

MANUALE DI PROGRAMMAZIONE COL LINGUAGGIO

HTML

Data ultimo rilascio: 4 maggio 2010

Indirizzo dal quale è possibile scaricare questo manuale: <http://www.menduni.info>

Premessa

Manuale di programmazione col linguaggio HTML
<http://www.menduni.info>

Questo manuale è stato concepito per essere di ausilio a chi si avvicina per la prima volta al linguaggio HTML. Contiene gli elementi basilari per poter realizzare pagine web statiche. Non è assolutamente esaustivo di tutti gli argomenti. Per un maggiore approfondimento si rimanda a manuali più completi presenti sul mercato in forma cartacea oppure on line sui siti di settore. Tra queste pagine, quindi, saranno introdotti tutti gli elementi del linguaggio in maniera essenziale e concisa. Il consiglio che si fornisce al lettore è quello di esercitarsi nella creazione di documenti per mettere in pratica le conoscenze acquisite attraverso questo manuale e di integrare con altro materiale in base alle proprie esigenze e necessità. Si richiama infine al senso civico, consigliando di non stampare questo documento se non è necessario nel pieno rispetto dell'ambiente che ci ospita.

Il linguaggio HTML

Il nome è la contrazione di HyperText Markup Language (linguaggio di marcatura per ipertesti) ed è stato ideato per consentire la realizzazione di documenti ipertestuali da poter diffondere sulla rete internet. La caratteristica fondamentale di questo linguaggio è l'utilizzo dei "marcatori" (che più avanti chiameremo *tag*), ossia dei delimitatori entro i quali si inseriscono parti del documento. La sua sintassi è relativamente semplice se paragonata ai linguaggi di programmazione di alto livello.

Cosa sono i Tag

All'interno di ogni pagina sono presenti i marcatori (TAG) a cui viene affidata la visualizzazione. Hanno differenti nomi a seconda della loro funzione. Vanno inseriti tra parentesi angolari <TAG> e la chiusura del tag viene indicata con una "/" cioè : </TAG>. Esistono alcuni tag che non hanno contenuto come il tag per le immagini, per andare a capo e quindi non hanno neanche bisogno di una chiusura. I tag possono essere annidati uno dentro l'altro. Ogni tag, infine, può contenere degli attributi relativi all'elemento HTML che delimitano e la sintassi è la seguente:

```
<TAG parametro1="valore1" parametro2="valore2"> parte di documento </TAG>
```

Si vedrà poi nel dettaglio quali e quanti sono gli attributi (parametri) che si possono associare ad ogni tag.

La struttura della pagina HTML

Una pagina web è sempre racchiusa tra i tag <HTML> aperto e chiuso ed è normalmente divisa in due parti: la testata ed il corpo. La testata fornisce tutta una serie di informazioni relative alla pagina (il titolo, il nome dell'autore, gli stili utilizzati nella pagina), mentre il corpo contiene l'effettivo contenuto del documento ossia ciò che poi viene effettivamente visualizzato.

La testata è delimitata dei tag <HEAD> aperto e chiuso, mentre il corpo è racchiuso tra i tag <BODY> aperto e chiuso.

```
<HTML>
  <HEAD>
    Informazioni relative alla pagina
  </HEAD>
  <BODY>
    Contenuto della pagina
  </BODY>
</HTML>
```

I tag della testata

All'interno della testata (HEAD) va inserito il tag "<TITLE>" aperto, "</TITLE>" chiuso. Il title è il titolo della pagina e compare in alto sulla barra del browser. È bene compilarlo da subito, onde evitare poi di avere pagine senza titolo.

```
<HTML>
  <HEAD>
    <TITLE> html.it </TITLE>
    <STYLE></STYLE>
  </HEAD>
  <BODY>
    Contenuto della pagina
  </BODY>
</HTML>
```



I meta tag

I meta tag servono per fornire informazioni relative al documento. Queste non sono visualizzate dal browser, ma consentono di fornire indicazioni ai motori di ricerca per una opportuna catalogazione della pagina.

Gli attributi principali dei meta tag sono i seguenti:

- **name** = identifica un nome di proprietà. Queste specifiche non elencano valori legali per tale attributo.
- **content** = specifica un valore di proprietà. Le presenti specifiche non elencano valori legali per questo attributo.
- **http-equiv** = può essere usato al posto dell'attributo name. I server HTTP usano questo attributo per raccogliere informazioni per le intestazioni dei messaggi di risposta http.

Vediamo un esempio di come si inseriscono i meta tag nella sezione HEAD del documento.

```
<HTML>
  <HEAD>
    <TITLE> html.it </TITLE>
    <META http-equiv="Expires" content="Tue, 20 Aug 1996 14:25:27 GMT">
    <META name="description" content="viaggi in Italia organizzati">
    <META name="keywords" lang="it" content="vacanze, Italia, viaggio">
    <META name="keywords" lang="en-us" content="vacation, Italy, journey">
    <META http-equiv="Content-Type" content="text/html; charset=ISO-8859-5">
  </HEAD>
  <BODY>
    Contenuto della pagina
  </BODY>
</HTML>
```

In questo esempio troviamo il meta tag “**description**” che fornisce indicazioni sul contenuto del documento, quello “**keywords**” che contiene le parole chiave per una ricerca su un motore quando la pagina viene indicizzata.

I primi Tag che si utilizzano nella sezione BODY

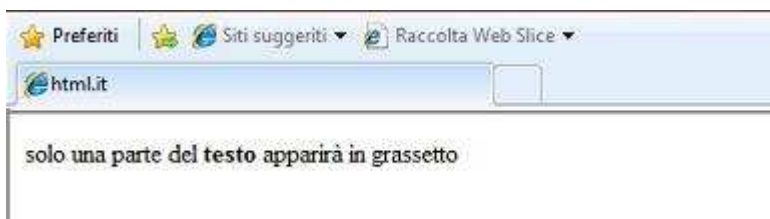
I tag più semplici sono quelli che si applicano al testo libero per trasformare il testo normale in corsivo, grassetto o sottolineato. Questi funzionano e si utilizzano solo nella sezione BODY della pagina HTML, quindi agiscono solo sulla parte visibile della stessa.

Il tag

Con si delimita un testo che si vuole rendere in grassetto, quindi se scriviamo il seguente codice nella parte BODY

```
solo una parte del <B>testo</B> apparirà in grassetto.
```

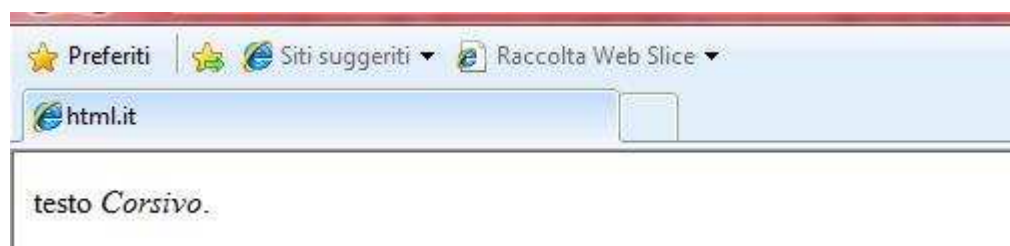
avremo il come risultato una stringa di questo tipo



Il tag <I>

In maniera del tutto simile al tag si utilizza il tag <I> per visualizzare il testo racchiuso tra l'apertura e la chiusura in corsivo (italic).

```
testo <I>Corsivo</I>.
```



Il tag <U>

Alla stessa stregua funziona il tag `<U>` che si usa per il sottolineato (underline).

```
testo <U>Sottolineato</U>.
```



Il tag `<CENTER>`

Se vogliamo centrare un testo o un qualsiasi altro oggetto all'interno della pagina, dobbiamo utilizzare questo tag, racchiudendo ciò che vogliamo centrare tra `<CENTER>` e `</CENTER>`.

```
testo<BR>
```

```
<CENTER>Centrato</CENTER>
```



• Il tag `
`

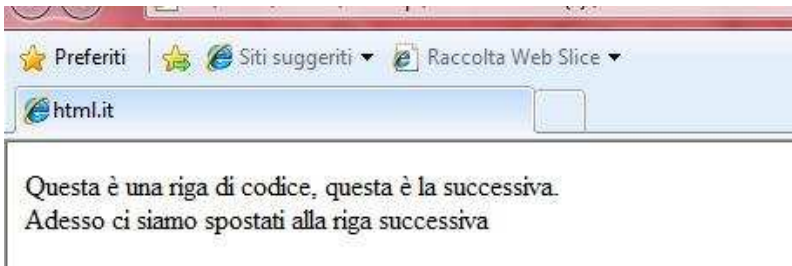
Questo tag si usa alla fine di una riga di testo, quando vogliamo andare a capo. Si tenga presente che indipendentemente da come scriviamo il testo all'interno di una pagina HTML, il browser andrà a capo solo quando viene indicato con `
`, facciamo un esempio:

```
Questa è una riga di codice,
```

```
questa è la successiva.<BR>
```

```
Adesso ci siamo spostati alla riga successiva.
```

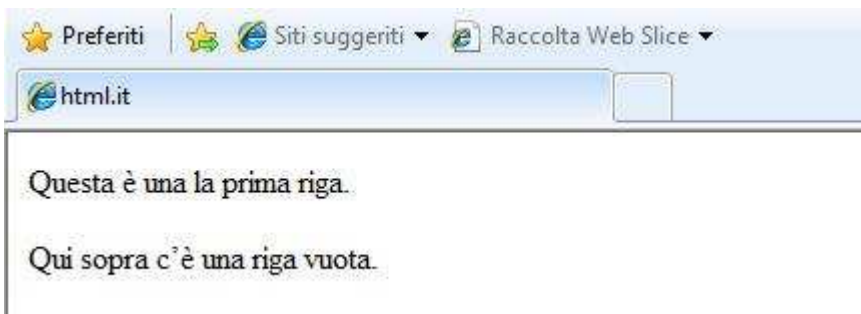
sarà rappresentato così



Quindi anche se le prime due righe sono separate all'interno della pagina HTML codificata, in fase di visualizzazione il browser terrà conto solo del tag `
` per andare ad una nuova riga. Aggiungiamo che un solo tag `
` porta il cursore alla riga successiva, ma se vogliamo lasciare una riga vuota tra due parti di testo dobbiamo usarne due, come nell'esempio qui sotto.

```
Questa è una la prima riga.<BR><BR>
Qui sopra c'è una riga vuota.
```

sarà rappresentato così



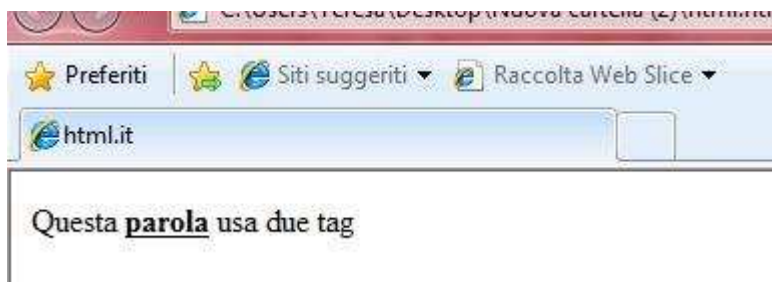
Il tag `
` è uno dei casi di tag che non hanno la chiusura. Quindi non esisterà mai un tag `</BR>`.

Nidificazione dei tag

Possiamo utilizzare più tag contemporaneamente. Supponiamo di voler scrivere un testo sia in grassetto che sottolineato, dobbiamo usare sia che <U>, facendo attenzione ad una regola fondamentale: i tag vanno inseriti uno dentro l'altro, ossia **nidificati**, quindi l'ultimo tag che abbiamo aperto è il primo che dobbiamo chiudere, come riportato nell'esempio successivo

Questa <U>parola</U> usa due tag.

che apparirà così



mentre la formulazione

Questa <U>parola</U> usa due tag.

è **errata** e può generare risultati imprevisti. Quindi bisogna sempre ricordarsi che tutti i tag vanno **innestati l'uno dentro l'altro**.

Delimitare porzioni di documento

Al fine di rendere il documento più leggibile e ordinare il suo contenuto, il linguaggio HTML prevede una serie di tag che perseguono questa finalità.

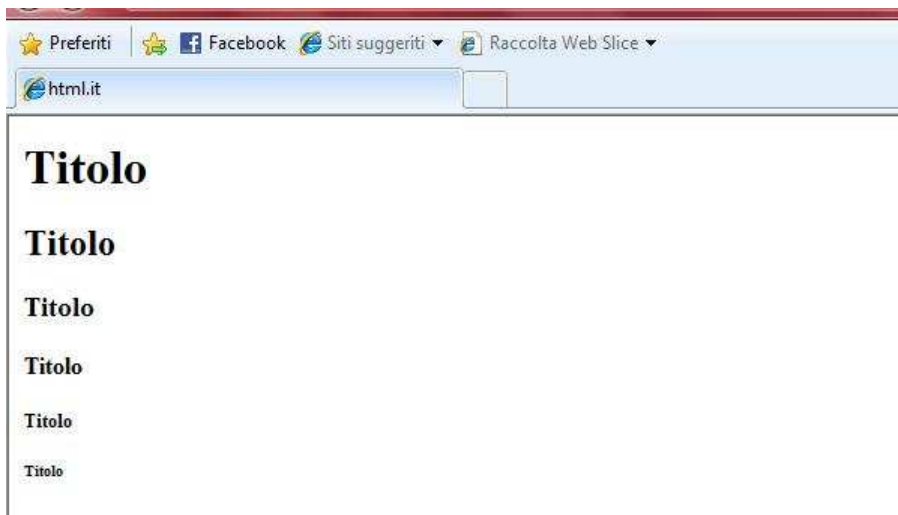
Il tag <Hn>

Per racchiudere parti di testo che fanno da intestazione, si usa il tag <Hn> dove la “n” è un valore compreso tra 1 e 6. Ognuna di queste intestazioni visualizza intestazioni di grandezza diversa. Dall’<H1>, che è il più importante, si va via via degradando fino all’<H6>. Il tag <Hx>(sia esso H1 o H6) risulta formattato in grassetto e lascia una riga vuota prima e dopo di se. Vediamo l’esempio:

<HTML>

```
<HEAD>
  <TITLE> html.it </TITLE>
</HEAD>
<BODY>
  <H1>Titolo </H1>
  <H2>Titolo </H2>
  <H3>Titolo </H3>
  <H4>Titolo </H4>
  <H5>Titolo </H5>
  <H6>Titolo </H6>
</BODY>
</HTML>
```

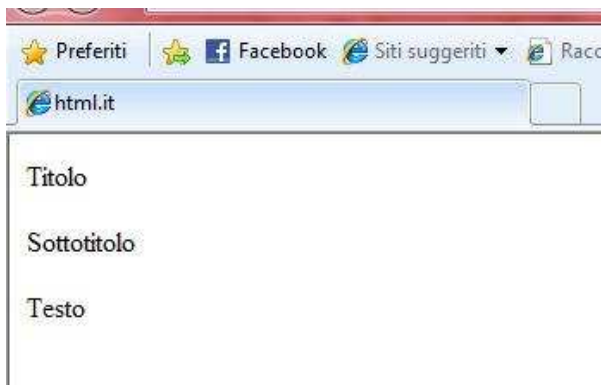
che produrrà il seguente risultato



Il tag <P>

Il paragrafo è l'unità di base entro cui suddividere un testo. Il tag <p> lascia una riga vuota prima della sua apertura e dopo la sua chiusura. Esempio:

```
<HTML>
  <HEAD>
    <TITLE> html.it </TITLE>
  </HEAD>
  <BODY>
    Titolo <p>Sottotitolo</p> Testo
  </BODY>
</HTML>
```

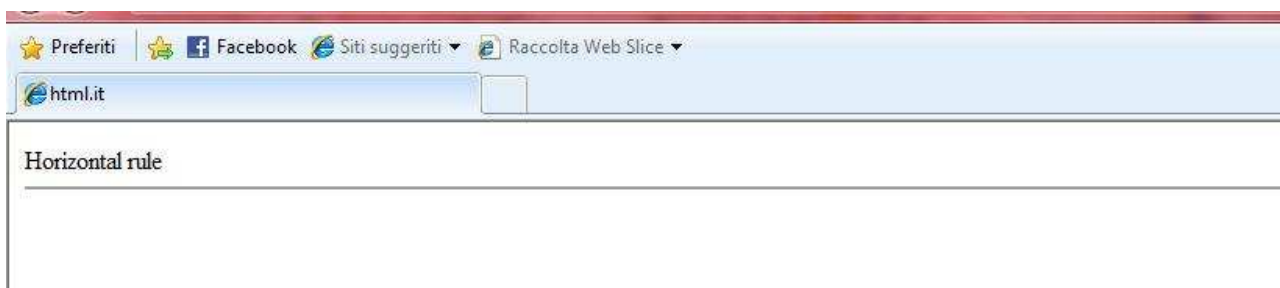


Il tag <HR>

Un altro valido tag per dividere la pagina in parti è il tag <hr> (“Horizontal rule”), che serve per tracciare una linea orizzontale. Questo tag ha anche alcuni attributi: l'attributo “noshade” che evita di sfumare la linea; “size” che indica l'altezza in pixel; “width” è la larghezza in pixel o in percentuale; “align” l'allineamento.

Esempio:

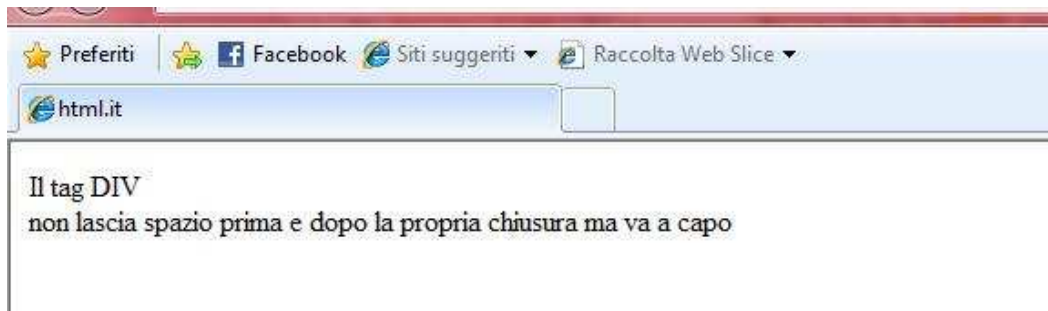
```
Horizontal rule
<Hr noshade size= "5" width= "50%" align= "center">
```



Il tag <DIV>

Il blocco di testo va a capo ma a differenza del paragrafo non lascia spazi prima e dopo la sua apertura. Il risultato sarà:

Il tag DIV <DIV> non lascia spazio prima e dopo la propria chiusura ma va a capo</DIV>



Il tag

La scelta del colore, delle dimensioni e del tipo di carattere del testo si effettuano attraverso l'utilizzo del tag **font**.

Per scegliere il tipo di carattere si deve utilizzare la seguente sintassi:

```
<FONT face= "Arial">Testo in Arial</FONT>
```

```
<FONT face= "Verdana">Testo in Verdana</FONT>
```

Per scegliere il colore del testo si deve utilizzare la seguente sintassi:

```
<FONT color= "red">Testo di colore Rosso</FONT>
```

```
<FONT color= "#ff0000">Testo di colore Rosso</FONT>
```

Per scegliere le dimensioni del testo (valori interni da 1 a 7 o valori relativi di default "3") si deve utilizzare la seguente sintassi:

```
<FONT size= "1"> Testo di grandezza 1</FONT>
```

```
<FONT size= "7"> Testo di grandezza 7</FONT>
```

Le tabelle

Le tabelle sono una delle parti più importanti di tutto il codice HTML. I tag necessari per creare una tabella sono:

`<TABLE>` apre la tabella

`<TR>` "table row": apre una riga

`<TD>` "table data": indica una cella all'interno di una riga`</TD>`

`</TR>`

`</TABLE>`

L'attributo **border** permette di specificare di quanti pixel deve essere il bordo delle tabelle. Per i bordi esiste anche l'attributo **bordercolor** che permette di scegliere il colore del bordo. L'attributo **width** permette di specificare la larghezza delle tabelle, l'attributo **height** specifica l'altezza della tabella. Il valore di questi attributi può essere specificato con una larghezza fissa in pixel (numero intero) o in percentuale (il numero deve essere seguito dal simbolo "%", in questo caso la tabella si adatterà allo spazio a disposizione). Mentre **cellpadding** indica la distanza tra il contenuto della cella e il bordo, **cellspacing** indica la distanza tra una cella e l'altra (la dimensione indicata, una volta specificata ha effetto su tutti i lati della cella). L'attributo **align** indica l'allineamento che può assumere i valori *center* (centrato), *left* (sinistra), *right* (destra). Allo stesso modo **valign** è utile per l'allineamento verticale delle celle, i valori possibili sono *top* (alto), *Middle* (in mezzo), *bottom* (in basso), *baseline* (alla linea di base) ad esempio:

```
<BODY>
```

```
<TABLE border= "1" bordercolor="blue" width="400px" height="20px"  
        cellspacing="1" cellpadding="5" align="left" valign="middle">
```

```
<TR>
```

```
<TD>prima cella</TD>
```

```
<TD>seconda cella</TD>
```

```
</TR>
```

```
<TR>
```

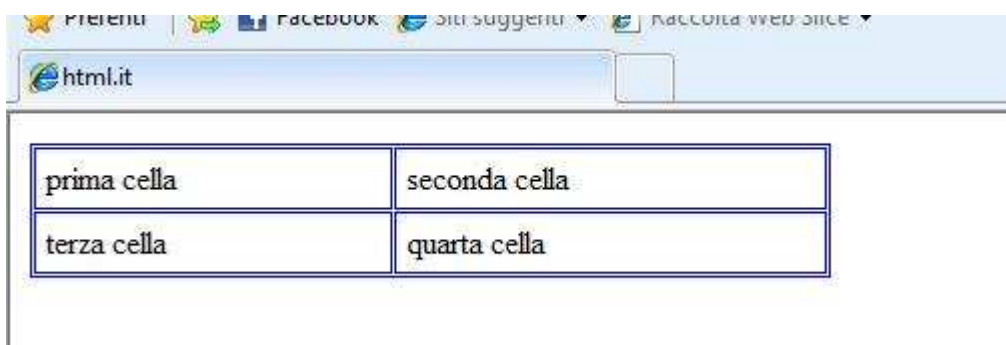
```
<TD>terza cella</TD>
```

```
<TD>quarta cella</TD>
```

```
</TR>
```

```
</TABLE>
```

```
</BODY>
```



La nidificazione delle tabelle

All'interno delle celle `<TD></TD>` possiamo inserire oltre al testo anche altri elementi HTML, anche delle tabelle. In questo caso bisogna fare molta attenzione a come queste ultime vengono nidificate. Vediamo un esempio:

```
<TABLE border= "1" bordercolor="#000000" bgcolor="#ffff00"
  cellspacing="10" cellpadding="10">
  <TR>
    <TD>Testata</TD>
  </TR>
  <TR>
    <TD>
      <TABLE border= "2" bordercolor="#0000ff" bgcolor="#ff0000">
        <TR>
          <TD bgcolor="#ffffff" align="center" valign="middle">Destra</TD>
          <TD>Sinistra</TD>
        </TR>
      </TABLE>
    </TD>
  </TR>
</TABLE>
```

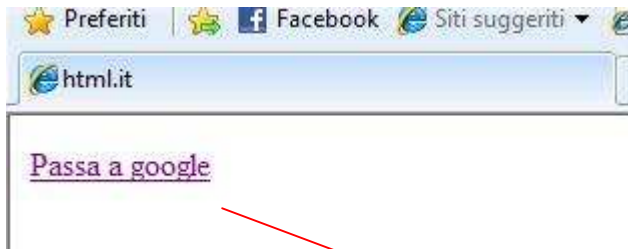
L'attributo **bgcolor** consente di selezionare il colore di sfondo della tabella, della riga o della cella. E' possibile specificare colori di sfondo differente per ognuno di questi elementi anche all'interno della stessa tabella. Quindi è possibile definire una tabella che ha uno sfondo rosso, con la prima riga con lo sfondo giallo e la prima cella della stessa riga con sfondo blu.

Per la cella o la riga (`<TD>` o `<TR>`) è possibile definire l'allineamento orizzontale e verticale del contenuto delle celle. L'attributo **align** può assumere i valori "left", "center" o "middle" se vogliamo un allineamento del contenuto a sinistra, al centro o a destra. Parimenti possiamo definire un allineamento orizzontale mediante l'attributo **valign** che può avere i valori "top", "middle" o "bottom" per ottenere un allineamento verticale in alto nella cella, al centro o in basso.

I link col tag <A>

I link consentono il passaggio da un documento all'altro o posizionano il browser su un'altra porzione dello stesso documento. Il parametro più importante è HREF che indica l'indirizzo del nuovo documento.

```
<A href="http://www.google.it/">Passa a google</A>
```



Il click su questo link provocherà l'apertura della nuova pagina



All'elemento <A> è possibile associare l'attributo **target** che ci indica dove deve essere caricata la pagina alla quale si fa riferimento con il link. Tra i valori che può assumere c'è "**_blank**" che fa in modo che cliccando sul link, si apra una nuova pagina del browser. Invece, specificando l'attributo title, quando il cursore del mouse passa sul link apparirà il testo contenuto che riportiamo tra apici doppi. Un esempio completo è il seguente:

```
<A href="http://www.google.it/" target="_blank" title="Collegamento a Google">
    Passa a google
</A>
```

Le immagini

Per inserire immagini nel documento si usa il tag che non deve essere chiuso. La sintassi è la seguente:

```
<img src= "http://www.harpo.com/materiale/bart.gif" width="224" height="69">
```



IMG sta per image, cioè immagine e src source, cioè origine. Dal momento che il browser normalmente non sa quali siano le dimensioni dell'immagine, finché questa non sia caricata correttamente, è opportuno indicare già nel codicella larghezza **width** e l'altezza **height** dell'immagine.

Lista puntata non ordinata

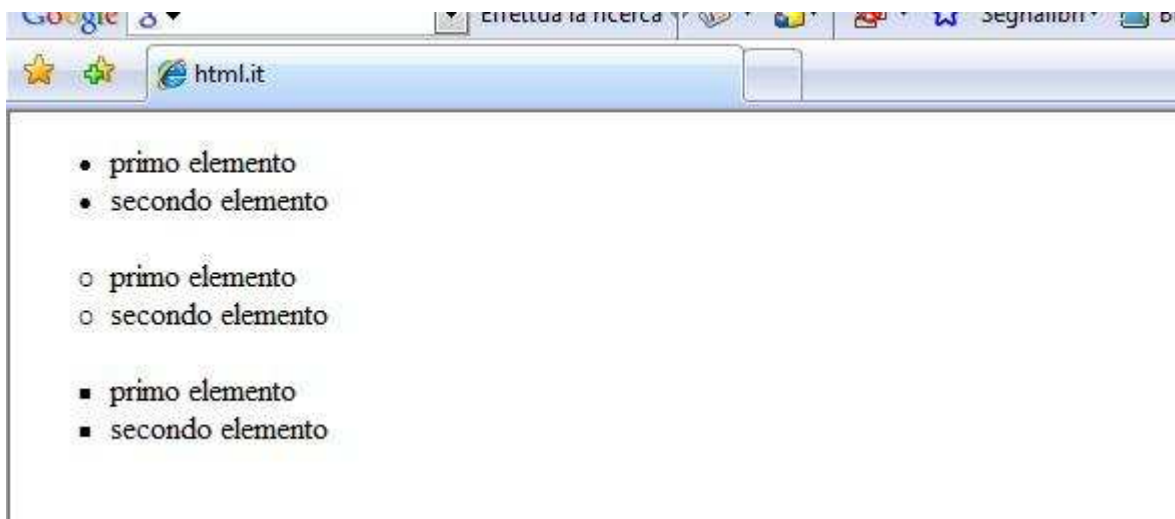
Gli elenchi non ordinati sono elenchi puntati individuati dal tag (unordered list) e gli elementi dell'elenco sono contraddistinti dal tag . Il tipo di segno grafico utilizzato per individuare gli elementi dell'elenco di default dipende dal browser ma di solito è un pallino pieno. I tipi di segno possono essere:

- type="disc" (visualizza un pallino pieno)
- type="circle" (visualizza un cerchio vuoto all'interno)
- type="square" (visualizza un quadrato pieno all'interno)

La porzione di codice sotto riportata

```
<UL type="disc">
  <LI>primo elemento</LI>
  <LI>secondo elemento</LI>
</UL>
<UL type="circle">
  <LI>primo elemento</LI>
  <LI>secondo elemento</LI>
</UL>
<UL type="square">
  <LI>primo elemento</LI>
  <LI>secondo elemento</LI>
</UL>
```

avrà il seguente risultato



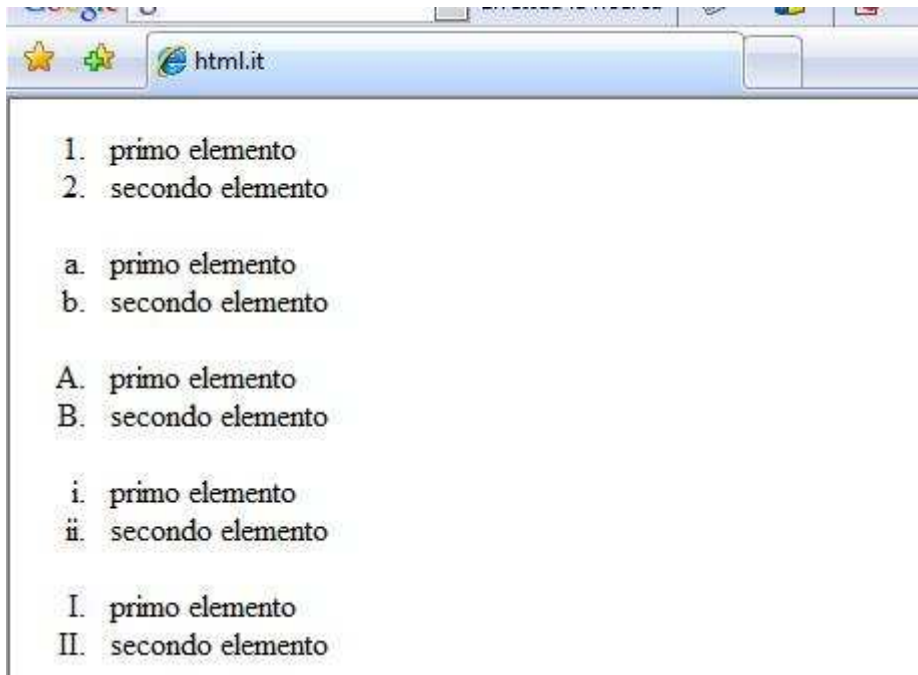
Lista puntata numerata

Gli elenchi ordinati sono contraddistinti dall'enumerazione degli elementi che compongono la lista ordinati e individuati in modo progressivo da numeri o lettere. Il tag da utilizzare per aprire un elenco ordinato è `` (ordered list) e gli elementi sono individuati anch'essi dal tag `` (list item). Lo stile di enumerazione visualizzata in default dal browser è quella numerica, ma è possibile indicare uno stile differente specificandolo per mezzo dell'attributo **type**.

Gli stili sono:

- `type="1"` (numeri arabi)
- `type="a"` (alfabeto minuscolo)
- `type="A"` (alfabeto maiuscolo)
- `type="i"` (numeri romani minuscoli)
- `type="I"` (numeri romani maiuscoli)

Riferiti all'esempio precedente produrranno una visualizzazione di questo tipo.



FORM

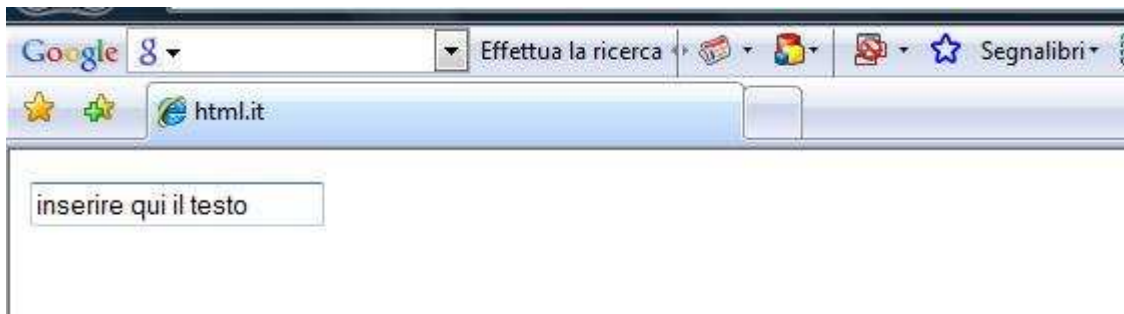
Uno dei fattori che ha decretato il successo del Web è la possibilità di interagire cioè di iscriversi a servizi di vario tipo e di partecipare a comunità virtuali. Per organizzare questo genere di servizi è necessario raccogliere i dati dell'utente e per farlo si utilizzano i moduli cioè i **form**.

Per creare una pagina con dei moduli bisogna utilizzare il tag <FORM> che è un elemento di blocco che lascia uno spazio prima dell'apertura e dopo la chiusura. Nel form vengono utilizzati due attributi: **name** che serve per indicare il nome del form e **action** che indica l'URL del programma o della pagina.

```
<FORM name="datiUtenti" action="paginaRisposta.php"></FORM>
```

Per i campi dei form il tag più utilizzato è l'<INPUT> che è senza chiusura e per specificare un determinato tipo di campo è sufficiente indicare il tipo di input ad esempio per creare un campo di testo si utilizza nel type "text", mentre per creare un bottone inserire nel type "button". I vari input oltre al tipo di campo (**type**) consentono di indicare anche il nome (**name**) e il valore (**value**).

```
<INPUT type="text" name="Testo" value="inserire qui il testo">
```



Non c'è form che si rispetti senza bottone di invio "submit" e senza bottone che consente di annullare ciò che è stato inserito e riporta il form allo stato originario "reset".

```
<INPUT type="text" value="inserire qui il testo">
```

```
<INPUT type="submit" value="tasto di invio">
```

```
<INPUT type="reset" value="cancella">
```



Altri elementi dei form

All'interno dei moduli è possibile inserire altri elementi. Vediamo il seguente esempio:

```
<html>
<head>
<title></title>
</head>
<body>
  <form name="Dati" action="registrazione.asp" method="post">
    Nominativo
    <input type="text" name="nominativo" size="30" maxlength="30"><br>

    Password
    <input type="password" name="password" size="20" maxlength="20"><br>

    Nazione
    <select name="nazione">
      <option value="Italia" selected> Italia</option>
      <option value="Francia"> Francia</option>
      <option value="Spagna"> Spagna</option>
    </select><br>

    Sesso
    <input type="radio" name="sesso" value="M"> M
    <input type="radio" name="sesso" value="F"> F<br>

    Iscritto <input type="checkbox" name="iscritto" value="si"><br>

    Note
    <textarea name="note" cols="30" rows="5"></textarea><br><br>

    <input type="reset" value="annulla">
    <input type="submit" value="Invia">
  </form>
</body>
</html>
```

Il form così costruito apparirà come nella figura seguente.

The image shows a web form with the following elements: a text input for 'Nominativo', a password input for 'Password', a dropdown menu for 'Nazione' with 'Italia' selected, two radio buttons for 'Sesso' labeled 'M' and 'F', a checkbox for 'Iscritto', a multi-line text area for 'Note', and two buttons at the bottom labeled 'annulla' and 'Invia'.

Ora esaminiamo gli ultimi elementi introdotti.

Nella categoria INPUT rientrano quelli di tipo **“password”** che è equivalente al tipo **“text”**, ma con la differenza che nel tipo password il contenuto della cella viene nascosto da una serie di pallini. Il tipo **“checkbox”** consente di gestire un valore che può essere selezionato oppure no e viene rappresentato da un quadrato in cui è possibile inserire una spunta. Con una serie di input di tipo **“radio”** è possibile rappresentare una serie di opzioni delle quali è possibile selezionare solo una. Per questi ultimi due tipi se impostiamo l’attributo **checked** l’elemento viene visualizzato come selezionato. La differenza sostanziale tra questi due elementi è la possibilità di selezionare più checkbox contemporaneamente, mentre dei radio se ne può selezionare solo uno.

In alternativa al radio button è possibile utilizzare l’elemento `<SELECT>` all’interno del quale possiamo inserire una serie di `<OPTION>`. Visivamente questo elemento sarà rappresentato da un combo box dal quale è possibile selezionare una voce dal menu a tendina. Il valore di default lo si può indicare inserendo l’attributo *“selected”* all’interno di uno degli elementi option.

L’elemento `<TEXTAREA>` consente di inserire un testo su più righe. I due parametri fondamentali di questo tag sono **cols** che definisce quanto deve essere larga la cella di testo (in pratica quanti caratteri deve contenere in larghezza), e **rows** che indica quante righe conterrà la cella.

I fogli di stile

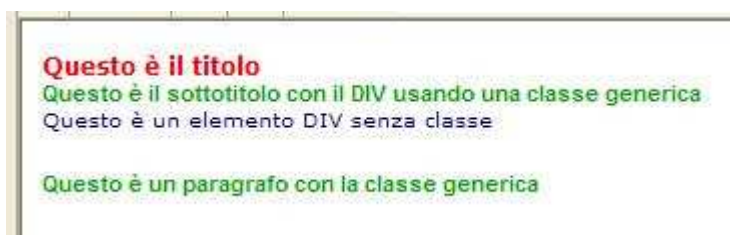
Secondo gli standard attuali di programmazione per il web, si tende a separare, nei documenti HTML, la parte presentazionale dai contenuti. Quindi all'interno della sezione BODY delle pagine si tende ad inserire solo ciò che effettivamente viene poi visualizzato, demandando tutte le impostazioni dei vari elementi alla sezione che viene definita foglio di stile. Le modalità con cui questi vengono realizzati sono diverse, in questo manuale prenderemo in esame solo la modalità embedded, cioè inserita all'interno della pagina stessa che ne fa uso, in alternativa si possono inserire in line, cioè in corrispondenza di ogni singolo elemento come riportato nel seguente esempio:

```
<div style="font-size: 10px; color: #ff0000">  
    Testo con carattere corpo 10 di colore rosso  
</div>
```

I fogli di stile in linea non presentano sostanziali differenze rispetto all'utilizzo degli attributi html, che vengono innestati direttamente in corrispondenza del tag. Nella programmazione web moderna però si tende a separare l'aspetto presentazionale dai contenuti, ossia si cerca di definire come un documento viene visualizzato in una sezione differente da quella che contiene il testo, le immagini e tutto ciò che il documento contiene. Ecco quindi che si usa l'elemento <STYLE></STYLE> all'interno dell'intestazione del documento definita dalla sezione <HEAD> </HEAD>.

Per quanto riguarda i fogli di stile possiamo definire stili legati ad un elemento HTML, possiamo definire una classe legata ad un elemento o possiamo definire una classe generica che può essere usata su più elementi. Chiariamo il tutto con un esempio.

La seguente pagina



E' il risultato di un documento HTML definito utilizzando i fogli di stile. Vediamolo.

```

<html>
<head>
  <title>Fogli di stile</title>
  <style>
    DIV {
      color : #000080;
      font-family : Verdana, Arial, Helvetica, sans-serif;
      font-size: 10px;}
    DIV.titolo {
      color : #ff0000;
      font-family : Verdana, Arial, Helvetica, sans-serif;
      font-size: 12px;
      font-weight: bold;}
    .sottotitolo {
      color : #00a000;
      font-family : Arial;
      font-size: 11px;
      font-weight: bold;}
  </style>
</head>
<body>
  <div class="titolo">
    Questo è il titolo
  </div>
  <div class="sottotitolo">
    Questo è il sottotitolo con il DIV usando una classe generica
  </div>
  <div>
    Questo è un elemento DIV senza classe
  </div>
  <p class="sottotitolo">
    Questo è un paragrafo con la classe generica
  </p>
</body>
</html>

```

Innanzitutto si comprende subito che, per utilizzare uno stile definito in una classe, dobbiamo inserire l'attributo **class** all'interno del tag nel quale vogliamo usarlo. Possiamo inoltre notare che tutti gli elementi <DIV> che non hanno una classe specificata vengono associati allo stile definito per l'elemento <DIV> generico ad eccezione di quelli che riportano una classe diversa. Infine la classe "*sottotitolo*" non è associata ad un elemento HTML specifico e quindi la possiamo usare legata a tutti gli elementi HTML, nel caso dell'esempio <DIV> e <P>.

Gli stili dei link

Alcuni elementi come i link possono avere definizioni di stili a seconda del contesto in cui si trovano, o per meglio dire del loro stato. Vediamo subito l'esempio per chiarire il concetto.

```
A:link {
    font-size: 11px;
    font-family: Verdana, Arial, Helvetica, Sans-Serif;
    text-decoration: none;
    color: #000000;}
A:visited {
    font-size: 11px;
    font-family: Verdana, Arial, Helvetica, Sans-Serif;
    text-decoration: none;
    color: #000000;}
A:hover {
    font-size: 11px;
    font-family: Verdana, Arial, Helvetica, Sans-Serif;
    text-decoration: underline;
    color: #505050;}
```

In pratica stiamo definendo come appaiono i link, però abbiamo definito tre stili diversi a seconda che il link sia relativo ad una pagina non ancora visitata ("**link**"), ad una pagina visitata ("**visited**") o che il cursore del mouse sia esattamente su di esso ("**hover**"). In questi casi l'elemento <A> inserito all'interno della sezione <BODY> non avrà una classe specificata ma sarà del tipo

```
<A HREF="http://www.menduni.info">Collegati al sito</A>
```

e la visualizzazione sarà differente in base allo stato del link.

Uso di stili nelle tabelle

Osserviamo la seguente figura:

Cognome	Nome
Rossi	Mario
Verdi	Giuseppe

Nel codice HTML possiamo notare che ancora una volta abbiamo una classe generica legata ad un elemento <TD> che viene applicata a tutte le celle ad eccezione di quelle definite con la classe “*testata*”.

```
<html>
<head>
  <title>Fogli di stile</title>
  <style>
    TABLE {
      width: 800px;
      background-color : #f0f0f0;
      border: 3px solid #0000ff;}
    TD.testata {
      color : #ffffff;
      font-family : Verdana, Arial, Helvetica, sans-serif;
      font-size: 12px;
      font-weight: bold;
      padding: 5px;
      text-align: center;
      vertical-align: middle;
      background-color : #ff0000;
      height: 30px;}
    TD {
      color : #000000;
      font-family : Verdana, Arial, Helvetica, sans-serif;
      font-size: 10px;
      padding: 3px;
      text-align: left;
      border: 1px solid #000000;
      height: 30px;}
  </style>
</head>
<body>
  <table>
    <tr>
      <td class="testata">Cognome</td>
      <td class="testata">Nome</td>
    </tr>
    <tr>
      <td>Rossi</td>
      <td>Mario</td>
    </tr>
  </table>
</body>
</html>
```

```
<tr>
  <td>Verdi</td>
  <td>Giuseppe</td>
</tr>
</table>
</body>
</html>
```

Attributi definibili nei fogli di stile

Dagli esempi riportati si comprende che la sintassi dei fogli di stile è leggermente diversa da quella degli attributi dei tag. Nei fogli di stile infatti la sintassi è la seguente:

```
attributo : valore ;
```

E' da sottolineare il fatto che nel caso dell'attributo "font-family" che indica il nome del font che voglia usare i valori sono più di uno e sono separati da virgole. Questo perché se vogliamo usare un determinato font ma non siamo sicuri che questo sia presente sul computer del client che accede al documento, indicando un elenco di font, il browser cercherà di utilizzare uno di quelli indicati, nell'ordine esatto in cui li abbiamo scritti e, se non ne trovasse nemmeno uno, utilizzerà quello di default del browser.

Passiamo ad una breve descrizione degli attributi utilizzati negli esempi:

- **color**: colore del testo;
- **font-family**: nome del font (o elenco dei font) che si vuole applicare;
- **font-size**: dimensione del carattere (espresso in pixel indicando sempre il suffisso "px");
- **font-weight**: assume valori "bold" o "none" e consente di specificare l'uso del grassetto;
- **text-decoration**: può essere "underline" o "none" e consente di specificare l'uso del sottolineato;
- **background-color**: indica il colore di sfondo;
- **border**: indica contemporaneamente lo spessore del bordo, il tipo ed il suo colore
- **padding**: indica la distanza tra il bordo della cella ed il suo contenuto;
- **text-align**: definisce l'allineamento orizzontale del testo ("left", "center" o "right");
- **vertical-align**: definisce l'allineamento verticale del testo ("top", "middle" o "bottom");
- **width**: definisce la larghezza in pixel dell'elemento;
- **height**: definisce l'altezza in pixel dell'elemento.

E' chiaro che ci sono molti altri attributi che possono essere definiti, quindi questo elenco non è assolutamente esaustivo. Per un approfondimento dell'argomento si rimanda ad un manuale più completo o ai siti inerenti il linguaggio HTML.