

Guida Html

Contents

1	HTML Base	3
1.1	Dare un titolo al documento	4
1.2	Aggiungere titoli e paragrafi	4
1.3	Aggiungere un pò di enfasi	5
1.4	Inserire immagini	6
1.5	Link ad altre pagine	7
1.6	Tre tipi di liste	9
2	HTML avanzato	11
2.1	Come forzare l'a capo.	11
2.2	Controllare la separazione delle parole	12
2.3	Inserimento di caratteri speciali	13
2.4	Collegamenti a metà di una pagina	13
2.5	Testo preformattato	14
2.6	Raggruppare il testo attorno alle immagini	15
2.7	Mappe cliccabili	15
2.7.1	Importanza dell'attributo <i>alt</i>	16
2.8	Tabelle	16
2.8.1	Cell Padding	17
2.8.2	Cell Spacing	17
2.8.3	Table Width	17
2.8.4	Allineamento del testo all'interno delle celle	17
2.8.5	Celle vuote	18
2.8.6	Celle che occupano più di una riga e di una colonna	18
2.8.7	Tabelle senza bordi	18
2.8.8	Colori in una tabella	18

Chapter 1

HTML Base

Questa è un brevissima guida ai concetti fondamentali del linguaggio HTML. E' possibile sperimentare direttamente gli esempi proposti utilizzando un qualsiasi editor di testo disponibile sul sistema utilizzato: come *notepad* sotto *Microsoft Windows* o *vi* sotto *Linux*. Anche se in avvenire non si intende editare il codice a mano, ma si pensa di adottare un ambiente di sviluppo come *Netscape Composer* o *Microsoft Frontpage* o *Dreamweaver*, credo che sarà comunque utile conoscere i tag¹ principali del linguaggio. Questo primo capitolo contiene qualche elemento base per cominciare a scrivere una semplice pagina HTML, il capitolo secondo conterrà qualcosa di più complesso e, infine, il terzo capitolo introduce il concetto di fogli di stile.

Il linguaggio HTML è molto flessibile o meglio, sono i browser² ad essere particolarmente flessibili, di solito. Ciò significa che difficilmente otterremo degli errori da un browser in seguito alla visualizzazione di un sorgente HTML con degli errori di sintassi. Il browser cerca sempre di visualizzare il contenuto della pagina, anche se i tag sono incompleti, non chiusi, o inesistenti. Naturalmente ciò non autorizza a scrivere volutamente codice non preciso confidando nel comportamento comunque "intelligente" del browser. In effetti nessuno ci garantisce (nessuno standard si intende) che il browser debba interpretare correttamente delle pagine sintatticamente errate.

Un'ottimo modo di imparare l'HTML e curiosare nel codice scritto da altri. Molti browser permettono di visualizzare il sorgente della pagina scaricata (Visualizza → HTML in IE o View → Page Source in Mozilla).

I diversi argomenti introdotti in questo primo capitolo sono:

- settare un titolo
- titoli e paragrafi
- enfatizzare il testo

¹Si vedrà più avanti cos'è un tag.

²Un browser è un programma in grado di interpretare del codice HTML, come *Internet Explorer*, *Netscape Navigator*, *Mozilla*, *Opera*, ...

- immagini
- link ad altre pagine
- diversi tipi di liste

1.1 Dare un titolo al documento

Questo è il codice associato ad una semplice pagina vuota, ma con un titolo “Salve, mondo.”:

```
<html>
<head>
  <title>Salve, mondo.</title>
</head>
<body>
</body>
</html>
```

Aprire un qualsiasi editor di testo a disposizione ed inserire il testo sopra. Salvare il documento (preferibilmente con estensione .htm o .html) ed aprirlo con un browser.

Ciò che si dovrebbe vedere è appunto un pagina vuota (il che non vuol dire necessariamente bianca), con il titolo “Salve, mondo.”. In molti browser (ho modo di sperimentare *Internet Explorer* e *Mozilla*) il titolo di una pagina compare nella barra del titolo della finestra del programma.

Il codice scritto sopra è costituito da un insieme di tag: alcuni sono tag di apertura (<html>, <head>, <title>, <body>); altri sono tag di chiusura (</html>, </head>, </title>, </body>). L'intero codice sorgente della pagina è compreso tra i tag <html> e </html>. Tra questi due si riconosce chiaramente una prima parte di intestazione (<head> ... </head>), e una seconda parte di corpo vero e proprio della pagina (<body> ... </body>).

All'interno dell'intestazione della pagina e tra i tag <title> e </title> si inserisce il titolo scelto per il documento. A questo punto dovrebbe essere evidente come modificare la pagina mettendo un titolo diverso.

1.2 Aggiungere titoli e paragrafi

Qualsiasi testo che si desideri inserire all'interno della pagina va messo nella sezione corpo vero e proprio del codice, ovvero tra i tag <body> e </body>.

Per ora ci preoccupiamo di come scrivere due tipi di testo all'interno del documento: i titoli ed i paragrafi (ossia il testo normale).

I titoli, a seconda della loro importanza, posso avere sei livelli di grandezza. Si realizzano inserendo il testo tra i tag <h1> e </h1>, oppure <h2> e </h2>, e così via fino a <h6> e </h6>. Ad esempio:

```
<h1>Questo è un titolo principale</h1>
<h2>Questo è un sottotitolo</h2>
<h3>Questo è un sottotitolo del sottotitolo</h3>
```

Ciò che non è un titolo è evidentemente un testo normale (un paragrafo), che va inserito fra i tag `<p>` e `</p>` (anche se, in questo caso, il tag di chiusura è facoltativo).

Riassumendo, una semplice pagina con un titolo ed un testo è fatta quindi così:

```
<html>
  <head>
    <title>Esempio</title>
  </head>
  <body>
    <h1>Capitolo 1</h1>
    <p>
      Il mattino ha l'oro in bocca
      Il mattino ha l'oro in bocca
        Il mattino ha l'oro in bocca
        Il mattino ha l'oro in bocca
      Il mattino ha l'oro in bocca
    </p>
  </body>
</html>
```

Vale la pena testare personalmente l'esempio sopra, notando tra l'altro come è in differente la maniera in cui il codice viene scritto: dove mettiamo gli spazi o dove andiamo a capo, in ogni caso la pagina viene visualizzata sempre nella stessa maniera.

1.3 Aggiungere un pò di enfasi

Per enfatizzare una porzione di testo, si può utilizzare il tag ``:

```
Questo e' un <em>nuovo</em> concetto.
```

Questo nuovo tag illustra un concetto estremamente interessante: la distinzione fra contenuto informativo di una pagina e maniera in cui lo stesso contenuto viene visualizzato.

Ciò che ci interessa nel momento in cui utilizziamo questo tag è riuscire ad enfatizzare, ossia distinguere dal resto, una porzione di testo. Non sappiamo come il risultato avverrà di preciso, ossia non stabiliamo che enfatizzare una parola significa scriverla in corsivo, oppure grassetto, oppure sottolineato. Lasciamo il browser libero di interpretare questo tag, a patto che il risultato sia comunque l'enfasi della parola. In questa maniera utenti diversi con browser diversi probabilmente visualizzeranno in maniera leggermente diversa la pagina, ma il nostro risultato sarà comunque raggiunto.

1.4 Inserire immagini

L'inserimento di un'immagine in un documento HTML avviene attraverso il tag ``. Diversamente dai precedenti, per ora il tag `` è l'unico tag che non ha una chiusura. Si supponga ad esempio di voler inserire in un certo punto del documento un file (*peter.jpg* ad esempio) memorizzato nella stessa directory del sorgente HTML. Supponendo che le dimensioni dell'immagine (in pixel) siano di 200 per 150, si avrà il seguente frammento di codice.

```

```

Si è già detto che questo tag si differenzia dagli altri per non avere una chiusura, inoltre questo tag ha anche dei parametri. In generale qualsiasi tag ha dei parametri, ogni parametro ha un nome ed un valore (racchiuso tra doppi apici). Nell'esempio precedente le coppie nome valore sono chiaramente:

Nome	Valore
src	peter.jpg
width	200
height	150

E' evidente che i parametri di un tag (siano essi obbligatori o facoltativi) servono per specializzare, rendere più preciso il significato di quel particolare tag. Nell'esempio mostrato il primo parametro (*src*) è senza dubbio fondamentale, serve infatti per indicare al browser quale immagine deve scaricare dal web server, per poter inserire in quel particolare punto della pagina. Omettendo il parametro *src*, o mettendo un valore sbagliato (corrispondente ad un nome di file inesistente) non si otterranno messaggi di errore, semplicemente la figura non verrà visualizzata dal browser. Gli altri due parametri (*width* e *height*) sono meno importanti: indicano al browser le dimensioni (in pixel) dell'immagine. Questi due parametri possono essere omessi nel caso in cui l'immagine venga visualizzata nella sua grandezza originale³, diversamente possono essere utilizzati per forzare l'immagine (eventualmente deformandola) ad avere quelle due dimensioni, anche se non sono le sue dimensioni originali.

Dal pezzo di codice descritto sopra sembra che sia possibile descrivere con una certa esattezza dove l'immagine dovrà andarsi ad inserire all'interno della pagina. Questa caratteristica è allo stesso tempo un limite ma anche una forza del linguaggio HTML.

Il linguaggio non fornisce nessun meccanismo per specificare ad esempio che l'immagine deve essere a 354 pixel dall'alto e a 12 pixel dal lato sinistro della finestra, e ciò potrebbe sembrare una limitazione di creatività; tuttavia in questa maniera è il browser che stabilisce dove è più opportuno inserire l'immagine. In questa maniera anche persone diverse con sistemi operativi diversi e browser diversi possono comunque visualizzare correttamente la pagina.

Un ultimo parametro molto importante da prendere in considerazione quando si ha a che fare con delle immagini è *alt*:

³Le informazioni sulla grandezza dell'immagine sono in questo caso intrinseche nel file.

```

```

Il parametro *alt* contiene una breve descrizione dell'immagine inserita, in questa maniera il testo descrittivo viene sostituito all'immagine in tutti questi casi in cui l'immagine non può essere caricata ad esempio:

- l'indirizzo dell'immagine è sbagliato
- l'immagine è stata rimossa dal server
- l'immagine è danneggiata
- l'utente ha disabilitato il download delle immagini dal proprio browser
- l'utente sta utilizzando un browser testuale⁴

La maggior parte dei browser supporta i formati grafici più tradizionali: GIF o JPEG. Non è raro che anche altri formati come PNG o BMP siano comunque supportati.

Chi progetta siti web non può essere esente da considerazioni riguardanti i tempi di scaricamento di una pagina: è quindi evidente che un'immagine in formato JPEG è di gran lunga preferibile alla stessa immagine in formato BMP. Non è saggio utilizzare i due parametri *width* e *hight* per ridurre le dimensioni di un'immagine, si supponga infatti di avere un file "luca.jpg" di 300 Kb, di dimensioni 300 per 150 pixel:

```

```

Volendo ridimensionare l'immagine, il pezzo di codice sopra realizza l'intento ma non in maniera efficiente. In effetti l'immagine viene comunque scaricata interamente (in tutti i suoi 300 Kb), e quindi viene ridimensionata dal browser. Una soluzione migliore sarebbe sicuramente stata quella di ridimensionare l'immagine utilizzando un programma di elaborazione di immagini portandola ad esempio da 300 Kb a 100 Kb e di caricare sul server la versione "ridotta" dell'immagine.

```

```

1.5 Link ad altre pagine

Ciò che rende l'HTML particolarmente adatto al Web o comunque alla realizzazione di certi tipi di documenti "facilmente consultabili" e la possibilità di descrivere degli *ipertesti*. Un ipertesto è un testo contenente dei collegamenti ad altri ipertesti, le modalità di passaggio da una pagina ad un'altra ad essa collegata tramite collegamento dipendono chiaramente dal browser in esame

⁴Spesso i browser testuali vengono utilizzati dai non vedenti perché l'output può essere reindirizzato ad una tastiera braille o ad un programma di sintesi vocale.

tuttavia, nella maggior parte dei browser⁵, è sufficiente cliccare con il mouse su un collegamento per invocare il caricamento della pagina di destinazione.

In HTML è sufficiente inserire un link, per creare un collegamento ad un'altra pagina:

```
<a href="Luca.html">Home page di Luca</a>
```

Si noti prima di ogni altra cosa che si hanno i due tag di apertura e chiusura `<a>` e ``. Il testo racchiuso tra tali tag ("Home page di Luca") rappresenta il collegamento vero e proprio (ossia dove cliccare per andare nella pagina di destinazione).

Si noti inoltre l'indispensabile parametro *href*, il cui valore è la pagina di destinazione del collegamento. Il tipo di collegamento utilizzato nell'esempio precedente è un "collegamento relativo": si assume cioè che il file di destinazione *Luca.html* si trovi nella stessa directory della pagina corrente. Leggermente diverso, ma altrettanto corretto, sarebbe stato l'utilizzo di un riferimento assoluto per indicare la pagina di destinazione.

```
<a href="http://www.w3.org/">W3C</a>
```

Vediamo un altro esempio per vedere cosa accade:

```
<html>
<head>
  <title>Un collegamento</title>
</head>
<body>
  A <a href="http://www.w3.org/">questa pagina </a>
  si trovano molte informazioni
  sull'HTML.
</body>
</html>
```

Dovrebbe essere evidente come apparirà questa pagina.

Molti tag possono essere composti in maniera annidata, questo ci permette ad esempio di utilizzare un'immagine come collegamento ad una diversa pagina. E' sufficiente inserire l'immagine (tramite il tag ``) tra i due tag `<a>` e ``:

```
<a href="luca.html"></a>
```

Un elemento nuovo introdotto nel precedente esempio riguarda la possibilità di identificare gli url relativi con il formalismo standard sotto unix: la scrittura *images/luca.gif* significa che il file *luca.gif* si trova, a partire dalla directory corrente, nella directory *images*. Allo stesso modo un url del tipo *../temp/file.html* significa che *file.html* si trova nella cartella *temp*, che si trova allo stesso livello della directory corrente.

⁵Internet Explorer, Mozilla, Netscape, Galeon, Opera, ...

1.6 Tre tipi di liste

In HTML è possibile realizzare tre tipi di liste.

Il primo tipo di lista che prendiamo in considerazione è la lista non ordinata di elementi, più nota come *unordered list*. Essa è compresa tra i tag `` e ``, ogni elemento della lista è invece compreso tra i tag `` e `` (presumo che stia per *list item*):

```
<ul>
  <li>Primo elemento della lista;</li>
  <li>secondo elemento della lista;</li>
  <li>ultimo elemento della lista.</li>
</ul>
```

Questo tipo di lista viene realizzata, in quasi tutti i browser, mediante un elenco puntato con un elemento per ogni item. Mentre è fondamentale aprire e chiudere correttamente i due tag `` e ``, per ogni elemento della lista il tag di chiusura `` è facoltativo, è infatti chiaro che il browser è comunque in grado di interpretare correttamente la lista.

Il secondo esempio di lista è quella con un insieme ordinato di elementi, ossia la *ordered list*. Essa viene realizzata mediante i due tag `` e ``, mentre i vari elementi della lista sono racchiusi sempre tra `` e ``:

```
<ol>
  <li>Primo elemento</li>
  <li>Secondo elemento</li>
  <li>Terzo elemento</li>
</ol>
```

Questo tipo di lista realizza la lista ordinata, e viene generalmente interpretata dal browser inserendo un numero davanti ad ogni item della lista.

Prima di introdurre il terzo tipo di lista, ricordiamo che molti tag HTML possono essere annidati, in questo caso questo ci permette evidentemente di creare liste annidate di elementi:

```
<ul>
  <li>item</li>
  <li>item</li>
  <li>
    <ul>
      <li>subitem</li>
      <li>subitem</li>
    </ul>
  </li>
  <li>item</li>
</ul>
```


In questo esempio abbiamo una *unordered list* in cui il terzo elemento è, a sua volta, un'altra *unordered list*. Molti browser realizzano ciò scrivendo la seconda lista più indentata verso destra rispetto alla prima.

Lo stesso esempio si può fare, ma relativamente alle *ordered list*:

```
<ol>
  <li>item</li>
  <li>item</li>
  <li>
    <ol>
      <li>subitem</li>
      <li>subitem</li>
    </ol>
  </li>
  <li>item</li>
</ol>
```

Infine è evidente come liste ordinate e non ordinate si possano mischiare assieme in maniera annidata.

Infine il terzo esempio di lista realizzabile in HTML è la lista di definizioni. L'intera lista è compresa tra i due tag `<dl>` e `</dl>`, ogni elemento della lista è costituito da un termine (compreso tra `<dt>` e `</dt>`) e da una definizione per quel termine (compresa fra `<dd>` e `</dd>`):

```
<dl>
  <dt>primo termine</dt>
  <dd>definizione per il primo termine</dd>

  <dt>secondo termine</dt>
  <dd>definizione per il secondo termine</dd>

  <dt>terzo termine</dt>
  <dd>definizione per il terzo termine</dd>
</dl>
```

Questo terzo tipo di lista non viene realizzata né come elenco puntato né come elenco numerato. I vari nomi con le loro definizioni vengono scritti uno sotto l'altro e indentati in maniera che si capisca bene quale è il nome e quale è la definizione.

Chapter 2

HTML avanzato

In questo secondo capitolo verranno introdotti alcuni nuovi argomenti “avanzati”, tra cui:

- A capo forzato
- Parole non spezzabili
- Caratteri speciali
- Link in mezzo ad una pagina
- Testo preformattato
- Testo attorno alle immagini
- Mappe cliccabili
- Tabelle
- Linguaggi di scripting

2.1 Come forzare l’a capo.

In generale all’interno di un sorgente HTML non è necessario specificare dove deve andare a capo il documento. Più precisamente i caratteri di “a capo”¹ vengono ignorati dal browser, così come gli spazi successivi presenti fra una parola e l’altra.

Capita così che i seguenti diversi frammenti di codice vengano interpretati esattamente nella stessa maniera:

¹Detti “new line”.

```
<p>Il      mattino ha
      l'oro
    in bocca
    .</p>
```

```
<p>Il mattino ha l'oro in bocca.</p>
```

Il browser dispone il testo all'interno di una finestra a suo piacere, a seconda delle dimensioni della finestra (che possono cambiare da utente a utente) e della grandezza del carattere.

Un primo modo per andare a capo nel punto desiderato è concludere il paragrafo:

```
<p>Questa è una riga.</p>
<p>Questa è sicuramente la riga sotto.</p>
```

Questo tipo di soluzione funziona sicuramente. Tuttavia siamo stati costretti a scrivere due paragrafi, in alcuni casi la cosa può non essere desiderabile², per cui si vorrebbe riuscire ad andare a capo senza concludere il paragrafo. In questo caso si può usare il tab `
`

```
<p>Questa è una prima riga.<br>
  Questa è una seconda riga ma dello
  stesso paragrafo.</p>
<p>Questa è una terza riga, ma su
  un nuovo paragrafo.</p>
```

2.2 Controllare la separazione delle parole

Il browser, interpretando il contenuto HTML di una pagina, riesce sempre ad andare a capo al termine di una riga senza spezzare a metà una parola. Ciò può essere una comodità, tranne in quei casi in cui si vuole vincolare il browser a non spezzare automaticamente una coppia di parole che devono essere scritte attaccate:

```
<p>In tutto il mondo troverai sempre la Coca Cola</p>
```

In alcune circostanze (dipendenti dalla dimensione della finestra o del carattere) è possibile che il browser spezzi la frase in maniera indesiderata, ad esempio scrivendo *Coca* al termine di una riga e *Cola* all'inizio della riga successiva. Volendo imporre che ciò non debba accadere, una soluzione è quella di utilizzare il comando ` `; al posto dello spazio separatore fra le parole che devono rimanere unite:

```
<p>In tutto il mondo troverai sempre la Coca&nbsp;Cola</p>
```

²In alcuni browser esiste una certa spaziatura fra un paragrafo e il successivo.

2.3 Inserimento di caratteri speciali

Capita spesso di voler inserire in un documento dei caratteri speciali, ossia caratteri che non sono direttamente stampabili utilizzando la tastiera, oppure caratteri che comunque non fanno parte del set di caratteri riconoscibili dal browser.

Ad esempio i simboli di copyright o di marchio registrato si possono ottenere rispettivamente con i comandi *©* e *®*:

```
<p>Copyright &copy; 1999 W3C</p>
<p>MagiCo &reg;</p>
```

Altri simboli “speciali” riguardano ad esempio l'utilizzo delle lettere accentate:

- *à* *á*;
- *è* *é*;
- *ì* *í*;
- *ò* *ó*;
- *ù* *ú*;

2.4 Collegamenti a metà di una pagina

In alcune circostanze, anziché costruire un insieme di pagine HTML, si preferisce inserire tutto il documento (eventualmente diviso in vari capitoli) in un'unica pagina (evidentemente di dimensioni considerevoli) ed inserire un menù di collegamenti ai veri capitoli, all'inizio della pagina. I vari collegamenti permettono evidentemente di spostarci a vari punti della stessa pagina. Come si realizza un meccanismo di questo tipo?

I vari punti di destinazione dei collegamenti, presenti all'interno del documento, hanno bisogno di essere identificati tramite un'ancora: una situazione di un documento diviso in capitoli (all'interno di uno stesso file) potrebbe ad esempio essere il seguente.

```
...
<h1>Capitolo 1</h1>
...
<h1>Capitolo 2</h1>
...
<h1>Capitolo 3</h1>
...
```

Affinché questi diversi punti possano essere riferiti da dei collegamenti, hanno bisogno di essere marcati con un'ancora:

```

...
<h1><a name="cap1">Capitolo 1</a></h1>
...
<h1><a name="cap2">Capitolo 2</a></h1>
...
<h1><a name="cap3">Capitolo 3</a></h1>
...

```

Le tre etichette *cap1*, *cap2* e *cap3* permettono ora di creare direttamente dei collegamenti a questi punti del documento in questa maniera:

```

<ul>
  <li><a href="#cap1">Capitolo 1</a></li>
  <li><a href="#cap2">Capitolo 2</a></li>
  <li><a href="#cap3">Capitolo 3</a></li>
  ...
</ul>

```

Si è creato un elenco di voci, in cui ogni voce è un collegamento ad ogni capitolo del documento. La sintassi descritta dall'esempio consiste nell'anteporre il carattere di cancelletto (#) all'etichetta che indica il nome del collegamento.

Se una pagina contiene dei collegamenti interni, essi possono essere riferiti da un file esterno in questa maniera:

```

<a href="http://www.libro.it/#cap1">Capitolo 1 del libro</a>

```

2.5 Testo preformatto

Come si è visto fino ad ora, in HTML non è possibile formattare il testo a piacimento. E' il browser che decide come andare a capo o come separare le diverse parole del documento.

Un modo per impedire al browser di applicare le sue convenzioni di formattazione è utilizzare i tag `<pre>` e `</pre>`: tutto il testo compreso tra questi due tag viene copiato nel browser rispettando le occorrenze di spazi, tabulatori e linee vuote.

```

<pre>
<html>
<head>
  <title>Ciao</title>
</head>
<body>
</body>
</html>
</pre>

```

I due tag `<pre>` e `</pre>` possono naturalmente essere utilizzati anche per scrivere in HTML degli esempi di codice HTML.

2.6 Raggruppare il testo attorno alle immagini

Ciò che normalmente accade quando si inserisce un'immagine in un frammento di codice del tipo

```
<p>Il mattino ha l'oro in bocca.</p>
```

E' che l'immagine viene inserita nella stessa riga del testo. Ciò significa che l'effetto risultante può non piacere, specialmente quando l'immagine è molto alta.

Una maniera di allineare il testo a destra dell'immagine (che quindi rimane allineata a sinistra) è utilizzare l'attributo *align* del tag ``:

```
<p>Il mattino ha l'oro in bocca.</p>
```

Analogo effetto si ha, naturalmente, con *align="right"*.

Infine si noti l'utilizzo del tag `
` in questi due casi:

```
<p>

Il mattino ha l'oro in bocca<br>
Il mattino ha l'oro in bocca
</p>
```

```
<p>

Il mattino ha l'oro in bocca<br clear="all">
Il mattino ha l'oro in bocca
</p>
```

Nel secondo caso, la seconda riga di testo viene posizionata subito sotto l'immagine, anche se ci sarebbe spazio per scriverla accanto all'immagine.

2.7 Mappe cliccabili

E' possibile realizzare una mappa cliccabile inserendo un'immagine e definendo quali parti dell'immagine devono essere trasformate in collegamenti ipertestuali, e a che cosa.

Vediamo e commentiamo un esempio di codice:

```
<p align="center">
  
```

```

        usemap="#sitemap">
<map name="sitemap">
  <area shape="rect" coords="0, 0, 50, 50"
    href="uno.html" alt="Link a uno.html">
  <area shape="circle" coords="60, 60, 10"
    href="due.html" alt="Link a due.html">
  <area shape="poly" coords="70, 70, 80, 80, 90, 90"
    href="tre.html" alt="Link a tre.html">
</map>
</p>

```

L'immagine inserita contiene un riferimento (tramite l'attributo *usemap*) ad un frammento di codice contenente le informazioni sulla mappa. I due tag `<map>` e `</map>` contengono le informazioni sulle varie zone della mappa.

Ci sono tre distinte aree cliccabili, ognuna ha una forma, delle coordinate e un indirizzo di destinazione. Se la forma è un rettangolo (attributo con valore *rect*) le coordinate sono quelle del vertice in alto a sinistra e del vertice in basso a destra. Nel caso del cerchio (*circle*) si hanno le coordinate del centro e la grandezza del raggio. Infine, nel caso di un poligono generico (*poly*), si hanno le coordinate dei vari vertici.

Si noti che le coordinate si misurano in un sistema di riferimento con l'origine in alto a sinistra, con l'ascissa che cresce a destra e l'ordinata che cresce in basso.

2.7.1 Importanza dell'attributo *alt*

Nel caso delle mappe cliccabili è evidente come sia importante l'utilizzo dell'attributo *alt*. Se infatti per qualche motivo la mappa non potesse essere visualizzabile, senza tale attributo non sarebbe assolutamente utilizzabile.

2.8 Tabelle

Le tabelle vengono largamente utilizzate in HTML, in parte come tabelle vere e proprie, in parte per risolvere i problemi di layout dovuti ai limiti di espressività del linguaggio.

Una semplice tabella si può realizzare inserendo il seguente frammento di codice:

```

<table border="1">
  <tr><th>Lettere</th><th>Cifre</th></tr>
  <tr><td>Uno</td><td>1</td></tr>
  <tr><td>Due</td><td>2</td></tr>
  <tr><td>Tre</td><td>3</td></tr>
</table>

```

Le varie righe della tabella sono delineate dai tag `<tr>` e `</tr>` (*table row*). All'interno di ogni riga si possono poi inserire le colonne: `<th>` e `</th>` per le

celle di intestazione (*table heading*); `<td>` e `</td>` per le normali celle di dati (*table data*).

Si noti come la grandezza della tabella dipenda di fatto dal contenuto della stessa.

2.8.1 Cell Padding

L'attributo *cellpadding* di una tabella permette di importare ad un valore fisso la distanza del contenuto di una cella dalla stessa, facendo quindi in modo che la tabella non sia attaccata al suo contenuto:

```
<table border="1" cellpadding="10">
```

2.8.2 Cell Spacing

Diversamente dal precedente, *cellspacing* permette di impostare la distanza fra le celle di una stessa tabella:

```
<table border="1" cellpadding="10"
  cellspacing="10">
```

2.8.3 Table Width

Una tabella viene normalmente dimensionata in base alla grandezza del suo contenuto. Diversamente si può utilizzare l'attributo *width* per specificare esplicitamente la larghezza di una tabella.

```
<table border="1" width="80%">
```

La larghezza di una colonna può essere espressa in pixel, o in percentuale della larghezza della pagina.

2.8.4 Allineamento del testo all'interno delle celle

Di default il testo viene allineato al centro per le celle di intestazione (`<th>`) e a sinistra per le celle di dati (`<td>`). Naturalmente si possono modificare queste impostazioni di allineamento utilizzando l'attributo *align*, sia per quanto riguarda le righe che le singole celle. I valori ammissibili, naturalmente, *left*, *center* o *right*:

```
<tr align="center">
  <td>Uno</td><td>Due</td>
</tr>
```

Insieme ad *align* si può utilizzare anche *valign* per settare l'allineamento verticale all'interno di una cella o per tutta una riga. I valori ammissibili per *valign* sono *top*, *middle* e *bottom*. Di default le celle di intestazione utilizzano *middle*, mentre le altre utilizzano *top*.

2.8.5 Celle vuote

Una cella vuota non si realizza scrivendo `<td></td>`, ma utilizzando il seguente frammento di codice:

```
<td>&nbsp;</td>
```

2.8.6 Celle che occupano più di una riga e di una colonna

Si può fare in modo che una cella si espanda per più di una riga ed una colonna, riuscendo così a realizzare tabelle che non sono una semplice intersezione di righe e di colonne.

I due attributi *rowspan* e *colspan* specificano se una certa cella si espanda per più di una riga o più di una colonna rispettivamente. Ad esempio:

```
<table border="1" cellpadding="10" width="80%">
  <tr align="center"><th rowspan="2">A</th><th colspan="3">B</th></tr>
  <tr align="center"><th>C</th><th>D</th><th>E</th></tr>
  <tr align="center"><td>1</td><td>2</td><td>3</td><td>4</td></tr>
  <tr align="center"><td>5</td><td>6</td><td>7</td><td>8</td></tr>
</table>
```

2.8.7 Tabelle senza bordi

Non mettere un bordo ad una tabella significa creare lo stesso una tabella, ma invisibile. Di solito può essere utilizzata una tabella di questo tipo per risolvere problematiche di layout all'interno di una pagina.

2.8.8 Colori in una tabella

L'attributo *bgcolor* permette di settare il colore di sfondo di una cella:

```
<table border="0" cellspacing="0" cellpadding="10">
  <tr>
    <th bgcolor="#CCCC99">Years</th>
    <th bgcolor="#CCCC99">Sales</th>
  </tr>
  <tr>
    <td bgcolor="#FFFF66">2000</td>
    <td bgcolor="#FFFF66">$18M</td>
  </tr>
  <tr>
    <td bgcolor="#FFFF66">2001</td>
    <td bgcolor="#FFFF66">$25M</td>
  </tr>
  <tr>
    <td bgcolor="#FFFF66">2002</td>
```

```
<td bgcolor="#FFFF66">$36M</td>
</tr>
</table>
```