GUIDA HTML

di Wolfgang Cecchin
w.cecchin@html.it

L'HTML è il principale linguaggio di pubblicazione di pagine Web. Seguendo questa guida imparerai a realizzare siti in HTML, implementando moduli, costruendo tabelle e altro ancora. La guida è rivolta a coloro che vogliono imparare l'HTML ma fornisce spunti anche per gli utenti esperti.

Percorso consigliato: principianti
Livello di difficoltà:

Navigatore

Introduzione

[torna su]

Come è fatta una pagina HTML

[torna su]

L'HTML e i fogli di stile (CSS)

[torna su]

Lo sfondo di un documento HTML

[torna su]

Il testo di un documento HTML

[torna su]

Il cuore del WWW: i Link

[torna su]

Le immagini e le mappe di immagine

[torna su]

Le tabelle

[torna su]

33. **Raggruppare celle con rowspan e colspan**
Come dare alle tabelle una struttura più elastica raggruppandone gli elementi

34. **Annidare tabelle**
Come annidare una tabella dentro l'altra

35. **Attributi del tag table**
Perfezioniamo la visualizzazione di una tabella: border, cellspacing e cellpadding

36. **Attributi di <table>, <tr>, <td>**
Gli attributi per le dimensioni, lo sfondo l'allineamento, i bordi, l'a capo

37. **Impaginare un layout con le tabelle**
Usare le tabelle per disegnare il layout del sito: fisso o fluido

I Frame

[torna su]

---

38. **Comporre una pagina in frame**
Cosa sono i frame; la struttura di un frameset; i frameset annidati

39. **Attributi dei frames per la visualizzazione**
Gli attributi: scrolling, noresize, frameborder, marginheight e marginwidth

40. **I link in un frameset e il tag noframes**
Come gestire i collegamenti all'interno di un frameset; il tag NOFRAME

41. **L'iframe**
Un particolare tipo di frame: gli iframe o inline frame

42. **Vantaggi e svantaggi dei frames**
Quali motivi per usare o non usare una struttura a Frame

43. **Impaginare a livelli con i CSS**
Un altro metodo di impaginazione: i livelli e i CSS

I Moduli (forms)

[torna su]

---

44. **Struttura del tag form**
La struttura del tag Form e i suoi principali utilizzi

45. **Un po' d'ordine: raggruppare i moduli**
Come rendere più eleganti e razionali i nostri moduli con il tag fieldset

46. **Il tag Input**
I campi dei form: come utilizzare il tag input

47. **I bottoni (sumbit, reset, button, image)**
Inserire i bottoni o bottoni di immagine per inviare i moduli

Includere elementi multimediali e script

[torna su]

Linguaggi e strumenti

[torna su]


TORNA SU

Cos’è l’HTML

HTML è l'acronimo di **Hypertext Markup Language** ("Linguaggio di contrassegno per gli Ipertesti") e non è un linguaggio di programmazione (sono linguaggi di programmazione il C, il C++, il Pascal, il Java, e sono linguaggi di scripting il PHP, l'ASP, il PERL, il JavaScript).

Si tratta invece di un **linguaggio di contrassegno** (o 'di marcatura'), che permette di indicare come disporre gli elementi all'interno di una pagina: le indicazioni vengono date attraverso degli appositi marcatori, detti "tag".

Ciò significa che l'HTML **non ha meccanismi che consentono di prendere delle decisioni** ("in questa situazione fai questo, in quest'altra fai quest'altro"), e non è in grado di compiere delle iterazioni ("ripeti questa cosa, finché non succede questo"), né ha altri costrutti propri della programmazione.

Il linguaggio HTML, pur essendo dotato di una sua sintassi, **non presuppone la logica ferrea e inappuntabile dei linguaggi di programmazione**: se vi dimenticate di chiudere un tag, non verranno prodotti dei messaggi di errore; se non rispettate la sintassi probabilmente non otterrete la visualizzazione della pagina che desiderate, ma nient'altro. A volte vi troverete persino a dover adottare dei "trucchetti", non proprio da manuale, pur di visualizzare la pagina correttamente con ogni browser.

Suggerimenti: Può succedere - soprattutto a chi è alle prime armi - di continuare a modificare un file, ma di non riuscire a vederne le modifiche. Questo succede perché la pagina visualizzare è sempre quella vecchia memorizzata nella cache. Quando state elaborando pagine per il web, ricordatevi di impostare la cache del vostro browser in modo che il file html venga ricaricato ogni volta che richiamate la pagina.

In Internet Explorer:

Strumenti > Opzioni Internet > Generale > Impostazioni > Ricerca versioni più recenti delle pagine memorizzate:
- all'apertura della pagina

In Mozilla:

Modifica > Preferenze > Avanzate > Cache > Confronta la pagina nella cache con la pagina in rete:
- ogni volta che vedo una pagina

Lezione successiva

[Sommario]

TORNA SU

Prima di cominciare davvero: lo standard HTML

L'organizzazione che si occupa di standardizzare la **sintassi del linguaggio HTML** (il W3C: Word Wide Web Consortium) ha rilasciato diverse versioni di questo linguaggio (HTML 2.0, HTML 3.2, HTML 4.0); e - da un certo punto in poi - l'HTML si è evoluto in XHTML (si tratta dell'HTML riformulato come linguaggio XML - ne sono già state rilasciate due versioni).

La versione dell'HTML che esamineremo in questo corso è l'ultima rilasciata: si tratta dell'HTML 4.01 del 24 dicembre 1999.

Anche se abbiamo detto che l'HTML si è evoluto in XHTML ci sono delle ottime ragioni per incominciare a studiare l'HTML e non l'XHTML:

- di fatto l'HTML **verrà utilizzato ancora per diversi anni** come linguaggio principe delle pagine web
- alcuni concetti dell'XHTML richiedono già **una certa comprensione dei problemi** che si acquisisce solo con l'esperienza. L'HTML è più immediato e consente di incominciare subito a produrre documenti web
- **chi conosce l'XHTML non può non conoscere l'HTML**. La conoscenza dell'HTML è infatti il prerequisito essenziale di ogni webmaster. Comunque le differenze tra i due linguaggi non sono così marcate e passare dall'uno all'altro non dovrebbe richiedere molta fatica.

Per gli approfondimenti sulle differenze tra i vari linguaggi vi rimando tuttavia all'appendice di questa guida.

Un'ultima avvertenza: in molte lezioni è presente una sezione denominata "approfondimenti". Chi inizia adesso a studiare HTML ed è alla sua prima lettura può tranquillamente ignorare quel paragrafo. Le indicazioni ivi contenute vi torneranno utili a una seconda lettura, o man mano che prendete confidenza con l'HTML e l'arte di sviluppare siti web.

Lezione successiva

[Sommario]

TORNA SU

I TAG dell'HTML: come scriverli

Struttura di un tag

Abbiamo detto che all'interno di ogni pagina è presente una serie di marcatori (i **TAG**), a cui viene affidata la visualizzazione e che hanno differenti nomi a seconda della loro funzione. I tag vanno inseriti tra parentesi uncinate (**<TAG>**), la chiusura del tag viene indicata con una "/" (è il simbolo comunemente detto "slash". Quindi: **</TAG>**). Il contenuto va inserito tra l'apertura e la chiusura del tag medesimo, secondo questa forma:

```
<TAG attributi>contenuto</TAG>
```

Ecco un esempio, con una sintassi che serve a disporre un testo giustificato a destra:

```
<P align="right">testo</P>
```

dall'esempio è evidente che la struttura di un attributo è: **attributo="valore"**

Quindi in definita la struttura di un tag sarà:

```
<TAG attributo_1="valore1" attributo_2="valore2">contenuto</TAG>
```

Alcuni particolari tag non hanno contenuto - perché ad esempio indicano la posizione di alcuni elementi (come il tag delle immagini) -, conseguentemente questi tag non hanno neanche chiusura. La loro forma sarà dunque:

```
<TAG attributi>
```

Ecco un esempio di immagine:

```
<IMG widht="20" height="20" SRC="miaImmagine.gif" ALT="alt">
```

come si vede il tag non viene chiuso. Questo tipo di tag viene detto "empty", cioè "vuoto".

Annidamento e indentazione

Una caratteristica importante del codice HTML è che i tag possono essere annidati l'uno dentro l'altro. Anzi molto spesso è necessario farlo.

Ad esempio:

```
<TAG1 attributi>
        contenuto 1

        <TAG2>
                contenuto 2
        </TAG2>

</TAG1>
```

Potremmo quindi avere ad esempio:

```
<P align="right">
        testo 1

                <P align="left">
                        testo 2
                </P>

</P>
```

L'annidamento ci permette quindi di attribuire formattazioni successive al testo che stiamo inserendo.

Come si può vedere già nell'esempio, è una buona norma utilizzare dei **caratteri di tabulazione** (il tasto tab a sinistra della lettera Q) per far rientrare il testo ogni volta che ci troviamo in presenza di un annidamento e man mano che entriamo più in profondità nel documento.

In pratica apertura e chiusura del tag si trovano allo stesso livello, mentre il contenuto viene spostato verso destra di un tab: non si tratta soltanto di un fattore visivo, ma l'allineamento di apertura e chiusura tag viene mantenuto anche se scorriamo in verticale il documento con il cursore.

Questa procedura si chiama **indentazione**, e grazie ad essa il codice HTML risulta più leggibile. Si confronti ad esempio:

```
<P align="right">testo 1<P align="left">        testo 2 </P></P>
```

con:

```
<P align="right">

        testo 1
                <P align="left">
                        testo 2
                </P>

</P>
```

per il browser i due esempi sono equivalenti, ma per l'utente umano è evidente che la differenza è notevole: pensate ad una pagina complessa visualizzata in un unico blocco di testo: sarebbe del tutto illeggibile!

Lezione successiva

[Sommario]

TORNA SU

I commenti

Un’altra strategia importante, per rendere il nostro codice più leggibile è quella di inserire dei **"commenti"** nei punti più significativi: si tratta di indicazioni significative per il webmaster, ma invisibili al browser. Inserendo i commenti in punti specifici del documento ci permette di mantenere l’orientamento anche in file molto complessi e lunghi. La sintassi è la seguente:

```
<!-- questo è un commento -->
```

e ci permette di "commentare" i vari punti della pagina. Ad esempio:

```
<!-- menu di sinistra -->

<!-- barra in alto -->

<!-- eccetera -->
```

Lezione successiva

[Sommario]

TORNA SU

Maiuscolo o minuscolo?

L'HTML è "**case unsensitive**", cioè indipendente dal formato. Questo significa che è del tutto indifferente se scrivere i tag in maiuscolo o in minuscolo.

```
<P ALIGN="RIGHT">
```

e

```
<p align="right">
```

vengono letti allo stesso modo dal browser.

Fino a qualche tempo fa, per aumentare la leggibilità del codice, era buona norma scrivere in maiuscolo il nome del tag (es: **<P>**) e in minuscolo gli attributi (es: **align="right"**). Quindi:

```
<P align="right">
```

Tuttavia oggi, per analogia con l'XHTML (che è figlio dell'XML e dell'HTML ed è **"case sensitive",** sensibile al formato) è consigliabile scrivere tutto in minuscolo, per abituarsi già al linguaggio che verrà. Maiuscolo e minuscolo, in ogni caso non costituiscono errore.

Fino a questo momento - per rendere più chiare le differenze - abbiamo utilizzato la vecchia abitudine di alternare maiuscolo e minuscolo differenziando tag e attributi, d'ora in poi invece tutta la sintassi HTML della guida sarà in minuscolo.

Lezione successiva

[Sommario]

TORNA SU

Struttura della pagina

Basandoci sulle indicazioni precedenti, incominciamo a scrivere la nostra prima pagina html.

Per prima cosa inseriamo una riga che indica che stiamo utilizzando le specifiche del Word Wide Web Consortium che riguardano il codice HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//IT">
```

esamineremo ulteriormente questa riga nell'appendice, per ora lasciamola così.

Poi apriamo il nostro primo tag, che indica che quanto è compreso tra apertura e chiusura è in codice HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//IT"> <html>

... altri tag ...

</html>
```

Un documento HTML è normalmente diviso in due sezioni:

Testa (<head>)	Contiene informazioni non immediatamente percepibili, ma che riguardano il modo in cui il documento deve essere letto e interpretato. Questo è il luogo dove scrivere - ad esempio - i meta-tag (alcuni sono ad esclusivo beneficio dei motori di ricerca), script JavaScript o VbScript, fogli di stile, eccetera
Corpo (<body>)	Qui è racchiuso il contenuto vero e proprio del documento

Ci occuperemo in seguito della head (l'argomento verrà ripreso poi nella conclusione della guida. Per ora facciamo riferimento soltanto a due tag che devono essere presenti in questa sezione:

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

indica al browser che deve caricare il set di caratteri occidentale (e non - ad esempio - il set di caratteri giapponese).

```
<title>Nome del sito</title>
```

Il title è il titolo della pagina e compare in alto sulla barra del browser (se guardate in alto a sinistra del browser noterete la scritta "Struttura della pagina | Guida HTML | HTML.it"). È bene compilarlo da subito, onde evitare poi di avere pagine senza titolo. Da quanto abbiamo detto la nostra prima pagina sarà questa, che è consultabile anche nell'esempio allegato:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//IT">

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <title>HTML.it</title>
```

```
</head>
<body>

  <!-- Scriveremo qui -->

  Qui il nostro contenuto

</body>
</html>
```

D'ora in poi i vari tag che impareremo all'interno della guida andranno scritti all'interno del body, quando non sia indicato diversamente.

L e z i o n e s u c c e s s i v a

[S o m m a r i o]

TORNA SU

Impostare il colore di sfondo

Incominciamo a vedere come ottenere la nostra prima pagina HTML nel modo in cui desideriamo visualizzarla.

Se vogliamo impostare un colore di sfondo è necessario impostare il relativo attributo del tag body. Così:

```
<body bgcolor="blue">
```

bgcolor sta per "background color", cioè "colore di sfondo". Molti colori sono disponibili utilizzando le corrispondenti parole chiave in inglese.

Qui potete trovare un esempio della pagina con lo sfondo blu

Tuttavia non è consigliabile inserire la notazione del colore facendo riferimento a questo tipo di sintassi, dal momento che non possiamo sapere esattamente a quale tonalità di colore corrisponda il blu del computer dell'utente. È preferibile in molti casi utilizzare la corrispondente codifica esadecimale del colore, che ci permette – tra le altre cose – di scegliere anche tonalità di colore non standard. Con la notazione esadecimale il nostro esempio diventa:

```
<body bgcolor="#0000FF">
```

Ecco una tabella con la notazione di alcuni colori (molti di essi sono disponibili anche nelle varianti "dark" e "light", ad esempio: "darkblue", "lightblue"):

colore	parola chiave	notazione esadecimale
arancione	orange	#FFA500
blu	blue	#0000FF
bianco	white	#FFFFFF
giallo	yellow	#FFFF00
grigio	gray	#808080
marrone	brown	#A52A2A
nero	black	#000000
rosso	red	#FF0000
verde	green	#008000

viola	violet	#EE82EE

Il numero di colori che l'utente ha a disposizione dipende dalla scheda video. Oggi si va da una risoluzione minima di 256 colori a una risoluzione che prevede svariati milioni di colori.

Per capire di cosa stiamo parlando, provate a visualizzare questa pagina cambiando il numero di colori visualizzati sul monitor. Per fare ciò, in Windows, andate in: **Pannello di controllo > Schermo > Impostazioni** e cambiate il numero dei colori, applicate i cambiameni e tornate a visualizzare la pagina. Come si vede la visualizzazione della tonalità di colore è sensibilmente diversa passando da 256 a 65.536 colori (16 bit).

Poiché non c'è modo di sapere quale scheda video abbia l'utente (o come l'abbia impostata), i webdesigner per molto tempo hanno fatto riferimento alla "palette sicura" dei 256 colori che sicuramente l'utente è in grado di visualizzare. Si tratta della cosiddetta palette web safe.

C'è però da dire che oramai la stragrande maggioranza dei computer è impostata per visualizzare almeno migliaia di colori, dunque l'utilizzo della palette "web safe" non è più così strettamente necessaria (lo era nei primi anni del web).

Lezione successiva

[Sommario]

TORNA SU

Inserire un’immagine di sfondo

Per inserire un’immagine come sfondo è sufficiente utilizzare la seguente sintassi:

```
<body background="imgSfondo.gif">
```

Per ora presupponiamo che l’immagine di sfondo si trovi nella stessa cartella della nostra pagina HTML, vedremo in seguito (quando parleremo delle immagini) come inserire immagini che si trovano in altre cartelle.

L’immagine di sfondo verrà ripetuta in orizzontale e in verticale.

È anche possibile combinare i due attributi, in modo che mentre l’immagine di sfondo viene caricata, venga comunque visualizzata una colorazione della pagina:

```
<body bgcolor="#0000ff" background="imgSfondo.gif">
```

Ecco subito un esempio di pagina impostata con lo sfondo.

È importante **assegnare sempre un colore alla pagina** anche quando lo sfondo della pagina è bianco (al massimo assegnare **bgcolor="#FFFFFF"**). Infatti, come impostazione predefinita, il browser assegna alla pagina il colore di sfondo che l’utente ha impostato nella finestra del sistema operativo: quindi se l’utente ha impostato uno sfondo nero e voi non avete assegnato nessun colore di sfondo alla pagina, la vostra pagina sarà nera.

Se usate Windows, per fare una prova provate a impostare diversamente il tema delle finestre. Dal pannello di controllo: **Schermo > Aspetto > Combinazione** e poi scegliere:

"nero a contrasto elevato", oppure "prugna".

Infine visualizzate questa pagina - che è senza sfondo – e vedrete che la pagina HTML prenderà la colorazione che avete impostato nel tema delle finestre.

Lezione successiva

[Sommario]

TORNA SU

Impostare il colore del testo e dei link per tutta la pagina

Il testo

Se non impostate nessun colore per il testo, di default il testo di una pagina è nero.

Tuttavia il nero non sempre è leggibile con tutti i colori di sfondo. Immaginate ad esempio di volere utilizzare come sfondo il colore nero: con una pagina nera e testo nero non leggeremmo nulla!

Abbiamo allora la possibilità di assegnare un colore per il testo di tutta la pagina, semplicemente utilizzando questo attributo del tag body:

```
<body text="red">
```

Quindi potremo avere, ad esempio:

```
<body bgcolor="#0000ff" text="#ffffff">
```

come nell'esempio consultabile in questa pagina.

I link

Non c'è bisogno di spiegare che cosa siano i link: l'esperienza della navigazione nel web ci ha infatti insegnato che il link è un collegamento, un ponte tra una pagina è l'altra.

Non tutti però sanno che i link testuali hanno diversi stati:

Status	Codifica in HTML 4.01	Descrizione
Collegamento normale	link	Normalmente il link quando si trova "a riposo" viene evidenziato in qualche maniera all'interno della pagina HTML, in modo che sia facile per l'utente individuarlo. Nell'HTML tradizionale il link è sempre sottolineato (è possibile eliminare la sottolineatura soltanto usando i CSS). Di default i link sono blu (**#0000FF**).

Collegamento visitato	visited	Un link è visitato, quando l'URL della pagina compare nella cronologia dell'utente. Di default i link visitati sono di color violetto (più esattamente: **#800080**).
Collegamento attivo	active	Il collegamento è attivo nel momento in cui il link è stato cliccato e sta avvenendo il passaggio da una pagina all'altra. Non si tratta di una caratteristica particolarmente utile oggi, ma quando i modem avevano una velocità molto inferiore a quella odierna, vedere un link "attivo" era comunque un'indicazione sul fatto che qualcosa stava avvenendo. Con Internet Explorer è possibile vedere anche una linea tratteggiata attorno al collegamento attivo. Un ulteriore condizione in cui un link si rileva "attivo" è quando si utilizza il tasto destro del mouse su di lui. Insomma un link è attivo quando "ha il focus".
Collegamento al passaggio del mouse	non presente ("**hover**" nei CSS)	Con l'HTML 4.01 al passaggio del mouse sul link si può fare ben poco, coi fogli di stile invece è possibile creare qualche effetto di visualizzazione.

Abbiamo dunque tre stati canonici dei link (link a riposo, link attivo e link visitato) e una condizione aggiuntiva introdotta dai fogli di stile (status del link al passaggio del mouse):

Anche il colore dei link di tutta la pagina può essere tramite gli attributi del body:

I link secondo le impostazioni predefinite sono blu, per cambiare colore:

```
<body link="red">
```

Per cambiare colore ai link visitati (di default viola):

```
<body vlink="green">
```

i link visitati vengono memorizzate nella cronologia del browser, quindi se volete ripristinare il colore originario dei link, è sufficiente cancellare la cronologia.

Per cambiare colore ai link attivi:

```
<body alink="yellow">
```

La sintassi completa per impostare i link è quindi:

```
<body link="red" alink="yellow" vlink="green">
```

Lezione successiva

[Sommario]

TORNA SU

Titoli, paragrafi, blocchi di testo e contenitori

Nulla ci vieta di scrivere direttamente all'interno del tag body, come già abbiamo visto negli esempi precedenti, senza utilizzare nessun tag.

A dire la verità è però più pratico racchiudere il testo in appositi tag a seconda della funzione che il testo sta svolgendo. La nostra pagina risulterà più semplice da leggere, quando dovremo modificarla, e inoltre potremo ottenere la formattazione che desideriamo.

Come abbiamo detto dall'inzio, i tag sono infatti dei marcatori che ci permettono di mantenere ordine nella pagina e ottenere il layout che desideriamo.

I principali tag-contenitori da utilizzare per "racchiudere" il testo sono:

Nome tag	Visualizzazione codice	Descrizione
titolo 1 titolo 2 titolo 3 titolo 4 titolo 5 titolo 6	**titolo 1** **titolo 2** **titolo 3** **titolo 4** **titolo 5** **titolo 6**	"H" sta per "heading", cioè titolo: le grandezze previste sono sei. 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 sé. Si tratta dunque di un elemento di blocco.

paragrafo **Esempio:** paragrafo 1 paragrafo 2	paragrafo 1 paragrafo 2	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.
Blocco di testo **Esempio:** blocco 1 blocco 2	blocco 1 blocco 2	Il blocco di testo va a capo, ma - a differenza del paragrafo – non lascia spazi prima e dopo la sua apertura.
contenitore **Esempio:** contenitore 1 contenitore 2 contenitore 3	contenitore 1 contenitore 2 contenitore 3	Lo span è un contenitore generico che può essere annidato (ad esempio) all'interno dei DIV. Si tratta di un elemento inline, che cioè non va a capo e continua sulla stessa linea del tag che lo include. Avrete modo di utilizzare lo <SPAN> soprattutto quando incomincerete ad usare i fogli di stile.

Le differenze tra <P>, <DIV> e <SPAN> sono quindi che:

- **<P>** lascia spazio prima e dopo la propria chiusura
- **<DIV>** non lascia spazio prima e dopo la propria chiusura, ma - essendo un elemento di blocco - va a capo
- **<SPAN>** -essendo un elemento inline - non va a capo

Un esempio dovrebbe chiarire il tutto.

Per quel che riguarda il tag heading (**<h1>**, …, **</h6>**) è da notare che la grandezza del carattere varia a seconda delle impostazioni che l'utente ha sul proprio computer.

Con Internet Explorer, ad esempio, basta andare in: Visualizza > Carattere

Per vedere il titolo crescere o decrescere.

Allineare il testo

Tutti i "tag-contenitori" che abbiamo appena visto (e molti altri tag di quelli che vedremo) permettono di allineare il testo utilizzando semplicemente l'attributo **align**.

Se avete seguito finora la presente guida, avrete anche indovinato che l'attributo "**align**" è **disapprovato dal W3C**, dal momento che per allineare il testo bisognerebbe invece utilizzare i fogli di stile.

In ogni caso, vediamo come potremmo ad esempio allineare il testo di un paragrafo:

Allineamento	**Sintassi**	**Visualizzazione codice HTML**
Testo allineato a sinistra	testo	Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ché la diritta via era smarrita
Testo allineato a destra	testo	Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ché la diritta via era smarrita
Testo giustificato	testo	Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ché la diritta via era smarrita

Andare a capo

Per andare a capo molti webmaster utilizzando l'apertura arbitraria di paragrafi che non contengono nulla e che vengono lasciati aperti. Ad esempio:

Si tratta in buona sostanza di un errore, visto che per andare a capo esiste il tag
 ("break", cioè "interruzione").

Per andare a capo è quindi sufficiente scrivere un <br>. Per saltare una riga ne occorrono due:

Un altro valido tag per dividere la pagina in parti è il tag **<hr>** ("horizontal rule"), che serve per tracciare una linea orizzontale. Ecco il tag in azione:

Questo tag ha anche alcuni attributi (deprecati, perché la formattazione andrebbe fatta con i CSS):

L'attributo "**noshade**" evita di sfumare la linea, "**size**" indica l'altezza in pixel, "**width**" è la larghezza in pixel o in percentuale, "**align**" l'allineamento. Con Internet Explorer si riesce persino a impostare il colore:

```
<hr noshade size="5" width="50%" align="center" color="red">
```

Risultato:

L e z i o n e s u c c e s s i v a

[S o m m a r i o]

TORNA SU

Scegliere lo stile (grassetto, corsivo & C.)

Nella grafica cartacea con "stile di un testo" si intende la variante del "tondo", del "corsivo", o del "grassetto" di un carattere tipografico.

Nel parlare di stili del testo in HTML solitamente si suddividono i tag in grado di attribuire lo stile al testo in **stili fisici** e **stili logici:**

- vengono definiti come **fisici** quei tag che definiscono graficamente lo stile del carattere, indipendentemente dalla funzione del contenuto del tag
- vengono definiti come **logici** quei tag che forniscono anche informazioni sul ruolo svolto dal contenuto del tag, e in base a questo adottano uno stile grafico

Gli stili fisici

I principali stili fisici sono:

Codice HTML	**Visualizzazione**	**Descrizione**
testo in grassetto **Esempio:** Questo testo è in grassetto	Questo **testo** è in grassetto	Formatta il testo in grassetto.
testo in corsivo **Esempio:** Questo testo è in corsivo	Questo *testo* è in corsivo	Formatta il testo in corsivo. Tuttavia bisogna evitare di evidenziare in corsivo dei blocchi di lunghezza considerevole, perché la leggibilità del corsivo nel web lascia a desiderare. Meglio limitarsi a poche parole.

<pre>testo preformattato</pre>
 	
Esempio:
 	
<pre>
Nel mezzo del cammin di nostra vita
 	
mi ritrovai per una selva oscura
 	
ché la diritta via era smarrita.
 	
</pre>
	Nel mezzo del cammin di nostra vita
	mi ritrovai per una selva oscura
	ché la diritta via era smarrita.
		Il motore di rendering del browser restituisce il testo così come è stato inserito nel file html dall'autore stesso (preformattato quindi), senza riformattarlo.
		
		Di fatto è un tag poco usato.

<u>testo sottolineato</u> Questo testo è sottolineato **Esempio:** Questo <u>testo</u> è sottolineato	Questo testo è sottolineato	Sottolinea il testo presente nel tag. Nel web le sottolineature del testo sono da evitare, per non confondere il lettore con i link.
<strike>testo barrato</strike> **Esempio:** Questo <strike>testo</strike> è barrato	Questo ~~testo~~ è barrato	Con il testo barrato, vengono indicate (ad esempio) le correzioni.
^{testo in apice} **Esempio:** E=mc²	E=mc^2	"Superscript": indica al browser di portare il testo al di sopra della linea di scrittura. Utile per formule matematiche (ad esempio le potenze)

_{testo in pedice} **Esempio:** H₂O	H_2O	"Subscript": indica al browser di portare il testo al di sotto della linea di scrittura (utile ad esempio per i simboli chimici)

Di fatto i tag <b> e <i> sono molto utilizzati, perché consentono di cambiare lo stile del testo al volo.

Gli stili logici

Come abbiamo visto gli **stili logici** forniscono anche informazioni sul contenuto e la loro formattazione è spesso lasciata al browser con risultati a volte deludenti. Proprio per questo gli stili logici sono entrati in disuso e sono poco usati.

Riportiamo di eseguito i principali stili logici, per completezza, ma non sarà necessario ricordarseli.

Codice HTML	**Visualizzazione**	**Descrizione**
<abbr>abbreviazione</<abbr> **Esempio:** <abbr>C/A</abbr> HTML.it	C/A HTML.it	Indica un abbreviazione. Nessun rendering del testo particolare
<acronym>acronimo</acronym> **Esempio:** <acronym>HTML</acronym>	HTML	Indica un acronimo. Nessun rendering del testo particolare
<address>indirizzo</address> **Esempio:** <address>HTML.it - via dei Castani 183/185 – 00172 Roma</address>	HTML.it - via dei Castani 183/185 – 00172 Roma	Serve per indicare gli indirizzi: siano essi e-mail, o indirizzi fisici. Il testo viene visualizzato in corsivo

<blockquote>blocco di citazione</blockquote> **Esempio:** <blockquote> Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ché la diritta via era smarrita </blockquote>	Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ché la diritta via era smarrita	Sono blocchi di citazione. Il testo viene rientrato verso destra.
<cite>citazione</cite> **Esempio:** <cite> Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ché la diritta via era smarrita **</cite>**	*Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura ché la diritta via era smarrita*	Per citazioni brevi: il testo è visualizzato in corsivo.
<code>codice</code> **Esempio:** <code>if (document.all) alert ("ciao");</code>	if (document.all) alert ("ciao");	Indica un blocco di codice in linguaggio di programmazione. Nessun rendering del testo particolare
<dfn>definizione</dfn> **Esempio:** <dfn>L'HTML è un linguaggio di contrassegno</dfn>	*L'HTML è un linguaggio di contrassegno*	Indica una definizione: il testo è visualizzato in corsivo
enfasi **Esempio:** Ti ho detto questo!	Ti ho detto *questo!*	Serve per porre l'enfasi su un'espressione: il testo è visualizzato in corsivo

<kdb>keyboard</kdb> **Esempio:** <kdb>digitazione da tastiera</kdb>	digitazione da tastiera	Indica una digitazione da tastiera: il testo viene visualizzato a spaziatura fissa
<q>citazione all'interno della frase</q> **Esempio:** Come diceva Don Abbondio: <q>***"***Il coraggio, uno non se lo può dare***"***</q>	Come diceva Don Abbondio: "Il coraggio, uno non se lo può dare"	Indica una citazione breve all'interno del testo. Nessun rendering del testo particolare
<samp>esempio</samp> **Esempio:** <samp>ecco un esempio di "samp"</samp>	`ecco un esempio di "samp"`	Indica un esempio. Il testo viene visualizzato a spaziatura fissa.
rafforzamento **Esempio:** Ecco un testo rafforzato	Ecco un **testo rafforzato**	Inseriamo i dati nella variabile temporanea *temp* ...
<var>variabile</var> **Esempio:** Inseriamo i dati nella variabile temporanea <var>temp</var> ...	Inseriamo i dati nella variabile temporanea *temp* ...	La variabile viene visualizzata in corsivo.

Approfondimenti

Come si può vedere molti tag (logici e fisici) tradiscono l'origine scientifica e informatica del Web (sono presenti tag per blocchi di codice di programmazione, per definizioni, per l'indicazione delle variabili…).

Sorprendentemente nessuno dei tag fisici o logici è stato dichiarato "deprecato" dal W3C, ma

anzi tutti questi tag sono passati dall'HTML 3.2 originario fino all'XHTML (passando illesi attraverso l'HTML 4).

Per quel che riguarda i tag fisici: a rigor di logica lo stile "grassetto" dovrebbe essere ottenuto con i fogli di stile (così come tutte le formattazioni), ma evidentemente la possibilità di ottenere un testo in grassetto semplicemente scrivendo **"testo"** è troppo comoda per poter essere considerata obsoleta.

Per quel che riguarda i tag logici: in realtà questo tipo di tag offrono un ulteriore aiuto al webmaster anche in un approccio a fogli di stile. Se infatti si ha l'accortezza di ridefinire i tag all'interno della definizione degli stili, si hanno molte occasioni di utilizzare una formattazione mirate a seconda della funzione del contenuto: in quest'ottica, il fatto che alcuni tag logici non restituiscano nessun rendering particolare è addirittura un invito a ri-definire lo stile del tag.

Lezione successiva

[Sommario]

TORNA SU

Scegliere il font del testo

La presente lezione tratta la scelta del colore, delle dimensioni e del tipo di carattere del testo attraverso l'utilizzo del tag "font". Si tratta di un **argomento obsoleto**, perché la formattazione del testo in tutti i siti moderni viene attribuita attraverso i fogli di stile. L'utilizzo del tag **<font>** inoltre è disapprovato dal W3C, e dunque sta cadendo in disuso. In ogni caso si tratta di un argomento che un buon webmaster non può ignorare: come già detto per studiare i fogli di stile ci sarà tempo, e comunque è un passo che viene dopo la conoscenza dell'HTML.

Il tipo di carattere (cioè il "font") che il browser visualizza di default è il "Times".

Purtroppo questo carattere (ottimo per la carta stampata) non è adatto a essere visualizzato sul monitor di un computer: è una questione di "grazie" (le grazie sono quegli abbellimenti tipografici delle lettere, che dovrebbero servire per rendere più leggibile il carattere).

Dal momento che i caratteri con grazie non ottengono il risultato voluto sul monitor (quello cioè di rendere le lettere maggiormente riconoscibili e di conseguenza il testo più leggibile), ma anzi ottengono l'effetto contrario, si preferisce di solito utilizzare dei caratteri senza grazie come il "Verdana", l'"Arial" o l'"Helvetica" (si veda l'articolo I font e la tipografia del testo).

Per scegliere il tipo di carattere con cui un font deve essere visualizzato è sufficiente usare la sintassi:

`<font face="Arial">testo in Arial</font>`	testo in Arial
`<font face="Verdana">testo in Verdana</font>`	testo in Verdana
`<font face="Geneva">testo in Geneva</font>`	testo in Geneva

Tuttavia è bene sottolineare da subito che non è possibile far sì che l'utente visualizzi un testo in un carattere fantasioso scelto da noi. Allo stato attuale dell'arte l'utente che naviga in internet può visualizzare solo i caratteri che sono installati nel suo sistema: in Windows si tratta dei caratteri presenti in: **Pannello di controllo > Tipi di caratteri**.

Se ad esempio scarichiamo da http://font.html.it il carattere Hackers, lo scompattiamo e lo inseriamo nella cartella dei caratteri, saremo poi in grado di visualizzare sul nostro computer il testo in Hackers.

Ma quando metteremo il nostro sito nel web gli utenti visualizzeranno un semplicissimo Times. Come nell'esempio sotto indicato:

`<font face="hackers">testo in hackers</font>`	**testo in hackers**

Per questo motivo è bene tener conto di due accorgimenti:

- scegliere caratteri "sicuri" , che siano cioè senz'altro presenti sul pc dell'utente
- non indicare un solo carattere, ma una serie di caratteri che gradualmente si allontanano dal risultato che vorremmo ottenere, ma non di molto, fino ad indicare la famiglia a cui il nostra carattere appartiene. In questo modo il browser dell'utente cercherà di trovare nella propria cartella dei fonts il primo carattere indicato, se non lo trova passerà al secondo, e solo come ultima spiaggia sceglierà si utilizzare il carattere predefinito (il famigerato "Times")

Vediamo alcuni esempi di famiglie "sicure" di caratteri:

`<font face="Verdana, Arial, Helvetica, sans-serif">Verdana e caratteri simili</font> `	**Verdana e caratteri simili**
`<font face="Arial, Helvetica, sans-serif">Arial e caratteri simili</font> `	**Arial e caratteri simili**
`<font face="Times New Roman, Times, serif">Times e caratteri simili</font>  `	**Times e caratteri simili**
`<font face="Courier New, Courier, mono">Curier e caratteri simili</font> `	**Curier e caratteri simili**
`<font face="Georgia, Times New Roman, Times, serif">Georgia e caratteri simili </font>  `	**Georgia e caratteri simili**
`<font face="Geneva, Arial, Helvetica, sans-serif">Geneva e caratteri simili</font>`	**Geneva e caratteri simili**

È vero: l'impossibilità di scegliere i caratteri che preferiamo limita terribilmente le nostre possibilità espressive, ma il bello di sviluppare per il web è proprio accettare di creare con delle regole ben definite, e a volte anche molto vincolanti.

Per i titoli delle pagine, i menu, e quant'altro potremmo poi sempre utilizzare delle immagini con il nostro carattere tipografico preferito (ad esempio delle "gif").

L e z i o n e s u c c e s s i v a

[S o m m a r i o]

TORNA SU

Scegliere il colore del testo

Adesso che abbiamo scelto il carattere con cui scrivere il nostro testo possiamo scegliere il colore, con la sintassi:

```
<font color="blue">testo blu</font>

ovvero:

<font color="#0000FF">testo blu</font>
```

testo blu

ovvero

testo blu

La scelta del colore può essere effettuata nello stesso momento in cui si sceglie il tipo di carattere (dal momento che "face" e "color" sono entrambi attributi del tag "font"). La sintassi è:

```
<font face="Verdana, Arial, Helvetica, sans-serif" color="blue">

testo blu in Verdana

</font>
```

testo blu in Verdana

Una volta scelto il colore possiamo sempre decidere di cambiarlo:

```
<font face="Verdana, Arial, Helvetica, sans-serif" color="blue">

testo blu in Verdana</font><br>

<font face="Verdana, Arial, Helvetica, sans-serif" color="red">

testo rosso

</font>

o meglio ancora:
```

testo blu in Verdana
testo rosso

o meglio ancora:

testo blu in Verdana
testo rosso

```
<font face="Verdana, Arial,
Helvetica, sans-serif" color="blue">

testo blu in Verdana<br>

<font color="red">

testo rosso

</font>

</font>
```

La seconda sintassi è preferibile alla precedente, perché la scelta del tipo di carattere viene effettuata una sola volta, evitando così di scrivere del codice inutile. Da notare che per evitare la ripetizione i due tag sono annidati l'uno dentro l'altro.

Le dimensioni del testo

Le dimensioni del testo si attribuisco mediante l'attributo "**size**" del tag font.

Ci sono due modi per dare attribuire le dimensioni al testo tramite il tag <font>:

- **valori interi da 1 a 7**
- **valori relativi** alla dimensione di base del tag font (di default "3")

Nel caso dei **valori interi**, ecco la scala di grandezza:

```
<font size="1">testo di
grandezza 1</font><br>

<font size="2">testo di
grandezza 2</font><br>

<font size="3">testo di
grandezza 3</font><br>

<font size="4">testo di
grandezza 4</font><br>

<font size="5">testo di
grandezza 5</font><br>

<font size="6">testo di
grandezza 6</font><br>

<font size="7">testo di
grandezza 7</font><br>
```

testo di grandezza 1

testo di grandezza 2

testo di grandezza 3

testo di grandezza 4

testo di grandezza 5

testo di grandezza 6

testo di grandezza

	7

Nel caso dei valori relativi alla dimensione di base è possibile "spostarsi" nella scala di grandezza del <font> utilizzando i segni "+" e "-".

Abbiamo detto che la grandezza del font di base di default nel browser è 3.

Dunque se utilizziamo un size="+2", vuol dire che la dimensione del font deve essere di 2 misure più grande della dimensione del font di base, quindi avremo un font di grandezza 5. Vediamo l'esempio:

```
<font size="+2">
Testo di grandezza +2 rispetto al
font di base (3).<br>
Cioè font di grandezza 5.
</font>
<br><br>
<font size="5">
Testo di grandezza 5.
</font>
```

Testo di grandezza +2 rispetto al font di base (3).
Cioè font di grandezza 5.

Testo di grandezza 5.

Come si può vedere le due sintassi sono equivalenti.

La grandezza del font di base può anche esser cambiata:

```
<basefont size="1">

<font size="+2">
Testo di 2 grandezze superiore al font di base, sopra definito.
</font>
<br>
<font size="3">
Testo di grandezza 3.
</font>
<br><br>

<basefont size="2">

<font size="+2">
Testo di 2 grandezze superiore al font di base, sopra ridefinito.
</font>
<br>
<font size="3">
Testo di grandezza 3.
</font>
```

Come si può vedere nella pagina esemplificativa.

È importante evitare di cadere nell'errore di pensare che la dimensione relativa faccia riferimento al precedente tag font. La dimensione relativa fa sempre riferimento alla dimensione del font di base:

Ecco un esempio corretto, ma che non darà il risultato desiderato, perché la dimensione relativa fa sempre riferimento al <basefont>: Testo di grandezza 7 testo di grandezza inferiore di 1 al font di base (che di default è 3), NON al tag precedente 	Testo di grandezza 7 testo di grandezza inferiore di 1 al font di base (che di default è 3), NON al tag precedente

Anche se non è corretto farlo, Internet Explorer consente di utilizzare il tag **<basefont>** per impostare in una sola volta il tipo di carattere del testo e il suo colore, come si può vedere nell'esempio.

Tuttavia questo tipo di trucco non funziona correttamente né con Mozilla (e quindi neanche con Netscape 6 o superiore, dal momento che eredita il motore di rendering di Mozilla), né con Opera.

NOTA BENE

Quando state utilizzando il tag <font> - sia che utilizziate il size i valori interi, sia che utilizziate le i valori relativi al tag di base -, in realtà la grandezza del carattere **dipende dalle impostazioni del browser dell'utente** (come già abbiamo visto per i tag "heading").

Con Internet Explorer ad esempio andando in: **Visualizza > Carattere**.

Se cambiate le dimensioni del carattere, vedrete cambiare le dimensioni dei font.

Questo appunto per le grandezze da 1 a 7 sono grandezze anch'esse relative.

Questa caratteristica da un lato è positiva (permette di ingrandire testi piccoli), dall'altra può risultare molto fastidiosa per il webmaster.

L'unico modo per fissare il carattere è (ancora una volta) quello di utilizzare i fogli di stile, esprimendo le dimensioni in pixel.

Lezione successiva

[Sommario]

TORNA SU

Gli elenchi nell'HTML

Se abbiamo la necessità di inserire un elenco di termini, possiamo utilizzare le "liste", che sono sostanzialmente di tre tipi:

- **Elenchi ordinati**
- **Elenchi non ordinati**
- **Elenchi di definizioni**

Tutti e tre i tipi di elenchi funzionano nel medesimo modo: si apre il tag, si elencano i vari elementi della lista (ciascuno con il proprio tag), si chiude il tag dell'elenco. La sintassi ha quindi questa forma:

```
<elenco>
        <elemento>nome del primo elemento
        <elemento>nome del secondo elemento
</elenco>
```

come si può vedere, il tag che individua l'elemento della lista non ha bisogno di chiusura (la sua chiusura, in questo caso, è opzionale).
Le liste di definizioni hanno una struttura leggermente diversa che vedremo a breve.

Gli elenchi ordinati

Gli elenchi ordinati sono contraddistinti dall'enumerazione degli elementi che compongono la lista. Avremo quindi una serie progressiva ordinata e individuata da lettere o numeri (se utilizzate un programma di videoscrittura, siete abituati a chiamarli **elenchi numerati**).
Il tag da utilizzare per aprire un elenco ordinato è **<ol>** ("ordered list") e gli elementi sono individuati dal tag **<li>** ("list item"):

```
<div>Testo che precede la lista
  <ol>
    <li>primo elemento
    <li>secondo elemento
    <li>terzo elemento
  </ol>testo che segue la lista
</div>
```

Testo che precede la lista

1. primo elemento
2. secondo elemento
3. terzo elemento

testo che segue la lista

Da notare che il tag che individua l'elenco lascia una riga di spazio prima e dopo il testo che eventualmente lo circonda (come avviene per il **<p>**); fa eccezione però l'inclusione di un nuovo elenco all'interno di un elenco preesistente: in questo caso non viene lasciato spazio, né prima, né dopo.
Gli elementi dell'elenco sono sempre rientrati di uno spazio verso destra: tutto questo serve a individuare in modo inequivocabile l'elenco.

Lo stile di enumerazione visualizzata di default dal browser è quello numerica, ma è possibile

indicare uno stile differente specificandolo per mezzo dell'**attributo type**. Ad esempio:

```
<ol type="a">
      <li>primo elemento
      <li>secondo elemento
      <li>terzo elemento
</ol>
```

Gli stili consentiti sono:

Valore dell'attributo type	Stile di enumerazione		
type="1" (è così di default)	**numeri arabi**	<ol type="1"> primo secondo terzo 	1. primo 2. secondo 3. terzo
type="a"	**alfabeto minuscolo**	<ol type="a"> primo secondo terzo 	a. primo b. secondo c. terzo
type="A"	**alfabeto maiuscolo**	<ol type="A"> primo secondo elemento terzo 	A. primo B. secondo C. terzo
type="i"	**numeri romani minuscoli**	<ol type="i"> primo secondo elemento terzo 	i. primo ii. secondo iii. terzo
type="I"	**numeri romani maiuscoli**	<ol type="I"> primo secondo elemento terzo 	I. primo II. secondo III. terzo

Gli elenchi non ordinati

Gli elenchi non ordinati sono individuati dal tag **<ul>** ("unordered list"), e gli elementi dell'elenco sono contraddistinti anch'essi dal tag <li> (in buona sostanza si tratta di quello che i programmi di videoscrittura chiamano **elenchi puntati**):

```
<ul>
  <li>primo elemento
  <li>secondo elemento
  <li>terzo elemento
 </ul>
```

il tipo di segno grafico utilizzato per individuare gli elementi dell'elenco di default dipende dal browser, ma di solito è un "pallino pieno". È possibile comunque scegliere un altro tipo di segno:

Valore dell'attributo type	Stile di enumerazione		

type="disc" (è così di default)	visualizza un **"pallino" pieno**. È la visualizzazione di default	`<ul type="disc">`
		`<li>primo`
		`<li>secondo`
		`<li>terzo`
		`</ul>`
			• primo
			• secondo
			• terzo

type="circle"	visualizza un **cerchio vuoto** al proprio interno	`<ul type="circle">`
		`<li>primo`
		`<li>secondo`
		`<li>terzo`
		`</ul>`
			○ primo
			○ secondo
			○ terzo

type="square"	Visualizza un **quadrato pieno** al proprio interno	`<ul type="square">`
		`<li>primo`
		`<li>secondo elemento`
		`<li>terzo`
		`</ul>`
			▪ primo
			▪ secondo
			▪ terzo

Da notare inoltre che il tipo di segno grafico, varia in automatico al variare dell'annidamento della lista. Ad esempio:

```
<ul>
<li>primo della 1a lista
<li>secondo della 1a lista

<ul>
  <li>primo della 2a lista
  <li>secondo della 2a lista

  <ul>
    <li>primo della 3a lista
  </ul>

  </ul>

  <li>terzo della 1a lista

</ul>
```

- primo della 1a lista
- secondo della 1a lista
  - primo della 2a lista
  - secondo della 2a lista
    - primo della 3a lista
- terzo della 1a lista

Elenchi di definizioni

Gli elenchi di definizioni sono individuati dal tag **<dl>**. Gli elementi dell'elenco (a differenza delle

liste ordinate, e delle liste non ordinate) questa volta sono formati da due parti:

<dt>	**definition term**: indica il termine da definire. A differenza dell'elemento ** ** in questo caso non c'è rientro.
<dd>	**definition description**: è la definizione vera e propria del termine. L'elemento è rientrato.

Vediamo un esempio:

```
<p>Ecco i principali tag per delimitare il testo:
<dl>
<dt><p>
<dd>individua l'apertura di un nuovo paragrafo

<dt><div>
<dd>individua l'apertura di un nuovo blocco di testo

<dt><span>
<dd>individua l'apertura di un elemento inline,
cui attribuire una formattazione atraverso gli stili
</dl>
ci sono poi altri tag che...
</p>
```

Ecco i principali tag per delimitare il testo:

<p>
: individua l'apertura di un nuovo paragrafo

: individua l'apertura di un nuovo blocco di testo

: individua l'apertura di un elemento inline, cui attribuire una formattazione atraverso gli stili

ci sono poi altri tag che...

Approfondimenti

Ovviamente la scelta del tipo di elenco attraverso l'attributo **type** è deprecato dal W3C, perché si tratta di una caratteristica che riguarda la formattazione, e dunque andrebbe effettuata utilizzando i CSS. Con i fogli di stile c'è anche la possibilità di scegliere un'immagine (ad esempio una GIF) come segno distintivo per l'elenco puntato. Chi fosse interessato ad approfondire può consultare la relativa lezione della guida ai fogli di stile.

Lezione successiva

[Sommario]

TORNA SU

I link e l'ipertestualità

Una delle caratteristiche che ha fatto la fortuna del web è l'essere costituito non da **testi** ma da **ipertesti** (un'altra delle caratteristiche che hanno fatto grande il web è senz'altro la possibilità di interagire, ma questo è un altro discorso).

I link sono "il ponte" che consente di passare da un testo all'altro. In quanto tali, i link sono formati da due componenti:

• il contenuto che "nasconde" il collegamento (non importa se si tratta di testo o di immagine)	È la parte visibile del link, e proprio per questo l'utente deve essere sempre in grado di capire quali sono i collegamenti da cliccare all'interno della pagina
• la risorsa verso cui il collegamento punta	Si tratta di un'altra pagina (sullo stesso server o su un server diverso), oppure è un collegamento interno a un punto della pagina stessa

Di solito per spiegare che cosa sono i link si utilizza la metafora dell'ancora con "la testa" all'interno del documento stesso, e la "coda" in un altro documento (o all'interno di un altro punto del documento stesso).

Link che puntano ad altri documenti

Ecco la sintassi per creare un link con riferimento a un sito web:

```
Le risorse per webmaster sono su <a href="http://www.html.it/">HTML.IT</a>.
```

Che dà come risultato: 'Le risorse per webmaster sono su HTML.IT'.

Come si può intuire la testa della nostra àncora è il testo "HTML.IT", mentre la coda, cioè la destinazione (specificata dall'attributo **href**) è il sito web verso cui il link punta, cioè http://www.html.it.

È indifferente che la destinazione dell'ancora sia una pagina HTML di un sito, un'immagine, un file pdf , un file zip, o un file exe: il meccanismo del link funziona allo stesso modo indipendentemente dal tipo di risorsa; poi il browser si comporterà in modo differente a seconda della risorsa. Ad esempio:

Immagine .gif, .jpg, .png	Viene visualizzata nel browser
Documento .html, .pdf, .doc	La pagina è visualizzata nel browser. Nel caso dei documenti **.doc** e **.pdf** l'utente deve avere installato sul proprio pc l'apposito plugin (nella maggior parte dei casi è sufficiente che abbia installato rispettivamente Microsoft Word e Adobe Acrobat Reader). Se non è installato il plugin il sistema chiederà all'utente se salvare il file.
File .zip, file .exe	Viene chiesto all'utente di scaricare il file **NOTA bene: per motivi di sicurezza non è possibile eseguire un file ".exe" direttamente dal web; l'utente dovrà sempre prima scaricarlo sul proprio PC.**

Potete anche specificare un indirizzo è mail. In questo caso si aprirà direttamente il client di posta dell'utente con l'indirizzo e-mail pre-impostato. La sintassi è la seguente:

```
<a href="mailto:tuaMail@nomeTuoSito.it">Mandami un'e-mail</a>
```

Che dà come risultato: Mandami un'e-mail

Lezione successiva

[Sommario]

TORNA SU

I percorsi assoluti e relativi

Percorsi assoluti

Fino a quando ci troviamo nella condizione di creare un sito web di dimensioni ridotte (poche pagine) non avremo problemi di complessità, e possiamo anche ipotizzare di lasciare tutti i nostri file in una medesima cartella. È evidente però che – man mano che il nostro sito web cresce – avremo bisogno di un maggior ordine. Si presenterà allora l'esigenza di inserire le immagini del sito in una cartelle diverse (in modo da averle tutte nella medesima locazione), e magari sarà opportuno dividere il sito in varie sezioni, in modo da avere tutti i documenti dello stesso tipo all'interno di un contesto omogeneo.

I siti web sono dunque organizzati in strutture ordinate: non a caso si parla di **albero di un sito**, per indicare la visualizzazione della struttura alla base del sito.

Poiché l'organizzazione di un sito in directory e sottodirectory è una cosa normalissima, dobbiamo imparare a muoverci tra i vari file che costituiscono il sito stesso, in modo da essere in grado di creare collegamenti verso i documenti più reconditi, destreggiandoci tra le strutture più ramificate.

Per farlo esistono due tecniche:

- **indicare un percorso assoluto**
- **indicare un percorso relativo**

Nel caso in cui il documento a cui vogliamo puntare si trovi in una particolare directory del sito di destinazione, con i percorsi assoluti non abbiamo che da indicare il percorso per esteso.
Se esaminiamo:

```
Leggi le risorse sui <a href="http://www.html.it/css/index.html">fogli di stile</a>
```

Possiamo vedere chiaramente che il link indica un percorso assoluto e fa riferimento a una particolare directory. Nella fattispecie:

http://	Indica al browser di utilizzare il protocollo per navigare nel web (l'http)
www.html.it/	Indica di fare riferimento al sito www.html.it
css/	Indica che la risorsa indicata si trova all'interno della cartella "css"
index.html	Indica che il file da collegare è quello chiamato "index.html"

Insomma, per creare un collegamento assoluto è sufficiente fare riferimento all'url che normalmente vedete scritto nella barra degli indirizzi. I percorsi assoluti si usano per lo più, quando si ha la necessità di fare riferimento a risorse situate nei siti di

terze persone.

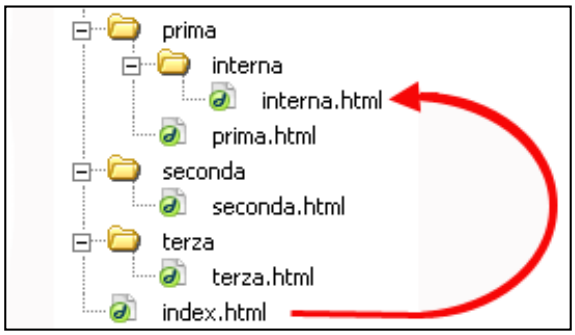
Percorsi relativi

Spesso vi troverete tuttavia a fare riferimento a documenti situati nel vostro stesso sito, e – se state sviluppando il sito sul vostro computer di casa (cioè “in locale”) – magari non avete ancora un indirizzo web, e non sapete di conseguenza come impostare i percorsi. È utile allora capire come funzionano i percorsi relativi.

I percorsi relativi fanno riferimento alla posizione degli altri file rispetto al documento in cui ci si trova in quel momento.
Per linkare due pagine che si trovano all’interno della stessa directory è sufficiente scrivere:

```
<a href=”paginaDaLinkare.html”>collegamento alla pagina da linkare nella stessa directory della pagina presente</a>
```

Poniamo ora di trovarci in una situazione di questo genere:

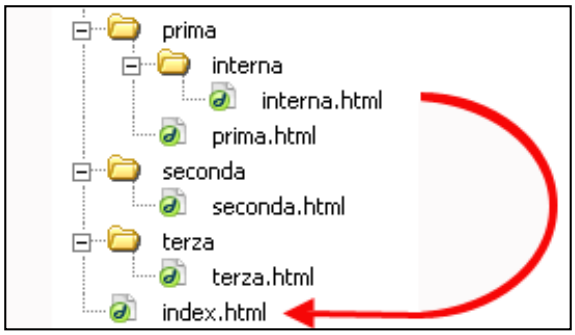


Dalla pagina “index.html” vogliamo cioè far riferimento al file “interna.html”, che si trova all’interno della directory “interna”, che a sua volta si trova all’interno della directory “prima”.

La sintassi è la seguente:

```
<a href=”prima/interna/interna.html”>Visita la pagina interna</a>
```

Vediamo adesso l’esempio opposto: dalla pagina interna vogliamo far riferimento a una pagina (“index.html”) che si trova più in alto di due livelli:



La sintassi è la seguente:

```
<a href=”../../index.html”>Visita la pagina interna</a>
```

Come si vede, con i percorsi relativi valgono le seguenti regole generali:

Per far riferimento a un file che si trovi all'interno della stessa directory basta linkare il nome del file	collegamento alla pagina
Per far riferimento a un file contenuto in una cartella di livello inferiore alla posizione corrente, basta nominare la cartella seguita dallo "slash", e poi il nome del file. Secondo la formula: **cartella/nomeFile.html**	Visita la pagina interna
Per tornare su di un livello, è sufficiente utilizzare la notazione: **../nomeFile.html**	Visita la pagina interna

Grazie a questi accorgimenti potete agevolmente navigare all'interno delle directory del vostro sito: se ce ne fosse bisogno potrete per esempio tornare su di un livello rispetto alla posizione del file, scegliere un'altra cartella, e poi scegliere un altro file:

```
../altraCartella/nuovoFile.html
```

Per approfondimenti potete consultare la pagina d'esempio.

Approfondimenti

A volte potrete incontrare la notazione:

```
Leggi le risorse sui <a href="/css/index.html">fogli di stile</a>
```

Se il vostro sito è all'interno di un server Unix (ma la sintassi funziona anche in sistemi Windows, basta che non siano in locale), questa notazione non deve stupirvi: il carattere / indica la directory principale del sito, altrimenti detta **"root"**. Dunque <a href="/css/index.html"> è un altro modo di esprimere i percorsi assoluti all'interno del proprio sito.

Un'altra cosa importante da sapere è che quando metterete il vostro sito all'interno dello spazio web, l'indicazione della index all'interno di una directory è facoltativa. Al posto di questo:

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

è sufficiente indicare la directory:

```
http://www.html.it/css/
```

Verificate solo con il vostro gestore dello spazio web (cioè "hosting"), se le pagine index della directory devono avere forma **index.html, index.htm, index.asp, index.php, home.asp**, o altro.

Consigli per i nomi dei file

Quando mettere nel web il vostro sito internet, vi accorgerete che esistono due famiglie di sistemi operativi: **Windows** e **Unix**. Questi due sistemi operativi utilizzano differenti modi per gestire i file, dunque alcuni accorgimenti sono necessari:

- è consigliabile non lasciare spazi vuoti nei nomi dei file (gli spazi vuoti non sempre vengono interpretati correttamente), meglio ovviare a questa necessità con un “trattino basso” (cioè “_”). Ad esempio: **mio_file.html**
- maiuscole e minuscole possono fare la differenza (in ambiente Unix spesso la fanno), quindi controllate il modo in cui avete scritto i file

Inoltre quando create un collegamento state attenti a non avere una notazione simile a questa:

```
<a href="file:///C|percorso\nomeFile.html">testo</A>
```

significa che state facendo un riferimento (assoluto) al vostro stesso computer: chiaro che quando metterete i file nel vostro spazio web, le cose non funzioneranno più.

L e z i o n e s u c c e s s i v a

[S o m m a r i o]

TORNA SU

I link interni o ancore

È possibile anche creare un indice interno al documento, utilizzando le àncore. Ciascuna àncora può avere infatti un nome:

```
<a name="primo">Stiamo per esaminare la struttura…. Eccetera…</a>
```

Da notare che in mancanza dell'attributo che indica il collegamento (href) le àncore non vengono viste come link, ma la loro formattazione è indistinguibile dal "normale" testo.

In un ipotetico indice è allora possibile far riferimento all'àncora presente all'interno del documento attraverso un link che punti ad essa:

```
<a href="#primo">vai al primo paragrafo</a>
```

il cancelletto indica che il collegamento deve cercare un àncora chiamata "primo" all'interno della pagina stessa.

Se non si specifica il nome dell'àncora a cui si vuol puntare, viene comunque creato un link che punta ad inizio pagina (viene cercata un'àncora il cui nome non è specificato). Questo infatti è un ottimo escamotage per creare link "vuoti" (in alcuni casi vi occorreranno). Ad esempio:

```
<a href="#">link vuoto</a>
```

Per creare un indice interno alla pagina si procede dunque in due fasi distinte:

- creazione dell'ancora a cui puntare (**<a name="mioNome">**)
- creazione del collegamento all'ancora appena creata e riferimento attraverso il cancelletto (**<a href="#mioNome">**)

È bene non confondere le due fasi.

Un esempio di quanto appena esposto lo potete trovare nella pagina dell'esempio.

Lezione successiva

[Sommario]

TORNA SU

Inserire le immagini

Finora abbiamo visto come inserire e formattare il testo all'interno delle nostre pagine Web. Naturalmente possiamo inserire anche delle immagini: diagrammi e grafici, fotografie, e in genere immagini create con un programma di elaborazione grafica (come The GIMP, Photoshop o Paint Shop Pro).

I formati ammessi nel Web sono sostanzialmente tre:

- **GIF** (**Graphic Interchange Format**). Le GIF sono immagini con non più di 256 colori (dunque con colori piatti e senza sfumature), come grafici o icone
- **JPG**: è l'acronimo del gruppo di ricerca che ha ideato questo formato (il **Joint Photographic Experts Group**), idoneo per le immagini di qualità fotografica
- **PNG** (**Portable Network Graphic**). Il PNG è un tipo di immagine introdotto abbastanza di recente, elaborato dal W3C per risolvere i problemi di copyright del formato GIF (che è appunto proprietario); tuttavia oggi il PNG è letto oramai da tutti i browser e offre alcune caratteristiche che gli altri formati non hanno (come il supporto al canale alfa, caratteristica questa non ancora perfettamente supportata da ogni browser).

Non provate dunque a inserire un file ".psd" (è il formato nativo di Photoshop) all'interno della vostra pagina HTML: con grande probabilità il browser non vi caricherà il file che vorreste includere (dovete infatti prima convertire il file in uno dei formati sopra-indicati).

Inoltre è importante ricordare che il codice HTML fornisce delle indicazioni al browser su come visualizzare il testo e le immagini - ed eventualmente i video e i suoni - all'interno della pagina: il testo (come abbiamo visto) è scritto direttamente nel file HTML, le immagini invece sono caricate insieme alla pagina. Attenzione dunque a non inserire immagini troppo pesanti (ricordatevi di ottimizzare sempre i file); bisogna evitare inoltre di sovraccaricare la pagina con troppe immagini. Allo stato attuale dell'arte infatti la maggior parte degli utenti (e non soltanto quelli italiani) naviga ancora con un modem analogico da 56 Kbs: inserire troppe immagini significa dunque creare pagine lente da caricare. Per ottenere un sito web dalla grafica accattivante, spesso è sufficiente giocare con i colori dello sfondo e delle scritte.

La sintassi per inserire un'immagine è:

```
<img src="miaImmagine.gif">
```

dove:

- **img** significa **image**, cioè **immagine**
- **src** significa **source**, cioè origine

Il tag è un tag "vuoto", che non ha la necessità di essere chiuso.

Ecco ad esempio come inserire il logo di HTML.it in una pagina dallo sfondo blu (si presuppone che il logo si trovi nella stessa cartella del file HTML):

```
<img src="logo.gif">
```

Resta valido il discorso sui percorsi relativi ed assoluti. Avremo ad esempio:

```
<img src="../img/logo.gif">
<img src=" http://www.html.it/img/logo.gif ">
```

Dal momento che il browser normalmente non sa quali siano le dimensioni dell'immagine, finché questa non sia caricata completamente, è un'ottima abitudine quella di indicare già nel codice la larghezza (**width**) e l'altezza (**height**) dell'immagine: in questo modo si evita di vedere la pagina costruirsi man mano che viene caricata, poiché stiamo dando al browser un'idea dell'ingombro. Ad esempio:

```
<img src="logo.gif" width="224" height="69">
```

L'attributo **alt** è utile per specificare il **testo alternativo** (**alternative text**), fintanto che l'immagine non viene caricata o nel caso in cui non lo sia affatto:

```
<img src="logo.gif"
alt="HTML.it"
width="224" height="69">
```

L'attributo **alt** è di estrema utilità per rendere il sito accessibile a tutti gli utenti: i disabili che non sono in grado di vedere nitidamente le immagini sullo schermo potrebbero avere delle difficoltà, nel caso in cui l'attributo alt non sia specificato. Gli ipo-vedenti e i non-vedenti sono infatti in grado di comprendere il contenuto delle immagini grazie a dei software appositi (gli **screen reader**) che "leggono" lo schermo tramite un programma di sintesi vocale. Non specificare il testo alternativo significa rendere impossibile la navigazione.

Nel caso in cui la spiegazione dell'immagine sia particolarmente lunga, è possibile espandere la descrizione sintetica - fornita tramite l'attributo "alt" - grazie ad un altro attributo: si tratta di **longdesc** (**long description**), che permette di specificare un file con una spiegazione estesa dell'immagine. Ecco la sintassi:

```
<img src="logo.gif" alt="HTML.it" longdesc="descrizione.html" width="224"
height="69">
```

Nell'esempio allegato è possibile visualizzare il codice di una pagina con la descrizione estesa dell'immagine. Nel caso in cui si utilizzi questo attributo è anche buona norma utilizzare un link esplicito alla pagina della descrizione.

longdesc dovrebbe essere utilizzato soprattutto nel caso in cui si usino delle immagini mappate (argomento che analizzeremo in seguito), in modo da fornirne una spiegazione esauriente in ogni contesto.

In realtà l'attributo **alt** non serve, come molti credono, a visualizzare un etichetta esplicativa dell'immagine nel caso in cui il cursore del mouse si soffermi sopra essa: questo semmai è un effetto collaterale che si verifica con Internet Explorer. L'attributo corretto per far visualizzare un testo che commenti l'immagine è infatti **title**:

```
<img src="logo.gif"
title="HTML.it"
alt="HTML.it"
width="224" height="69">
```

È inoltre possibile specificare la grandezza (in pixel) del bordo attorno all'immagine:

```
<img src="logo.gif"
border="3"
width="224" height="69">
```

Si noti che i link lasciano **sempre** di default un bordo di un pixel attorno all'immagine (il colore sarà quello espresso nel body dall'attributo **link**, oppure quello default – quindi blu – se non specificato altrimenti):

```
<a href="http://www.html.it"
target="_blank">
  <img src="logo.gif"
   width="224" height="69">
</a>
```

Dunque, nel caso dei link se non si desidera avere i bordi, sarà necessario impostarli a **"0"**:

```
<a href="http://www.html.it"
target="_blank">
  <img src="logo.gif"
   border="0"
   width="224" height="69">
</a>
```

Lezione successiva

[Sommario]

TORNA SU

Tabella: struttura di base

Le tabelle sono una delle parti più importanti di tutto il codice HTML: nate sin dagli inizi del Web per impaginare dati aggregati, si sono poi trasformate in uno strumento indispensabile per gestire i layout grafici.
Il loro ampio utilizzo all'interno dei documenti ha fatto sì che – nel passaggio dall'HTML 3.2 all'HTML 4 - le specifiche delle tabelle venissero estese con una serie di notazioni destinate a "far ordine" all'interno di un codice che rischiava di diventare troppo vasto.

Immaginiamo la nostra prima tabella come una griglia formata da righe e colonne. I tag necessari per creare una tabella sono:

<table>	apre la tabella
<tr>	**"table row"**: indica l'apertura di una riga
<td>	**"table data"**: indica una cella all'interno di una riga

In questi nostri primi esempi presupponiamo che il numero delle celle all'interno di ciascuna riga sia costante: ogni riga avrà cioè lo stesso numero di celle. Ci sono dei metodi per variare il numero delle celle all'interno di una riga, ma li vedremo in seguito.

L'attributo **border** permette di specificare di quanti pixel deve essere il bordo delle tabelle. Ad esempio:

```
<table border="2">
```

Lo useremo in questi esempi, altrimenti non percepiremmo la struttura di quanto stiamo costruendo. Ecco un primo esempio di tabella:

```
<table border="1>
   <tr>
      <td>prima cella</td>
      <td>seconda cella</td>
  </tr>

  <tr>
     <td>terza cella</td>
     <td>quarta cella</td>
  </tr>
</table>
```

Che viene visualizzato così:

prima cella	seconda cella
terza cella	quarta cella

Possiamo specificare la larghezza e l'altezza delle tabelle tramite gli attributi **width** e **height** che possono essere riferiti a tutti e tre i tag (**<table>**, **<tr>**, **<td>**). Il valore di questi attributi può essere specificato con una larghezza fissa (in pixel: in questo

caso basta indicare un numero intero), oppure in percentuale (il numero deve essere allora seguito dal simbolo “%”): in questo caso la tabella si adatta secondo lo spazio a disposizione.

```
<table width="300" height="200" border="1>
   <tr>
      <td>prima cella</td>
      <td>seconda cella</td>
  </tr>

  <tr>
     <td>terza cella</td>
     <td>quarta cella</td>
  </tr>
</table>
```

Che viene visualizzato così:

prima cella	seconda cella
terza cella	quarta cella

Oppure:

```
<table width="75%" border="1>
   <tr>
      <td width="25%">prima cella</td>
      <td width="75%">seconda cella</td>
  </tr>

  <tr>
     <td width="25%">terza cella</td>
     <td width="75%">quarta cella</td>
  </tr>
</table>
```

Che viene visualizzato così:

prima cella	seconda cella
terza cella	quarta cella

Di solito la larghezza e l’altezza globali della tabella sono espresse nel tag **<table>**, mentre la larghezza delle varie celle viene espressa nei **<td>** della prima riga. L’altezza in percentuale non sempre è visualizzata correttamente da tutti i browser.

Come detto inizialmente le tabelle vanno immaginate come delle griglie, tutto sommato abbastanza rigide: l’eventuale larghezza specificata nelle celle della prima riga avrà effetto dunque anche sulle celle delle righe sottostanti. Viceversa non è possibile variare arbitrariamente le dimensioni delle celle: le misure specificate nelle righe sottostanti non avranno infatti effetto, come si può vedere nell’esempio allegato,

che non è corretto.

Le dimensioni espresse non devono tuttavia essere in contraddizione ma mano che si procede verso l'interno della tabella: in un caso simile infatti "vincerebbe" il valore specificato nel tag genitore, come si può vedere nella pagina di esempio.

Inoltre (come si evince dagli esempi) la visualizzazione dei layout con indicazioni non corrette è a discrezione del browser, quindi si rischia di ottenere risultati diversi da quelli voluti.

Lezione successiva

[Sommario]

TORNA SU

Creare gruppi di righe: <caption>, <thead>, <tfoot>, <tbody>

Come dicevamo, la struttura delle tabelle ha letteralmente invaso le pagine HTML, che si sono riempite di **<tr>** e di **<td>**. Per portare un po' di ordine in questo caos nelle specifiche sono state introdotti dei tag (opzionali) che consentono di capire facilmente quali siano le diverse parti della tabella.

Per individuare facilmente i gruppi di righe sono stati introdotti i seguenti tag:

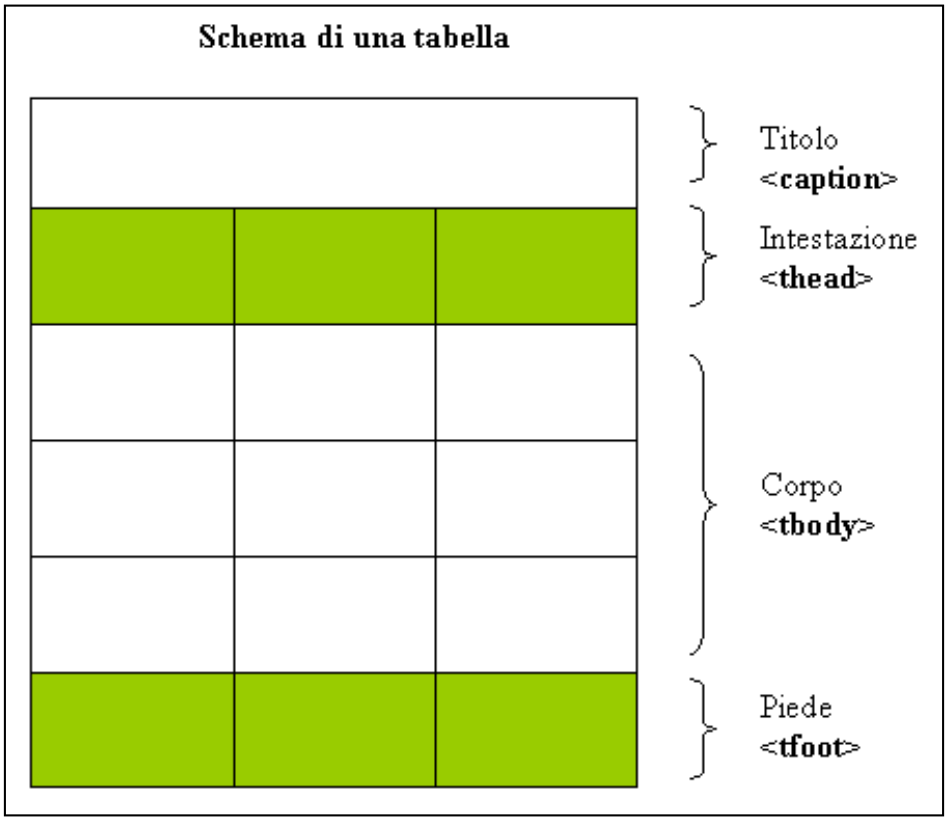
<caption>	è l'intestazione, il titolo con un commento esplicativo sulla tabella
<thead>	è la testa, la parte iniziale della tabella, quella che contiene ad esempio indicazioni sul contenuto delle celle
<tfoot>	è il piede, la conclusione della tabella, quella che consente ad esempio di tirare le somme
<tbody>	è il corpo, la parte centrale con il contenuto vero e proprio della tabella

<thead>, **<tfoot>**, **<tbody>** sono tag che consentono di individuare gruppi di righe (**"row group"**).

Da notare che – contrariamente a quello che si potrebbe pensare – il tag **<tfoot>** che chiude la tabella, è **anteposto** rispetto al **<tbody>**. L'idea di base è che il browser nell'eseguire il rendering del codice tenga conto della parte iniziale e della parte finale della tabella, e il corpo vero e proprio sia sviluppato nella sua interezza tra le due estremità.

Un'altra particolarità è che le celle all'interno del tag **<thead>** possono essere indicate con **<th>** (**"table header"**), al posto del consueto **<td>** (**"table data"**), in questo caso il contenuto delle celle è automaticamente formattato in grassetto e centrato.

Ecco comunque uno schema che riassume la struttura delle tabelle secondo l'HTML 4:



A questa pagina è possibile visualizzare un esempio.

L e z i o n e s u c c e s s i v a

[S o m m a r i o]

TORNA SU

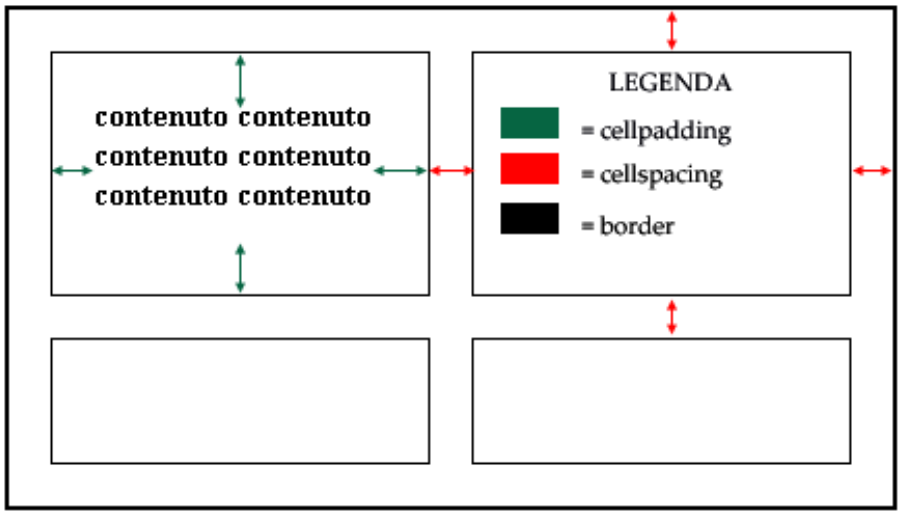
Attributi del tag table

Per quel che riguarda il tag **<table>**, i seguenti attributi che ci permettono di regolare le distanze tra i margini della tabella (o della cella) e il contenuto:

border	(che abbiamo già visto) specifica la larghezza dei bordi di una tabella (in pixel)
cellspacing	specifica la distanza (in pixel) tra una cella e l'altra, oppure tra una cella e il bordo. Di default è un pixel, dunque occorrerà sempre azzerarlo esplicitamente, quando non lo si desideri
cellpadding	indica la distanza tra il contenuto della cella e il bordo. Se il valore viene indicato con un numero intero, la distanza è espressa in pixel; il cellpadding tuttavia può anche essere espresso in percentuale. Di default la distanza è nulla

La dimensione indicata nel **cellpadding** e dal **cellspacing** - una volta specificata – ha effetto su tutti i lati della cella.

I rapporti tra gli attributi che abbiamo appena esaminato sono regolati come segue:



Con questa sintassi ad esempio si imposta una tabella con bordo di 1 pixel, senza spazio tra le celle e con il contenuto che è distanziato dai bordi della cella di 10 pixel:

```
<table width="75%" border="1" cellpadding="10" cellspacing="0">
```

come si può vedere nell'esempio.

per quel che riguarda l'attributo **border,** a partire da Internet Explorer 4 e da Netscape Navigator 6 è possibile modificare l'aspetto dei bordi esterni della tabella (tramite l'attributo **frames**) e delle righe fra le celle (tramite l'attributo **rules**).

Vediamo quali sono i possibili valori e i relativi esempi:

esempio	spiegazione
<table border="1" frame="above"> (nelle pagine di esempio qui a fianco le righe interne tra le celle sono state azzerate per facilitare la comprensione, dunque ci troveremo nella situazione in cui **rules="none"**)	Il bordo della tabella è presente: • **void**: in nessun lato. È il valore di default • **above**: solo nel lato superiore • **below**: solo nel lato inferiore • **hsides**: solo nei lati superiore e inferiore • **vsides**: solo a sinistra e a destra • **lhs**: solo nel lato sinistro (left-hand side) • **rhs**: solo nel lato destro (right hand side). • **box**: in tutti e quattro i lati • **border**: in tutti e quattro i lati
<table border="1" rules="rows"> (nelle pagine di esempio qui a fianco i bordi esterni della tabella sono stati azzerati per facilitare la comprensione, dunque ci troveremo nella situazione in cui **frame="void"**)	Le righe interne alle celle sono presenti: • **none**: da nessuna parte. È il valore di default • **groups**: le righe separano i gruppi (siano essi gruppi di righe: <thead>, <tfoot>, <tbody> - o gruppi di colonne: <colgroup>) • **rows**: le righe separano i vari <tr> • **cols**: le righe separano le colonne • **all**: le righe separano tanto i <tr>, quanto le colonne

Lezione successiva

[Sommario]

TORNA SU

Attributi di <table>, <tr>, <td>

I seguenti attributi invece hanno invece valore per tutti gli elementi della tabella (**<table>**, **<tr>**, **<td>**), li presenteremo quindi in un medesimo contesto.

Dimensioni

Abbiamo già esaminato gli attributi **width** e **height** che determinano la larghezza e l'altezza (in pixel o in percentuale) di tabelle, righe o celle.

Lo sfondo

Possiamo assegnare un colore di sfondo tramite l'attributo **bgcolor**, oppure un'immagine tramite **background**, come abbiamo già visto a proposito del tag <body>.

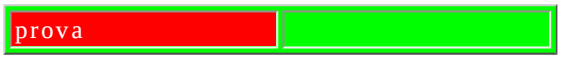
Vediamo un esempio:

bgcolor

codice:

```
<table width="75%" border="1"
align="center" bgcolor="#00FF00">
    <tr>
         <td width="50%" bgcolor="#FF0000">
              <font color="#FFFFFF">prova</font>
          </td>
         <td width="50%"> 
         </td>
     </tr>
</table>
```

visualizzazione:



background

codice:

```
<table width="75%" border="1"
align="center" bgcolor="#00FF00">
    <tr>
         <td width="50%"
             background="tabelle/sfondo.gif">
              <font color="#FFFFFF">prova</font>
          </td>
         <td width="50%"> 
         </td>
     </tr>
</table>
```

visualizzazione:

Come già nel **<body>** l'immagine di sfondo viene ripetuta, ed è possibile specificare entrambi gli attributi (**bgcolor** e **background**) all'interno dello stesso tag:

```
<td width="50%" bgcolor=”#0000FF” background="tabelle/sfondo.gif">
```

L'allineamento

L'attributo **align**, se riferito al tag **<table>**, sposta la tabella rispettivamente a sinistra (**align="left"** – è così di default), a destra (**align="right"**), o al centro (**align="center"**) del documento. Es:

```
<table align=”right”>
```

Se riferito a **<tr>** o a **<td>** è invece il contenuto delle celle ad essere allineato a sinistra, al centro oppure a destra. Es:

```
<td align=”right”>
contenuto
</td>
```

Allo stesso modo **valign** è utile per l'allineamento verticale delle celle. I valori possibili sono **top** (alto), **middle** (in mezzo – è il valore di default), **bottom** (in basso), **baseline** (alla linea di base). Es:

```
<td height=”100” valign=”middle”>
contenuto
</td>
```

Colori dei bordi

Per i bordi esistono gli attributi **bordercolor**, **bordercolorlight**, **bordercolordark**. Ad esempio:

```
<table border="2" bordercolor="blue" bordercolorlight="#00CCFF"
bordercolordark="#000099">
```

Questi attributi - che consentono di creare degli effetti bellissimi - sono visualizzati correttamente soltanto da Internet Explorer, mentre con gli altri browser (Mozilla, Opera) verranno visualizzati in modo parziale se non scorretto.
In realtà il modo corretto per attribuire un colore al bordo è quello di utilizzare i CSS.

Ci sono tuttavia delle soluzioni - utilizzate dagli sviluppatori sin dall'HTML 3 – che permettono di mostrare un filetto colorato attorno alle tabelle. La tecnica di solito è quella di lasciar trasparire il colore di sfondo attraverso lo spazio fra le celle. Vediamo un esempio:

```
<table width="450" bgcolor="#00CCFF" cellpadding="10" cellspacing="1">
  <tr bgcolor="FFFFFF">
      <td width="50%"><b>contenuto</b></td>
      <td width="50%"> </td>
```

```
    </tr>
</table>
```

che dà:

contenuto	

nowrap

Grazie all'attributo **nowrap** si può far sì che il contenuto di una cella non vada a capo, a meno che non lo forziamo espressamente con un **
** (è un **"break**", cioè un' "interruzione"):

```
<table width="100" border="1">
  <tr>
    <td nowrap>
        Se non lo vogliamo non va a capo.<br>Qui va a capo.
    </td>
  </tr>
</table>
```

cioè:

Se non lo vogliamo non va a capo. Qui va a capo.

Approfondimenti

Da notare che quando una cella non viene riempita con un qualsiasi elemento non tutti i browser visualizzeranno i bordi allo stesso modo:

```
<table width="200" border="1">
  <tr>
    <td width="50%">
    </td>
    <td width="50%">contenuto
    </td>
  </tr>
</table>
```

cioè:

	contenuto

Dunque è opportuno riempire sempre le celle con qualcosa, sia pure un ** ** (è la notazione per indicare un **"non-breaking space"**, cioè uno "spazio che non va a capo"), o un **
. Attenzione che questi caratteri speciali prendono le dimensioni del tag **<font> all'interno di cui sono contenuti.

Con Netscape 4 per ottenere la visualizzazione desiderata è spesso necessario introdurre una **gif trasparente di 1 pixel x 1 pixel** (detta "**shim**") come sfondo della cella.

Ovviamente per ottenere il layout desiderato di bordi e tabelle sarebbe più opportuno utilizzare i fogli di stile. Ecco alcuni link interessanti: i margini, il padding, i bordi, lo sfondo.

L e z i o n e s u c c e s s i v a

[S o m m a r i o]

TORNA SU

Comporre una pagina in frame

I frame (“riquadri”) comparvero per la prima volta con Netscape Navigator 2: si tratta della possibilità di suddividere una medesima finestra del browser in vari riquadri indipendenti.

Questa soluzione all’epoca si rivelò un successo, dal momento che permetteva notevoli vantaggi:

- Fino a qualche tempo fa la velocità di navigazione era ben poca cosa, e si navigava con modem analogici molto lenti (anche da 14.4 kbs): i frame hanno l’indubbio vantaggio di non costringere a ricaricare tutta quanta la pagina, accelerando così la navigazione dell’utente all’interno di un sito web. D’altro canto il fatto che solo una parte del contenuto sia ricaricata fa risparmiare banda anche dal punto di vista del server che deve erogare le pagine
- Per quel che riguarda i webmaster, i frame hanno la caratteristica di utilizzare una struttura che consente di non ripetere le parti comuni nelle varie pagine di un sito, dal momento che il contenuto della pagina (per sua natura) è organizzato “a zone”
- Il fatto di poter mantenere fisso su un lato del monitor il menu di navigazione e far scorrere sull’altro lato il contenuto piace a molti utenti, soprattutto quando la risoluzione del monitor è bassa (800 x 600 o 640x480, magari su un monitor da 15’’)

Tutte queste caratteristiche hanno causato un vero e proprio boom dei frames, tanto che subito dopo l’invenzione della Netscape, anche Microsoft si trovò a “copiare” la possibilità di strutturare le pagine in questo modo; in seguito (con l’HTML 4) i frames divennero una specifica ufficiale del W3C.

Struttura di un frameset

Per utilizzare i frame, è necessario creare una pagina che contenga la dichiarazione della struttura che vogliamo utilizzare. Vediamo subito il codice:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//IT"
       "http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <title>HTML.it</title>
</head>
<frameset rows=”50%,50%” cols="50%, 50%">
    <frame src="prima.html">
    <frame src="seconda.html">
    <frame src="terza.html">
    <frame src="quarta.html">
    <noframes>
        <p>Qui può essere indicato il link a<a href="senzaFrame.html">
            una versione del sito</a> che non utilizzi un layout a frame</p>
        </noframes>
</frameset>
</html>
```

L’esempio completo si trova qui.

Come vi sarete accorti, rispetto alle pagine che abbiamo studiato finora cambiano alcune cose.
In primo luogo cambia il **doctype**, cioè il tipo di documento di riferimento.
All'inizio del documento al posto di questa riga:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//IT">
```

compare ora infatti questa dicitura:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//IT"
        "http://www.w3.org/TR/html4/frameset.dtd">
```

stiamo indicando semplice al browser che facciamo riferimento alle specifiche che servono per regolare il comportamento dei frame.

Avrete notato inoltre che scompare il tag **<body>** e al suo posto troviamo il tag **<frameset>** ("set di riquadri"), che ci permette di indicare come devono essere indicati i frames all'interno della pagina

Il tag <frameset> ha sostanzialmente due importanti attributi: **rows** e **cols**:

- **rows** permette di specificare il numero e la grandezza delle righe, nel caso in cui venga omesso, significa che ci troviamo di fronte a una struttura a colonne . Ad esempio:

  ```
  <frameset cols="33%, 33%,*">
  ```

- **cols** permette di specificare il numero e la grandezza delle colonne e, nel caso in cui venga omesso significa che ci troviamo di fronte una struttura a righe

  ```
  <frameset rows="33%, 33%,*">
  ```

Nell'indicare la grandezza di ciascuna riga (o colonna) possiamo poi lasciare che una o più righe si auto-dimensionino, occupando tutto lo spazio che rimane, in questo caso utilizzeremo l'asterisco ("wild card"); normalmente invece potremo esprimere la grandezza dei riquadri secondo uno dei seguenti sistemi di misura (da scegliere a nostra discrezione):

dimensione fissa	Questa sintassi crea un frameset di 2 righe con 3 colonne ciascuna, per un totale di 6 riquadri: `<frameset rows="150,*" cols="100,200,*">` L'altezza della 1a riga è di 150 pixel, mentre la seconda si adatta al resto della pagina. Le tre colonne sono larghe rispettivamente: 100 pixel, 200 pixel, e la terza colonna si adatta al resto della pagina

percentuale	Questa sintassi crea un frameset che si adatta alla risoluzione. La grandezza dei riquadri è espressa in percentuale: `<frameset rows="20%,80%" cols="25%,25%,50%">` come si può vedere la prima riga occupa il 20% dell'altezza, la seconda il rimanente 80%. Le 3 colonne si dividono la larghezza: la prima colonna occupa il 25%, la seconda nuovamente il 25%, la terza il 50% dello spazio
proporzionale	In questo caso la ripartizione è proporzionale: `<frameset rows="1*,3*" cols="3*,2*,1*">` • per quel che riguarda le righe: l'altezza viene suddivisa in 4 parti (1+3); la prima riga ne occupa 1 parte e la seconda riga ne occupa 3 • per quel che riguarda le colonne: l'altezza viene suddivisa in 6 parti (3+2+1); la prima colonna occupa 3 parti, la seconda riga ne occupa 2 e la terza 1

Una volta creata la nostra griglia con il tag **<frameset>**, incrociando le righe e le colonne, dobbiamo specificare dove si trova il file di origine di ciascun frame. Possiamo farlo con la sintassi:

```
<frame src="prima.html">
```

come si può vedere l'origine di ciascun frame è un documento HTML standard (come quelli che abbiamo analizzato finora): avrà dunque la sua dichiarazione di documento, la sua <head> e il suo <body>.
Se le dimensioni del riquadro non sono sufficienti a mostrare il documento nella sua interezza, il frame avrà delle barre di scorrimento, a meno che non sia stato esplicitamente specificato il contrario negli attributi (che vedremo tra poco).

Per visualizzare il codice HTML di ciascun frame è sufficiente andare nel riquadro desiderato e poi digitare il tasto destro del mouse. Quindi:

- Con Internet Explorer:
  selezionare HTML
- Con Mozilla:
  selezionare **this frame > view frame source**

È possibile anche individuare un frame in modo più preciso, assegnandogli un nome:

```
<frame id="primoRiquadro" name="primoRiquadro" src="prima.html">
```

la sintassi corretta per dare un nome a un frame dovrebbe essere:
id="primoRiquadro"

tuttavia per questioni di retro-compatibilità (con Netscape 4) è oramai entrato nell'uso utilizzare anche **name="primoRiquadro"**.

Frameset annidati

È possibile annidare diversi frames l'uno dentro l'altro. In questo caso, al posto di uno dei tag <frame> è sufficiente includere le indicazioni del nuovo frameset. Così:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//IT"
      "http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
     <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
     <title>HTML.it</title>
</head>
    <frameset rows="15%,70%,15%">
        <frame src="11.html">
        <frameset cols="25%,50%,25%">
            <frame src="21.html">
            <frame src="22.html">
            <frame src="23.html">
        </frameset>
        <frame src="12.html">

    <noframes>
        <p>Qui può essere indicato il link a <a href="senzaFrame.html">
            una versione del sito</a> che non utilizzi un layout a frame</p>
    </noframes>
</frameset>
</html>
```

L'esempio completo si trova qui.

L e z i o n e s u c c e s s i v a

[S o m m a r i o]

TORNA SU

Attributi dei frames per la visualizzazione

Attributi del frameset

Il tag frameset non ha (secondo le specifiche ufficiali) attributi per la visualizzazione. Alcuni attributi tuttavia sono entrati nell'uso e sono correttamente supportati dai browser attuali:

`<frameset frameborder="no" cols="25%,75%">`	L'attributo **frameborder** (di default impostato a "yes") permette di specificare se nel frameset devono essere presenti i bordi. L'esempio è qui
`<frameset framespacing="20" border="20" cols="25%,75%">`	**framespacing** funziona solo con Internet Explorer e permette di impostare lo spazio tra un frame e l'altro. Di fatto equivale all'attributo **border**, che permette di specificare lo spessore dei bordi in pixel. Per mantenere la compatibilità con Internet Explorer 4 (che non legge l'attributo border), di solito si specificano sia il **framespacing**, sia il **border**. L'esempio è qui
`<frameset border="10" framespacing="10" bordercolor="#FF0000" cols="25%,75%">`	**bordercolor** permette di specificare il colore dei bordi del frameset. L'esempio è qui

Attributi dei frame

A differenza degli attributi del tag frameset, che sono dovuti alla convenzione, i seguenti attributi del tag frame sono invece descritti nelle specifiche del W3C e permettono di modificare l'aspetto dei riquadri e il modo in cui l'utente può interagire con essi:

`<frame src="prima.html" scrolling="no">` `<frame src="prima.html" scrolling="auto">`	L'attributo **scrolling** (di default impostato a **"yes"**) indica se si vuol consentire all'utente di poter scorrere il frame oppure no. Nel caso sia impostato a **"no"**, il frame non ha la barra di scorrimento anche nel caso in cui il contenuto della pagina HTML vado oltre la cornice, come si può vedere nell'esempio. L'esempio è qui **scrolling** può anche essere impostato ad **"auto"**. In questo caso la barra di scorrimento compare in automatico, ma solo se necessario
`<frame src="prima.html" scrolling="no">` `<frame src="prima.html" scrolling="no" noresize>`	**noresize** impedisce al singolo frame di essere ridimensionato. L'esempio è qui Se usato in unione con **scrolling="no"**, di fatto "blocca" il contenuto del frame. L'esempio è qui Un uso maldestro di questa tecnica potrebbe però impedire all'utente la corretta visualizzazione dei contenuti

`<frame src=”prima.html” frameborder=”0”>`	**frameborder** conesente di far apparire o meno i bordi del frame (i valori ammessi sono "0" e "1", ovvero "no" e "yes"). Se **frameborder** è impostato a “0” i bordi non sono visibili L'esempio è qui Attenzione però a come impostate i bordi nei vari frame, dal momento che i bordi di frame adiacenti non sempre vanno d’accordo (provate) Questo attributo permette di specificare un valore differente da quello impostato nel **frameborder** del **<frameset>**
`<frame marginwidth="50" marginheight="50" src="prima.html">`	**marginheight** e **marginwidth** permettono di impostare la distanza verticale (**marginheigth**) e orizzontale (**marginwidth**) tra i bordi del frame e il suo contenuto. L'esempio è qui

Approfondimenti

Ovviamente sarebbe meglio impostare i bordi e gli spazi tra i frame attraverso i CSS. Nella lezione dedicata ai bordi con i CSS è spiegato come fare.

Lezione successiva

[Sommario]

TORNA SU

Struttura del tag form

Uno dei fattori che ha decretato il successo del Web è senz'altro la possibilità di interagire: la possibilità cioè di iscriversi a servizi di vario tipo (ad esempio mailing list), ma soprattutto di partecipare a vere e proprie comunità virtuali, come il forum di HTML.it.

Per organizzare questo genere di servizi è necessario raccogliere in qualche modo i dati dell'utente: per farlo si utilizzano, in maniera molto semplice, i moduli (cioè i form).

L'invio dei dati è solitamente organizzato in due parti:

- una **pagina principale** che contiene i vari campi dei form, che consentono all'utente di effettuare delle scelte, scrivere del testo, inserire un'immagine
- una **pagina secondaria** che viene richiamata dalla principale e che effettua "il lavoro" vero e proprio di processare e raccogliere i dati. Di norma si tratta di una pagina di programmazione che si trova sul server. Può essere un cgi, oppure una pagina asp, php, jsp o altro

Noi ci occuperemo della sola pagina principale, dal momento che il modo in cui struttura una pagina di programmazione esula dagli obiettivi della presente guida.

Name e action

Per creare una pagina con dei moduli, bisogna utilizzare l'apposito tag **<form>**: si tratta di un elemento di blocco, come il **<p>**, quindi il tag **<form>** lascia uno spazio prima dell'apertura e dopo la chiusura.

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

Nel caso in non si desideri avere dello spazio superfluo è possibile modificare i bordi del tag utilizzando i fogli di stile. Con questa semplice sintassi:

```
<form name="datiUtenti" style="border:0px" action="paginaRisposta.php">
```

Come si può vedere, **"name"** serve per indicare il nome del form, **"action"** indica l'URL del programma o della pagina di risposta che processerà i dati.

Grazie all'**"action"** è anche possibile far sì che i dati vengano inviati in e-mail al webmaster (si tratta infatti a tutti gli effetti di un riferimento a un URL). Il codice è questo:

```
<form action="mailto:tuamail@nomeDominio.it?subject=Oggetto predefinito"
enctype="text/plain" method="POST">
```

vedremo in una delle prossime lezioni come utilizzare concretamente questa sintassi.

Method

Quando creiamo un form possiamo scegliere due metodi di invio: **GET** e **POST**.

Con il metodo **GET** la pagina di risposta viene contattata e i dati vengono inviati in un unico step.
Nell'URL della pagina di risposta potremo allora vedere tutti i parametri nella barra degli indirizzi (più precisamente nella "**query string**", cioè nella "**stringa di interrogazione**") secondo questa forma:

```
http://www.html.it/esempioForm/paginaRisposta.php?nome=Wolfgang&cognome=Cecchin&datiInviati=prova+invio
```

i dati (nella forma **nome del campo = valore del campo**) sono appesi alla pagina dopo il punto interrogativo.
Alcuni server hanno tuttavia hanno delle limitazioni per quel che riguarda il metodo **GET** e non consentono di inviare form con valori superiori a **255 caratteri** complessivi. Il metodo **GET** è dunque particolarmente indicato per form con pochi campi e pochi dati da inviare. La sintassi per l'invio in get è:

```
<form name="datiUtenti" action="paginaRisposta.php" method="GET">
```

Nel metodo **POST** invece l'invio dei dati avviene in due step distinti: prima viene contattata la pagina sul server che deve processare i dati, e poi vengono inviati i dati stessi. Per questo motivo i parametri non compaiono nella query string (dunque se non si desidera che i parametri siano mostrati all'utente questo metodo è preferibile).
In questo caso non ci sono limiti sulla lunghezza dei caratteri. La sintassi è:

```
<form name="datiUtenti" action="paginaRisposta.php" method="POST">
```

Enctype (tipo di codifica)

Prima di passare i dati alla pagina di risposta, che si trova sul server, questi vengono codificati dal browser in modo da non poter dare adito ad errori (ad esempio gli spazi vengono convertiti in "+"). Normalmente non è necessario specificare come si vuole effettuare la codifica dei dati, perché è sottinteso l'invio di semplice testo. A volte però, come quando è necessario inviare un'immagine, è tuttavia indispensabile dichiarare espressamente quali dati vogliamo inviare. Per farlo è necessario utilizzare l'attributo **"enctype"** (**"encoding type**", cioè **"tipo di codifica"**).

Come dicevamo normalmente non è necessario farlo, perché viene sottinteso questo tipo di sintassi:

```
<form name="datiUtenti" action="paginaRisposta.php" enctype="text/plain">
```

Ma nel caso di invio di immagini dovremo dichiarare:

```
<form name="datiUtenti" action="paginaRisposta.php" method="post" enctype="multipart/form-data">
```

Target

Grazie all'attributo **"target"** è possibile far aprire i dati del form in una pagina differente rispetto a quella corrente (o in una determinata parte di un frameset):

```
<form name="datiUtenti" action="paginaRisposta.php" method="get" target="_blank">
```

L e z i o n e s u c c e s s i v a

[S o m m a r i o]

TORNA SU

Un po' d’ordine: raggruppare i moduli

Per la loro natura di "raccoglitori di informazioni", i moduli tendono a ingigantirsi e diventare lunghissimi. Per questo, con l’HTML 4 sono stati introdotti dei tag per fare un po’ d’ordine all’interno dei form.

Grazie al tag **<fieldset>** possiamo creare delle macro-aree all’interno dei form, e grazie al tag **<legend>**, possiamo indicare il nome di ciascuna macro-area.

Poniamo ad esempio di dover raccogliere i dati di un utente, raccogliendo dati anagrafici, residenza, domicilio e reperibilità sul lavoro.

Possiamo farlo con la seguente sintassi:

```
<fieldset>
    <legend>Dati anagrafici</legend>
    <br><br><br>
</fieldset>

<fieldset>
    <legend>Residenza</legend>
    <br><br><br>
</fieldset>

eccetera
```

che dà:

Dati anagrafici

Residenza

come si può vedere vengono creati dei riquadri con un indicazione del tipo di contenuto.

Un altro tag particolarmente utile - si può utilizzare con ogni tipo di campo che vedremo d’ora in poi - è il tag **<label>**, che permette di indicare un’etichetta per il nome del campo.

Ad esempio:

```
<fieldset>
   <legend>Dati anagrafici</legend>
   <label>Anno di nascita: <input type="text"></label>
</fieldset>
```

che dà:

Dati anagrafici Anno di nascita:

oppure (cambiando la posizione del testo):

```
<fieldset>
   <legend>Dati anagrafici</legend>
    <label><input type="text">: anno di nascita</label>
</fieldset>
```

che dà:

Dati anagrafici [ ]: anno di nascita

Come si può vedere il campo su cui si vogliono dare delle indicazioni deve essere compreso all’interno del tag **label** stesso.

Lezione successiva

[Sommario]

TORNA SU

Il tag Input

Per quel che riguarda i campi dei form il tag più utilizzato è l'**<input>**, che è senza chiusura. Per specificare un determinato tipo di campo è sufficiente indicare il tipo di input.

Ad esempio:

```
<input type="text">
```

crea un campo di testo.

```
<input type="button">
```

crea un bottone.

I vari **<input>** sono dotati di attributi che consentono di indicare il tipo di campo, il nome (ad esempio per interagire con JavaScript), e il valore (per lo più il testo visualizzato).

```
<input type="text" name="tuoTesto" value="qui il tuo testo">
```

che dà:

Lezione successiva

[Sommario]

TORNA SU

I bottoni (sumbit, reset, button, image)

Non c'è form che si rispetti senza bottone di invio.

La sintassi tradizionale per creare un bottone di invio è:

```
<input type="submit" value="invia I dati">
```

Ad esempio:

```
<form action=http://www.html.it target="_blank">
   <input type="submit" value="visita HTML.it">
</form>
```

cioè:

visita HTML.it

Un altro bottone utile è il **"reset"** che – una volta premuto - consente di riportare il form allo stato originario, cancellando ogni cosa scritta finora dall'utente. Ecco un esempio:

```
<form>
   <input type="text"><br>
   <input type="reset" value="cancella">
</form>
```

cioè

cancella

Esiste infine un tipo di bottone generico, che non esegue nessuna azione particolare, ma che può essere ad esempio utilizzato per associare degli eventi tramite JavaScript.

```
<form>
   <input type="button" value="bottone generico">
</form>
```

che dà:

bottone generico

Il tag <button>

Con l'HTML 4 è stato introdotto il tag **<button>** che offre la possibilità di creare dei bottoni con un aspetto particolarmente ricco.

Il tag **<button>**, a differenza del tag **<input>**, dà la possibilità di inserire il testo del bottone tra l'apertura e la chiusura del tag medesimo. Questo ci consente di specificare anche del codice HTML all'interno del tag.

I bottoni che abbiamo appena visto dovrebbero dunque avere questa forma:

```
<form action=http://www.html.it target="_blank">

    <input type="text"><br>

    <button type="button">
          bottone generico
    </button>

    <button type="reset">
          cancella
     </button>

     <button type="submit">
         invia
     </button>

</form>
```

cioè:

bottone generico cancella invia

Ed ecco un esempio complesso:

```
<button name="vai" type="submit">
     invia
    <img src="puntoEsclamativo.gif" width="23" height="67" title="invia adesso"
border="1" vspace="5" alt="invia adesso" align="middle">
    <b>adesso</b>
</button>
```

che dà:

invia  **adesso**

Grazie all'attributo **"disable"** è infine possibile disabilitare i bottoni.

Es:

```
<input type="submit" value="invia" disabled>
```

o anche:

```
<button type="submit" disabled>
invia
</button>
```

cioè:

invia

invia

Il campo image

Il campo **"image"** ci consente di utilizzare come bottoni del form delle vere e proprie immagini e assegnare loro un valore grazie a JavaScript; in questo caso non si tratta propriamente di un bottone ma la funzionalità è la medesima. Ecco il codice:

```
<input name="invia" type="image" src="invia.gif" alt="invia il modulo" title="invia il modulo" width="78" height="38">
```

cioè:

come si può vedere, se non si specifica nulla, l'immagine ha valore di submit.
Gli attributi del campo immagine sono molto simili a quelli del tag **<img>**.

L e z i o n e s u c c e s s i v a

[S o m m a r i o]

TORNA SU

Inserire testo (campo testo, textarea, password)

Per consentire all'utente di inserire del testo è possibile utilizzare un **campo testo**. Se il campo è su una singola linea avremo:

```
<input name="mioTesto" type="text" value="qui il tuo testo" size="40" maxlength="200">
```

qui il tuo testo

"maxlenght" indica il numero massimo di caratteri che l'utente può inserire, con **"size"** si esprimono invece le dimensioni del campo di testo (la larghezza è data dal numero di caratteri).

Se si ha la necessità di indicare un campo che consenta di inserire una grande quantità di testo conviene invece utilizzare una **"textarea"** (**"area di testo"**). Ecco la sintassi:

```
<textarea name="testo" cols="40" rows="10">qui puoi scrivere il tuo testo</textarea>
```

qui puoi scrivere il tuo testo

"rows" indica il numero di righe della textarea, **"cols"** il numero di caratteri (cioè di colonne) che ogni riga può contenere.
Come si può vedere, se si vuol indicare del testo predefinito in questo caso bisogna inserirlo fra l'apertura e la chiusura del tag.

Esiste infine il **campo password** che codifica i caratteri insieriti con degli asterischi:

```
<input name="mioTesto" type="password" size="18" maxlength="8">
```

che dà:

da notare che la codifica fornisce una protezione soltanto per chi eventualmente stia sbirciando sul monitor dell'utente. L'invio dei dati attraverso il web avviene,se non vengono adottate altre misure di sicurezza, 'in chiaro'.

I campi di testo possono essere anche di sola lettura. Ad esempio:

```
<input name="mioTesto" type="text" value="leggere l'informativa" size="25" maxlength="8" readonly>
```

che dà:

leggere l'informativa

O disabilitati:

```
<input name="mioTesto" type="text" value="leggere l'informativa" size="25" maxlength="8" disabled>
```

cioè

leggere l'informativa

Lezione successiva

[Sommario]

TORNA SU

Consentire delle scelte (checkbox, radio, select)

Checkbox

Con i checkbox possiamo consentire all'utente di operare delle scelte multiple. Ad esempio:

```
<fieldset>
    <legend>Linguaggi conosciuti</legend><br>
     <input type="checkbox" name="html" value="html"> html

     <br>
     <input type="checkbox" name="css" value="css"> css

     <br>
     <input type="checkbox" name="javascript" value="javascript"> JavaScript
</fieldset>
```

che dà:

Linguaggi conosciuti

- ☐ html
- ☐ css
- ☐ JavaScript

Si possono anche selezionare uno o più valori di default:

```
<input name="html" type="checkbox" value="html" checked>
```

cioè

☑

ed è possibile disabilitare una casella:

<input name="html" type="checkbox" value="html" **disabled**>

cioè:

☐

Radio button

I **"radio button"** (**"bottoni circolari"**) invece consentono di effettuare una scelta esclusiva. In questo caso quindi una scelta esclude l'altra. Per ottenere questo effetto i campi devono avere lo stesso nome e differente valore:

```
<fieldset>
```

```
    <legend>Linguaggi conosciuti</legend>
    HTML<input type="radio" name="linguaggio" value="html">
    CSS  <input type="radio" name="linguaggio" value="css">
    JavaScript <input type="radio" name="linguaggio" value="javascript">
</fieldset>
```

che viene così visualizzato:

Linguaggi conosciuti HTML ○ CSS ○ JavaScript ○

Anche in questo caso è possibile assegnare un valore di default o disabilitare un pulsante.

```
<input type="radio" name="linguaggio" value="html" checked disabled>
```

cioè:

⊙

Menu di opzioni (select)

Grazie al tag **<select>** è possibile costruire dei menu di opzioni. In questo caso ciascuna voce deve essere compresa all'interno del tag **<option>** (la chiusura del tag è opzionale) e il valore deve essere specificato attravero l'attributo **"value"**. Con l'attributo **"selected"** si può indicare una scelta predefinita:

```
<fieldset>
    <legend>Siti per webmaseter</legend>

    <select name="siti" >
        <option value="http://www.html.it" selected>www.html.it
        <option value="http://freephp.html.it">frephp.html.it
        <option value="http://freasp.html.it">freasp.html.it
    </select>
</fieldset>
```

che da luogo a:

Siti per webmaseter [www.html.it]

Siccome i menu di scelta tendono a diventare particolarmente lunghi, nell'HTML 4 è stato introdotto il tag **<optgroup>** che consente di suddividere le varie possibilità di scelta in gruppi tramite l'utilizzo di apposite etichette. Ecco l'esempio:

```
<select name="siti" >
   <optgroup label="siti per webmaster">
        <option value="http://www.html.it">www.html.it
        <option value="http://freephp.html.it">frephp.html.it
        <option value="http://freasp.html.it">freasp.html.it
   </optgroup>

   <optgroup label="risorse per webmaster">
      <option value="http://font.html.it">font.html.it
      <option value="http://cgipoint.html.it">cgipoint.html.it
    </optgroup>
</select>
```

che dà luogo al seguente menu:

www.html.it

Infine con il tag **select** è possibile impostare anche delle scelte multiple. Come si può vedere, utilizzando l'attributo **"multiple"** l'aspetto del tag select cambia notevolmente:

```
<label>Quale siti visiti?<br>

<select name="siti" multiple>
    <option value="http://www.html.it">www.html.it
    <option value="http://freephp.html.it">frephp.html.it
    <option value="http://freasp.html.it">freasp.html.it
    <option value="http://font.html.it">font.html.it
    <option value="http://cgipoint.html.it" >cgipoint.html.it
</select>

</label>
```

cioè:

Quale siti visiti?

www.html.it
frephp.html.it
freasp.html.it
font.html.it

utilizzando il tasto **"ctrl"** l'utente può così effettuare delle scelte multiple.

Tramite l'attributo **"size"** si può specificare il numero delle voci che devono comparire nel menu, e conseguentemente regolare l'altezza del menu, aggiungendo o togliendo la barra di scorrimento verticale.

```
<label>Quale siti visiti?<br>

<select name="siti" size="5" multiple>
    <option value="http://www.html.it">www.html.it
    <option value="http://freephp.html.it">frephp.html.it
    <option value="http://freasp.html.it">freasp.html.it
    <option value="http://font.html.it">font.html.it
    <option value="http://cgipoint.html.it" >cgipoint.html.it
</select>

</label>
```

che viene così visualizzato:

Quale siti visiti?

www.html.it
frephp.html.it
freasp.html.it
font.html.it
cgipoint.html.it

Lezione successiva

[Sommario]

TORNA SU

Altri campi (file e hidden)

Potremmo avere la necessità di passare dei parametri "di servizio", senza far percepire la loro presenza all'utente. In questo caso possiamo utilizzare dei campi nascosti, presenti all'interno del form ma invisibili all'utente (ricordiamoci sempre di specificare la coppia **"nome-valore"**):

```
<input type="hidden" name="urlDiProvenienza" value="www.html.it">
```

Il campo **"file"**, consente invece di inviare un file sul server, nel caso in cui la pagina di risposta sia stata programmata correttamente. La sintassi è:

```
<input name="fileUtente" type="file" size="20">
```

che dà:

"size" indica la larghezza del campo. Come si può vedere, a fianco del modulo compare il pulsante **"sfoglia"** o **"browse"** (a seconda della lingua del browser dell'utente).

Un esempio concreto

Riprendendo un esempio accennato in precedenza, possiamo vedere come sia possibile consentire all'utente di inviarci il contenuto di un questionario tramite e-mail. Dal punto di vista dell'utente si aprirà un messaggio che domanda se si vuole inviare una mail, ma ciò è inevitabile se si utilizza questo metodo: per evitare questa eventualità bisognerebbe infatti usare dei programmi che inviino e-mail lato-server.

```
<form name="datiUtente" enctype="text/plain" method="POST"
action="mailto:tuamail@nomeDominio.it?subject=Questionario proveniente dal
web">

   <fieldset>
      <legend>Dati Utente</legend>
      <label>Nome: <input name="nome" type="text" size="20"
maxlength="30"></label>
      <label>Cognome: <input name="cognome" type="text" size="20"
maxlength="30"></label>
      <label>Professione: <input name="cognome" type="text" size="20"
maxlength="30"> </label>
   </fieldset>

<br><br>

   <fieldset>
     <legend>Questionario</legend>
          <label>Siti visitati:<br>
          <select name="siti" size="5" multiple>
               <option value="http://www.html.it">www.html.it
               <option value="http://freephp.html.it">frephp.html.it
               <option value="http://freasp.html.it">freasp.html.it
               <option value="http://font.html.it">font.html.it
```

```
                    <option value="http://cgipoint.html.it" >cgipoint.html.it
                </select>
        </label>
</fieldset>

<br><br>

<fieldset>
        <legend>Campo libero</legend>
        <label>Lasciami un parere:<br><textarea name="parere" cols="20"
rows="10"></textarea></label>
</fieldset>

<br><br>

<button type="reset">annulla</button>
    
<button type="submit">invia</button>

</form>
```

che dà luogo a questo form:

Dati Utente Nome: Cognome:

Professione:

Questionario Siti visitati:

www.html.it
frephp.html.it
freasp.html.it
font.html.it
cgipoint.html.it

Campo libero Lasciami un parere:

annulla invia

L e z i o n e s u c c e s s i v a

[S o m m a r i o]

TORNA SU

Approfondimenti sui form

L’attributo tabindex

Utilizzando il tasto **"tab"** della tastiera l’utente può passare da un campo del form all’altro. Per varie ragioni di impaginazione l’ordine così ottenuto potrebbe però non essere quello desiderato. Grazie all’attributo **"tabindex"** che si applica ai campi dei moduli è possibili specificare in quale ordine deve avvenire il passaggio da un campo all’altro. Il valore di questo attributo può variare tra 0 e 32767. Vediamo un esempio:

```
<form name="datiUtente">
   <fieldset >
   <legend>Dati utente</legend>

   <table width="300" border="1" cellspacing="0" cellpadding="5">
      <tr>
         <td>
            <label>Nome: <input tabindex="1" name="nome" type="text" size="30"
maxlength="30"></label>
         </td>
          <td>
            <label>Professione: <input tabindex="3" name="professione"
type="text" size="30" maxlength="100"></label>
          </td>
      </tr>

      <tr>
         <td>
            <label>Cognome: <input tabindex="2" name="cognome" type="text"
size="30" maxlength="30"></label>
         </td>
         <td> </td>
      </tr>
</table>
</fieldset>
</form>
```

che viene così visualizzato:

Dati utente

Nome:	Professione:

Cognome:	

come si può vedere, digitando il tasto **"tab"**, l'ordine di passaggio da un campo all'altro non è quello indicato nell'HTML, ma è modificato secondo il valore di **"tabindex"**.

Il layout dei form

Se siete alle vostre prima pagine HTML, può apparire difficile avere il controllo perfetto dei form. Qui potete trovare due articoli che fanno al vostro caso:

- I Form: segreti e trucchi di personalizzazione
- I Form: risposte a domande frequenti

L e z i o n e s u c c e s s i v a

[S o m m a r i o]

TORNA SU

HTML.it

I meta tag

I meta tag

Adesso che abbiamo terminato il nostro sito possiamo occuparci di farlo trovare dai motori di ricerca.

È utile allora impostare correttamente i meta tag all interno della **<head>** del documento: si tratta di una serie di parole chiave e descrizioni, che aiutano i motori di ricerca a classificare il sito.

Abbiamo già visto il **<title>**, che è il titolo della pagina; ma il testo ivi contenuto può comparire anche in seguito alla ricerca in un motore, come titolo del link. Sarà dunque importante impostarlo in modo pertinente:

```
<title>HTML.it – il sito italiano sul webpublishing</title>
```

C'è poi il **meta-tag "description"** che permette di impostare una descrizione sintetica del sito stesso. Anche in questo caso, la descrizione compare talvolta nei risultati della ricerca:

```
<meta name="description" content="HTML.it - il sito italiano sul Web publishing">
```

Infine il meta-tag "**keywords"** permette di indicare alcuni contenuti relativi al sito stesso.
Le keywords (a seconda del webmaster) compaiono separate da virgola, da punto e virgola, oppure senza alcun segno di interpunzione:

```
<meta name="keywords" content="html wml xml smil javascript js dhtml dynamic
xhtml vbscript coldfusion photoshop paint shop pro risorse webmaster webdesigner
flash grafica css applet java asp cgi perl guida free corso php mysql tutorial lezioni
sql database realizzazione siti web leggi mailing list newsletter gif jpg publishing
editor iis webserver apache linux raccolte script news chat forum fogli di stile hdml
wap linux mac apple palmari computer c++ delphi visual basic vb vbasic">
```

È fortemente sconsigliabile l'inserimento di keyword "astute" non relative al contenuto effettivo del sito per migliorare il posizionamento (tipo le ricercatissime "sesso", "mp3", ecc.). Quando i motori di ricerca se ne accorgono, per lo più cancellano il sito dalle loro liste.

Su HTML.it sono presenti molte risorse sull'argomento. L'articolo I Meta-tag: come scriverli correttamente, ad esempio, è un approfondimento su come impostare i meta-tag.

Un buon posizionamento all'interno dei motori di ricerca è una meta difficile da raggiungere e l'argomento non si esaurisce certo in poche righe. Per cui, se siete interessati all'argomento, vi rimando alla sezione Motori di Ricerca di PRO.HTML.it e al forum di discussione dedicato a Motori di Ricerca e Web Marketing.

Lezione successiva

[Sommario]

TORNA SU