Benvenuto nel wikibook:

Linguaggio HTML

Edizione: 0.3 12/06/2007

Versione attuale del libro>>

Autori: Diablo, Ramac, The Doc, Pietrodn, Sky .

Indice generale

Introduzione....3
Breve storia dell'HTML....3
Premesse....3
Tag, attributi e valori....4
Struttura generale di una pagina HTML....5

Elementi block e in-line....6
Inserire dei commenti....6
Consigli sulla scrittura....6
Indentazione....6
Tag vuoti ed XHTML....7
Definire l'intestazione....7
Il Titolo della pagina....7
I meta tag....8
Il DocType....10
Formattare il testo....11
Font....11
Colore....11
Stile....12
Dimensioni....12
Organizzare il testo....13
Titoli....13
Paragrafi....13
Linee divisorie....14
Andare a capo....14
Testo preformattato....15
Elenchi....15
Elenchi ordinati....15
Elenchi non ordinati....15
Elenchi di definizioni....16
Testo dinamico....16
Testo lampeggiante....16
Testo scorrevole....17
Links....17
Collegamenti esterni....17
Collegamenti interni....17
Posta elettronica....18
Attributi dei link....18
Immagini....19
Inserire un'immagine....19
Definire l'allineamento....20
Immagine come link....20
Mappa d'immagine....20
Immagine nella barra degli indirizzi....20
Tabelle....21
Attributi delle tabelle....21
Tabelle preformattate....22
Sfondo....22
Colore di sfondo....22
Immagine di sfondo....22
Multimedia....23
Suoni....23
Moduli....23
Menu a scelta....24
Area di testo....25
Pulsanti....25
I Frames....26
A cosa servono i frame?....26

Creare una pagina....26
Fuori dai frame....27
Attributi del frameset....27
Attributi dei frame....27
Caricare una pagina....28
Iframe....28
Caratteri speciali....29
Includere altri linguaggi....30
Fogli di stile....30
Il tag <style>....30
Inclusione da file esterno....30
Inclusione inline....30
Flash....31
Javascript e VBScript....31
I Verificatori....31
Verificatore W3C....31
L'evoluzione del markup....32
XHTML....32
CSS....32
Licenza....33

Introduzione

HTML (**HyperText Markup Language**) è il primo linguaggio ideato (e il più utilizzato) per realizzare pagine ipertestuali, più comunemente note come pagine internet.

Una comune pagina HTML è un documento di testo contenente codice HTML ed avente le seguenti estensioni: **.htm** o **.HTML**. La differenza tra le due estensioni è che la prima veniva usata nel mondo DOS perché esso supportava estensioni con al massimo 3 lettere. L'estensione dice al browser che si tratta di un pagina HTML e di interpretarla di conseguenza.

L'HTML è un linguaggio che descrive il **contenuto** di una pagina web ma non la forma, in passato veniva anche usato massicciamente ed erroneamente per definire la visualizzazione della pagina stessa compito oggi eseguito molto più efficacemente dai fogli di stile.

In questo wikibook si potranno apprendere la basi minime di conoscenza dell'HTML.

Breve storia dell'HTML

Il linguaggio HTML come lo intendiamo oggi è in realtà la versione 4, resa pubblica dal World Wide Web Consortium il 24 dicembre 1999, ed sta ormai per essere soppiantato da linguaggi come l'XHTML.
L'ideazione e nascita dell'HTML sono merito di due fisici del CERN di Ginevra, Tim Berners-Lee e Robert Caillau, che nel 1989 ebbero l'idea di creare un sistema di informazioni collegate tra di loro accessibile attraverso la rete in uso al centro stesso.

Fondamenti di HTML

Premesse

Questa premessa si rende necessaria per poter consultare e capire nel modo migliore questo wikibook. Molti degli elementi cui si farà cenno potranno essere compresi solamente leggendo i prossimi due paragrafi.

Ecco una lista di termini di cui si presuppone la conoscenza:

- Browser o Web Browser, che alcune volte verrà chiamato interprete.
- W3C o W3.

I tag non saranno scritti in modo particolare ma nel contesto di una frase saranno scritti su sfondo grigio. Gli attributi dei tag verrano scritti in grassetto. I valori saranno scritti in corsivo.

Ogni tematica verrà accompagnata da un esempio di codice che si potrà copiare e incollare sul proprio editor di testo preferito. La struttura di una pagina rimane sempre la seguente:

```
<html>
 <head></head>
  <body>
  </body>
</html>
```

Qualora tutti questi elementi non fossero presenti ma un esempio fosse costituito solo dal seguente codice:

```
<body>codice</body>
```

è semplicemente per rendere la formattazione del testo il meno estesa possibile. Quindi il codice di cui sopra dovrà essere inserito nella struttura generale della pagina:

```
<html>
 <head></head>
  <body>codice</body>
</html>
```

Tag, attributi e valori

Prima di cominciare a lavorare a stretto contatto con il codice è bene soffermarsi su questi tre elementi: il *tag*, l'*attributo*, e il *valore*.

Il tag è l'unità fondamentale, la potremmo definire l'istruzione che fa capire al browser come interpretare il codice e visualizzarlo sul monitor. Attraverso i tag possiamo definire molti elementi di un documento: paragrafi, colore del testo, collegamenti ipertestuali e quant'altro. Ogni tag è caratterizzato da tre componenti:

- < il segno di minore
- nome il suo nome
- > il segno di maggiore

Un tag di chiusura ha la particolarità che dopo il segno di minore ha uno slash / che comunica al browser che quello è, per l'appunto, un tag di chiusura.

All'interno dei tag vi è poi il contenuto che è quello che verrà formattato secondo le regole del tag che lo comprende. Per fare il punto della situazione, la sintassi comune alla maggior parte dei tag è la seguente `<nome tag>contenuto che verrà formattato</nome tag>`

Ad esempio il tag `<h1>` serve per determinare i titoli e quindi rendere il testo al suo interno più

grande rispetto al contenuto della pagina. Ecco ad esempio un codice che visualizzerà sul nostro browser un titolo (non vi preoccupate del resto del codice, verrà analizzato in seguito).

```
<html>
  <head>
  </head>
<body>
  <h1>Wikibooks</h1>
</body>
</html>
```

In realtà non tutti i tag necessitano di essere chiusi ve ne sono alcuni che sono a se stanti e verrano affrontati nel corso del wikibook.

La maggior parte dei tag, quasi tutti, supportano gli attributi ossia dei parametri che servono a definire ad esempio l'allineamento del testo, il suo colore. Ogni attributo è costituito da un valore che determinerà, ad esempio, se questo verrà visualizzato a destra o a sinistra, o se il testo sarà di colore rosso piuttosto che nero.

Ecco la struttura di un tag con tutti gli elementi che abbiamo definito:

```
<nometag attributocolore="valore" attributoallineamento="valore">Testo che verrà
visualizzato
sulla pagina dell'utente</nometag>
```

Ora che abbiamo capito com'è strutturato un tag possiamo passare ad analizzare la struttura di una pagina.

Struttura generale di una pagina HTML

Tutte le pagine web in cui navighi hanno la seguente struttura a livello di codice HTML:

```
<html>

 <head>
   codice
 </head>

 <body>
   codice
 </body>

</html>
```

Il tag `<html>` indica all'interprete, in questo caso il browser, che il documento è stato formattato in HTML e pertanto il suo contenuto dovà essere interpretato secondo le specifiche del linguaggio.

I tag `<head>` d'apertura e `</head>` di chiusura, servono a definire l'intestazione del documento, ossia tutte quelle informazioni che servono a definire il contenuto della pagina.

All'interno dei tag `<body>` e `</body>` è presente invece il resto del documento, il *corpo* della pagina, ciò che effettivamente poi verrà visualizzato sul browser.

Il tag di chiusura `</html>` serve invece per comunicare all'interprete che il codice HTML è terminato è pertanto tutto ciò che sarà scritto successivamente a questo tag verrà interpretato come normale testo.

L'HTML è un linguaggio case-unsensitive pertanto scrivere i tag tutti in maiuscolo o in minuscolo è indifferente, anche alternandoli non cambia nulla. I seguenti esempi codificano una pagina identica, sia dal punto di vista sintattico sia da quello dell'interpretazione del browser e pertanto ciò che vedrà l'utente.

Esempi:

```
<html>
 <head></head>
  <body>
  </body>
</html>

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

<htMl>
 <HeAd></hEAD>
  <bODY>
  </BOdy>
</HTml>
```

Queste tre pagine mostreranno lo stesso risultato sul browser dell'utente.

Elementi block e in-line

Una distinzione importante da fare tra gli elementi di una pagina HTML sono è quella tra gli elementi cosiddetti block-elements (elementi blocco) e gli in-line elements (elementi in linea) chiamati anche text-elements (elementi testo).

Le diversià tra questi due tipi di elementi potrebbero sembrare non importanti ma è molto utile capire la loro differenza; il concetto di elementi block e in-line è molto importante inoltre nel Linguaggio CSS.

- generalmente, gli elementi **block** possono contenere elementi in-line e altri elementi block e, quando vengono inseriti, iniziano una nuova riga
- generalmente, gli elementi **in-line** possono contenere solo testo e altri elementi in-line. Gli elementi in-line, come suggerisce la parola stessa, vengono visualizzati sulla riga corrente e non vanno perciò a capo.

Inserire dei commenti

Come ogni buon programmatore sa è molto importante inserire dei commenti all'interno del codice, così facendo infatti il codice sarà comprensibile anche a distanza di tempo. I commenti sono molto utili e non verranno letti dal browser di conseguenza la pagina non verrà influenzata dalla presenza o meno dei commenti. Questi potranno però essere visualizzati da chiunque solamente attraverso il codice sorgente del sito.

Nell'HTML inserire commenti è molto semplice infatti basta usare la seguente sintassi: `<!-- il mio commento -->`

Ecco un esempio:

```
<html>

 <head>
  <!-- Qui ci va il titolo -->
 </head>

 <body>
  <!-- qui ci va il contenuto della pagina -->
 </body>
```

```
</html>
```

Consigli sulla scrittura

Indentazione

L'indentazione è la tecnica attraverso cui il codice viene rientrato a mano a mano che si entra in sotto-tag. Nel seguente wikibook si cercherà di adottare questa tecnica nel miglior modo possibile e si invita chiunque ad utilizzarla in quanto rende il codice più leggibile che non se si trovasse tutto su una riga, o suddiviso senza criterio. Ecco due esempi:

Sintassi sconsigliata:

```
<html>
<head></head>
<body><div>testo testo</div>
<a href="http://it.wikibooks.org/">Wikibooks</a><a
href="http://it.wikipedia.org/">Wikipedia</a>
</body>
</html>
```

Sintassi consigliata:

```
<html>
  <head></head>
    <body>
     <div>testo testo</div>
     <a href="http://it.wikibooks.org/">Wikibooks</a>
     <a href="http://it.wikipedia.org/">Wikipedia</a>
    </body>
</html>
```

Il fatto che l'HTML sia un linguaggio unsensitive è molto utile nel momento in cui bisogna scrivere molto codice e ognuno potrà utilizzare lo stile che preferisce. Tra gli stili più usati ne elenchiamo alcuni:

- **Tutto maiuscolo**: si scrivono tutti i tag indiscriminatamente grandi e gli attributi e i valori in minuscolo, ciò aumenta la leggibilità del testo se correttamente indentato
- **Maiuscoli alcuni tag**: si scrivono in maiuscolo solamente i tag principali come `<html>`, `<head>` e `<body>`.
- **Tutto minuscolo**: tutti i tag vengono scritti in minuscolo, con il rischio di confondersi con gli attributi, una corretta indentazione tuttavia risolve il problema. Quest'ultimo stile è inoltre consigliato se si vuole mantenere la compatibilità con l'XHTML.

Tag vuoti ed XHTML

Per definizione, alcuni tag non possono contenere nemmeno marcare porzioni di testo: non hanno quindi il tag di chiusura e pertanto sono detti **tag vuoti**.

I seguenti sono alcuni esempi di tag vuoti:

```
<img src = 'images/fiore.png'>
<hr>
<p>Questo è un paragrafo<br>che continua sulla riga seguente</p>
```

Sempre per motivi di compatibilità verso l'XML, che non ammette tag vuoti e privi dell'informazione di fine-tag, è buona norma terminare ogni tag vuoto inserendo una barra prima del

simbolo > terminale.
Per garantire la piena compatibilità con i browser attuali (che nella maggioranza dei casi supportano intrinsecamente l'HTML e non l'XML), nella codifica XHTML la barra finale dev'essere preceduta da uno spazio.
In XHTML, i precedenti esempi di tag vuoti si scriveranno quindi:

```
<img src = 'images/fiore.png' />
<hr />
<p>Questo è un paragrafo<br />che continua sulla riga seguente</p>
```

Definire l'intestazione

Il Titolo della pagina

Per far sì che venga visualizzato un titolo del documento nella barra del titolo del browser è sufficiente inserire all'interno dei tag <head>,</head> il tag <title>seguito dal titolo che vorreste dare alla pagina e successivamente il tag di chiusura </title>.

Ecco un esempio:

```
<html>

 <head>
   <title>Inserire qui il titolo della pagina</title>
 </head>

 <body>
   codice
 </body>

</html>
```

I meta tag

I meta tag sono dei tag speciali che posti all'interno dei tag <head></head> permettono di definire il contenuto della pagina web, in modo, ad esempio, da rendere l'indicizzazione da parte dei motori di ricerca più rapida ed efficace. I meta tag non producono alcun tipo di effetto grafico nella visualizzazione della pagina. I meta tag sono costituiti da due parti:

- Nome del meta tag
- Valore del meta tag

Ecco quindi la loro sintassi generale: <meta name="nome meta tag" content="valore del meta tag">.

Ecco una tabella riassuntiva dei meta tag utilizzabili:

Nome Tag	Valore	Funzione
DC.Title	Titolo del documento	Serve a definire il titolo del documento
description	Descrizione del sito	Serve per descrivere il contenuto del sito o della pagina
creation_date	Data --> 13/07/2006	Indica la data di creazione della pagina o quando essa verrà aggiornata
keywords	Ogni parola chiave separata da *;* -->	Serve a dichiarare le parole chiave del sito

	Wikibooks; Wiki;	o della pagina
robots	Vedi in fondo alla tabella	Serve a dare indicazione ai motori di ricerca per l'indicizzazione
revisit-after	n°di giorni seguito da days --> 15 days	Serve a dire al motore di ricerca dopo quanti giorni dovrà rivisitare la pagina
generator	L'editor usato per scrivere il codice --> Amaya	Indica il nome dell'editor che si è usato per scrivere il codice della pagina
copyright	Il proprietario del contenuto del sito --> Wikimedia foundation	Indica a chi spetta il copyright del contenuto del sito
author	L'autore della pagina --> Wikibooksiani	Indica l'autore della pagina e eventualmente il suo indirizzo e-mail
owner	Proprietario del sito --> Wikimedia foundation	Indica il proprietario del sito
language	Una o più lingue --> it, eng	Indica la lingua o le lingue del sito
DC.Language	Una o più lingue --> it, eng	Indica al motore di ricerca di indicizzare le pagine in base alla lingua

Il meta tag *robots* come si è detto in precedenza serve a dare alcune indicazioni ai motori di ricerca, più nello specifico, serve a comunicare allo spider del motore di ricerca come si deve comportare durante l'indicizzazione di una pagina. I suoi possibili valori sono:

- *all*: e' l'insieme tra follow e index, indicizza la pagina che sta visitando e prosegue attraverso i links.
- *none*: non viene indicizzata la pagina e tutto ciò che è al suo interno
- *index*: indicizza solo la pagina che il motore di ricerca sta visitando
- *noindex*: non indicizza la pagina che il motore di ricerca sta visitando
- *follow*: non indicizza la pagina che sta visitando prosegue attraverso i link della pagina
- *nofollow*: indicizza la pagina che sta visitando ma salta i link della pagina

Ci sono altri meta tag un po' particolari detti *HTTP-EQUIV meta tag* perché il server web nella sua risposta HTTP al browser conterrà questi meta tag nella parte iniziale, il cui nome è *HTTP header block*. La loro sintassi, in linea generale, è la seguente: `<meta http-equiv="nome meta tag" content="valore del meta tag">`.

Nome Tag	Valore	Funzione
expires	Ora nel formato GMT --> Tue, 13 Jul 2006 10:30:00 GMT	Indica la data di scadenza della validità della pagina.
reply-to	Indirizzo mail --> wiki@wikibooks.it	Definisce un indirizzo e-mail a cui gli utenti possono fare riferimento
Set-Cookie	Vedi in fondo alla tabella	Salva un cookie sul computer del visitatore
refresh	Vedi in fondo alla tabella	Serve per ricaricare la pagina dopo un determinato tempo
content-Type	Vedi in fondo alla tabella	Indica il set di caratteri in uso nella pagina
Pragma	no-cache, costringe il browser a svuotare la cache e ricaricarla	Solo per NetScape. Forza il browser a non leggere il sito della cache

imagetoolbar	yes abilita la visualizzazione della toolbar, no la disabilita	Da IE 6.0 in su, abilita/disabilita la toolbar che appare sulle immagini.
distribution	global se è un contenuto di interesse generale, in caso contrario, usare local	Indica se il contenuto della pagina è di interesse generale o specifico

Il meta tag *Set-Cookie* serve, come suggerisce il suo nome, a salvare un cookie sul computer di chi sta visitando la pagina. La sua sintassi è la seguente: `<meta http-equiv="Set-Cookie" content="cookievalue=nome cookie; expires=data di scadenza; path=percorso nella cache">`

- **cookievalue**: attraverso questo parametro impostiamo il nome che prenderà il cookie sul computer dell'utente.
- **expires**: impostando una data in formato GMT si può determinare la data dopo la quale il cookie non avrà più effetto.
- **path**: dove verrà salvato all'interno della cache

Esempio di settaggio di un cookie:

```
<head>
<meta http-equiv="Set-Cookie" content="cookievalue=xxx;
expires=Tuesday, 13-Jul-06 23:00:00 GMT path=/">
</head>
```

Il meta tag *refresh* ricarica il contenuto della pagina dopo un determinato intervallo di tempo, si può aggiungere anche un parametro mediante il quale il ricaricamento porterà ad un'altra pagina (reindirizzamento). La sua sintassi è: `<meta http-equiv="REFRESH" content="intervallo di tempo per ricaricare; url=pagina a cui porterà il ricaricamente, opzionale">`

Esempio di reindirizzamento:

```
<head>
 <meta http-equiv="refresh" content="4; url=http://it.wikibooks.org">
</head>
```

Il meta tag *content-Type* serve per dichiarare il set di caratteri usati all'interno della pagina. La sua sintassi è la seguente: `<meta http-equiv="content-Type" content="tipo di documento; charset=set di caratteri">`.
Il tipo di documento nel nostro caso è *text/html* con l'avvento dell'xml potrebbe esserci qualche nuovo valore.

Esempio di come utilizzare il meta tag *content-Type*:

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

Il DocType

La prima riga di ogni pagina web (quindi ancora prima di qualsiasi tag, nel caso dell'HTML prima del tag `<html>`), che tenga conto delle specifiche, è il *DocType*. Questa riga fornisce al browser tutte le informazioni per capire le caratteristiche della pagina che interpreterà. Il DocType, nella sua forma più estesa, ha bisogno di diverse informazioni:

- Il linguaggio usato all'interno della pagina
- Una dichiarazione in cui si determina se il documento è di dominio pubblico o meno
- Una nota che veicoli a quali specifiche del W3C si sta facendo riferimento
- La lingua del documento

La maggior parte delle pagine web, attualmente, ha questo tipo di DocType:

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

Il linguaggio usato all'interno della pagina è l'html, il documento è di dominio pubblico e rispetta le specifiche del W3C del tipo HTML Transitional definiti nel DTD (Document Type Definition) `loose.dtd`, la lingua impostata è l'italiano.

È ora di approfondire a quali specifiche del W3C si può fare riferimento:

- **Frameset**: questa DTD è utilizzata dai siti che al loro interno contengono i frames.
- **Strict**: è una DTD che applica rigide regole ed applica quindi i nuovissimi standard del web. Ad esempio i tag deprecati non sono ammessi e gli elementi grafici di una pagina web devono essere definiti tramite i fogli di stile.
- **Transitional**: come è facilmente intuibile dal nome questo è documento che contiene delle specifiche di transizione da uno standard all'altro, è il tipo maggiormente utilizzato ora sul web. Il documento non applica rigide regole ed è quindi possibile utilizzare tag deprecati.

Esempio di DocType per un sito con i frames:

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

Esempio di DocType per un sito che applica rigidamente le ultime direttive del W3C:

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

Approfondimento

Non è obbligatorio specificare il DocType infatti la pagina verrà comunque visualizzata dal browser, e nella maggior parte dei casi correttamente. Tuttavia consigliamo caldamente l'uso di questo tag, così facendo la pagina sarà in armonia con gli standard minimi di validità

Formattare il testo

Ad eccezione dei tag di stile, l'insieme dei tag che stiamo per esaminare è stata *sostituita* dall'uso dei fogli di stile che consentono di scrivere una mole di codice decisamente inferiore e hanno molte altre caratteristiche che li fanno preferire all'uso di questi tag.
Se è realmente necessario mantenere una totale compatibilità verso i browser più vecchi (quali Explorer 3, Netscape 4.7 etc.) che praticamente non supportavano i fogli di stile se non in maniera molto approssimata, occorre comunque continuare ad usare i tag ed i loro attributi.

Font

E' possibile scegliere l'aspetto che assumerà il testo attraverso l'attributo *face* i cui valori sono tutti i nomi di font esistenti. Tuttavia è preferibile scegliere sempre le famiglie generiche di font (ogni font appartiene a una famiglia) in maniera tale da garantire la massima compatibilità con tutta l'utenza rinunciando ad un font che probabilmente l'utente non avrà installato sul proprio pc. I possibili valori dell'attributo **face** sono molteplici tra cui: *Arial*, *Verdana*, *Helvetica*, *Times*, più tutte le famiglie generiche di font e così via.

Famiglia di font	Caratteristiche	Esempi di font

serif	Sono proporzionati e hanno le grazie	Times, Georgia
sans-serif	Sono proporzionati e non hanno le grazie	Helvetica, Geneva, Verdana,Arial
monospace	Non sono proporzionati, con o senza grazie	Courier, Courier New
cursive	Hanno le grazie	Vivaldi, Comic Sans
fantasy	Non sono classificabili	Woodblock

*Per avere una lista completa dei font vai su Wikipedia, vedi la voce **lista di font**.*

Colore

Il colore si può impostare attraverso l'attributo **color** i cui valori possono essere tutti i colori disponibili sia in forma nominale che in forma esadecimale.

Esempio:

```
<body>
  <font color="red">Questo testo sarà di colore rosso</font>
  <font color="#FF0000">Questo testo sarà di colore rosso</font>
</body>
```

*Per avere una lista completa dei colori con i rispettivi codici esadecimali vai su Wikipedia, vedi la voce **lista dei colori**.*

Stile

Nell'HTML vi sono due stili utilizzabili per determinare il modo in cui sarà visualizzato il testo:

- *Stile fisico*: non si preoccupa della funzione che ha il testo all'interno del contesto pagina e si limita a definire solo gli attributi del testo
- *Stile logico*: gli stili logici al contrario si preoccupano anche di veicolare all'interprete la funzione del blocco di testo che è presente al loro interno.

Gli stili logici sono quelli maggiormente supportati anche dai vecchi browser in quanto presenti sin dalle prime versioni dell'HTML.

Da un punto di vista concettuale è grandemente preferibile usare gli stili logici, specialmente in vista di un utilizzo esteso dei fogli di stile.

Ad esempio, marcando col il tag <code> una porzione di testo che racchiude del codice sorgente, tale testo sarà normalmente visualizzato con caratteri non-proporzionali.
Se in un secondo tempo si desidera che tutti i testi marcati <code> siano visualizzati diversamente (per esempio, su uno sfondo colorato e delimitato da un bordo), sarà sufficiente ridefinire il tag <code> in un foglio di stile, con gli attributi necessari.

Ci sono diversi valori per impostare lo stile di un testo, di seguito una tabella riassuntiva.

Stile logico	Stile fisico	Significato
`<strong>`	`<b>`	**b** rende il testo in grassetto; **strong** è usualmente (ma non sempre) visualizzato in grassetto

	<i>	Rende il testo in corsivo
<cite>	<u>	Rende il testo sottolineato
<code>	<tt>	Rende il testo monospaziato
<kbd>	<tt>	Rende il testo monospaziato come <code>

Dimensioni

Le dimensioni sono determinate, su una scala arbitraria, dal valore dell'attributo **size**, che può essere compreso tra *1* e *7*. Il valore predefinito è *3*. Qualora si inserisse un valore minore di uno o maggiore di sette, verrano interpretati dal browser come dimensione 1 o 7.

E' possibile anche determinare una dimensione base del font attraverso il tag <basefont> e il suo attributo **size** e lo si può ingrandire come si desidera semplicemente inserendo gli operatori + e - seguiti dal valore per cui si desidera incrementare o decrementare la dimensione base. Se non viene determinato il valore base di basefont **size**="valore" è di *3*.

Ecco un esempio:

```
<body>
  <font size="3">Questo testo sarà di dimensione 3</font>
</body>
```

Ecco un esempio con basefont:

```
<body>
  <basefont size="4">
  <font size="+2">Questo testo sarà di dimensione 6</font>
  <font size="-2">Questo testo sarà di dimensione 2</font>
</body>
```

Ecco un altro esempio senza il tag basefont che quindi assumerà valore 3:

```
<body>
  <font size="+2">Questo testo sarà di dimensione 5</font>
  <font size="-2">Questo testo sarà di dimensione 1</font>
</body>
```

Organizzare il testo

Titoli

I titoli sono molto utili nelle pagine per dare maggior visibilità a un elemento che, solitamente, racchiude ciò che sarà possibile leggere nei paragrafi che seguono. Vi sono 6 livelli di titoli, il primo è quello più grande e man mano che si va verso il sesto il titolo risulterà più piccolo.

I titoli sono automaticamente visualizzati dall'interprete in grassetto e lasciano prima e dopo di sè, un margine considerevole (in modo che risulti più leggibile). La sintassi per definire un titolo è la seguente: <h1>Inseriamo il titolo del paragrafo</h1>.
Al posto del numero uno ovviamente si possono inserire i numeri: 2,3,4,5,6 a seconda della grandezza che si desidera visualizzare.

Paragrafi

Vi sono essenzialmente tre tag che svolgono la funzione di suddivisione del testo: <p>, ,e

. Le differenze tra questi tag sono minime ma abbastanza determinati, nel contesto della pagina, per ciò che verrà visualizzato dal browser.

Tag	Significato
<p>	Elemento block, di default ha dei margini superiori e inferiori che lo separano dal resto dei contenuti.
	Elemento block. Di default il testo non ha margini.
	Elemento in-line, serve a raggruppare il testo concettualmente

- <p> sta per *paragraph* ed è infatti l'elemento che rappresenta la nostra concezione di paragrafo con dei margini inferiori e superiori.
- sta per *division* e definisce un blocco di testo al suo interno. In realtà il tag è, prima che un separatore di testo, un contenitore di elementi HTML. Il suo utilizzo è spesso associato ai fogli di stile.
- fa da contenitore al testo presente al suo interno. La sua utilità è evidente quando si usano i fogli di stile.

Ecco un esempio:

```
<body>
 <p>Questo è un paragrafo</p>
 <div>A cui seguono due blocchi di testo, questo è il primo.</div>
  <div>Questo è il secondo</div>
 <span>Questo è un contenitore niente di più</span>
</body>
```

Approfondimento

I tag <p>, ,e supportano sia l'attributo **class** sia **id**, essenziali per l'interazione con i fogli di stile.

Linee divisorie

Per impaginare meglio un testo si potrebbe rendere necessaria una linea che divida ad esempio un articolo dall'altro il tag che svolge questa funzione è <hr>. I suoi principali attributi sono **width** che ne indica la larghezza, richiede un valore in pixel o in percentuale; **size** che ne regola le dimensioni, il cui valore viene espresso in pixel di default 2; e infine **align** che ci permetterà di allineare la barra rispettivamente a destra, al centro o a sinistra.

Ecco un esempio:

```
 <body>
   <h1>Wikibooks</h1>
   <hr width="100%" size="3">
   <div>Questo è una guida a contenuto aperto</div>
 </body>
```

Nota

Il tag <hr> non prevede un tag di chiusura

Andare a capo

Il tag che svolge la funzione per andare a capo è <br>. Alcuni webmaster gli preferiscono dei tag

<p> senza il tag di chiusura perché sono più veloci da scrivere però così facendo lasciano il paragrafo aperto incorrendo ad un errore sintattico.

Esempio di codice sintatticamente sbagliato:

```
<body>
  <h1>Wikibooks</h1>
  <p>
  <p>
  <p>
  <div>Questo è una guida a contenuto aperto</div>
</body>
```

Ecco la versione corretta:

```
<body>
  <h1>Wikibooks</h1>
  <br>
  <br>
  <br>
  <div>Questo è una guida a contenuto aperto</div>
</body>
```

Nota

Il tag <br> non prevede un tag di chiusura

Testo preformattato

Indica del testo che verrà visualizzato così come è scritto all'interno del documento HTML, rispettando rigorosamente gli spazi. Questo tag è molto usato ad esempio nei forum per rappresentare il sorgente delle applicazioni.
La sua sintassi è la seguente: <pre>Inserite del testo con diversi spazi di separazione tra le parole e il browser lo riporterà così come l'avete scritto.</pre>

Elenchi

Per definire un elenco o *lista* nell'HTML abbiamo a disposizione diversi tag che ci permetto di creare tre tipi di elenchi: *elenco ordinato*, *elenco non ordinato*, e infine *elenco di definizione*.

Elenchi ordinati

Gli elenchi ordinati vengono definiti attraverso il tag <ol> che sta per *Ordered List*. Gli elementi dell'elenco, devono essere inclusi all'interno dei tag ossia *List Item*. Ogni elemento verrà automaticamente preceduto da un numero.
È utile l'uso dell'attributo *start*, il quale specifica il numero da cui partirà la numerazione dell'elenco.

Esempio di elenco ordinato:

```
<body>
<ol>
  <li>1° Elemento in ordine numerico</li>
  <li>2° Elemento in ordine numerico</li>
  <li>3° Elemento in ordine numerico</li>
</ol>
</body>
```

Elenchi non ordinati

Gli elenchi non ordinati, le cui voci sono precedute da un pallino, vengono definiti attraverso il tag `<ul>` che sta per *Unordered List*. Gli elementi dell'elenco, devono sempre essere inclusi all'interno dei tag `<li></li>`.

Esempio di elenco non ordinato:

```
<body>
<ul>
  <li>Elemento uno</li>
  <li>Elemento due</li>
  <li>Elemento tre</li>
</ul>
</body>
```

E' possibile anche cambiare l'immagine del pallino attraverso l'attributo **type** i cui possibili valori sono:

- *circle*: visualizzerà un pallino vuoto dentro (bianco al suo interno).
- *disc*: visualizzerà un pallino pieno (nero al suo interno) è il valore di default.
- *square*: visualizzerà un quadratino pieno (nero al suo interno).

I browser che non supportano questo attributo lo ignoreranno.

Esempio di elenchi con diversi stili:

```
<body>
  <ul type="disc">
     <li>Elemento con pallino</li>
     <li>Elemento con pallino 2</li>
  </ul>
  <ul type="circle">
     <li>Elemento con pallino vuoto</li>
     <li>Elemento con pallino vuoto due</li>
   </ul>
   <ul type="square">
     <li>Elemento con quadratino pieno</li>
     <li>Elemento con quadratino pieno 2</li>
   </ul>
</body>
```

Elenchi di definizioni

Gli elenchi di definizioni sono degli elenchi un po' particolari che prevedono due parti:

- Un elemento di testo
- Una spiegazione dell'elemento

Questa caratteristica renderà questo tipo di elenchi utili per piccoli glossari, o anche per la gestione delle FAQ sul proprio sito.

Per poterle usare sono necessari tre tag: `<dl>`, *Definition List*; `<dt>`, *Definition Term*; e `<dd>`, *Definition Defined*. Il primo tag serve per contenere l'elenco ed indicare di che tipo è, in questo caso di definizione. Il secondo tag serve per specificare la parola che verrà spiegata o meglio definita. Il terzo tag serve per contenere la spiegazione che verrà rientrata rispetto alla parola da definire.

Vediamo un esempio che chiarirà meglio le idee:

```
<body>
 <dl>
   <dt>Wikibooks</dt>
     <dd>Stiamo sviluppando e distribuendo libri di testo,
```

```
   manuali e altri testi   educativi, tutti gratis e a contenuto aperto.</dd>
    <dt>Wikipedia</dt>
      <dd>Wikipedia è un'enciclopedia libera</dd>
  </dl>
 <body>
```

Testo dinamico

Testo lampeggiante

Un modo molto semplice per far apparire e scomparire rapidamente il testo, ottenendo un lampeggiamento, è l'uso del tag `<blink>`.

Ecco un esempio:

```
 <body>
   <blink>Testo lampeggiante</blink>
 </body>
```

Nota

Il tag <blink> non rientra nelle speficiche del W3C, pertanto ve ne sconsigliamo l'uso

Testo scorrevole

Per inserire un testo scorrevole è necessario far uso del tag <marquee> i cui principali attributi sono *direction*, i cui valori possono essere left e right; *height* e *width*, i cui valori possono essere espressi o in pixel o in percentuale e determinano l'area di sfondo su cui si muoverà il testo; *loop*, il cui valore può essere un numero e determina quante volte dovrà essere visualizzata la scritta (*-1* o *infinite* per far sì che venga mostrata all'infinito).

Ecco un esempio:

```
 <body>
   <marquee direction="left" width="100%" height="10%" loop="-1">Testo
scorrevole</marquee>
 </body>
```

Il testo si muoverà all'infinito da sinistra a destra.

Nota

Il tag <marquee> è proprietario di Microsoft, non rientra quindi nelle speficiche del W3C, pertanto ve ne sconsigliamo l'uso

I links sono una delle caratteristiche che hanno fatto le fortune del web, ossia la possibilità di passare da un documento ad un altro in maniera molto semplice: cliccando su un collegamento ipertestuale. I link sono costituiti da due componenti: il contenuto e la risorsa.

- Il contenuto può essere definito come la parola che risulta *cliccabile*, o come vedremo più avanti un'immagine, che contengono la risorsa.
- La risorsa è la pagina, il file, a cui punta l'url del contenuto.

Links

Collegamenti esterni

La sintassi per creare un collegamente ipertestuale è molto semplice ed è la seguente: `<a href="qui va inserito l'url">e qui la parola che risulterà cliccabile</a>`.

Ecco un esempio:

```
<body>
   <a href="http://it.wikibooks.org">Homepage di Wikibooks</a>
</body>
```

Collegamenti interni

I collegamenti interni o *àncore* sono collegamenti che non rimandano a una pagina esterna come abbiamo visto in precedenza, bensì a un contenuto disponibile nella stessa pagina. La loro sintassi è leggermente più complicata dei collegamenti esterni. Per prima cosa, si deve creare un collegamento con l'attributo **name** che farà sì che il collegamento non venga visto come un reale link ma al contrario come un'àncora di cui faremo uso per poter definire un collegamento che punti a quel determinato testo della pagina.

Ecco l'esempio:

```
<body>
   <a href="#wiki">Vai al paragrafo su Wikibooks</a>
   <a name="wiki">Questo non è un link ma il testo a cui siamo stati
rimandati</a>
</body>
```

Dunque ricapitolando per poter realizzare un collegamento interno bisogna eseguire le seguenti operazioni:

- Definire l'àncora attraverso l'attributo **name**.
- Richiamarla da un altro link inserendo come valore dell'attributo **href** il valore dell'attributo **name** preceduto dal segno #.

E' possibile anche creare in questo modo link vuoti richiamando un'àncora senza valore:
(ad esempio per creare un link in fondo alla pagina che vi faccia tornare all'inizio della stessa senza doverla ricaricare)

```
<body>
   <a href="#">Questo link è vuoto</a>
</body>
```

Posta elettronica

Attraverso un link è anche possibile aprire il client di posta dell'utente affinché questo possa mandare una mail con il campo *A:* precompilato. Ecco la sintassi: `<a href="mailto:indirizzo mail">`Parola da visualizzare`</a>`

Ecco un esempio:

```
<body>
   <a href="mailto:prova@prova.it">Invia  Mail</a>
</body>
```

Inoltre, è possibile stabilire soggetto e corpo del messaggio. Ecco un esempio:

```
<body>
   <a href="mailto:prova@prova.it?subject=Soggetto&body=Corpo del
Messaggio">Invia  Mail</a>
</body>
```

Attributi dei link

La seguente è una tabella riassuntiva:

Attributo	Valori	Significato
`link`	Qualsiasi colore	Indica il colore di tutti i link presenti nella pagina, il tipo di colore può essere specificato sia col nome inglese sia col relativo codice esadecimale.
`alink`	Qualsiasi colore	Indica tutti i link attivi, il colore, come il precedente può essere espresso sia col nome in inglese sia col relativo valore esadecimale.
`vlink`	Qualsiasi colore	Indica tutti i link gia visitati, il colore, come gli altri può essere espresso sia col nome in inglese sia col relativo valore esadecimale.
`accesskey`	Qualsiasi lettera	Determina le scorciatoie da tastiera, se l'utente clicca la combinazione *ALT+valore* viene mostrato il link
`title`	Qualsiasi lettera/parola	Il valore dell'attributo *title* verrà mostrato quando il cursore sarà sopra il link. Serve per spiegare dove porterà il link.
`target`	_blank (apre in uona nuova pagina) o **_main** (aperto nella stessa pagina)	Determina come verrà visualizzata la risorsa del link: o nella stessa pagina o in una nuova finestra
`hreflang`	Diversi valori es.: ita, eng	Indica la lingua del documento, utile se si ha un sito multilingua. Facilita anche l'indicizzazione da parte dei motori di ricerca.

Immagini

Inserire un'immagine

La sintassi di base per inserire un'immagine all'interno di una pagina web è la seguente `<img src="url dell'immagine" alt="testo alternativo">`

Attributo	Significato
`border`	identifica il bordo che a 0 non è presente, salendo con i numeri aumenta di spessore
`title`	testo informativo, visualizzato posizionando il cursore sopra l'immagine
`width`	forza la dimensione della foto in larghezza, riducendo i tempi di caricamento

`height`	forza la dimensione della foto in altezza, riducendo i tempi di caricamento
`alt`	permette di specificare un testo alternativo, descrittivo dell'immagine

È fortemente consigliato inserire testo alternativo per ogni immagine, in modo da migliorare l'accessibiltà del sito per i non vedenti in quanto il suo contenuto può essere 'letto' da browser adatti. Il testo alternativo verrà inoltre visualizzato in automatico se il browser non riesce a visualizzare l'immagine.

I valori di **width** e **height** possono essere espressi in pixel od come percentuale della larghezza della pagina.

Esempio di inserimento di un'immagine:

```
<body>
  <img src="http://it.wikibooks.org/skins-1.5/monobook/user.gif" alt="immagine
utente"
  height="30" width="20" title="Immagine Utente" />
</body>
```

Nota

Il tag <img> non prevede un tag di chiusura, quindi richiede la barra prima del > finale, per la validazione da parte del WWW Consortium per l'XHTML

Definire l'allineamento

L'HTML permette di definire l'allineamento delle immagini rispetto al testo tramite l'attributo `align` che può assumere i seguenti valori:

- `bottom` (default): il bordo inferiore dell'immagine risulta allineato verticalmente con la prima linea del testo.
- `middle`: la prima riga del testo è allineato con il centro dell'immagine. Dopo la prima riga, il testo scorre sotto l'immagine.
- `top`: la prima riga del testo è allineata con il bordo superiore dell'immagine.
- left
- right

Immagine come link

Se si vuole creare un collegamente ipertestuale su di un'immagine bisogna inserire il tag `<img>` incluso tra i tag `<a href="url">` e `</a>`.

Ecco un esempio:

```
<body>
 <a href="http://wikimediafoundation.org">
  <img src="http://it.wikibooks.org/images/wikimedia-button.png" alt="immagine
di wikimedia"
  height="50" width="50" title="Wikimedia" />
 </a>
</body>
```

Mappa d'immagine

Le mappe d'immagine sono delle mappe costituite da un'immagine sulla cui area sono disposti diversi link. L'esempio più eclatante è la penisola italiana: cliccando, ad esempio, sull'immagine del Lazio si verrà indirizzati verso le pagine inerenti al Lazio; al contrario, cliccando sull'immagine

della Sardegna si verrà indirizzati verso le pagine inerenti alla Sardegna.

Ecco un esempio (è stata definita l'area dell'immagine da mappare, e inserita la mappa da usare (map1) all'interno del tag immagine):

```
<body>


  <MAP NAME="map1">
  <AREA HREF="pagina.html" ALT="Pagina" TITLE="Pagina" SHAPE=RECT
COORDS="6,116,97,184">
  </MAP>
  <img src="immaginedamappare.jpg" usemap="#map1">


</body>
```

Immagine nella barra degli indirizzi

L'HTML ci permette attraverso il tag `<link>` di inserire un'immagine nella barra degli indirizzi accanto all'url del sito che si sta visitando (in Mozilla Firefox l'immagine verrà visualizzata anche accanto al titolo della pagina nei tab). L'immagine che vorremmo usare a tale scopo dovrà avere necessariamente il nome di `favicon`, dovrà essere grande 16x16px o maggiore e potrà essere dei seguenti formati: .gif, .ico o .jpg.

Ecco un esempio su come utilizzare favicon:

```
<head>
  <link rel="icon" href="http://<percorso>/favicon.ico">
</head>
```

Precedemente favicon era un elemento supportato solamente da Internet Explorer e per poterlo utilizzare era necessaria la seguente sintassi: `<link rel="icon" href="http://<percorso>/favicon.ico">`. Tuttavia come già detto, questa sintassi è stata dichiarata deprecata dal W3C e il primo esempio che vi abbiamo mostrato è quello definito negli standard.

Tabelle

Il tag usato per la creazione delle tabelle è `<table>` in coppia con il suo tag di chiusura `</table>`. E' tra questi due tag che si devono inserire colonne e righe, le prime si creano attraverso il tag `<tr>` (*Table Row*); le seconde, attraverso il tag `<td>` (*Table Data*). Se si vuole creare una cella d'intestazione con il contenuto in neretto e centrato si può usare il tag `<th>` (*Table Header*), ideato appositamente per questo tipo di funzione. Possibili attributi di colonne e righe sono align (*alignment*), valign (Vertical alignment), colspan (*Column span*), e rowspan.

- **align**: allinea il testo della cella a destra (*right*), sinistra (*left*) e centrato (*center*).
- **valign**: allinea il testo della cella sul margine superiore (*top*), sul margine inferiore (*bottom*), e in mezzo (*middle*)
- **colspan** e **rowspan**: il primo indica l'estensione di una colonna, il secondo di una riga

Le celle e le colonne supportano anche gli attributi **height** e **width**.

Vi è un tag interessante che è `<caption>` che permette di aggiungere una descrizione alla tabella (una sorta di didascalia) che supporta l'attributo **align** pertanto si posizionerà o sopra o sotto la tabella secondo il valore specificato in tale attributo.

Attributi delle tabelle

Attributo	Significato
`border`	Stabilisce la dimensione del bordo esterno alla tabella. Il valore di default è 0 (ossia assente).
`width`	Stabilisce la larghezza della tabella
`height`	Stabilisce la lunghezza della tabella
`cellspacing`	Stabilisce la spaziatura tra le celle
`cellpadding`	Stabilisce la spaziatura tra il testo e la cella che lo contiene

Esempio di tabella:

```
<table width="100%" height="30%" cellspacing="4" cellpadding="2" border="1">
   <caption align="bottom">Questa è una tabella di esempio</caption>
  <tr>
    <th>Intestazione 1</th>
    <th>Intestazione 2</th>
  </tr>
  <tr>
    <td align="center">Cella 1 colonna 1 testo centrato</td>
    <td valign="top">Cella 1 colonna 2 allineata sopra</td>
  </tr>
  <tr>
    <td>Cella 2 colonna 1</td>
    <td width="70%">Cella 2 colonna 2</td>
  </tr>
 </table>
```

Tabelle preformattate

E' possibile creare delle tabelle anche attraverso spazi e senza l'ausilio del tag `<table>`, ci si può servire infatti del tag `<pre>` che come abbiamo detto in precedenza, dirà all'interprete di visualizzare il testo così com'è stato scritto all'interno del documento HTML, pertanto con i medesimi spazi e a capo. L'unico inconveniente è probabilmente che il testo all'interno del tag `<pre>` verrà visualizzato monospaziato e non potrebbe soddisfare la grafica della pagina. Per determinare la larghezza massima della tabella è sufficiente specificare l'attributo **width** con un valore numerico in pixel o in percentuale.

Ecco un esempio di tabella preformattata:

```
<body>
    <pre>
       prima colonna    | seconda colonna | terza colonna
       primo elemento   | primo elemento  | primo elemento
       secondo elemento | secondo elemento| secondo elemento
   </pre>
</body>
```

Sfondo

Colore di sfondo

Per poter settare un colore come sfondo di una pagina è sufficiente servirsi dell'attributo **bgcolor** che va inserito all'interno del tag `<body>`. L'attributo bgcolor sta per *background color* che com'è

facile intuire corrisponde all'italiano: colore di sfondo. Come valore dell'attributo bgcolor si può impostare qualsiasi colore, sia attraverso il suo valore nominale che attraverso il suo valore esadecimale.

La sintassi per l'uso di **bgcolor**:

```
<body bgcolor="#FFFF00">
  La pagina avrà uno sfondo di colore giallo (valore esadecimale)
</body>

<body bgcolor="black">
  La pagina avrà uno sfondo di colore nero (valore nominale)
</body>
```

Immagine di sfondo

Una volta settata un'immagine come sfondo di una pagina essa verrà ripetuta sia orizzontalmente che verticalmente, bisognerà quindi stare attenti alla colorazione del testo in modo che questo sia in ogni caso leggibile. E' possibile anche combinare un'immagine di sfondo con una colorazione di modo che nell'attesa del caricamento dell'immagine l'utente possa comunque leggere il testo. L'attributo per poter inserire un'immagine di sfondo è **background** che va inserito sempre all'interno di `<body>`, il suo valore ovviamente sarà l'url che porterà all'immagine in questione.

Ecco un esempio:

```
<body bgcolor="#FFFF00" background="immagine.jpg">
  Questa pagina contiene un'immagine come sfondo, tuttavia mentre essa verrà
caricata
  vedrai uno sfondo di colore giallo
</body>
```

Multimedia

Attraverso l'HTML si possono anche inserire contenuti multimediali all'interno delle proprie pagine. Per permettere di scaricare un file è sufficiente creare un link che come *risorsa* ha il file che vorreste far scaricare.

Ecco un esempio:

```
<body>
   <a href="http://www.url.it/file.mpg">Scarica il video!</a>
</body>
```

Si può usare la stessa sintassi per qualsiasi altro tipo di file: audio, archivi e così via.

Suoni

Se si desidera far ascoltare ai propri utenti un suono di sottofondo alla pagina è sufficiente inserire il tag `<bgsound>` in cui attributi sono **src** e **loop**. La sintassi è la seguente: `<bgsound src="url del file audio" loop="numero di volte da riprodurre">`. L'attributo **loop** serve a comunicare all'interprete quante volte dovrà ripetere il file audio specificato nell'attributo **src**. Per poter riprodurre il file audio all'infinito si può usare il valore *infinite* o *-1*, anche se ciò è sconsigliabile perché può dar fastidio all'utente.

Ecco un esempio di integrazione del suono

```
<body>
   <bgsound src="prova.mid" loop="2">
```

```
   <div>Questa pagina contiene un file audio che verrà ripetuto per due volte
consecutive</div>
</body>
```

Nota

Il tag <bgsound> è proprietario di Microsoft, non rientra quindi nelle speficiche del W3C, pertanto ve ne sconsigliamo l'uso

Moduli

Il tag <form> si occupa di definire il modulo, tutto ciò che è tra il suo tag d'apertura e quello di chiusura è parte del modulo stesso. Possiamo definire diversi tipi di elementi: campi di testo, checkbox, area di testo e box di selezione. Gli attributi del tag `<form>` sono action e method.

- **action**: serve per specificare a quale file verranno inviati i dati
- **method**: serve a indicare il metodo attraverso i quali saranno spediti

Tipi di campi

Per poter creare un campo di testo, o una casella ci serviremmo del tag `<input>`. I principali attributi di `<input>` sono **type** (che determina il tipo di elemento), **name** (che determinano un nome per ogni campo in modo da far comprendere a uno script quali opzioni sono state scelte) e **value** (per inserire del testo precompilato nei campi).

All'interno dei *form* è possibile inserire uno o più campi e di vario tipo, facendo uso del tag `<input>` i cui attributi per definire i tipi di campi sono i seguenti:

Valore	Tipo di campo
text	Campo di testo
radio	Voci da selezionare, mutuamente esclusive
checkbox	Voci per selezione multipla

Il valore *text* creerà una finestra in cui sarà possibile scrivere ciò che si desidera. Vi sono altri due attributi del tag `<input></input>` che possono essere molto utili e sono:

- **size**: stabilisce la lunghezza del campo da compilare
- **maxlenght**: stabilisce il numero massimo di caratteri che il campo acceterà

Se si vuole oscurare il contenuto di un campo (ad esempio per permettere di inserire una password senza che esa venga visualizzata da altri) è sufficiente inserire *password* come valore dell'attributo **type**.

Ecco un esempio di ciò che si è visto finora:

```
<body>
 <form action="prova.php" method="post">
  Inserire un nome:
   <input type="text" name="nome" size="20" maxlenght="15"><br>
  Inserire un cognome:
   <input type="text" name="cognome" size="40"><br>
  Inserire una password:
   <input type="password" name="pass" size="40">
 </form>
</body>
```

Menu a scelta

E' possibile creare dei menù a discesa all'interno dei moduli questo permetterà all'utente di scegliere tra voci predefinite. Per creare un menù a discesa si usa il tag `<select>`, `</select>` al cui interno verrà definita ogni voce attraverso il tag `<option>`, `</option>`.

La sintassi per creare un menù a discesa è la seguente: `<select name="nome del menù"><option>Testo che si vedrà nel menù</option></select>`

Esempio di menù a discesa:

```
<body>
 <form action="prova.php" method="post">
  <select name="menu">
   <option>Wikibooks</option>
   <option>Wikipedia</option>
   <option>Wikisource</option>
  </select>
 </form>
</body>
```

Si può anche creare un altro tipo di menù in cui non vi è un menù a cascata ma si scorre semplicemente un elenco. La sintassi per creare questo tipo di menù prevede l'uso dell'attributo **size**, all'interno del tag <select>, che conterrà come valore il numero delle opzioni del menù.

Esempio di elenco da cui scegliere:

```
<body>
 <form action="prova.php" method="post">
  <select name="menu" size="3">
   <option>Wikibooks</option>
   <option>Wikipedia</option>
   <option>Wikisource</option>
  </select>
 </form>
</body>
```

Area di testo

Le aree di testo si possono creare utilizzando il tag `<textarea>` che viene accompagnato dal suo tag di chiusura `</textarea>`. Gli attributi del tag `<texarea>` sono principalmente due:

- **rows**: indica il numero di righe che conterrà l'area di testo
- **cols**: indica il numero di colonne che conterrà l'area di testo

Ovviamente il valore di questi due attributi sarà un numero a nostra discrezione.

Altri due attributi sono **name** che dà un nome all'area di testo e **wrap** (un attributo senza valore) che manderà automaticamente a capo il testo che verrà scritto all'interno della *textarea*.

Ecco un esempio di textarea:

```
<body>
  Commenti su wikibooks?<br>
 <form action="prova.php" method="post">
  <textarea name="messaggio" rows="20" cols="50" wrap></textarea>
 </form>
</body>
```

Pulsanti

Esistono due pulsanti che si possono definire tramite l'HTML uno per inviare i dati a un'altra pagina e un'altro per resettare i campi di un modulo per poterli ricompilare, il primo è contraddistinto dal valore *submit* nell'attributo **type**; il secondo dal valore *reset* sempre nell'attributo **type**.

Esempio di pulsante che invia i dati:

```
<body>
 <form action="prova.php" method="post">
  <input type="submit" value="Invia!">
 </form>
</body>
```

Esempio di pulsante che cancella i dati:

```
<body>
 <form action="prova.php" method="post">
  <input type="reset" value="Cancella Tutto">
 </form>
</body>
```

I Frames

A cosa servono i frame?

I frames che si possono rendere in italiano con il termine *riquadri* sono dei porzioni di pagina indipendenti l'una dell'altra. Ad esempio si può dividere la pagina in 3 frame: uno a sinistra, per il menù; una a destra, per i links; e infine la parte centrale per il contenuto della pagina. L'obiettivo dei frame è quello di evitare di dover riscrivere interamente ogni pagina in tutte le sue componenti ma ad esempio, suddividendo la pagina come suggerito prima, per lasciare intatti i frame menù e links, si potrà ricaricare solamente il contenuto del frame centrale, permettendo così di gestire una mole di codice nettamente minore.

Approfondimento

L'uso dei frame per quanto possa sembrare vantaggioso, è sconsigliabile per i seguenti motivi:

- Per mantenere una parte di pagina invariata si può usare un linguaggio lato server come PHP, e con veramente poco codice avrete lo stesso risultato dei frame.
- Per riuscire a disegnare dei frame graficamente accattivanti è necessario saper utilizzare i fogli di stile.

Creare una pagina

Per poter creare una pagina con i frame utilizzeremo il tag `<frameset>` che inizializza la struttura di una pagina con i frame. All' interno del tag `<frameset>` useremo il tag `<frame>` i cui attributi principali sono **id**, per dare un nome al frame; e **src**, per indicare la pagina che verrà caricata all'interno del frame stesso. Gli attributi del tag `<frameset>` ci permettono invece di dare delle dimensioni ai frame essi sono due:

- **cols**: specifica la dimensione delle colonne di ogni frame. Se viene omesso la struttura della pagina sarà a righe.
- **rows**: specifica la dimensione delle righe di ogni frame. Se viene omesso la struttura della pagina sarà a colonne.

Il codice dei seguenti esempio e in generale la struttura dei frame non necessita di trovarsi all'interno del tag <body>. Lo si deve inserire dopo il tag di chiusura </head>

Ecco un esempio con la struttura a colonne:

```
<frameset cols="20%,60%,20%">
 <frame src="pagina.htm">
 <frame src="pagina2.htm">
 <frame src="pagina3.htm">
</frameset>
```

Ecco un esempio con la struttura a righe:

```
<frameset rows="20%,60%,20%">
 <frame src="pagina.htm">
 <frame src="pagina2.htm">
 <frame src="pagina3.htm">
</frameset>
```

I valori di questi due attributi possono essere espressi in percentuale (rispetto alla pagina) o in pixel. C'è un elemento complementare * (asterisco) che si può usare ad esempio per riempire tutta la porzione di pagina non compresa dagli altri frame.

Ecco un esempio:

```
<frameset cols="30%,50%,*">
 <frame src="pagina.htm">
 <frame src="pagina2.htm">
 <frame src="pagina3.htm">
</frameset>
```

Il terzo frame sarà equivalente al 20% della pagina, ossia la porzione restante di pagina non occupata dagli altri due frame.

Fuori dai frame

Per poter scrivere al di fuori della struttura dei frame è necessario servirsi del tag <noframes> seguito da tutta la formattazione html necessaria ai vostri scopi </noframes>

Ecco un esempio di uso del tag <noframes>:

```
<frameset cols="30%,50%,*">
   <frame src="pagina.htm">
   <frame src="pagina2.htm">
    <noframes>
    Se non ti piacciono i frame usa questo menù:
      <a href="pagina.htm">Pagina 1</a>

      <a href="pagina2.htm">Pagina 2</a>

      <a href="pagina3.htm">Pagina 3</a>
    </noframes>
   <frame src="pagina3.htm">
  </frameset>
```

Attributi del frameset

Ecco una tabella riassuntiva degli attributi del tag <frameset>

Attributo	Significato

`frameborder`	Specifica se si desiderano i bordi, *yes* (opzione di default), o meno *no*
`framespacing`	Specifica lo spazio tra un frame e un altro (solo IE)
`bordercolor`	Specifica il colore dei bordi del frameset
`border`	Specifica lo spazio tra un frame e un altro, espresso in pixel

Attributi dei frame

Ecco una tabella riassuntiva degli attributi del tag `<frame>`

Attributo	Significato
`scrolling`	Specifica se si desiderano i bordi, *yes* (opzione di default), o meno *no*
`noresize`	Impedisce il ridimensionamento di un frame (no necessita di valori)
`marginheight`	Specifica la distanza in verticale tra il bordo del frame e il suo contenuto
`marginwidth`	Specifica la distanza in orizzontale tra il bordo del frame e il suo contenuto
`frameborder`	Specifica se i bordi sono visibili o meno, rispettivamente i valori 1 e 0

Caricare una pagina

Attraverso l'attributo target del tag `<a>` è possibile caricare i contenuti di un link in un frame specifico o in nuova finestra. Questa è la sintassi: `<a href="url alla pagina da inserire nel frame" target="nome del frame">Testo del link</a>`

Ecco un esempio:

```
<body>
 <a href="pagina.htm" target="centro">Il link verrà caricato all'interno del
frame 'centro'</a>
</body>
```

L'attributo target può avere, oltre al nome del frame dichiarato negli attributi **name** o **id**, i seguenti valori:

- *_blank*: carica il contenuto in una nuova finestra.
- *_top*: carica il contenuto sovrastando la struttura del frameset.
- *_self*: carica il contenuto nello stesso frame del link.
- *_parent*: carica il contenuto nel frameset genitore.

Ecco un esempio:

```
<body>
 <a href="pagina.htm" target="_blank">Il link verrà caricato in una nuova
finestra</a>
</body>
```

E' possibile anche dare un valore di default all'attributo target mediante il tag `<base>`.

Ecco un esempio:

```
<head>
 <base target=”nomeframe”>
</head>
```

Questo farà si che i link si aprano all'interno del frame che si è indicato, a meno che nell'attributo **target** di un link non sia specificato un altro valore.

Iframe

Iframe sta per *Inline frame* e permette di inserire un frame anche in una pagina non strutturata in frame. E' sufficiente servirsi del tag `<iframe>` i cui attributi sono **width**, che ne specifica la larghezza; **height**, che ne specifica l'altezza; e **src** che specifica il documento da caricare al suo interno (anche un url esterno se necessario). Per la sua comodità questo tag è stato subito incluso nelle specifiche dal W3C.

Ecco un esempio di iframe:

```
<body>
 <iframe width="50%" height="30%" src="http://it.wikipedia.org">
 Se il vostro browser non supporta i frane verrà mostrato questo testo
 </iframe>
</body>
```

Il tag `<iframe>` supporta inoltre tutti gli attributi del tag `<frame>`.

Caratteri speciali

I caratteri speciali sono dei caratteri che per poter essere visualizzati come ci si aspetterebbe necessitano di un particolare codice, in caso contrario l'interprete potrebbe non visualizzarlo correttamente. Per poter codificare un carattere, come per gli stili del testo, vi sono due metodi:

- codice numerico o *Numerical Reference*
- codice nominale o *Entity Name*

I codici nominali sono molto usati da chi è anglofono perché in sostanza il loro codice corrisponde al nome del carattere nella lingua inglese. I codici numerici forse saranno più semplici da usare per i non-anglofoni. Non c'è un motivo valido per preferire un set di codificazione rispetto all'altro tuttavia si tenga a mente che i codici numerici sono maggiormente supportati dai browser, proprio perché la loro comprensione è universale.

I codici in entrambi i caratteri sono costituiti da tre parti:

- &
- valore numerico o nominale
- ;

Ecco un tabella dei caratteri speciali

Carattere	Codice nominale	Codice numerico
&	&amp;	&#38;
÷	&divide;	&#247;
¢	&cent;	&#162;
©	&copy;	&#169;
>	&gt;	&#62;
<	&lt;	&#60;

µ	&micro;	&#169;
·	&middot;	&#183;
£	&pound;	&#163;
§	&sect;	&#167;
¥	&yen;	&#165;
®	&reg;	&#174;
±	&plusmn;	&#177;
¶	&para;	&#182;
á Á	á Á	á Á
à À	à À	à À
â Â	&acric; &Acric;	â Â
å Å	å Å	å Å
ã Ã	ã Ã	ã Ã

Includere altri linguaggi

Fogli di stile

Il tag <style>

E' possibile scrivere direttamente nell'header (quindi bisogna includerle tra i tag `<head> e </head>`)le istruzioni che desideriamo grazie al tag `<style>` la cui sintassi è `<style type="tipo">istruzioni di stile</style>`.

Ecco un esempio:

```
<head>
  <style type="text/css">DIV {FONT-FAMILY: Verdana,Helvetica;FONT-SIZE:11px}
   </style>
</head>
```

Questo renderà tutto il testo contenuto tra i tag `<div>,</div>`, del font Verdana o in caso di sua assenza di Helvetica, famiglia generica. Il testo sarà grande 11px.

Nota

ll tipo di foglio di stile non deve essere necessariamente *text/css* ma dipende da foglio di stile che si desidera utilizzare, ad oggi i CSS sono sicuramente i più usati ma verranno presto soppiantati o comunque affiancati dagli XSLT

Inclusione da file esterno

E' possibile creare dei veri e propri fogli di stile che contengono solamente istruzioni di stile con totale assenza di HTML (nel caso in cui fosse presente, il foglio di stile non funzionerà). Per poter includere i fogli di stile all'interno dell'HTML è necessario il tag `<link>`, la cui sintassi è la

seguente `<link rel="parola" href="url del foglio di stile" type="tipo">`. Il tag deve essere necessariamente incluso tra i tag `<head></head>`.

Ecco un esempio:

```
 <head>
  <link rel="stylesheet" href="http://www.prova.it/fogliodistile.css"
type="text/css">
 </head>
```

Inclusione inline

La maggior parte dei tag supportano l'attributo **style** i cui valori possono essere istruzioni di stile CSS di qualsiasi tipo. In questo caso sarà necessario omettere le parentesi graffe per far sì che il codice venga interpretato nella maniera corretta dal browser. Gli elementi inline solitamente prevalgono sugli altri metodi di inclusione di istruzioni di stile, quindi se in un foglio di stile esterno scriviamo di voler avere tutto il testo della pagina in blu, possiamo modificare il colore dei soli titoli attraverso una dichiarazione inline, rendendoli ad esempio, neri.

La sintassi è: `<nometag style="istruzioni di stile"></nometag>`

Ecco un esempio:

```
<body>
   <div style="color: red;">questo testo sarà rosso</div>
</body>
```

Flash

Per inlcudere un'animazione in flash è necessario usare il tag `<object>` al cui interno è necessario inserire il tag `<param>`.

Ecco un esempio:

```
<body>
  <object type="application/x-shockwave-flash" data="animazione.swf" width="10%"
height="10%">
   <param name="movie" value="animazione.swf">
   </object>
</body>
```

Javascript e VBScript

Per includere uno script di uno di questi due linguggi si può procdere come segue:

```
<body>
   <script language="vbscript">MsgBox "Sei su wikibooks",0,"Wikibooks"</script>
</body>
```

Questo codice stamperà a video una finestra di dialogo, all'interno dei tag ovviamente è possibile inserire qualsiasi istruzione del linguaggio in questione. Per inserire uno javascript si usa la medesima procedura:

```
<body>
   <script language="javascript">
 function stampa()
  {
   if (window.print)
   {
```

```
    window.print();
  }
  else
  {
    alert('Funzione non supportata dal browser.');
  }
 }</script>
</body>
```

Questo codice permette, una volta richiamata la funzione, di stampare il testo della pagina. Se Javascript non è supportato dal browser si otterrà il seguente messaggio d'errore `Funzione non supportata dal browser`.

I Verificatori

I verificatori o *validatori* sono dei programmi che verificano la correttezza della sintassi dell'HTML facendo così risaltare eventuali errori.

Verificatore W3C

Il verificatore o *validatore* messo a disposizione sia online che offline dal W3C è un programma che si preoccupa di verificare che la sintassi dell'HTML di un documento rispetti gli standard definiti dal W3C stesso.

Per verificare una pagina, è possibile accedere al validatore a questo indirizzo; tramite questa pagina è possibile verificare un file salvato sul proprio disco, una pagina pubblicata on-line oppure del codice immesso direttamente in una apposita casella di testo.

Dal momento che esistono specifiche più nuove rispetto a quelle dell'HTML, come ad esempio quelle dell'XHTML, il validatore fornisce la possibilità di verificare le pagine secondo diverse direttive; tuttavia, se non viene specificata la specifica in uso della pagine in questione, il validatore è in grado di selezionarne una automaticamente.

Una volta ottenuto il seguente messaggio "This Page Is Valid HTML 4.01 Transitional!" (che apparirà ovviamente solo se la pagina ha un doctype impostato su transitional) è possibile pubblicare sul proprio sito la seguente immagine che attesta la validità del documento:

Il seguente codice permette di cliccare sull'immagine e verificare la validità del sito (codice preso dal sito del W3C).

```
<body>
  <a href="http://validator.w3.org/check?uri=URL DEL SITO">
  <img src="http://www.w3.org/Icons/valid-html401"
  alt="Valid HTML 4.01 Transitional" height="31" width="88"></a>
</body>
```

Ovviamente a `URL DEL SITO` bisognerà sostituire l'indirizzo del proprio sito.

L'evoluzione del markup

Le specifiche HTML più recenti, quelle della versione 4.01, furono rilasciate dal W3C il 24 dicembre 1999.
Da quel momento il linguaggio di markup per la progettazione delle pagine web si è notevolmente evoluto, tanto da rendere obsolete alcune funzionalità dell'HTML. In questo modulo verrà fornita una breve panoramica dell'evoluzione dei linguaggio per la creazione di pagine web, che si basano

comunque sull'HTML.

XHTML

 *Per approfondire, vedi la voce **Programmare in XML**.*

La parola **XHTML** è l'acronimo per **Extensible HyperText Markup Language** ed è una fusione tra i nomi di due linguaggi, l'HTML e l'XML.

L'idea dell'XHTML è quella di rimappare le regole dell'HTML secondo le regole dell'XML; il vantaggio è quello di ottenere documenti XML per il web e leggibili da tutti i browser che possono quindi usufruire di particolari funzionalità di XML (come l'uso degli XSLT) e di creare documenti ben strutturati (questo ha anche lo scopo, ad esempio, di una migliore lettura della pagina da parte del DOM JavaScript).

CSS

 *Per approfondire, vedi la voce **Linguaggio CSS**.*

Un linguaggio che ha reso obsoleto gran parte dei tag di formattazione HTML è stato il **CSS**, acronimo di **Cascading Style Sheet** (fogli di stile a cascata).

L'idea che sta alla base di questo linguaggio è la separazione concettuale e pratica del contenuto della pagina dalla sua struttura (X)HTML. Il CSS permette infatti di definire per ogni elemento o gruppo di elementi della pagina determinati stili, ad esempio la formattazione del testo grassetto o sottolineato. Facendo un piccolo esempio, quello che con HTML puro sarebbe:

```
<b>testo grassetto</b>
```

potrebbe diventare

```
<span style="font-weight:bolder;">testo grassetto</span>
```

Proprio a seguito della nascita dei fogli di stile il W3C ha deprecato (cioè indicato come obsoleti) molti tag HTML (anche font, ad esempio) in favore del CSS, che è più potente e più flessibile.

Licenza

it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- **C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.
- **I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- **K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

- **M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- **N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- **O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant

Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.