Riga 2, cella 3</TD>
```

```
</TR>
</TABLE>
</BODY>
...
```

L'elemento Form (FORM)

Costruisce un modulo, ovvero una serie di campi in cui l'utente può inserire o selezionare dei valori. Richiede due attributi: METHOD, che specifica il metodo HTTP con cui trasmettere i dati (GET vs. POST), e ACTION, che indica l'URL del programma da attivare per l'elaborazione dei dati.
Esempio:

```
...
<BODY>
<FORM METHOD="GET" ACTION="script.cgi">
...
</FORM>
</BODY>
...
```

All'interno di FORM sono annidati gli elementi corrispondenti ai vari tipi di interfaccia di input. I principali sono i campi di testo, le password, le aree di testo, i checkbox e i bottoni radio, i menu a tendina: a ciascuno di essi va associato un nome, che identificherà poi il valore inserito all'interno del programma specificato in ACTION.

```
...
<FORM METHOD="GET" ACTION="script.cgi">
<!-- inserisco un campo di testo -->
<INPUT TYPE="text" NAME="campo1">
<!-- inserisco un campo per l'immissione di una password -->
<INPUT TYPE="password" NAME="campo2">
<!-- inserisco un'area di testo, utile per i testi lunghi come ad es. commenti o messaggi -->
<TEXTAREA NAME="campo3"></TEXTAREA>
<!-- inserisco un menu a tendina con tre opzioni -->
<SELECT NAME="campo4">
<OPTION>Opzione 1
<OPTION>Opzione 2
<OPTION>Opzione 3
</SELECT>
<!-- inserisco tre opzioni con checkbox, non mutuamente esclusivi -->
<INPUT TYPE="checkbox" NAME="campo5" VALUE="valore1">Opzione 1
<INPUT TYPE="checkbox" NAME="campo5" VALUE="valore2">Opzione 2
<INPUT TYPE="checkbox" NAME="campo5" VALUE="valore3">Opzione 3
<!-- inserisco tre opzioni con bottoni radio, mutuamente esclusivi -->
```

```
<INPUT TYPE="radio" NAME="campo6" VALUE="valore1">Opzione 1
<INPUT TYPE="radio" NAME="campo6" VALUE="valore2">Opzione 2
<INPUT TYPE="radio" NAME="campo6" VALUE="valore3">Opzione 3
</FORM>
...
```

Alla fine delle varie interfacce di input, inserisco i bottoni per l'invio e l'azzeramento dei dati immessi.
L'impiego del primo è necessario al funzionamento della transazione, il secondo invece è un elemento facoltativo ma suggerito.

```
...
<FORM METHOD="GET" ACTION="script.cgi">
...
<!-- inserisco un tasto di invio dei dati-->
<INPUT TYPE="submit">
<!-- inserisco un tasto di azzeramento dei dati -->
<INPUT TYPE="reset">
</FORM>
...
```

Appendice 1: pagine con frames

Originariamente introdotti come elemento proprietario di Netscape, i frames per un certo periodo hanno goduto di una certa popolarità tra i webdesigner. Essi permettono infatti di suddividere la schermata in zone, ciascuna delle quali richiama un documento HTML separato: tramite alcuni accorgimenti, è possibile impostare regole di navigazione che permettono di cambiare il documento di alcune zone (tipicamente, l'area di visualizzazione del contenuto principale della schermata), e di tenere invece fisse altre (come l'intestazione e il menu principale di navigazione).
Assumendo che i singoli documenti da referenziare siano già stati creati, la sintassi prevede la creazione di un documento HTML contenitore, in cui l'elemento BODY viene sostituito da FRAMESET:

```
<HTML>
<HEAD>
...
</HEAD>
<FRAMESET>
...
</FRAMESET>
</HTML>
```

Dentro FRAMESET occorre specificare la suddivisione della schermata: ciò avviene tramite l'utilizzo dell'attributo COLS (Columns), se si vuole creare una ripartizione per colonne, oppure ROWS, che impone invece una ripartizione in righe. Entrambi gli attributi specificano il numero e l'ampiezza (in pixel o percentuale) di ciascuna

partizione. Esempio (schermata con due colonne, la prima sempre larga un terzo dell'area di visualizzazione):

```
…
<FRAMESET COLS="33%,*">
…
</FRAMESET>
…
```

Annidati in FRAMESET vanno i singoli elementi FRAME, con gli attributi SRC (l'URL del documento da visualizzare inizialmente all'interno del frame) e NAME (l'identificativo del frame ai fini della navigazione, scelto arbitrariamente). Quindi, ipotizzando che nell'esempio vogliamo mettere un menu nel frame di sinistra e il testo principale che cambia nel frame di destra, potremmo scrivere:

```
…
<FRAMESET COLS="33%,*">
<FRAME SRC="menu.html" NAME="navmenu">
<FRAME SRC="page1.html" NAME="maincontent">
</FRAMESET>
…
```

Per creare concretamente i link che dal frame "navmenu" modificano il contenuto di "maincontent", occorre inserire nei singoli elementi A presenti in "navmenu" l'attributo TARGET, che specifica il nome del frame di destinazione. Quindi, ammettendo che il menu prevede tre voci di navigazione, il codice di menu.html prevederà delle istruzioni di questo tipo:

```
…
<BODY>
<A HREF="page1.html" TARGET="maincontent">Link che visualizza la prima schermata</A><BR>
<A HREF="page2.html" TARGET="maincontent">Link che visualizza la seconda schermata</A><BR>
<A HREF="page3.html" TARGET="maincontent">Link che visualizza la terza schermata</A><BR>
</BODY>
…
```

Concepiti originariamente come un artificio per facilitare la navigazione, i frames si sono di fatto rivelati uno strumento abbastanza complesso da gestire e non esente da problemi di usabilità. Di conseguenza, dopo il periodo d'oro (1996-1999), i frames oggi sono un po' caduti in disuso, e in effetti il loro impiego va scelto con cautela.

Appendice 2: i colori

HTML comunica al browser le informazioni di colore utilizzando il modello RGB: vale a dire, ogni colore è definito in base alle componenti di rosso, verde e blu (ciascuna indicata con due cifre, nel formato #RRGGBB – RR=Red GG=Green BB=Blue). I singoli

valori sono specificati con il codice esadecimale a due cifre, di conseguenza gli estremi saranno “#000000” (totale assenza dei tre colori, quindi nero) e “#FFFFFF” (massima saturazione dei tre colori, quindi bianco). Se volessimo un rosso saturato al massimo, il codice sarebbe “#FF0000”, se invece volessimo un marrone, potremmo inserire “#993300” (un valore di rosso un po’ meno saturato, un po’ di verde e niente blu).
Esiste un numero limitato di colori che i due browser principali sono in grado di visualizzare allo stesso modo: si tratta della “Web-safe color palette”, che comprende 216 colori (quelli con i valori in coppia per multipli di tre, es. “#000000”, “#333333”, “#006699”…). Per questo motivo Dreamweaver offre in genere un numero limitato di colori, ai quali conviene attenersi.

Appendice 3: i caratteri non-ASCII

Come sappiamo, sistemi diversi possono essere configurati con un diverso sistema di decodifica dei caratteri. Di conseguenza, per quanto riguarda i caratteri estranei ai 128 del codice ASCII standard, conviene utilizzare le cosiddette entità di carattere: stringhe predeterminate che il browser è in grado di sostituire automaticamente con determinati caratteri. In particolare, per noi italiani si pone il problema delle lettere accentate:

- à diventa à
- è diventa è
- ì diventa ì
- ò diventa ò
- ù diventa ù
- é diventa é

Quindi, riprendendo il primo esempio, per produrre codice “a prova di sistema operativo” dovremmo scrivere:

Questo è un <A HREF=”http://www.pippo.com”>link</A>
Fortunatamente gli editor WYSIWYG sostituiscono automaticamente i caratteri durante la digitazione (tranne quando si lavora nella finestra di editing del codice sorgente).