

HyperText Markup Language (HTML)

Il linguaggio HTML descrive *documenti strutturati*.

Un documento scritto in linguaggio HTML è un puro testo ASCII formato da:

1. il testo vero e proprio del documento;
2. i *tag* HTML che indicano gli elementi, la struttura, la formattazione e i link ipertestuali.

Tipicamente i tag sono di due tipi:

```
<tagname>Text</tagname>  
oppure  
<tagname> ...
```

Esempio

```
<HTML>  
  <HEAD>  
    <TITLE>Linguaggi e Traduttori</TITLE>  
  </HEAD>  
  <BODY>  
    <H1>  
      Benvenuti a Linguaggi e Traduttori  
    </H1>  
  </BODY>  
</HTML>
```

Questo esempio sarà salvato in un file ASCII usando una delle seguenti estensioni:

```
.htm  
.html
```

Ad esempio, `welcome.htm`

A questo punto il file è pronto per essere interpretato da un web browser, ad esempio Netscape.

La formattazione del testo HTML serve solo per facilitare la lettura del file, ma il browser-interprete la ignorerà. Con l'eccezione del tag <PRE> il browser procede alla formattazione automatica del testo.

Struttura di un documento HTML

Un documento HTML deve rispettare una precisa struttura. La struttura è definita mediante tre tag:

1. <HTML>
2. <HEAD>
3. <BODY>

<HTML>

Indica che il file in oggetto è un file HTML. La sintassi generale è:

```
<HTML>
...document_text
</HTML>
```

<HEAD>

È il delimitatore del prologo del file. All'interno di questo tag possono comparire solo alcuni tag, tra cui quello che indica il titolo.

Nel prologo non deve essere inserita nessuna altra porzione di testo.

Es.

```
<HTML>
<HEAD>
  <TITLE>Linguaggi e Traduttori</TITLE>
</HEAD>
...
</HTML>
```

<BODY>

Il corpo vero e proprio del documento è racchiuso tra i tag <BODY>. Questo tag deve trovarsi in una ben precisa posizione:

```
<HTML>
<HEAD>
  <TITLE>Linguaggi e Traduttori</TITLE>
</HEAD>
<BODY>
...

```

```
</BODY>
</HTML>
```

Titolo

Ogni documento HTML ha un titolo. Per assegnare il titolo a un documento si usa il tag <TITLE>.

Questo tag deve stare obbligatoriamente nella regione delimitata dal tag <HEAD>.

In un documento deve essere presente un solo titolo e il titolo può essere composto solo da un testo semplice, ovvero non possono essere specificati altri tag all'interno del titolo.

La sintassi generale è:

```
<TITLE>title_name</TITLE>
```

Scegliere sempre nomi significativi per il titolo di un documento HTML.

Intestazioni

Le intestazioni si usano in un documento per suddividerne le parti. Anche HTML prevede la possibilità di usare intestazioni. La sintassi generale delle intestazioni è:

```
<H#>header_name </H#>
```

dove

:= 1..6

Ad esempio

```
<H1>Manutenzione motore</H1>
  <H2>Cambio olio</H2>
  <H2>Regolazione valvole</H2>
  <H2>Sostituzione candele</H2>
    <H3>Estrazione candele</H3>
    <H3>Preparazione nuove candele</H3>
      <H4>Eliminazione protezione</H4>
      <H4>
        Regolazione distanza elettrodi
      </H4>
      <H4>Applicazione lubrificante</H4>
      <H4>Installazione candele</H4>
    <H2>Regolazione anticipo</H2>
```

Le intestazioni, a differenza dei titoli, possono essere formate da più righe di testo.

L'intestazione <H1> è tipicamente usata per ribadire il titolo all'interno del documento.

In genere è consigliabile disporre gerarchicamente le intestazioni, senza saltare livelli.

Paragrafi

Per separare i paragrafi di un testo si usa il tag <P>. Con il tag paragrafo si salta un rigo e si va a capo.

Questo tag può essere usato come tag singolo oppure come contenitore:

```
Paragrafo con tag singolo <P>
secondo le vecchie specifiche<P>
HTML Level One<P>
```

```
<P>Paragrafo con tag singolo </P>
<P>secondo le nuove specifiche</P>
<P>HTML Level Three</P>
```

Si raccomanda la seconda notazione.

Commenti

È possibile introdurre commenti in un file HTML così come avviene con tutti i normali linguaggi di programmazione. La sintassi generale è:

```
<!-- This is a comment -->
```

I commenti sono spesso inseriti prima della sezione <HEAD>:

```
<HTML>
<!------->
<!--      author:  abel                      --->
<!--      e-mail:  abel@mathema.it           --->
<!--      created: 12/03/1996                 --->
<!--      modified: 13/03/1996                --->
<!--      file:    bio.html                   --->
<!--      purpose: abel's biography           --->
```

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

Anche per i commenti si raccomandano serietà e rigore.

Link

In HTML è possibile creare documenti e collegamenti ipertestuali tra documenti. Questi collegamenti sono in gergo detti *link*.

Per creare un link occorre specificare:

1. l'indirizzo della risorsa verso cui collegarsi;
2. il testo da selezionare per collegarsi alla risorsa.

Nel documento visualizzato dal browser si vede solo il secondo elemento generalmente sottolineato.

Per creare un link si usa il tag <A>.

Esempio:

```
Ritorna al <A HREF="../menu.htm">Menu</A>
```

Effettua un collegamento dal file attuale al file `menu.htm`.

Esempio

Consideriamo i due file `menu.htm` e `alim.htm`:

menu.htm

```
<HTML>
<HEAD>
<TITLE>Come Allevare Struzzi</TITLE>
</HEAD>
<BODY>
<H1>Come Allevare gli Struzzi</H1>
<P>Cercate di voler bene agli struzzi. Sono
bestie buone e delicate.</P>
<!-- Segue un Menu rozzo -->
<P><A HREF="alim.htm">
    Alimentazione degli struzzi
</A></P>
```

```
<P>Addestramento degli struzzi</P>
<P>Pulizia degli struzzi</P>
</BODY>
</HTML>
```

alim.htm

```
<HTML>
<HEAD>
<TITLE>Struzzi: Alimentazione</TITLE>
</HEAD>
<BODY>
<H1>Alimentazione degli Struzzi</H1>
<P>Questa sezione descrive come alimentare gli
struzzi</P>
<H2>Cosa dare da mangiare agli struzzi</H2>
<P>Gli struzzi mangiano una ciotola di mangime
Struzzin due volte al giorno.</P>
<!-- Ritorno al Menu->
<P><A HREF="menu.htm">
    Ritorno al menu principale
</A></P>
</BODY>
</HTML>
```

Link a documenti locali

L'esempio degli struzzi era un classico esempio di collegamento tra file locali, cioè tra due documenti residenti sul medesimo sistema. Inoltre i due file erano anche residenti nella medesima directory.

Per effettuare il link tra documenti locali si possono usare due metodi di indirizzamento:

1. indirizzamento relativo;
2. indirizzamento assoluto.

Indirizzamento relativo - Si specifica il percorso che si deve effettuare per raggiungere il file di collegamento a partire dalla directory attuale, cioè quella dove si trova il documento da cui inizia il link.

Ad esempio:

```
HREF="pippo.htm"
```

il file `pippo.htm` è nella directory corrente.

`HREF="mydir/pippo.htm"`

il file `pippo.htm` è nella directory `mydir`, sottodirectory della directory corrente.

`HREF="../pippo.htm"`

il file `pippo.htm` si trova nella directory che contiene la directory corrente, ovvero è nella directory di livello gerarchico superiore.

Per muoversi lungo i percorsi relativi usare sempre i simboli:

`/`

`..`

anche se non si lavora sotto Unix.

Nel caso si lavori sotto DOS/Windows si ricordi che il nome delle unità a disco tipicamente denotato da lettera e due punti

`C:`

`D:`

È invece denotato da lettera e barra

`C|`

`D|`

Ad esempio

`HREF="../../../d|/pippo.plu/bob/rogn.htm`

Indirizzamento assoluto - Si specifica il percorso che si deve effettuare per raggiungere il file di collegamento a partire dal livello superiore della gerarchia di directory. I nomi di percorsi con indirizzamento assoluto iniziano sempre con la barra (/).

Ad esempio:

`HREF="/usr/abel/bio.htm"`

il file `bio.htm` si trova nella directory `/usr/abel`

`HREF="/~abel/bio.htm"`

il file `bio.htm` si trova nella home directory dell'utente `abel` (Unix).

`HREF="/d|/clienti/html/terzani.htm"`

il file `terzani.htm` si trova nella directory `clienti/html` del disco `d:` (DOS).

Suggerimenti - Evitare gli indirizzamenti assoluti. Gli indirizzamenti relativi sono più flessibili, favoriscono la manutenzione e il trasferimento di documenti HTML dall'ambiente di sviluppo locale al web server.

Link a documenti remoti

È possibile collegare un documento HTML ad un'altra risorsa (eventualmente un altro documento) disponibile su Internet, sempre usando il tag `<A>`.

La sintassi che viene tipicamente usata è:

```
<A HREF="http://www.mathema.it/"> Mathema </A>
```

In questo modo si effettua un collegamento alla home page del web server *www.mathema.it*.

Esempio

menu.htm

```
<HTML>
<HEAD>
<TITLE>Come Allevare Struzzi</TITLE>
</HEAD>
<BODY>
<H1>Come Allevare gli Struzzi</H1>
<P>Cercate di voler bene agli struzzi. Sono
bestie buone e delicate.</P>
<!-- Segue un Menu rozzo -->
<P><A HREF="alim.htm">
    Alimentazione degli struzzi
</A></P>
<P>Addestramento degli struzzi</P>
<P>Pulizia degli struzzi</P>
<P>Ulteriori informazioni sugli struzzi si
trovano presso lo
<A
                                HREF="http://www.zoo.palo-
alt.ca.us/ostriches/home.html">Zoo Palo Alto</A>
</P>
</BODY>
</HTML>
```

Liste

In HTML si possono specificare diversi tipi di liste:

- liste numerate o ordinate;
- liste non ordinate con contrassegno
- liste di definizione;
- liste a menu;
- liste a directory.

Gli ultimi due tipi di liste furono definite nel HTML Level One e sono oggi considerate obsolete e non saranno descritte.

I tag per la creazione di liste hanno alcuni elementi in comune:

- ogni lista è delimitata dai tag di apertura e chiusura specifici di quel tipo di lista (ad es. `<OL>` e `</OL>`, `<MENU>` e `</MENU>`);
- ogni voce all'interno della lista è contraddistinta da un proprio tag: `<DT>` e `<DD>` per le liste di definizione e `<LI>` per tutte le altre liste.

Esempio-

```
<P>La Divina Commedia</P>
<UL>
  <LI>Inferno
  <LI>Purgatorio
  <LI>Paradiso
</UL>
```

Liste numerate

Le liste numerate dette anche ordinate, *Ordered List* (`OL`), sono formate da voci di lista, *List Item* (`LI`), che vengono numerate e indentate:

Per imparare il linguaggio HTML occorrono:

1. un editor
2. un browser
3. tanta pazienza

In HTML avremmo:

```
<P>Per imparare il linguaggio HTML occorrono:</P>
<OL>
<LI>un editor
<LI>un browser
<LI>tanta pazienza
</OL>
```

In generale la sintassi è:

```
<OL>
<LI>item_1
<LI>item_2
...
<LI>item_N
</OL>
```

Liste non numerate

Le liste non numerate dette anche non ordinate, *Unordered List* (UL), sono formate da voci di lista, *List Item* (LI), che vengono contrassegnate da un *bullet* (•) e indentate:

Per imparare il linguaggio HTML occorrono:

- un editor
- un browser
- tanta pazienza

In HTML avremmo:

```
<P>Per imparare il linguaggio HTML occorrono:</P>
<UL>
<LI>un editor
<LI>un browser
<LI>tanta pazienza
</UL>
```

In generale la sintassi è:

```
<UL>
<LI>item_1
<LI>item_2
...
```

```
<LI>item_N  
</UL>
```

Liste di definizione

Le liste di definizione , *Definition List* (DL), sono formate da voci di lista composte da due elementi:

1. il termine, *Definition Term* (DT);
2. la definizione vera e propria (DD).

Ad esempio:

cliente

utente interno e/o esterno che acquista il programma

fornitore

sviluppatore interno e/o esterno che realizza il programma

In HTML avremmo:

```
<DL>  
<DT>cliente<DD>utente interno e/o esterno che  
acquista il programma  
<DT>fornitore<DD>sviluppatore interno e/o esterno  
che realizza il programma  
</DL>
```

In generale la sintassi è:

```
<DL>  
<DT>term_1<DD>definition_1  
<DT>term_2<DD>definition_2  
...  
<DT>term_N<DD>definition_N  
</DL>
```

Liste nidificate

In HTML è possibile annidare una lista dentro l'altra. In pratica per far ciò basta specificare una lista le cui voci sono a loro volta liste:

```
<OL>  
<LI>Internet
```

```

<LI>Web Server
<LI> Il linguaggio HTML
    <UL>
    <LI>Introduzione
    <LI>Struttura Documenti
    <LI>Intestazioni
    <LI>Paragrafi
    </UL>
<LI> Il linguaggio Java
</OL>

```

Per avere l'esatta percezione di che cosa accade annidando due o più liste occorre provare con il proprio browser.

Attributi di lista

Il browser Netscape consente di definire utili attributi di lista.

In particolare si ha:

Liste non numerate

```
<UL TYPE=value>
```

dove

value:= DISC | CIRCLE | SQUARE

DISC	è il pallino pieno
CIRCLE	è il pallino vuoto
SQUARE	è il quadrato vuoto

Liste numerate

```
<OL TYPE=number>
```

dove

number:= A | a | I | i | 1

A	numerazione con lettere maiuscole
a	numerazione con lettere minuscole
I	numerazione romana maiuscola

i numerazione romana minuscola
1 numeri

Voci di lista

`<LI TYPE=itemtype VALUE=itemvalue>`

dove

itemtype := DISC | CIRCLE | SQUARE | A | a | I | i | 1

l'impostazione del tipo di voce vale anche per tutte le voci che seguono.

valuetype := numero naturale

vale solo per le liste `<OL>`. Il numero naturale assegna un valore alla corrispondente voce e reinizializza il contatore per le voci a seguire.

Testo centrato

Un tag molto utile è quello per la centratura del testo. La sintassi generale di questo tag è:

`<CENTER>Text to be centered</CENTER>`

Il testo che viene centrato da questa sezione può contenere oltre al normale testo anche paragrafi e intestazioni. È molto utile per la definizione dei titoli di pagina.

Esempio

```
<HTML>
<HEAD>
<TITLE>Linguaggi e Traduttori</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Benvenuti a Linguaggi e Traduttori</H1>

<P>Divertimento assicurato per tutti</P>
<P>Almeno si spera</P>
</CENTER>
</BODY>
</HTML>
```

Attributo ALIGN

Il tag <CENTER> è purtroppo fuori standard perchè adottato solo da Netscape.

Per specificare un testo centrato si può usare anche l'attributo ALIGN=CENTER.

Tra i tag che supportano questo attributo ci sono:

```
<P ALIGN=CENTER>Mi hanno messo in mezzo</P>
```

e tutti quelli dell'intestazione:

```
<H2 ALIGN=CENTER>The HTML Programmer</H2>
```

L'attributo ALIGN=CENTER è standard HTML 3.0.

Inoltre per l'attributo ALIGN dell tag <P> sono ammessi anche i valori LEFT e RIGHT.

Caratteri con stile

Le sezioni che vengono delimitate dai tag che definiscono le liste, i paragrafi e le intestazioni prendono il controllo sul testo che contengono decidendone spaziatura, font e contrassegno.

È possibile riprendere il controllo dei caratteri all'interno di queste sezioni per mezzo di un insieme di tag di stile. Tipici esempi sono il grassetto e la sottolineatura.

L'aspetto di un insieme di caratteri nel testo può essere variato con riferimento a:

- *lo stile logico;*
- *lo stile fisico.*

Lo *stile logico* si riferisce alla funzione del testo, al suo significato. Ad esempio occorre uno stile per una citazione, per rappresentare il codice sorgente di un programma o semplicemente per evidenziare una frase.

Lo *stile fisico* si riferisce alla formattazione dei caratteri del testo, come ad esempio il grassetto e il corsivo.

Stile logico - Il tag indica per quale scopo deve essere utilizzato il testo e non come deve essere esattamente visualizzato.

In HTML esistono otto diversi stili logici:

`<EM>..</EM>`

Il testo delimitato da questi tag deve essere in qualche modo evidenziato rispetto al testo circostante. Di solito si riduce ad un normale corsivo.

Es.

`<P>Mi raccomando:<EM> Studiare la sintassi</EM></P>`

`<STRONG>..</STRONG>`

Come `<EM>` ma ancora più forte. Di solito si riduce a un grassetto.

Es.

`<P>Prima di parlare <STRONG>attiva il cervello!</STRONG></P>`

`<CODE>..</CODE>`

Si usa per rappresentare il codice sorgente di un listato. Di solito usa il font Courier.

Es.

`<P><CODE>printf("Hello World");</CODE></P>`

`<SAMP>..</SAMP>`

Si usa per rappresentare il testo negli esempi.

Es.

`<P>La nostra URL:<SAMP>http://www.mathema.it/</SAMP></P>`

`<KBD>..</KBD>`

Si usa nei manuali utente per contraddistinguere un testo che deve essere immesso dall'utente del programma.

Es.

`<P>Digitare il comando:<KBD>a:\setup</KBD></P>`

`<VAR>..</VAR>`

Si usa nei libri sui linguaggi di programmazione per contraddistinguere le variabili. Di solito si riduce ad un normale corsivo.

Es.

`<P>Assegnando il valore 0 alla variabile <VAR>pippo</VAR> si esce dal ciclo principale</P>`

`<DFN>..</DFN>`

Evidenzia una parola o una frase in una definizione.

Es.

`<P>Una complicazione inutile è detta <DFN>UCCS</DFN></P>`

`<CITE>..</CITE>`

Si usa per le citazioni.

Es.

`<P><CITE>Abbiamo preso mezza fava con due piccioni (Abel 1994)</CITE></P>`

Stile fisico - Indica come deve essere visualizzato il testo denotato dal tag.

In HTML esistono quattro tag di stile fisico:

`<B>..</B>`, per il grassetto,

`<I>..</I>`, per il corsivo,

`<TT>..</TT>`, per il carattere tele-type (macchina da scrivere

`<U>..</U>`, per il sottolineato (non sempre funziona).

Esempio

`<P>Ordine tassativo: <B>Passare tutti</B></P>`

`<P>Come si permette. <I>Sovversivo!</I></P>`

`<P>Premere <TT>start</TT> per avviare la macchina</P>`

`<P>Lo sottolineo:<U>Sei un UCCS</U>`

Dimensione dei font

Due utilissimi tag fuori standard introdotti da Netscape e non supportati da tutti i browser sono `<BASEFONT>` e `<FONT>`.

Il primo `<BASEFONT>` cambia le dimensioni standard del font. La sintassi generale è:

`<BASEFONT SIZE=fontvalue>...</BASEFONT>`

dove

fontvalue:=1..7

L'attributo `SIZE` è obbligatorio. Il valore del font è compreso tra 1 e 7. Il valore di default standard è 3. Tutte le eventuali successive variazioni relative di font partono dalla base definita da `<BASEFONT>`.

Il secondo definisce la dimensione del font del testo, sia in modo assoluto che relativo rispetto al font base del testo circostante. La sintassi generale è:

```
<FONT SIZE=fontvalue> . . . </FONT>
```

dove

fontvalue := *absolute_value* | *relative_value*

absolute_value := 1..7

relative_value := ± *absolute_value*

L'attributo SIZE è obbligatorio. Il valore assoluto del font è compreso tra 1 e 7. Il valore assoluto di default standard è 3. Si possono effettuare variazioni relative al font di base definito da <BASEFONT> usando un + o un -. Ad esempio:

```
<FONT SIZE=+2>very big<FONT>
```

Altri stili per i font

Altri utili tag impiegati come elementi di stile per i font sono:

<BIG> (**big print**) Il testo nella sezione <BIG> dovrebbe essere visualizzato, se conveniente, usando un font "grosso", rispetto al testo cosiddetto normale;

<SMALL> (**small print**) Il testo nella sezione <SMALL> dovrebbe essere visualizzato, se conveniente, usando un font "piccino", rispetto al testo cosiddetto normale;

<SUB> (**subscript**) Il testo nella sezione <SUB> dovrebbe essere visualizzato come un pedice usando un font "piccino", rispetto al testo cosiddetto normale. L'attributo ALIGN per il tag <SUB> ha significato solo per i tag MATH (HTML 3.0).

<SUP> (**superscript**) Il testo nella sezione <SUP> dovrebbe essere visualizzato come un esponente usando un font "piccino", rispetto al testo cosiddetto normale. L'attributo ALIGN per il tag <SUP> ha significato solo per i tag MATH (HTML 3.0).

Testo preformattato

Esiste in HTML la possibilità di riprodurre un testo così come è formattato nel file ASCII del documento. È questa una tipica funzione di molti linguaggi di markup quali il Tex e il Latex. Sfortunatamente il testo cosiddetto preformattato è riprodotto in un font mono-spaziato come il Courier. Il tag per il testo preformattato è `<PRE> . . . </PRE>`.

Esempio (la mucchina):

```
<PRE>
<CENTER>
<H1>W la mucchina<SUB>mu</SUB></H1>
</CENTER>
      (    )
Muu    (00)
        V-----\
        ||         |\
        ||--W-||  *
        ||         ||
</PRE>
```

I tag HTML all'interno di `<PRE>` saranno riconosciuti dal browser ed eseguiti.

Il tag `<XMP>` invece agisce come `<PRE>` con l'unica differenza che i tag HTML nella sua sezione non sono eseguiti ma riprodotti alla lettera.

Righe Orizzontali

Le righe orizzontali servono per interrompere un testo, per separarlo in modo più razionale e piacevole.

Il tag tradizionale che si usa per le righe orizzontali, *Horizontal Row* (HR), è `<HR>`.

Esempio

```
<HR>
<H2>Elenco rogne</H2>
<UL>
<LI>Compilare il 740
<LI>Installare Windows NT
<LI>Programmare in Pascal
</UL>
<HR>
```

Netscape propone una versione più completa per la riga orizzontale:

```
<HR SIZE=sizevalue WIDTH=widthvalue ALIGN=align  
NOSHADE>
```

dove

sizevalue è lo spessore in pixel della linea

widthvalue è la larghezza della linea che può essere specificata in pixel o in percentuale (25%, 50% etc.)

align è l'allineamento della linea (nel caso di linea più piccola dello spazio disponibile). Valori ammessi sono LEFT, RIGHT e CENTER.

L'attributo NOSHADE mostra poi una linea senza ombreggiatura.

Righe spezzate

Il tag
 (*Break Row*) consente di interrompere una riga di testo e di ricominciare da capo. Dove da capo può significare anche all'inizio dell'indentazione se si è spezzata la riga di un elemento indentato.

Esempio

```
<P>Mi hai spezzato<BR>  
in due righe</P>
```

Netscape ama spezzare le righe in modo ancora più articolato:

```
<BR CLEAR=clear_value>
```

dove

clear_value:=LEFT | RIGHT | ALL

L'attributo

è usato quando si vuole introdurre una riga vuota sotto una porzione composta da testo e immagini allineate. Il significato dei valori di questo attributo è:

LEFT ricomincia il testo dove trova il primo margine sinistro libero;

RIGHT ricomincia il testo dove trova il primo margine destro libero;

ALL ricomincia il testo sulla prima linea con entrambi i margini liberi;

Netscape prevede altri tag per il fine riga. Questi sono:

<NOBR>...</NOBR> Il testo inserito in questa sezione non viene spezzato dal browser

<WBR> Suggerisce al browser il punto in cui spezzare una parola o una frase (*Word Break Row*). Netscape però si riserva di fare come meglio gli piace. Può essere usato anche dentro la sezione <NOBR>.

Indirizzi

Il tag <ADDRESS>...</ADDRESS> serve per "firmare" la pagina.
Un tipico esempio:

```
<HR>
<ADDRESS>
Alessandro Bellini abel@mathema.it <BR>
This is a research developed by Abel<BR>
Copyright &#169; Abel, 1996<BR>
</ADDRESS>
```

Citazioni

Per inserire una citazione nel testo HTML offre il tag <BLOCKQUOTE>...</BLOCKQUOTE>. In genere la citazione sarà centrata o comunque evidenziata a piacere del browser.
Esempio (un bell'esempio):

```
<BLOCKQUOTE>
"Siamo realisti, esigiamo l'impossibile"-CHE
GUEVARA
</BLOCKQUOTE>
```

Caratteri speciali

Un testo HTML è scritto usando i caratteri disponibili su una tastiera americana. Le lettere accentate come "è", "ì", "ù" etc. della lingua italiana debbono essere specificati per mezzo di un codice.

I caratteri che debbono essere necessariamente specificati con un codice sono detti *caratteri speciali*.

Esistono due metodi di specifica dei caratteri speciali:

1. specifica per nome;
2. specifica per numero.

Il formato delle specifiche per nome è:

&name;

dove

name

indica il nome del carattere speciale.

Ad esempio:

é

corrisponde al carattere "è"

è

corrisponde al carattere "é"

Il formato delle specifiche per numero è:

&#num;

dove *num* indica il numero corrispondente al carattere speciale.

Ad esempio:

è

corrisponde al carattere "è"

é

corrisponde al carattere "é"

Caratteri speciali perchè riservati alla sintassi HTML anche se disponibili sulla tastiera americana sono:

<	<
>	>
&	&
"	"

Ancore

In HTML è possibile stabilire non solo collegamenti (link) tra documenti diversi, ma anche collegamenti all'interno del medesimo documento. I collegamenti all'interno di un medesimo collegamento sono detti *ancore*. Per creare un ancora occorre contraddistinguere con una etichetta il testo destinazione dell'ancora.

Ad esempio la destinazione di un ancora potrebbe essere così specificata:

```
<A NAME="Sezione8">Sezione ottava: Calcio</A>
```

E il riferimento a questo testo etichettato potrebbe essere:

```
<A HREF="#Sezione8">Il Calcio</A>
```

Il simbolo "#" precede il nome destinazione dell'ancora.

Ovviamente è possibile riferirsi con un ancora da un documento a un altro.

Ad esempio:

```
<A HREF="../abel/sports/calcio.htm#Sezione8">Il Calcio</A>
```

Un testo può essere anche contemporaneamente destinazione di un'ancora e punto di partenza di un link, ovvero di due attributi HREF e NAME possono coesistere nel medesimo tag di apertura <A> .

URL

URL è l'acronimo di *Uniform Resource Locator*. Gli URL sono gli indirizzi per raggiungere una risorsa in Internet, dove per risorsa si intende un web server, un indirizzo di posta elettronica o il riferimento a un file transfer server.

Gli URL debbono rispettare un certo formato.

Il tipico formato di un URL è composto da parti non tutte necessarie:

`"method://host:port/file#anchor"`

dove

method è il protocollo:

http
ftp
gopher
...

host è il nome del sistema connesso in Internet:

http://www.mathema.it
http://www.ibm.com

port è il numero di porta a cui occorre connettersi dato quel metodo e quell'host:

http://www.mathema.it:81/

Nei sistemi multitasking è possibile indirizzare non solo la macchina ma anche particolari processi identificati da un ben determinato numero di porta.

file indica la directory e il nome del file corrispondente alla risorsa

http://www.mathema.it/customer/cna.html

anchor è l'ancora all'interno del documento contraddistinto da file.

HTTP (Hyper-Text Transfer Protocol)

È il tipo più diffuso di URL. Lo abbiamo già specificato e usato in molti esempi:

http://www.pin.it/

Se la componente file termina con il simbolo "/" ciò significa che essa denota una directory. In questo caso il protocollo http va a cercare in quella directory un file standard da trasferire. Il tipico nome di file standard è:

`index.html`

Un altro tipico nome è `home.html`. Il nome può coambiare a seconda del server.

Dunque

`http://www.pin.it/`

può corrispondere a:

`http://www.pin.it/index.html`

Evitare URL del tipo

`http://www.pin.it/varie`

dove `varie` è il nome di una directory.

Il web server potrebbe non andare a cercare il file standard `index.html`.

Codici di escape di un URL

Gli unici caratteri ammessi in un URL sono:

- lettere maiuscole e minuscole
- numeri da 0 a 9
- il dollaro, \$
- underline, _
- hyphen, -
- punto, .
- il segno più, +

Gli altri caratteri devono essere contrassegnati da codici speciali detti codici di escape.

I codici di escape non c'entrano nulla con i caratteri speciali dell'HTML.

I più comuni codici di escape sono:

Carattere	Codice di escape
spazio	<code>%20</code>
due punti, :	<code>%3A</code>
punto interrogativo, ?	<code>%3F</code>
barra, /	<code>%2F</code>

I codici di escape hanno il seguente formato:

`%hh`

dove

`hh` sono due caratteri esadecimali ASCII.

Anonymous FTP

Gli URL FTP puntano a un file o a una directory che si trova su un server FTP, tipicamente anonimo. Server FTP anonimo significa che è possibile collegarsi a quel server utilizzando la login generica `anonymous` e immettere come password il proprio indirizzo di posta elettronica.

Ad esempio:

```
ftp://ftp.what.ever/dir/subdir/filename.ext
```

dove

`ftp.what.ever` è l'indirizzo del server ftp

`dir/subdir` è il path del file

`filename.ext` è il file da ritrovare.

Se il file da ritrovare è un file html questo viene visualizzato come per il server http.

Se invece è un altro tipo di file questo viene trasferito e caricato sul disco locale di chi ha puntato dal suo browser a questo URL. In questo ultimo caso la tipica estensione del file è `.zip` o `.z`.

Se invece di puntare ad un file questo tipo di URL punta ad una directory intera:

```
ftp://ftp.what.ever/dir/subdir
```

allora il browser visualizzerà la directory con la lista dei file e direttori che essa contiene.

FTP non anonimo

È possibile puntare a file e directory di server FTP non anonimi, cioè dove l'utente deve essere stato accreditato con un nome (login) e una relativa password.

Il formato in questo caso è:

```
ftp://username:password@ftp.what.ever/dir/subdir/  
filename.ext
```

Ovviamente questo tipo di URL può far scoprire la vostra login e password.

Mailto

È possibile creare un link che attiva un modulo per la spedizione di un messaggio in posta elettronica. Il formato di questo tipo di URL è:

```
mailto:abel@mathema.it
```

dove

mailto è il nome del metodo o protocollo
abel@mathema.it è l'indirizzo di posta elettronica verso cui
spedire la missiva.

Un classico esempio:

```
<ADDRESS>  
Alessandro Bellini  
<A  
HREF="mailto:abel@mathema.it">abel@mathema.it</A>  
</ADDRESS>
```

Gopher

Per collegarsi a un sito gopher si usa il seguente tipo di URL:

`gopher://the.gopher.address`

dove

`gopher` è il nome del protocollo
`the.gopher.address` è l'indirizzo del server gopher.

Ad esempio il menu principale del server gopher della biblioteca di Berkeley è:

```
<A      HREF="gopher://library.berkeley.edu">UC  
Berkeley Library Gopher Server</A>
```

Usenet

Per riferirsi ad un newsgroup Usenet si usa il seguente tipo di URL:

`news:name.of.group`

dove

`news` è il nome del protocollo
`name.of.group` è il nome del newsgroup

Ad esempio per collegarsi al newsgroup `comp.infosystem.www` si può specificare:

```
<A      HREF="news:comp.infosystem.www">WWW      News  
Group</A>
```

Selezionando il trigger il browser visualizzerà tutti gli articoli più recenti.

Per visualizzare un particolare messaggio conservato in un newsgroup invece si usa il seguente formato:

`news:news.message.id`

dove

`news.message.id` è l'identificativo del messaggio nel newsgroup.

Ad esempio:

```
<A HREF="news:40tvmg$sqj@wn.aksi.net"> GOES </A>
```

File

Esiste anche un URL per indicare un file locale al sistema dove si sta lavorando. Il formato è:

```
file:///c|host/dir/sudbir/filename.ext
```

dove

file	è il nome del protocollo
host	è il nome dell'host (opzionale)
dir/subdir	è il nome del path del file
filename.ext	è il nome del file locale

Questo tipo di URL è usato solo per caricare una *home page* locale quando si avvia il browser.

Immagini

In HTML è possibile comporre testi con testo e immagini. Le immagini sono file conformi a un determinato formato.

Il tipico formato riconosciuto da tutti i browser è il GIF. Dipendentemente dal browser esistono altri formati riconosciuti quali: JPEG, PCX e XBM.

Si raccomanda comunque il formato GIF.

Per inserire una immagine *inline* insieme al testo si usa il tag `<IMG>`. Il tag `<IMG>` non ha un tag di chiusura.

Attributi di `<IMG>` sono:

```
<IMG  
SRC="..."  
ALIGN=...  
ALT="..."  
VSPACE=... HSPACE=...  
HEIGHT=... WIDTH=...  
BORDER=...  
LOWSRC="...">
```

Attributo SRC - Specifica il nome del file immagine da caricare nel documento:

```
<IMG SRC="cat.gif">  
<IMG SRC="../../animals/cat.gif">
```

La specifica del path del file immagine segue le medesime regole evidenziate per i link.

Il testo che segue la specifica di inclusione di una immagine viene inserito sulla medesima riga dell'immagine oppure sotto l'immagine a seconda delle dimensioni del file di immagine:

```
<H2><IMG SRC="cat.gif">My wonderful pussy  
cat</H2>
```

```
<BLOCKQUOTE>  
The world <IMG SRC="world.gif">is here<BR>  
then, the sun is included<IMG SRC="sun.gif"><BR>  
some other text<BR>  
and a beautiful flower <IMG SRC="flower.gif"><BR>  
</BLOCKQUOTE>
```

Attributo ALIGN- Stabilisce l'allineamento del testo e dell'immagine. Questo attributo può assumere i seguenti valori:

ALIGN=
LEFT
RIGHT
TEXTOP
TOP
ABSMIDDLE
MIDDLE
BASELINE
BOTTOM
ABSBOTTOM

dove

- **LEFT**, allinea l'immagine a sinistra e inserisce il testo seguente a destra
- **RIGHT**, allinea l'immagine a destra e inserisce il testo a sinistra
- **TEXTOP**, allinea il margine superiore dell'immagine con margine superiore del testo più alto sulla stessa linea

- TOP, allinea il margine superiore dell'immagine con l'elemento(testo o immagine) più in alto sulla stessa linea;
- ABSMIDDLE, allinea la linea centrale dell'immagine con la linea centrale dell'elemento più largo sulla stessa linea
- MIDDLE, allinea la linea centrale dell'immagine con il centro della linea base del testo
- BASELINE, allinea il margine inferiore dell'immagine con la linea base del testo
- BOTTOM, come BASELINE
- ABSBOTTOM, allinea il margine inferiore dell'immagine con il margine inferiore dell'elemento più in basso sulla stessa linea.

Attributo ALT - Specifica il testo da visualizzare in alternativa all'immagine nel caso in cui:

- il browser non sia grafico;
- il file immagine sia stato mal specificato;
- sia stata scelta l'opzione del browser che annulla le immagini.

Esempio

```
<IMG SRC="mycat.gif" ALT="I' mi gattino">
```

Attributi VSPACE e HSPACE - Stabiliscono quanto spazio libero lasciare intorno all'immagine.

- VSPACE definisce lo spazio in pixel che deve essere lasciato libero sopra e sotto l'immagine;
- HSPACE stabilisce lo spazio in pixel che deve essere lasciato libero a sinistra e destra dell'immagine.

Esempio:

```
<P><IMG SRC="uccs.gif" VSPACE=30 HSPACE=30  
ALIGN=LEFT>Questo &acute; un uccs. Meglio  
stargli alla larga!</P>
```

Attributi HEIGHT e WIDTH - Specificano altezza (HEIGHT) e larghezza (WIDTH) in pixel dell'immagine. Se questa specifica corrisponde alle esatte dimensioni dell'immagine che il browser deve visualizzare, l'eventuale testo che circonda l'immagine viene immediatamente presentato.

Se si specificano dimensioni in pixel inferiori a quelle reali dell'immagine il browser trasforma l'immagine in un'altra con le dimensioni specificate. Gli effetti della trasformazione possono essere sgradevoli.

Attributo BORDER - Stabilisce la dimensione in pixel della cornice da inserire intorno all'immagine.

Attenzione. Se si specifica BORDER=0 e l'immagine ha associato un link il link non è più visibile.

Attributo LOWSRC - Specifica il nome di un altro file immagine come l'attributo SRC:

```
<IMG SRC="bigcat.gif" LOWSRC="smallcat.gif">
```

Quando il browser visualizza la pagina prima mostra il testo e l'immagine specificata in LOWSRC. Poi dopo che ha finito comincia a visualizzare l'immagine in SRC, che va a rimpiazzare quella di LOWSRC.

In questo modo il cybernauta impaziente prima vede l'immagine generalmente a bassa risoluzione e poi quella a alta.

Link immagine

In HTML è possibile avere dei link associati a delle immagini:

```
<P>
<A      HREF="a.htm"><IMG      SRC="rarrow.gif"
ALIGN=BOTTOM>
</A> right arrow<BR>
<A      HREF="b.htm"><IMG      SRC="larrow.gif"
ALIGN=BOTTOM>
right arrow</A><BR>
</P>
```

Il primo link è associato a una immagine che sarà perciò presentata con la cornice di selezione.

Il secondo link è associato sia a un immagine, presentata con la cornice di selezione, che a un testo, sottolineato.

L'avvento del colore

Per cambiare colore allo sfondo del documento e scrivere a lettere colorate HTML offre una serie di attributi nel tag <BODY>. La struttura generale è:

```
<BODY  BGCOLOR="#rrggb" TEXT="#rrggb"
LINK="#rrggb" VLINK="#rrggb" ALINK="#rrggb">
```

dove

BGCOLOR è il colore del background, cioè dell'area dietro al testo

TEXT è il colore del testo normale e delle intestazioni

LINK è il colore del testo di un link (*trigger*) non ancora visitato

VLINK è il colore di un trigger già visitato

ALINK è il colore di un trigger attivo, ovvero il tasto del mouse è stato premuto, ma non ancora rilasciato

I colori sono specificati con una tripla di due cifre in esadecimale:

rrggbb

dove

rr rappresenta la quantità di rosso (*red*)
gg rappresenta la quantità di verde (*green*)
bb rappresenta la quantità di blu (*blue*)

Ad esempio:

```
<BODY    BGCOLOR="#0000CC" TEXT="#FFFF00"  
LINK="#00DD00" VLINK="#FF0000" ALINK="#00FF00">
```

Corrisponde a:

- testo giallo
- sfondo blu scuro
- trigger non visitato verde scuro
- trigger visitato rosso acceso
- trigger attivo verde acceso.

Attenzione agli sfondi troppo scuri. Rendono illeggibile il testo.

Immagini di sfondo

In HTML si può utilizzare una immagine come sfondo. Lo sfondo viene riempito di piastrelle disposte su righe (*tiled*), dove ogni piastrella è l'immagine.

Occorre allora fare molta attenzione e scegliere immagini che una volta disposte a piastrella danno luogo ad un *continuum*, senza evidenziare antiestetici punti di giunzione.

Per creare uno sfondo piastrellato si usa l'attributo BACKGROUND del tag <BODY> in modo analogo all'attributo SRC del tag :

```
<BODY BACKGROUND="tiles.gif">  
<BODY BACKGROUND="backgrounds/rosemarble.gif">
```

Attenzione a non creare immagini di sfondo illeggibili e pesanti.

Form

In HTML è possibile costruire moduli (form) da far compilare a un utente. I moduli sono costituiti da campi vuoti (blank) che l'utente deve riempire digitando sulla tastiera e da liste di opzioni che l'utente deve selezionare.

Il contenuto del form, così come stabilito dall'utente, viene poi spedito al server che, a sua volta, lo instrada verso un programma che si prende cura del contenuto del form. Ad esempio il form potrebbe rappresentare l'ordine di un libro. L'ordine è spedito al server che lo dirotta verso un programma di gestione ordini che aggiorna un database e rimanda all'utente una conferma di ordine tramite posta elettronica.

Consideriamo un semplice esempio di form:

```
<H2>Title of form</H2>

<FORM                                METHOD="post_or_get"
ACTION="url_for_processing">
Name  <INPUT TYPE="TEXT" NAME="fieldname" SIZE=25
VALUE="default text"><BR>

<!-- And other input elements -->

<INPUT TYPE="SUBMIT" VALUE="Submit">
<INPUT TYPE="RESET" VALUE="Reset">

</FORM>
```

Tecnicamente parlando un form è qualsiasi cosa compreso nella sezione <FORM>.

Analizziamo il generico esempio.

Il tag <H2> dà un titolo al form. Questo tag è stato inserito solo per motivi di leggibilità, non è parte della sintassi del form, ma ha una funzione di supporto.

```
<H2>Title of form</H2>
```

Analizziamo invece la testata del form:

```
<FORM METHOD="post_or_get"
ACTION="url_for_processing">
```

L'attributo `METHOD` stabilisce la modalità attraverso cui sarà spedito al server il contenuto dei campi e delle scelte del form. `METHOD` può assumere due valori:

- `GET`
- `POST`

Il significato di queste due opzioni sarà chiarito più avanti. Per ora basti sapere che `GET` è la modalità di trasmissione meno sicura, indicata se il form ha pochi campi con poca informazione e `POST` è la modalità di trasmissione sicura adatta per form con diversi campi.

L'attributo `ACTION` invece specifica la locazione e il nome del programma che si prenderà cura del contenuto del form, una volta spedito al server. I programmi che sono innescati dal server per il trattamento dei form sono detti *CGI script*. L'acronimo CGI sta per Common Gateway Interface. Un CGI può essere un file di comandi scritto nella shell del sistema operativo (ad esempio Unix), un programma in Perl, in C/C++, VisualBasic o qualsiasi altro linguaggio compilato e/o interpretato.

Anche delle CGI parleremo più diffusamente. Per ora basti sapere che le CGI ricevono in qualche modo i dati e le scelte del form, le elaborano e ritornano indietro una pagina HTML di risposta.

Tra il tag di apertura e quello di chiusura del form sono inseriti gli elementi del form per inserire dati e effettuare scelte.

Nell'esempio:

```
Name <INPUT TYPE="TEXT" NAME="fieldname" SIZE=25  
MAX=40 VALUE="default text"><BR>
```

viene presentato un campo per l'input dati. In questo campo:

Name	è l'etichetta che precede il campo;
INPUT	è il tag che introduce un elemento imputabile;
TYPE	è l'attributo che specifica il tipo di campo. Nell'esempio il valore "TEXT" corrisponde all'inserimento di un testo semplice su una riga;
NAME	è l'attributo che identifica il nome del campo. Questo nome sarà poi trasmesso insieme al corrispondente valore digitato dall'utente al CGI script che gestisce il form;
SIZE	è l'attributo che definisce la dimensione in caratteri del campo. Occorre dimensionare questo valore sulla stringa più lunga che si prevede possa essere inserita, altrimenti l'utente

non vedrà la parte finale del testo che sta inserendo. Se questo attributo non è specificato il browser riserva uno spazio di 20 caratteri.

MAX è l'attributo che specifica il numero massimo di caratteri che può essere digitato.

VALUE è l'attributo che specifica il valore di default che riempie il campo. Se non specificato il campo è presentato vuoto.

Una volta che l'utente ha riempito il form può prendere due decisioni:

- inviare il form (Submit);
- abortire il form e riportarlo allo stato originale (Reset).

A questo scopo i form mettono a disposizione due bottoni speciali:

```
<INPUT TYPE="SUBMIT" VALUE="Submit">
```

Se l'attributo TYPE del tag <INPUT> è valorizzato a "SUBMIT" il browser crea un bottone contrassegnato dall'etichetta il cui valore è descritto dalla stringa VALUE.

Selezionando il bottone facendo click con il tasto sinistro del mouse si inviano al CGI script indicato dall'attributo ACTION le coppie fieldname=user_value corrispondenti al contenuto del form.

Per sbianchettare il form invece si usa un bottone di RESET:

```
<INPUT TYPE="RESET" VALUE="Reset">
```

L'attributo VALUE contiene la stringa che sarà incollata sul bottone di reset.

Per dare un formato visivo più gradevole al form si può usare il tag <PRE>:

```
<PRE>
Name          <INPUT TYPE="TEXT" NAME="fieldname1"
VALUE="default text"><BR>
Address       <INPUT TYPE="TEXT" NAME="fieldname2"
SIZE=25 VALUE="default text"><BR>
Phone Number <INPUT TYPE="TEXT" NAME="fieldname3"
VALUE="default text"><BR>
</PRE>
```

Un form può essere più utilmente "formattato" con le tabelle.

Text Area

Un altro elemento dei form HTML è il campo text area. Una text area è un campo libero che si estende su una o più linee. La sintassi tipica è:

```
label_or_prompt_for_the_area:    <BR>    <TEXTAREA  
NAME="areaname" ROWS=num1 COLS=num2>default text  
for the area</TEXTAREA>
```

dove:

label_or_prompt_for_the_area:	è l'etichetta che dice all'utente il tipo di testo da inserire.
TEXTAREA	è il nome del tag per una text area;
NAME	è l'attributo che specifica il nome dell'area. Questo nome sarà utilizzato dal CGI script per decifrare il contenuto del form.
ROWS	è l'attributo che definisce di quante righe testo è composto il form;
COLS	è l'attributo che definisce di quante colonne è composto il testo, in caratteri;
default text for the area	è il testo di default con cui è inizializzata l'area.

Esempio:

```
Minuta dell'incontro:<BR>  
<TEXTAREA NAME="minuta"  
ROWS=20 COLS=40>Si &acute; deciso che:  
<\TEXTAREA>
```

Radio button

I radio button sono elementi di un form impiegati quando si vuole che l'utente scelga una sola alternativa data una lista di possibilità. Ad esempio:

```
label for the list:  
<INPUT TYPE="RADIO" NAME="groupname" VALUE="option_1  
CHECKED>label 1
```

```

<INPUT TYPE="RADIO" NAME="groupname" VALUE="option_2">
label 2
<INPUT TYPE="RADIO" NAME="groupname" VALUE="option_3">
label 3
<INPUT TYPE="RADIO" NAME="groupname" VALUE="option_4">
label 4
<INPUT TYPE="RADIO" NAME="groupname" VALUE="option_5">
label 5

```

A una lista viene associata una descrizione generale:

label for the list

e un gruppo di elementi radio button contraddistinti dalla tipica sintassi:

```

<INPUT TYPE="RADIO" NAME="groupname" VALUE="option_1"
CHECKED>label 1

```

dove,

INPUT	è il tag usato per i campi editabili e/o selezionabili.
TYPE="RADIO"	specifica che l'elemento è un radio button.
NAME="groupname"	specifica il nome della variabile che sarà inviata allo script CGI insieme al valore dell'opzione scelta.
VALUE="option 1"	specifica che valore assegnare alla variabile
NAME	nel caso in cui venga scelta effettivamente quella voce.
CHECKED	è l'attributo che stabilisce la scelta di default.

Ad esempio:

```

What size do you want?: <BR>
<INPUT TYPE="RADIO" NAME="size" VALUE="small">Small
<INPUT TYPE="RADIO" NAME="size" VALUE="med">Medium
<INPUT TYPE="RADIO" NAME="size" VALUE="large" CHECKED>
Large
<INPUT TYPE="RADIO" NAME="size" VALUE="xlarge"> Extra
Large

```

Checkbox

I checkbox sono elementi di un form impiegati quando l'utente deve poter scegliere più alternative data una lista di possibilità.

Ad esempio:

label for the list:


```
<INPUT TYPE="CHECKBOX" NAME="option_1"
CHECKED>label 1
```

```
<INPUT TYPE="CHECKBOX" NAME="option_2">label 2
```

```
<INPUT TYPE="CHECKBOX" NAME="option_3">label 3
```

```
<INPUT TYPE="CHECKBOX" NAME="option_4">label 4
```

```
<INPUT TYPE="CHECKBOX" NAME="option_5">label 5<BR>
```

A una lista viene associata una descrizione generale:

label for the list

e un gruppo di elementi checkbox contraddistinti dalla tipica sintassi:

```
<INPUT TYPE="CHECKBOX" NAME="option_1" CHECKED>label 1
```

dove,

INPUT	è il tag usato per i campi editabili e/o selezionabili.
-------	---

TYPE="CHECKBOX"	specifica che l'elemento è una checkbox.
-----------------	--

NAME="option_1"	specifica il nome della opzione che sarà inviata allo script CGI se selezionata.
-----------------	--

CHECKED	è l'attributo che stabilisce la scelta di default.
---------	--

Alternativamente i checkbox potrebbero anche essere trattati come i radiobutton usando tutti e due gli attributi NAME e VALUE.

Ad esempio:

What kind of media do you own?:


```
<INPUT TYPE="CHECKBOX"
```

```
NAME="media" VALUE="telephone">Telephone
```

```
<INPUT TYPE="CHECKBOX"
```

```
NAME="media" VALUE="tv">TV
```

```
<INPUT TYPE="CHECKBOX"
```

```
NAME="media" VALUE="radio" CHECKED>Radio
```

Questo formato però è più duro da interpretare a livello di CGI.

Selection list

Una lista di scelte può essere presentata anche usando un menu di tipo pull-down o pop-up. La sintassi generale è:

```
label or prompt for the list: <BR>
<SELECT NAME="listname" MULTIPLE SIZE="rows">
<OPTION SELECTED>first option
<OPTION NAME="second">second option
<OPTION>third option
</SELECT><BR>
```

L'argomento trattato dalla selection list è specificato da:

label or prompt for the list:

Il significato per ciascun elemento è:

SELECT	è il nome del tag di una selection list
NAME	è il nome della selection list usato come nome di variabile per spedire alla CGI l'opzione.
MULTIPLE	è l'attributo da specificare nel caso in cui si consenta più di una scelta.
SIZE	è l'attributo che specifica quante opzioni debbono essere visualizzate.
OPTION	è il tag che contraddistingue ciascuna voce della selection list.
NAME	è il nome del valore da associare alla scelta.

Ancora su INPUT

L'attributo TYPE del tag INPUT assume i seguenti valori:

TEXT
INT
CHECKBOX
RADIO
SUBMIT
RESET

HIDDEN

Gli unici non considerati fino ad ora sono INT e HIDDEN. Il valore INT è riservato a campi destinati a contenere valori numerici, mentre il valore HIDDEN fa sì che in corrispondente campo non venga visualizzato ma venga trasferito il contenuto specificato dall'attributo VALUE, senza che l'utente immetta alcun valore. In pratica il campo HIDDEN serve per comunicazioni interne che si propagano da un form a quello successivo o da uno script a quello successivo.

Input di file

Esiste un tipo particolare di form che consente la spedizione di un file:

```
<FORM ENCTYPE="multipart/form-data" ACTION="_URL_"  
METHOD=POST>
```

Send the file: <INPUT NAME="userfile" TYPE="file">

<INPUT TYPE="submit" VALUE="Send File">

</FORM>

Tabelle

Una tabella è una struttura bidimensionale che viene costruita per righe:

```
<TABLE BORDER>  
<CAPTION ALIGN=TOP>Caption for tabel</CAPTION>  
<TR>  
    <TD>Content of Cell 1A</TD>  
    <TD>Content of Cell 1B</TD>  
    <TD>Content of Cell 1C</TD>  
</TR>  
<TR>  
    <TD>Content of Cell 2A</TD>  
    <TD>Content of Cell 2B</TD>  
    <TD>Content of Cell 2C</TD>  
</TR>  
<\TABLE>
```

dove

TABLE	è il tag che contrassegna una tabella
BORDER	è l'attributo facoltativo per avere tabelle incorniciate
TR(table row)	è il tag che contrassegna una riga
TD(table data)	è il tag che contrassegna il contenuto di una cella
CAPTION	è il tag che contrassegna il titolo della tabella. Questo tag ha un attributo ALIGN che può assumere valore TOP, titolo in testa (caso più comune), o BOTTOM, titolo in fondo alla tabella.

Per costruire una cella vuota basta specificare:

```
<TD></TD>
```

La cella apparirà come una finestra chiusa.

Per realizzare vuota che appaia come una finestra aperta specificare:

```
<TD><BR></TD>
```

Le tabelle possono avere intestazioni sia orizzontali che verticali. Una cella che contenga una intestazione è contrassegnata dal tag TH (table heading). Le celle delle intestazioni sono specificate analogamente alle celle dati:

```
<TABLE BORDER>
<CAPTION ALIGN=TOP>Caption for tabel</CAPTION>
<TR>
  <TH></TH>
  <TH>Head A</TH>
  <TH>Head B</TH>
  <TH>Head C</TH>
</TR>

<TR>
  <TH>Head 1</HD>
  <TD>Content of Cell 1A</TD>
  <TD>Content of Cell 1B</TD>
  <TD>Content of Cell 1C</TD>
</TR>

<TR>
  <TH>Head 2</TH>
  <TD>Content of Cell 2A</TD>
  <TD>Content of Cell 2B</TD>
```

```

        <TD>Content of Cell 2C</TD>
    </TR>
</TABLE>

```

Allineamento del contenuto di una cella- È possibile posizionare il contenuto di una cella con due attributi, ALIGN (horizontal align) e VALIGN (vertical align). La sintassi è:

```
<TD ALIGN=hpos VALIGN=vpos>Cell Content</TD>
```

dove

hpos= LEFT | CENTER | RIGHT

e

vpos= BOTTOM | MIDDLE | TOP

ALIGN è l'allineamento in orizzontale

VALIGN è l'allineamento in verticale.

Celle multiriga e multicolonna - È possibile specificare celle che si estendono su più di una riga e di una colonna usando gli attributi ROWSPAN e COLSPAN per i tag TH e TD.

Esempio:

```

<TABLE BORDER>
<CAPTION ALIGN=TOP>Caption for tabel</CAPTION>
<TR>
    <TH ROWSPAN=2 COLSPAN=2</TH>
    <TH COLSPAN=2>Head A-B</TH>
    <TH COLSPAN=2>Head C-D</TH>
</TR>
<TR>
    <TH>Head A</TH>
    <TH>Head B</TH>
    <TH>Head C</TH>
    <TH>Head D</TH>
</TR>

<TR>
    <TH ROWSPAN=2>Head 1-2</TH>
    <TH>Head 1</TH>

```

```

        <TD>Content of Cell 1A</TD>
        <TD>Content of Cell 1B</TD>
        <TD>Content of Cell 1C</TD>
        <TD>Content of Cell 1D</TD>
</TR>
<TR>
    <TH>Head 2</TH>
    <TD>Content of Cell 2A</TD>
    <TD>Content of Cell 2B</TD>
    <TD>Content of Cell 2C</TD>
    <TD>Content of Cell 2D</TD>
</TR>
<TR>
    <TH ROWSPAN=2>Head 3-4</TH>
    <TH>Head 3</TH>
    <TD>Content of Cell 3A</TD>
    <TD>Content of Cell 3B</TD>
    <TD>Content of Cell 3C</TD>
    <TD>Content of Cell 3D</TD>
</TR>
<TR>
    <TH>Head 4</TH>
    <TD>Content of Cell 4A</TD>
    <TD>Content of Cell 4B</TD>
    <TD>Content of Cell 4C</TD>
    <TD>Content of Cell 4D</TD>
</TR>
</TABLE>

```

Le estensioni Netscape - Netscape estende la sintassi delle tabelle consentendo regolazioni più fini sulla dimensione delle tabelle.

La prima estensione concerne l'attributo WIDTH che può essere incluso nei tag <TABLE>, <TH> e <TD>. L'attributo WIDTH specifica per il tag <TABLE> la larghezza della tabella espressa sia in pixel che in percentuale della larghezza di pagina.

Per i tag <TH> e <TD> l'attributo WIDTH specifica la larghezza in modo assoluto in pixel, o in modo relativo come percentuale della larghezza complessiva della tabella.

L'attributo BORDER del tag <TABLE> può essere valorizzato con un valore numerico. BORDER=1 è il default, BORDER=0 equivale ad omettere l'attributo BORDER.

L'attributo numerico CELLSPACING del tag <TABLE> determina la larghezza della linea ombreggiata che separa due celle contigue. Il valore di default è 2.

L'attributo numerico CELLPADDING del tag <TABLE> determina la larghezza della linea ombreggiata che separa due celle contigue. Il valore di default è 2.

Image Map

In HTML è possibile predisporre mappe reattive secondo due modalità:

server-side;
client-side.

Nella prima modalità la coordinata della selezione sull'immagine è gestita dal server; nella seconda dal client.

Una mappa reattiva è una image scomposta in parti (punti, rettangoli, cerchi, polilinee) dove ad ogni parte nell'immagine è associabile un link.

Server-side image map

La procedura da seguire per la costruzione di image map di tipo server-side è:

1. scegliere l'immagine reattiva;
2. specificare le aree selezionabili dell'immagine reattiva in un file .map;
3. collegare l'immagine reattiva, il file .map e il Gateway script che riceve la coordinata selezionata per confrontarla con il file .map.

Il primo passo viene svolto indipendentemente dal tipo di server adottato, mentre gli ultimi due passi cambiano a seconda che si usi un server CERN o uno NCSA.

La scelta dell'immagine

Poichè le aree selezionabili sono figure geometriche ben definite:

- il punto (si specificano x, y)
- il rettangolo (si specificano le coordinate dell'angolo in alto a sinistra e dell'angolo in basso a destra);

- il cerchio (si specificano le coordinate del centro e la lunghezza del raggio);
- il poligono (si specificano le coordinate di ogni angolo)

è consigliabile scegliere immagini che si prestano a suddivisioni geometriche nette. Ad esempio la fotografia di un animale è un pessimo soggetto, mentre un .gif con l'immagine di uno scaffale di libri potrebbe andare bene.

Si ricorda inoltre che le coordinate/lunghezze sono espresse in pixel e che gli assi coordinati hanno origine nell'angolo in alto a sinistra e l'asse delle Y positive si sviluppa verso il basso.

Server CERN

La sintassi del file .map per la specifica delle aree selezionabili all'interno della figura è:

```
default URL
circle (x,y) r URL
rectangle (x,y) (x,y) URL
polygon (x1,y1) (x2,y2)...(xn,yn) URL
```

Ad esempio

```
circle (10,15) 20 /abel/banana.htm
circle (346,23) 59 /abel/pesca.htm
rectangle (57,57) (100,210) /abel/pere/williams.htm
polygon (192,3) (192,170) (115,217) /abel/mele/cotogne.htm
default /abel/uccs.htm
```

L'ordine di definizione delle aree selezionabili è importante: se una certa zona è coperta da più di un'area allora vince sempre la prima definizione.

La definizione contrassegnata dalla parola chiave default ha il significato di "tutti gli altri punti dell'immagine non inclusi in una delle aree definite".

La URL deve essere specificata con indirizzamento assoluto.

Il server CERN non ha aree di tipo punto semplice.

Il file .map che definisce le aree selezionabili, ad esempio frutta.map, deve essere incluso nel server nella directory (o in una sua sottodirectory) che contiene tutti i CGI script. Ad esempio un possibile scenario potrebbe essere:

/home/www/cgi-bin/
/home/www/maps/frutta.map
/home/www/abel/

Il server CERN mette a disposizione un ben preciso cgi script, denominato htimage, che intercetta la coordinata del punto selezionato sull'immagine, consulta il file .map associato all'immagine e attiva la corrispondente URL.

Nel documento HTML che contiene l'immagine reattiva occorre inserire il seguente frammento:

```
<A HREF="../cgi-bin/htimage/maps/frutta.map">  
<IMG SRC="macedon.gif" ISMAP>  
</A>
```

si noti che dopo il nome del cgi scrip htimage, specificato normalmente occorre indicare la directory assoluta, /maps/frutta.map, del file .map a partire dalla directory del server web.

L'attributo ISMAP del tag avverte il browser e in conseguenza il server di gestire le coordinate del punto selezionato.

Server NCSA

La sintassi del file .map per la specifica delle aree selezionabili all'interno della figura è:

default URL
circle URL x,y,r
rect URL x,y,x,y
poly URL x1,y1,x2,y2,...xn,yn
point URL x,y

Ad esempio

circle /abel/banana.htm 10,15,20
circle /abel/pesca.htm 346,23,59
rect /abel/pere/williams.htm 57,57,100,210
poly /abel/mele/cotogne.htm 192,3,192,170,115,217
point /abel/cliliegia.htm 100,100
default /abel/uccs.htm

L'ordine di definizione delle aree selezionabili è importante: se una certa zona è coperta da più di un'area allora vince sempre la prima definizione. La definizione contrassegnata dalla parola chiave default ha il significato di "tutti gli altri punti dell'immagine non inclusi in una delle aree definite".

La URL deve essere specificata con indirizzamento assoluto.

Il file .map che definisce le aree selezionabili, ad esempio frutta.map, deve essere incluso nel server nella directory (o in una sua sottodirectory) che contiene tutti i CGI script. Ad esempio un possibile scenario potrebbe essere:

```
/home/www/cgi-bin/  
/home/www/maps/frutta.map  
/home/www/abel/
```

Il server NCSA mette a disposizione un ben preciso cgi script, denominato imagemap, che intercetta la coordinata del punto selezionato sull'immagine, consulta il file .map associato all'immagine e attiva la corrispondente URL.

Nel documento HTML che contiene l'immagine reattiva occorre inserire il seguente frammento:

```
<A HREF="cgi-bin/imagemap/maps/frutta.map">  
<IMG SRC="macedon.gif" ISMAP>  
</A>
```

si noti che dopo il nome del cgi scrip htimage, specificato normalmente occorre indicare la directory assoluta, /maps/frutta.map, del file .map a partire dalla directory del server web.

L'attributo ISMAP del tag avverte il browser e in conseguenza il server di gestire le coordinate del punto selezionato.

Assicurarsi di usare sempre l'ultima versione dello script imagemap.

Client-side image map

Il difetto delle server-side image map è che non si possono costruire immagini reattive in locale. Netscape allora ha introdotto estensioni dell'HTML che consentono la gestione di mappe reattive in locale, cioè dalla parte del client.

L'attributo USEMAP del tag indica che l'immagine sarà gestita dalla parte del client"

```
<IMG SRC="../images/tech/pic1.gif" USEMAP="maps.html#map1">
```

L'argomento di USEMAP specifica quale mappa usare con l'immagine in un formato simile all'attributo HREF. Se l'argomento inizia con # allora si assume che la specifica delle aree selezionabili sia nel medesimo file del documento.

Le differenti regioni dell'immagine sono descritte usando il tag <MAP>:

```
<MAP NAME="name">
<AREA [SHAPE="shape"] COORDS="x,y,..." [HREF="reference"]
[NOHREF]>
</MAP>
```

Il valore name specifica il nome del file map a cui IMG deve riferirsi. Il valore shape definisce la forma dell'area. Valori ammessi sono:

rect, rettangolo;
poly, poligono;
circle, cerchio;
default, per tutto il resto.

Se SHAPE è omissso il default è rect. COORDS descrive le coordinate della forma usando i pixel come unità di misura.

L'attributo NOREF indica che il click in quella regione non dovrebbe dar luogo a niente.

Se due aree si intersecano vince sempre la definizione che appare per prima:

```
<MAP NAME="buttonbar">
<AREA SHAPE="RECT" COORDS="10,10,49,49" HREF="us.htm">
<AREA SHAPE="RECT" COORDS="60,10,99,49" HREF="as.htm">
<AREA SHAPE="RECT" COORDS="110,10,149,49" HREF="os.htm">
<AREA SHAPE="RECT" COORDS="0,0,159,59" NOHREF>
</MAP>
<IMG SRC="../images/tech/bar.gif" USEMAP="#buttonbar">
```

L'ultima area con il NOHREF prende tutta l'immagine e quindi stabilisce un default non reattivo. Di fatto questa specifica è inutile perchè se non si specifica default la mappa non è reattiva.

FRAME

Un documento Frame ha una struttura base simile ad un normale documento HTML, escluso per il fatto che il contenitore BODY è sostituito dal contenitore FRAMESET che descrive documenti sub-HTML, o Frame che allestiranno la pagina:

```
<HTML>
```

```
<HEAD>
```

```
</HEAD>
```

```
<FRAMESET>
```

```
</FRAMESET>
```

```
</HTML>
```

FRAMESET

La sintassi dei frame non è più rognosa di quella impiegata per le tabelle. Il tag <FRAMESET> ha i seguenti attributi:

ROWS="row_height_value_list"

```
<FRAMESET ROWS="20%,60%,20%">
```

```
<FRAMESET ROWS="100,*,100">
```

COLS="column_width_list"

Il tag FRAMESET può essere annidato entro altri tag FRAMESET. All'interno di un frameset si definiscono i singoli FRAME con il tag <FRAME>. Attributi di <FRAME> sono:

SRC="url"

l'URL del documento da visualizzare nel frame

NAME="window_name"

il nome da assegnare al frame e da usare con l'attributo TARGET.

MARGINWIDTH="value"

È il margine destro/sinistro in pixel del frame.

MARGINHEIGHT="value"

È il margine superiore/inferiore in pixel del frame.

SCROLLING="yes|no|auto"

yes scrollbar sempre visibili

no scrollbar non visibili

auto decide da solo (è il default)

NORESIZE

Il frame non è resizable da parte dell'utente.

Un altro tag utile è <NOFRAME> dove si inseriscono istruzioni per i browser che non interpretano i frame.

Esempio:

```
<FRAMESET COLS="50%,50%">
  <FRAMESET ROWS="50%,50%">
    <FRAME SRC="cell.html">
    <FRAME SRC="cell.html">
  </FRAMESET>
  <FRAMESET ROWS="33%,33%",33%">
    <FRAME SRC="cell.html">
    <FRAME SRC="cell.html">
    <FRAME SRC="cell.html">
  </FRAMESET>
</FRAMESET>
```

TARGET

Per indirizzare un documento in una finestra definita da un frame si usa l'attributo TARGET:

Targeted Anchor
Carica il link definito da HREF nella finestra specificata da TARGET.

<AREA SHAPE="shape" ... HREF="url" TARGET="window_name">
come per i link solo in seguito alla selezione dell'area.

<FORM ACTION="url" TARGET="window_name">

Il risultato della submit del form è inviato alla finestra specificata dal TARGET.

Nomi di TARGET

I target devono essere nomi alfanumerici. Ci sono alcuni nomi speciali che iniziano con `_`. Questi sono:

`TARGET="_blank"`

il link è caricato in una nuova finestra vuota. Questa finestra non ha un nome.

`TARGET="_self"`

il link è sempre caricato nella stessa finestra dove si è fatto il click dell'ancora.

`TARGET="_parent"`

il link è caricato nel primo FRAMESET babbo di questo documento. Si comporta come `_self` se il documento non ha babbo.

`TARGET="_top"`

Il link è caricato nel corpo completo della finestra. Si comporta come `_self` se il documento è già al top.