

Università degli Studi di Milano

**Corso di Laurea in
Sicurezza dei Sistemi e delle Reti Informatiche**

Lezioni 17 e 18 – Creazione di File HTML

FABIO SCOTTI

Laboratorio di programmazione per la sicurezza

Indice

1. RUDIMENTI DI HTML.....	3
1.1 Un semplice file HTML in 5 passi.	3
1.2 Cosa è il linguaggio HTML	4
1.3 Intestazione e corpo di un documento HTML	5
1.4 Inserire una immagine in un file HTML	5
1.5 Creare un link in una pagine HTML.....	7
1.6 Inserire delle liste ordinate e non ordinate	8
1.7 Inserire tabelle.....	9
2. CREARE FILE HTML CON IL C	13

1. Rudimenti di HTML

Obiettivo di questo paragrafo è di fornire le basi di HTML necessarie per il nostro corso. Iniziamo con vedere cosa è il linguaggio HTML, per cosa è nato e arriviamo a capire come è scritta una semplice pagina HTML contenente una tabella. Alcuni passi della dispensa sono stati realizzati in collaborazione con Roberto Sassi.

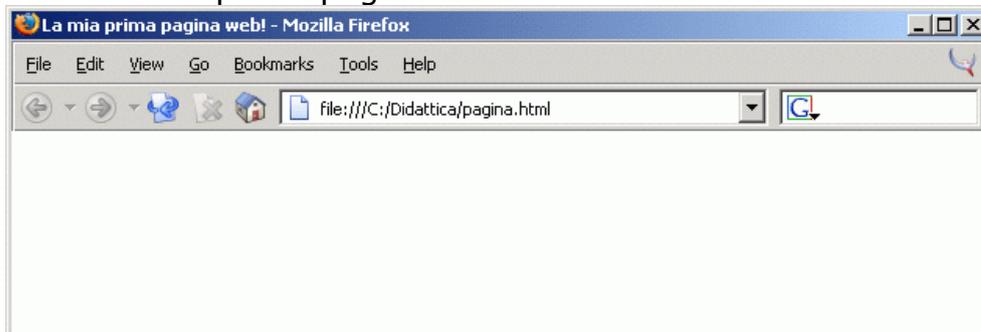
1.1 Un semplice file HTML in 5 passi.

Ecco i passi per creare un file HTML.

1. prendi un file di testo (pagina.txt)
2. scrivi le seguenti righe:

```
<HTML>
<HEAD>
<TITLE>La mia prima pagina web!</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

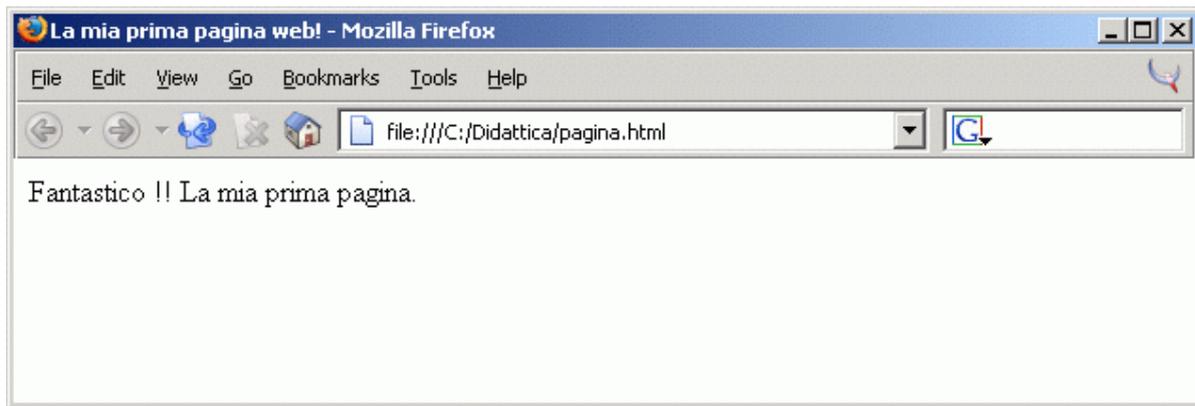
3. salva e rinomina con pagina.html
4. ecco fatta la prima pagina html. Visualizziamola con un browser:



5. scrivi dentro qualche cosa nel testo

```
<HTML>
<HEAD>
<TITLE>La mia prima pagina web!</TITLE>
</HEAD>
<BODY>
Fantastico !!
La mia prima pagina.
</BODY>
</HTML>
```

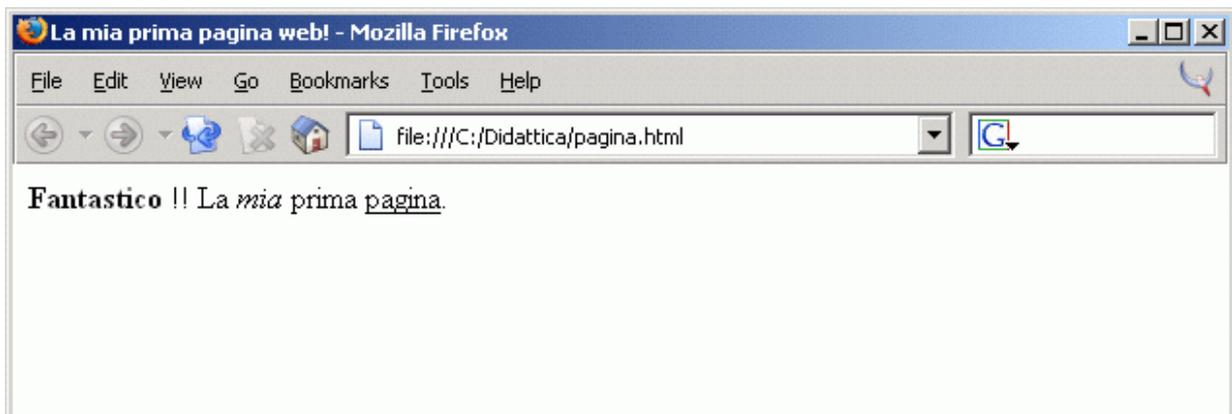
Ed ecco visualizzata la pagina



Possiamo ad esempio scrivere in grassetto, corsivo e sottolineato.

```
...  
<BODY>  
<b>Fantastico</b> !!  
La <i>mia</i> prima <u>pagina</u>.  
</BODY>  
...
```

Quello che stiamo dicendo al browser è di visualizzare in grassetto dal punto in cui mettiamo il tag `` fino al punto in cui inseriamo ``.



1.2 Cosa è il linguaggio HTML

HTML significa Hyper Text Mark-up Language. Non è un linguaggio di programmazione (come il C) né di scripting (come può essere il PHP). L'HTML non ha costrutti per il controllo di flusso (come `if-else`, `while`, `for` in C). E' solo un linguaggio di marcatura.

Un esempio elementare di marcatura è il modo in cui il progetto "Manuzio" dell'associazione "Liber Liber" aveva, attorno all'inizio degli anni '90, codificato il capitolo dell'[Inferno](#) della Divina Commedia di Dante Alighieri:

```
corsivo: ~abcdef~  
sottolineato: _abcdef_  
neretto: **abcdef**
```

Prova ad aprire un file di testo con il browser: lo visualizzerà senza problemi- Quindi, quando il browser trova i tag del linguaggio di marcatura (es:<HTML>) capisce come formattare i testi e quindi come procedere alla loro visualizzazione.

1.3 Intestazione e corpo di un documento HTML

Un documento HTML si divide in due parti fondamentali: *l'intestazione* e il *corpo* del documento.

Il corpo del documento contiene tutti gli elementi della pagina: il testo, le immagini, le applet Java, il codice Javascript e quant'altro viene materialmente visualizzato dal browser.

Al contrario, l'intestazione contiene una serie di informazioni necessarie al browser per una corretta interpretazione del documento, ma non visualizzate all'interno dello stesso. L'intestazione, quindi, ha un ruolo non apparente ma sicuramente fondamentale.

Nella semplice pagina che abbiamo creato tutto il codice viene ad essere racchiuso fra i tag <HTML> e </HTML>

L'intestazione è racchiusa fra i tag <HEAD> e </HEAD> e subito dopo inizia la parte del corpo racchiusa fra i tag <BODY> e </BODY>. Non è possibile invertire il loro ordine.

```
<HTML>
<HEAD>
<TITLE>La mia prima pagina web!</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

1.4 Inserire una immagine in un file HTML

Inserire **immagini** è veramente semplice. Dobbiamo specificare il percorso (dove si trova fisicamente l'immagine sul nostro disco) e la grandezza che desideriamo che l'immagine abbia nella pagina. Ad esempio:

```
...
<body>
Ecco un immagine:

<br />
La stessa dimezzata:

</body>
...
```



Nell'esempio precedente, `src="./immagini/cow.jpg"` significa che il browser cercherà l'immagine nella cartella "immagini" contenuta nella stessa cartella in cui si trova il documento html.

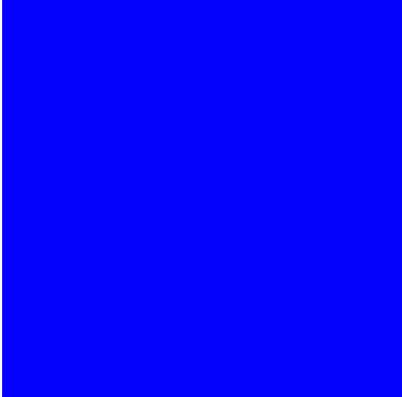
Come puoi vedere, anche se non sono state specificate le dimensioni, la prima immagine viene visualizzata correttamente. Inserire le dimensioni dell'immagine è importante perché in questo modo il browser predispose già lo spazio necessario ad ospitarla e mentre la pagina viene scaricata (ed inizia ad essere visualizzata).

Puoi cambiare le dimensioni dell'immagine indicandone di nuove. Da una piccola immagine di colore uniforme come questo piccolo quadrato -> .-< (un quadrato di 1 pixel per lato) si possono ottenere...

```
...  
<body>  
  
</body>  
...
```

```
...  
<body>  
  
</body>  
...
```

```
...  
<body>  
  
</body>  
...
```



La possibilità di ridimensionare efficacemente le immagini potrebbe essere utile ad esempio per creare istogrammi.

Che tipo di immagini includere? Come regola generale, se dobbiamo includere una fotografia è meglio utilizzare il formato JPEG. Se invece l'immagine è un grafico e contiene molte linee e pochi colori sono meglio i formati GIF o PNG (dimensioni inferiori a parità di qualità).

1.5 Creare un link in una pagine HTML

Creare un **link** è estremamente semplice (in fondo l'HTML è nato proprio per creare documenti ipertestuali):

```
...  
<body>  
Ecco una pagina con un link alla home page del motore di  
ricerca <a href="http://www.google.it">Google</a>.  
</body>  
...
```

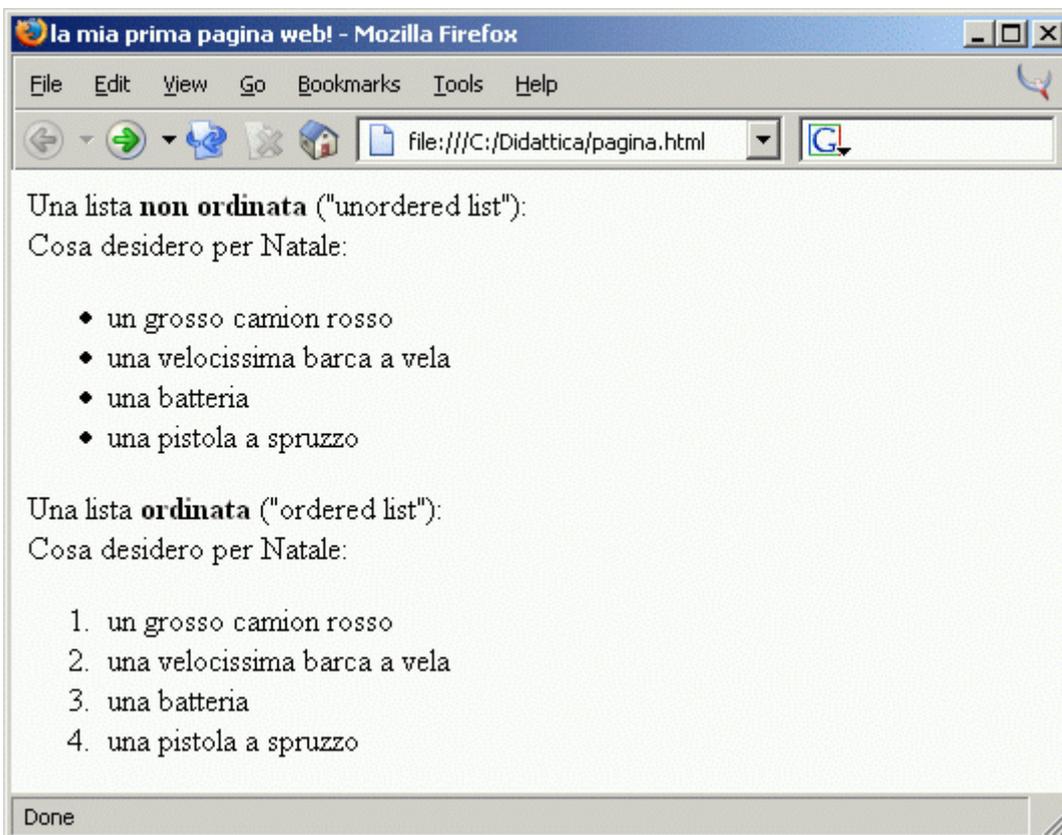


Il tag `<a>` è un "àncora" che pongo nel documento e a cui aggancio il collegamento. Cliccando sul collegamento il browser tenterà di caricare il file indicato dall'URL specificato con `href="URL"`.

1.6 Inserire delle liste ordinate e non ordinate

Al pari di tutti gli altri elementi che compongono una pagina HTML anche la lista è aperta e chiusa da un proprio tag, così come i suoi elementi (item). Ecco un esempio:

```
...  
<body>  
Una lista <b>non ordinata</b> ("unordered list"):<br />  
Cosa desidero per Natale:  
<ul>  
<li> un grosso camion rosso </li>  
<li> una velocissima barca a vela </li>  
<li> una batteria </li>  
<li> una pistola a spruzzo </li>  
</ul>  
Una lista <b>ordinata</b> ("ordered list"): <br />  
Cosa desidero per Natale:  
<ol>  
<li> un grosso camion rosso </li>  
<li> una velocissima barca a vela </li>  
<li> una batteria </li>  
<li> una pistola a spruzzo </li>  
</ol>  
</body>  
...
```

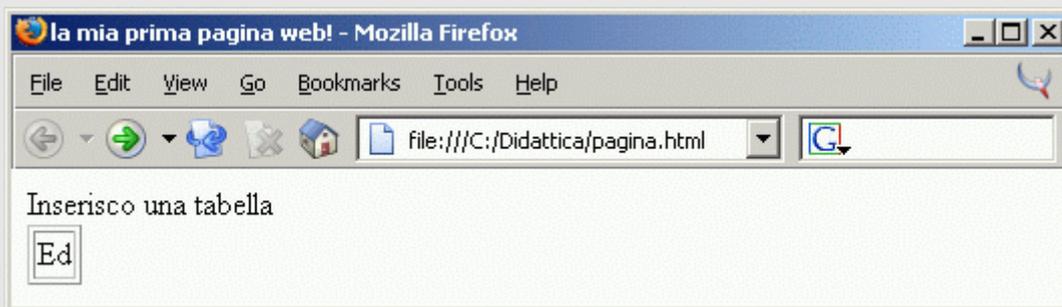


1.7 Inserire tabelle

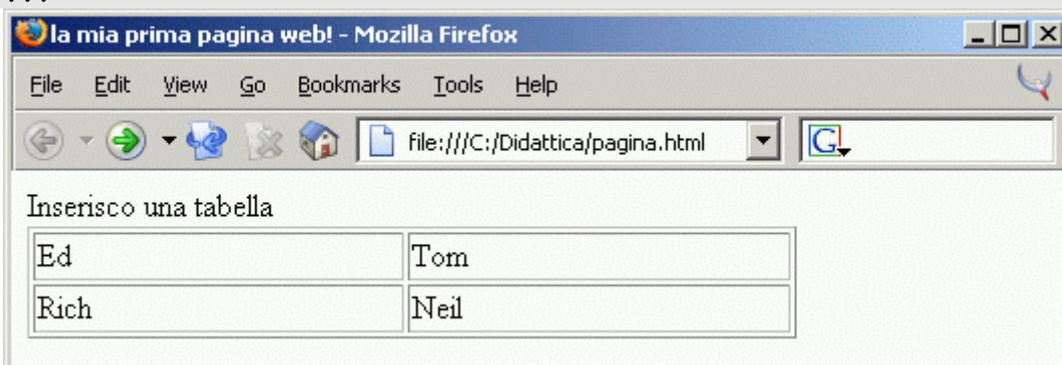
Esattamente come tutti gli elementi di una pagina HTML, anche le tabelle sono introdotte da uno specifico tag, come anche una loro riga ed un loro elemento (cella).

Ogni tabella ha bisogno di perlomeno UNA riga. Ogni riga necessita di perlomeno una cella di dati.

```
...  
<body>  
Inserisco una tabella  
<table border=1>  
<tr>  
<td>Ed</td>  
</tr>  
</table>  
</body>  
...
```

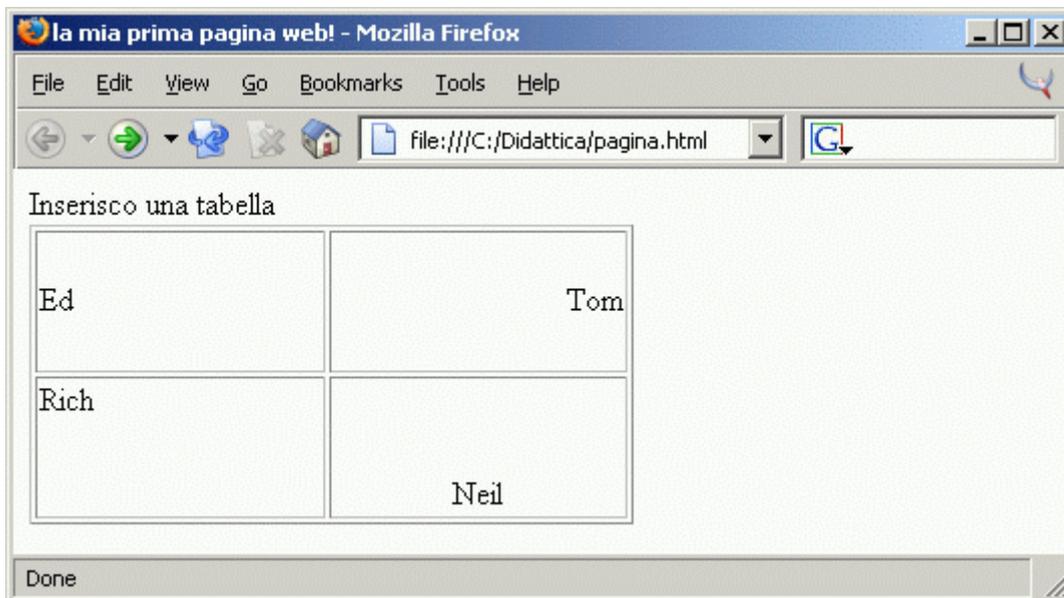


```
...  
<body>  
Inserisco una tabella  
<table border=1 width=75%>  
<tr>  
<td>Ed</td>  
<td>Tom</td>  
</tr>  
<tr>  
<td>Rich</td>  
<td>Neil</td>  
</tr>  
</table>  
</body>  
...
```



Se vogliamo allineare gli elementi di una cella esiste un apposito attributo del tag.

```
...  
<body>  
Inserisco una tabella  
<table border="1" width="300" height="150">  
<tr>  
<td align="left">Ed</td>  
<td align="right">Tom</td>  
</tr>  
<tr>  
<td valign="top">Rich</td>  
<td align="center" valign="bottom">Neil</td>  
</tr>  
</table>  
</body>  
...
```

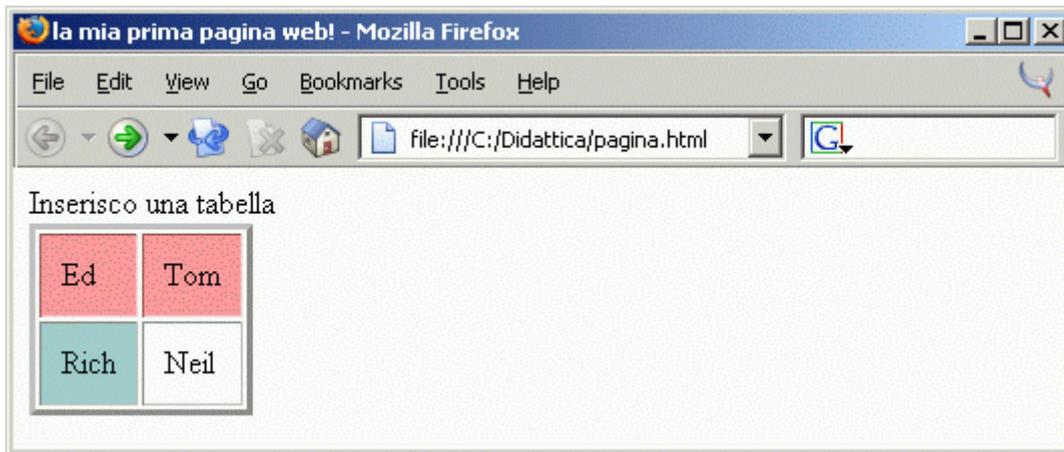


Anche modificare l'indentazione degli elementi è possibile. Si usa l'attributo chiamato **cellpadding** ovvero lo spazio che c'è fra il bordo della cella e il contenuto della cella stessa. **bgcolor** definisce il colore di sfondo di una cella. Per il colore viene utilizzata la codifica RGB ossia Rosso Verde Blu (Red Green Blue). Erroneamente alcune persone "italianizzano" il tutto e pensano che RGB corrisponda a Rosso, Giallo, Blu. Attenzione quindi a non compiere questo errore.

Se un'immagine ha il "colore RGB" significa che l'immagine stessa è stata creata dalla miscelazione di questi 3 colori, che uniti danno il bianco. Ogni colore primario viene espresso mediante un numero da 0 a 255 in base esadecimale (da x00 a xFF). Pertanto #FFFFFF indica il bianco, #FF0000 il rosso, #00FF00 il verde, #0000FF il blu e #000000 il nero, inteso come "assenza" di colore.

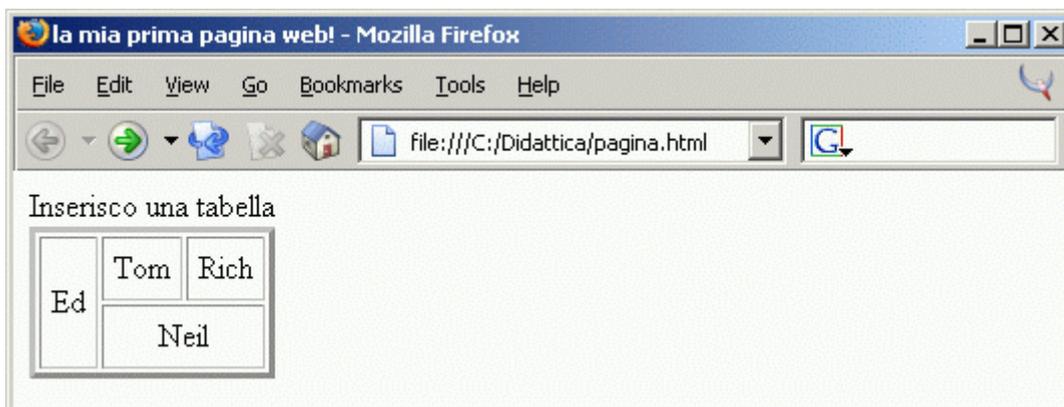
```
...  
<body>  
Inserisco una tabella  
<table border="3" cellpadding="10">  
<tr bgcolor="#FF9999">  
<td>Ed</td>  
<td>Tom</td>  
</tr>
```

```
<tr>
<td bgcolor="#99CCCC">Rich</td>
<td >Neil</td>
</tr>
</table>
</body>
...
```



E' possibile avere un campo che spazia per più di una riga o per più di una colonna

```
<html>
<head>
<title>la mia prima pagina web!</title>
</head>
<body>
Inserisco una tabella
<table border="3" cellpadding="5">
<tr >
<td rowspan="2">Ed</td>
<td>Tom</td>
<td>Rich</td>
</tr>
<tr>
<td colspan="2" align="center">Neil</td>
</tr>
</table>
</body>
</html>
```



Le tabelle sono molto utilizzate (anche se la tendenza è quella di spostarsi verso l'uso dei CSS) per disporre testo e immagini in una pagina.

```
<html>
<head>
<title>Allevamento in Italia</title>
</head>
<body>
<table border="0" width="100%" cellpadding="10">
<tr>

<td width="40%" align="right">

</td>

<td width="60%" valign="top">
In Italia ci sono:
<ul>
<li> 7,1 milioni di unità di bovini </li>
<li> 8 milioni di unità di suini </li>
<li> 8 milioni di unità di ovini </li>
<li> 1 milione di unità di caprini </li>
<li> 600 milioni di unità di polli </li>
</ul>
</td>

</tr>
</table>
</body>
</html>
```



2. Creare file HTML con il C

Creare un file HTML in C significa creare un normale file di testo contenente gli elementi necessari perché possa essere un file HTML. Limitando il file di testo da creare proprio all'essenziale troveremo almeno questo codice:

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

La corretta estensione del file di testo che il programma creerà dovrà essere `.html` o `.htm`. Questo affinché un browser possa aprirlo e visualizzarlo interpretando i tag che vi sono all'interno.

L'idea più semplice per scrivere del codice C che crea dei file HTML consiste nel creare delle etichette nel nostro programma contenenti le parti di testo HTML che devono essere presenti e poi stamparle su file con delle `fprintf`. Di seguito vengono proposti alcuni esempi di definizioni di etichette (`#define`) utili per stampare un file HTML del tipo:

```
<html>
<head>
<title>Creato da un mio programma C!</title>
</head>
<body bgcolor="#FFFFFF">
creato da un mio programma C,<br />
<b>Magnifico</b><br />
scrivo altre cose ..... chiudo il body ed il tag html
</body>
</html>
```

```
#define MIO_TITOLO "Creato da un mio programma C!"

// ATTENZIONE: Per spezzare il define su più di una riga dobbiamo
// utilizzare il carattere \
// Perché spezzare su più righe la costante stringa?
// Solo perché è molto più facile da leggere!
#define HTML_TOP "<html>\n<head>\n<title>" \
    MIO_TITOLO \
    "</title>\n</head>\n"
#define BODY_START "<body bgcolor=\"#FFFFFF\">\n"
#define HTML_BOTTOM "</body>\n</html>"
```

L'uso del C diviene più interessante quando occorre ripetere molte volte una porzione di codice HTML, come avviene nel caso di tabelle con molte righe. In questo caso la parte di tabella che viene ripetuta può essere stampata da un ciclo.

Questo procedimento è molto simile a quello utilizzato dai server che creano pagine attive su richiesta dei browser ad essi collegati. Il numero ed il tipo di elementi che comporranno la pagina HTML finale che verrà spedita al browser dell'utente dipende da condizioni e dati spediti al server dal browser. Ad esempio, un utente attraverso il browser clicca sul link di una pagina degli orari ferroviari delle prossime 2 ore. Il server accede ad un database,

crea una tabella con tutti i treni delle prossime 2 ore e manda il file HTML così creato al browser che lo avevo richiesto.

Come potrebbe essere composto il codice di una tabella con 100 righe contenenti un numero ed un nome?

-- 1--	Tom
-- 2 --	Neil
...	...
-- 100 --	Tom

Ecco un esempio:

```
<table border=1 width=75%>
<tr>
<td>-- 1 --</td>
<td>Tom</td>
</tr>
<tr>
<td>-- 2 --</td>
<td>Neil</td>
</tr>
...
<tr>
<td>-- 100 --</td>
<td>Tom</td>
</tr>
</table>
```

Le parti che sono costanti sono

```
<table border=1 width=75%>
```

e

```
</table>
```

La parte che deve essere ripetuta è

```
<tr>
<td>-- Numero --</td>
<td>nome</td>
</tr>
```

Il codice C che genera automaticamente la tabella di 100 righe e 2 colonne potrebbe essere simile al seguente:

```
fprintf(Fp1, "%s\n", "<table border=1 width=75%>");
for (i=0; i<100; i++)
{
    fprintf(Fp1, "<tr>\n");
    fprintf(Fp1, "<td>-- %d --</td>\n", i );
    fprintf(Fp1, "<td>-- %s --</td>\n", Nomi[i] );
    fprintf(Fp1, "</tr>\n\n");
}
fprintf(Fp1, "%s\n", "</table>\n");
```