

Appendice A

Quick reference per HTML 4.0

Questa appendice fornisce una breve panoramica su HTML 4.0. Questa versione di HTML è la prima a contenere gli elementi e attributi necessari per un uso efficace dei fogli di stile, in particolare **STYLE** e **CLASS**. In effetti ci sono due versioni di HTML 4.0: HTML 4.0 e HTML 4.0 Transitional. La seconda contiene alcuni tag (le cosiddette estensioni) che esistevano in HTML 3.2, ma che ora sono state deprecate in favore dei fogli di stile. Qui si parlerà solo di HTML 4.0, che non contiene estensioni.

Il testo seguente si concentra sugli elementi più usati in combinazione con i fogli di stile. Vengono menzionati anche altri elementi, ma senza spiegazioni. Per la stessa ragione, non tutti i possibili attributi vengono descritti. Le regole di annidamento degli elementi sono state stratificate tralasciando alcuni casi permessi in HTML 4.0. Il risultato è più semplice da spiegare delle regole complete, ma un po' più restrittivo.

Il testo che segue non è un tutorial su HTML e presuppone che si sia letto almeno il Capitolo I di questo libro. Questa appendice tratta un gruppo di elementi HTML e delle regole di come combinarli.

STRUTTURA DI UN DOCUMENTO

Un documento HTML 4.0 è costituito da elementi annidati in altri elementi. Ciascun elemento comincia con un *tag iniziale* `<...>` e finisce con un *tag finale* `</...>`.

In alcuni casi, un tag può essere omesso anche se, in generale, è preferibile includerli entrambi. Quando un documento viene creato utilizzando un editor dedicato, i tag vengono inseriti automaticamente dal programma.

Al livello più alto, un documento consiste di un elemento, chiamato **HTML**. Al suo interno ce ne sono altri due, **HEAD** e **BODY**:

Cascading Style Sheets

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

Gli elementi possono avere attributi, e questi hanno un nome e un valore, separati da un segno di uguale (=). Il valore viene delineato dalle virgolette ("..." o '!...'). Gli attributi vengono posti solo nel tag iniziale, mai in quello finale:

```
<EM CLASS="surname">
<A REL='copyright' HREF='copy.html'>
```

I nomi degli elementi e degli attributi possono essere scritti in maiuscolo o minuscolo, o un mix dei due. L'esempio precedente potrebbe essere scritto così:

```
<em class="surname">
<a rel='copyright' href='copy.html'>
```

L'ELEMENTO HEAD

L'elemento **HEAD** contiene informazioni sul documento e di norma non viene visualizzato nel browser. Gli elementi che possono essere posti in **HEAD** sono **BASE**, **ISINDEX**, **MAP**, **META**, **LINK**, **STYLE** e **SCRIPT**.

TITLE

È possibile avere un solo elemento **TITLE**, che può contenere esclusivamente testo e nessun altro elemento. Il titolo viene mostrato in genere nella barra del titolo della finestra del browser.

STYLE

L'elemento **STYLE** contiene un foglio di stile, come spiegato nel Capitolo 2. Deve essere aggiunto l'attributo **TYPE**, che dichiara il tipo di linguaggio di stile (per esempio, `<STYLE TYPE="text/css">`).

LINK

LINK specifica le relazioni fra il documento e altri documenti. La relazione viene espressa con una parola chiave posta nell'attributo **REL**. L'URL dell'altro elemento va posto nell'attributo **HREF**.

Appendice A: Quick reference per HTML 4.0

Possono essere presenti zero o più elementi **LINK**. L'elemento non ha contenuto e il tag finale deve essere omesso. Per associare un foglio di stile al documento viene usata la relazione **STYLE-SHEET**:

```
<LINK REL="stylesheet"
      HREF="http://example.com/sty/rf.css">
```

META

L'elemento **META** allega diversi tipi di metainformazioni al documento. L'elemento non ha contenuto e il tag finale deve essere omesso.

BASE

BASE può contenere l'URL del documento stesso o l'URL che fa da base agli URL relativi del documento. L'URL viene incluso come valore dell'attributo **HREF**. L'elemento non ha contenuto e il tag finale deve essere omesso.

Ecco un elemento **HEAD** tipico:

```
<HEAD>
  <TITLE>The oak tree</TITLE>
  <LINK REL="author" HREF="./people/Jones">
  <STYLE TYPE="text/css">
    H1 {font-family: Helvetica, sans-serif}
    BODY {font-family: Bodoni, serif}
  <STYLE>
</HEAD>
```

L'ELEMENTO BODY

BODY può contenere tre tipi di elementi: elementi container, elementi bridge ed elementi speciali. Gli elementi definiti *container* possono contenere esattamente gli stessi elementi di **BODY**. Questi elementi creano divisioni, moduli, lunghe citazioni e altre strutture a un livello alto della struttura. Gli elementi *bridge* contengono testo ed elementi text-level. Per esempio, paragrafi e titoli sono esempi di elementi bridge. Gli elementi *special* includono elenchi, tabelle e determinate parti dei moduli interattivi.

Tutti gli elementi che appaiono nel **BODY** possono essere presenti qualsiasi numero di volte. Tutti accettano questi cinque attributi: **CLASS**, **STYLE**, **ID**, **LANG** (per indicare la lingua utilizzata per scrivere il contenuto dell'elemento) e **DIR** (per indicare la direzione di scrittura: `ltr` significa da sinistra a destra (*left-to-right*), `rtl` da destra a sinistra (*right-to-left*, come in ebraico o in arabo).

Cascading Style Sheets

Elementi container

Gli elementi container (contenitori) sono **DIV**, **BLOCKQUOTE**, **ADDRESS**, **FORM** e **FIELDSET**.

DIV serve a creare una divisione di tipo generale, come un capitolo, un paragrafo, un sunto o delle note. È normale indicarne lo scopo con l'attributo **CLASS** (per esempio, `<DIV CLASS="versi">`).

BLOCKQUOTE rappresenta una citazione composta da uno o più paragrafi.

ADDRESS è il nome e/o l'indirizzo, di solito posizionato dopo un titolo o una citazione, oppure alla fine del documento, che indica l'autore della citazione o del documento.

FORM è un contenitore per un modulo interattivo. L'attributo **ACTION** contiene l'URL del server che elaborerà il modulo. L'attributo **METHOD** contiene il metodo utilizzato per inviare i dati nel modulo al server e può essere `get`, `post` o `put`.

L'elemento **FIELDSET** raggruppa parti di un modulo. Oltre a tutti gli altri elementi che possono essere presenti in **BODY**, può contenere un elemento **LEGEND** che, se presente, deve essere il primo elemento in **FIELDSET**. I browser usano **FIELDSET** per attivare il set degli elementi del modulo tutti insieme. L'elemento **LEGEND** appartiene alla categoria `bridge`.

Elementi bridge

Gli elementi bridge sono **P**, **H1**, **H2**, **H3**, **H4**, **H5**, **H6** e **PRE**. **P** è il normale paragrafo. Da **H1** a **H6** sono i titoli di differente livello. **H1** è il più importante. **PRE** significa preformattato. Tipicamente, la proprietà **white-space** di questo elemento è impostata a `pre` (si veda il Capitolo 6).

Elementi speciali

Gli elementi speciali sono **OL**, **UL**, **DL**, **HR** e **TABLE**. **OL** e **UL** sono elenchi. Possono contenere soltanto elementi **LI** (voce dell'elenco o lista), uno o più. L'elemento **LI** stesso funziona come fosse un container: accetta gli stessi contenuti di **BODY**. **UL** di norma viene visualizzato con un punto elemento come etichetta, mentre **OL** presenta dei numeri in successione. Per esempio (si veda la Figura A.1):

- | |
|---|
| <ul style="list-style-type: none">• primo elemento• secondo elemento <ol style="list-style-type: none">1) primo elemento2) secondo elemento |
|---|

Figura A.1 Liste non ordinate e ordinate.

```
<UL>
  <LI><P>primo elemento</P></LI>
  <LI><P>secondo elemento</P></LI>
</UL>
<OL>
  <LI><P>primo elemento</P></LI>
```

Appendice A: Quick reference per HTML 4.0

```
<LI><P>secondo elemento</P></LI>  
</OL>
```

termine A

Definizione del termine A

termine B1

termine B2

Definizione dei
termini B1 e B2

DL è una lista di definizioni. Contiene una o più definizioni, dove ogni definizione consiste di una o più elementi **DT** (termine), seguiti da uno o più elementi **DD** (definizione). L'elemento **DT** funziona come elemento bridge: contiene elementi di testo e text-level. Il **DD** è un container: può contenere gli stessi elementi di **BODY**. La Figura A.2 mostra come viene visualizzato il seguente **DL**:

```
<DL>  
<DT>termine A</DT>  
<DD><P>Definizione del termine A</P></DD>  
<DT>termine B1</DT>  
<DT>termine B2</DT>  
<DD><P>Definizioni dei termini B1 e B2</P></DD>  
</DL>
```

Figura A.2 Elenco di definizioni.

HR è un elemento senza contenuto e senza tag finale. Il suo scopo è separare i paragrafi senza raggrupparli. Normalmente viene renderizzato come filetto orizzontale o semplicemente come spazio bianco.

L'elemento **TABLE** crea una tabella. Una tabella ha una struttura complessa (si veda il Capitolo 17), ma la parte principale consiste di righe di celle. Il contenuto di una tabella inizia con tre parti opzionali: **CAPTION**, **THEAD** e **TFOOT** (in questo ordine); dopo questi elementi ci saranno uno o più **TBODY**, che contengono la tabella.

CAPTION è un elemento bridge: può contenere solamente testo ed elementi text-level. Definisce la didascalia che verrà mostrata sopra o sotto la tabella.

THEAD contiene la prima riga di una tabella, quella che contiene i titoli delle colonne. Ponendo questi titoli in **THEAD** si permette ad alcuni browser di gestirli in modo speciale, ma possono essere messi anche nel corpo della tabella. Anche **TFOOT** contiene i titoli delle colonne, possibilmente i medesimi, ma alla fine delle stesse. Per tabelle piccole, è possibile omettere **TFOOT** e collocare i titoli direttamente nel corpo della tabella.

Il corpo della tabella è contenuto in uno o più elementi **TBODY**. Nelle tabelle di grandi dimensioni, le righe possono essere raggruppate in **TBODY** multipli. In una tabella semplice di solito basta un solo **TBODY**.

THEAD, **TFOOT** e **TBODY** hanno la stessa struttura. Contengono uno o più elementi **TR** (*Table Row*, righe della tabella). Ogni **TR** contiene zero o più celle.

Ci sono due tipi di cella: **TH** e **TD**. Il primo elemento definisce i titoli della tabella e il secondo i dati. **TH** e **TD** funzionano da container: pos-

Cascading Style Sheets

sono contenere gli stessi elementi di **BODY**. **TH** e **TD** condividono due attributi: **COLSPAN** e **ROWSPAN**. Questi indicano, rispettivamente, su quante colonne e quante righe si espandono le celle. Il valore di default è 1.

Di seguito riportiamo un esempio di tabella semplice (per agevolare la lettura sono stati omessi i tag di chiusura, come permesso da HTML 4.0). La Figura A.3 mostra il risultato.

ANNO 1996				
	Q1	Q2	Q3	Q4
auto	365	320	258	191
bici	165	208	358	391
treni	35	45	53	72

Figura A.3 Una semplice tabella.

```
<TABLE>
  <TBODY>
    <TR>
      <TH ROWSPAN=2>
        <TH COLSPAN=4><P>Anno 1996
    <TR>
      <TH><P>Q1
      <TH><P>Q2
      <TH><P>Q3
      <TH><P>Q4
    <TR>
      <TH><P>auto
      <TD><P>365
      <TD><P>320
      <TD><P>258
      <TD><P>191
    <TR>
      <TH><P>bici
      <TD><P>165
      <TD><P>208
      <TD><P>358
      <TD><P>391
    <TR>
      <TH>treni
      <TD><P>35
      <TD><P>45
      <TD><P>53
      <TD><P>72
  </TABLE>
```

ELEMENTI TEXT-LEVEL

Gli elementi *text-level* sono contenuti nel testo all'interno degli elementi *bridge* e nell'elemento **DT**. Gli elementi *text-level* indicano la funzione di certe parole o frasi. Di seguito è riportato un esempio di paragrafo con del testo e diversi elementi *text-level*. Notate che gli elementi *text-*

Appendice A: Quick reference per HTML 4.0

level possono essere annidati uno dentro l'altro. La Figura A.4 mostra il risultato del codice d'esempio.

Un *quadrato* è un rettangolo in cui tutti i lati hanno uguale lunghezza. I *quadrati* **non** devono essere usati per risolvere il problema A.

Figura A.4 Gli elementi **DFN**, **EM** e **STRONG** all'opera.

```
<P>A <DFN>Un quadrato</DFN> è un rettangolo in cui tutti  
i lati hanno uguale lunghezza. <EM CLASS="instruction">I  
quadrati <STRONG>non</STRONG> devono essere usati per  
risolvere il problema A.</EM></P>
```

Tutti gli elementi text-level accettano gli attributi **CLASS**, **LANG**, **DIR** e **STYLE**.

Molti elementi text-level possono essere annidati arbitrariamente, ma con qualche restrizione. Gli elementi senza restrizioni sono **ACRONYM**, **ABBREV**, **B**, **BDO**, **CITE**, **CODE**, **DFN**, **EM**, **I**, **KBD**, **LABEL**, **Q**, **SPAN**, **STRONG**, **SUB**, **SUP**, **VAR** e **LEGEND**. Quelli con restrizioni sono **A**, **BR**, **IMG**, **INPUT**, **SELECT** e **TEXTAREA**.

Gli elementi text-level ed **EM** e **STRONG** marcano parole o frasi che richiedono enfasi o enfasi accentuata. **DFN** contiene una parola o un termine che deve essere definito. La prima occorrenza di un nuovo termine in un documento spesso viene marcata così. **CODE** e **KBD** sono molto usati parlando di argomenti relativi ai computer. Servono a indicare codice (come una parola di un programma o un comando) e testo da digitare con la tastiera. **VAR** indica una variabile, sia di un programma per computer sia in una formula. **SUB** e **SUP** sono per testo in pedice (come il 2 di H₂O) e apice (come il 2 in E=mc²). Anche se HTML non supporta direttamente le formule matematiche complesse, questi elementi possono essere d'aiuto per quelle semplici.

ACRONYM e **ABBREV** marcano acronimi (abbreviazioni che vengono pronunciate come fossero una singola parola, come NATO, NASA, UNICEF e Benelux) e abbreviazioni (come USA, viz., i.e., W3C e CSS). Marcando questi termini si permette al browser di espanderli, ma è una tecnica molto utile soprattutto per i browser vocali.

B e **I** indicano le parole che devono essere mostrate in grassetto o corsivo rispettivamente. Sono adatti al testo convertito da un formato che non permette di codificare il ruolo delle parole. Per un testo digitato direttamente in HTML, **EM** e **STRONG** di solito sono scelte migliori.

CITE delinea un riferimento bibliografico (un tipo di collegamento che non è un link), come (*Raggett 1996*). **Q** serve a delimitare una piccola citazione o una parola usata come metafora. Il browser inserirà le virgolette appropriate; per esempio, provate con: egli disse: <q>ciaio!</q>.

L'elemento **LABEL** viene usato in combinazione a un elemento contenuto in un modulo (**SELECT**, **TEXTAREA**, **INPUT**, **FIELDSET**) o con un elemento **OBJECT** per fornirne una descrizione. Normalmente si presenta vicino all'elemento. Possiede un attributo aggiuntivo, **FOR**, ri-

Cascading Style Sheets

chiesto e che contiene l'**ID** dell'elemento a cui è associato. Normalmente, in un browser, il clic su un etichetta attiva l'elemento a cui è unito e vi pone il cursore.

LEGEND si presenta solo in un fieldset e fornisce un'etichetta ai controlli del modulo delimitati dal fieldset.

L'elemento **BDO** è richiesto in certi rari casi che si verificano nei documenti scritti sia da sinistra a destra sia da destra a sinistra. Significa *Bi-Directional Override*. Possiede un attributo richiesto, **DIR**, che può valere `ltr` o `rtl`. Il valore dell'attributo indica al programma HTML la direzione del testo contenuto, anche se i caratteri nell'elemento di solito venissero utilizzati in senso inverso (è una cosa diversa dall'attributo **DIR** che tutti gli elementi possiedono, che indica solo la direzione di default per i caratteri che non hanno una direzione definita. Per maggiori informazioni, si vedano le specifiche CSS2.)

SPAN è un elemento di uso generico che deve essere usato quando non è disponibile alcun altro elemento text-level significativo. Può avere un attributo **CLASS** per indicare il ruolo dell'elemento o un attributo **STYLE** per definire lo stile contestualmente; per esempio, per marcare nomi propri, potete fare così:

```
<SPAN CLASS="person">Berners-Lee</SPAN>.
```

Elementi text-level con restrizioni

Questo tipo di elementi text-level differiscono dagli altri elementi text-level nella modalità di annidamento. A parte **A**, non tollerano alcun altro elemento text-level al loro interno.

A è probabilmente l'elemento HTML più importante: rappresenta l'ancora di un collegamento. Oltre ai normali attributi (**CLASS**, **ID**, **STYLE**, **LANG** e **DIR**), possiede l'attributo richiesto **HREF** che contiene l'URL dell'ancora di destinazione. Accetta al suo interno tutti gli elementi text-level eccetto un altro **A**.

BR indica una interruzione di riga forzata, senza aprire un nuovo paragrafo. Non ha contenuto né tag di chiusura.

L'elemento **IMG** serve a inserire immagini e altri semplici oggetti grafici nella pagina. Ha un attributo richiesto, **SRC**, che gestisce l'URL dell'immagine, e un altro attributo richiesto, **ALT**, che gestisce il testo che verrà mostrato in alternativa all'immagine quando questa non è accessibile (per esempio, ``). L'elemento non ha contenuto né tag di chiusura.

INPUT è un elemento utilizzato nei moduli interattivi. Crea pulsanti o piccoli campi di testo e dipende dal valore dell'attributo (richiesto) **TYPE**: **TEXT**, **PASSWORD**, **CHECKBOX**, **RADIO**, **SUBMIT**, **RESET**, **FILE**, **HIDDEN** o **IMAGE**. Possiede un attributo richiesto **NAME** e l'attributo **VALUE**, richiesto da alcuni valori dell'attributo **TYPE**. **INPUT**

Appendice A: Quick reference per HTML 4.0

non ha contenuto né tag di chiusura. Di seguito, alcuni esempi di elementi di input (la Figura A.5 ne mostra l'aspetto tipico).

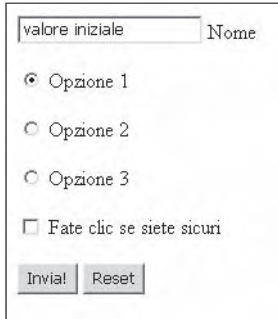


Figura A.5 Esempi di elementi **INPUT**.

```
<FORM ACTION="xx">
  <P><INPUT TYPE="text" NAME="fld9" VALUE="valore iniziale">
    Nome
  <P><INPUT TYPE="radio" NAME="r7" VALUE="a" CHECKED>
    Opzione 1
  <P><INPUT TYPE="radio" NAME="r7" VALUE="b">
    Opzione 2
  <P><INPUT TYPE="radio" NAME="r7" VALUE="c">
    Opzione 3
  <P><INPUT TYPE="checkbox" NAME="pu8">
    Fate clic se siete sicuri
  <P><INPUT TYPE="submit" VALUE="Invia!" NAME="sub-a">
  <INPUT TYPE="reset" VALUE="Reset">
</FORM>
```

SELECT è un elenco o un menu da utilizzare in un modulo interattivo. Può contenere solo uno o più elementi **OPTION** o **OPTGROUP** che, a turno, contengono elementi **OPTION**. Un elemento **OPTION** contiene esclusivamente testo, nessun altro elemento. L'elemento **SELECT** possiede l'attributo richiesto **NAME** che ha come valore del testo arbitrario. Riferitevi alle specifiche HTML per ulteriori attributi. La Figura A.6 mostra il risultato del codice d'esempio:

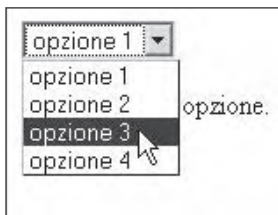


Figura A.6 Esempio di **SELECT**.

```
<FORM ACTION="xx">
  <P><SELECT NAME="hh44">
    <OPTION SELECTED>opzione 1</OPTION>
    <OPTION>opzione 2</OPTION>
    <OPTION>opzione 3</OPTION>
    <OPTION>opzione 4</OPTION>
  </SELECT>
  <P>Selezionate un'opzione.
</FORM>
```

TEXTAREA è un altro elemento per i moduli interattivi. Rappresenta un campo in cui è possibile scrivere una o più righe. Contiene testo e nessun altro elemento. Possiede tre attributi richiesti: **NAME**, **ROWS** e **COLS**. **NAME** contiene testo arbitrario. **ROWS** definisce il numero che indica l'altezza del campo. **COLS** definisce la larghezza del campo. Gli altri attributi sono opzionali:

```
<FORM ACTION="xx">
```

Cascading Style Sheets

```
<TEXTAREA NAME="kk12" ROWS=3 COLS=15>  
Modifica questo testo.  
</TEXTAREA>  
</FORM>
```

OBJECT è un elemento sofisticato. Può inserire immagini nel testo, include applet (programmi da eseguire all'interno del documento) e in generale permette di porre oggetti multimediali nel testo, anche nei documenti HTML. Come principio, rende **IMG** obsoleto, ma **IMG** viene mantenuto per i casi più semplici. L'attributo più importante di **OBJECT** è **DATA**, che contiene l'URL che punta all'oggetto esterno da includere. Possiede numerosi altri attributi, utilizzati per diversi tipi di oggetto. Poiché questa è solo una guida rapida, qui non verranno elencati tutti gli oggetti multimediali e i rispettivi attributi.

Sebbene **OBJECT** stesso sia un elemento text-level, i suoi contenuti sono gli stessi di **BODY**. Tuttavia, idealmente, questo contenuto non viene visualizzato. Lo scopo di questo contenuto viene mostrato solamente se l'oggetto stesso per qualche motivo non può essere caricato. Il contenuto costituisce un'alternativa, come l'attributo **ALT** di **IMG**. Tuttavia, il contenuto alternativo di **OBJECT** è più ricco, e potrebbe essere un documento HTML completo. Di seguito un semplice esempio:

```
<P>provate da soli:  
<OBJECT DATA="nonexistent.ngf">  
<P>Il browser non riesce a caricare un'immagine NGF. Se  
avesse funzionato, vedreste uno specchio d'oro che riflette  
la vostra faccia.<P>  
</OBJECT>  
</P>
```

BUTTON: un elemento container text-level

L'elemento **BUTTON** è un po' come una tabella inline costituita da una cella (si veda il Capitolo 17). Può essere presente solo un elemento bridge e text-level, ma è comunque un container. Viene utilizzato di solito per i pulsanti di invio o reset dei moduli, con la differenza che l'etichetta non ha il vincolo di una riga di testo senza markup, ma può essere invece qualsiasi testo HTML. Normalmente viene renderizzato con un bordo, ma i diversi browser lo potrebbero visualizzare in modo leggermente diverso. Di seguito, un esempio (Figura A.7):



Figura 1.7 Esempi di elementi **BUTTON**.

```
<P>Premete  
<BUTTON STYLE="width: 8em">  
<H2>Rosso</H2>  
<P>per selezionare rosso </P>
```

Appendice A: Quick reference per HTML 4.0

```
</BUTTON>  
o  
<BUTTON STYLE="width: 8em">  
  <H2>Verde</H2>  
  <P>per selezionare verde</P>  
</BUTTON>  
...</P>
```

CARATTERI SPECIALI

Alcuni caratteri possono essere indicati per nome. Questo può essere utile quando il file HTML deve essere spedito con vecchi sistemi di posta elettronica o la tastiera disponibile non permette di digitare questi caratteri in modo semplice. Per esempio, invece di é potete digitare é (inclusi & e ;). Di seguito è riportata la lista completa di questi caratteri.

Lettera	Codice	Lettera	Codice	Lettera	Codice
á	á	é	é	í	í
Á	Á	É	É	Í	Í
â	â	ê	ê	î	î
Â	Â	Ê	Ê	Î	Î
à	à	è	è	ì	ì
À	À	È	È	Ì	Ì
ä	ä	ë	ë	ï	ï
Ä	Ä	Ë	Ë	Ï	Ï
å	å	ó	ó	ú	ú
Å	Å	Ó	Ó	Ú	Ú
æ	æ	ô	ô	û	û
Æ	Æ	Ô	Ô	Û	Û
ç	ç	ò	ò	ù	ù
Ç	Ç	Ò	Ò	Ù	Ù
ð	ð	ö	ö	ü	ü
Ð	Ð	Ö	Ö	Ü	Ü
ñ	ñ	ø	ø	ý	ý

Cascading Style Sheets

Lettera	Codice	Lettera	Codice	Lettera	Codice
Ñ	Ñ	Ø	Ø	Ý	Ý
ß	ß	ö	õ	ÿ	ÿ
þ	þ	Ö	Õ	©	©
Þ	Þ	™	™	®	®
&	&	<	<	>	>
"	"	­		 	
	 	 		 	
—	—	–	–		
	‍	‍			

Un *non-breaking space* () è esattamente come un normale spazio eccetto che non può essere interrotto alla fine di una riga. Un *soft hyphen* (­) è un marcatore invisibile che indica dove è possibile spezzare una riga se necessario, nel qual caso si trasforma in un trattino di sillabazione per indicare che la parola continua nella riga successiva. Un *thin space* ( ) è uno spazio largo circa la metà di uno spazio normale.

Un *em space* ( ) è uno spazio largo come il valore em della font corrente. Un *en space* ( ) è largo la metà di uno spazio em.

Uno *zero-width non-joiner* (‍) è un marcatore invisibile fra due lettere che indica che le lettere non devono essere visivamente combinate. Viene utilizzato per evitare legature. Nelle lingue occidentali non è quasi mai necessario, ma in alcuni linguaggi orientali è piuttosto comune. Uno *zero-width joiner* (‍) fa l'opposto: combina due lettere che non dovrebbero essere legate in una legatura.

XHTML I

Dal 2000, HTML 4 ha un successore, chiamato XHTML I. Contiene esattamente gli stessi elementi di HTML 4, ma è scritto con la sintassi di XML. La sintassi XML renderà più semplice in futuro combinare XHTML con altri linguaggi basati su XML, come MathML (per le espressioni matematiche), SVG (per diagrammi, mappe e altri formati grafici) e SMIL (per il multimedia) in un singolo formato per diversi tipi di documenti.

In questo libro per gli esempi è stato utilizzato HTML 4, ma tutti i browser attuali accettano anche XHTML I. Le differenze di sintassi sono poche.

Appendice A: Quick reference per HTML 4.0

- Tutti gli elementi in XHTML devono essere scritti in minuscolo.

HTML	XHTML
<P>	<p>
<H1>	<h1>
	
<DIV>	<div>

e così via.

- In XHTML tutti gli elementi che non hanno un tag di chiusura devono avere uno slash (/) appena prima del carattere >.

HTML	XHTML
	
<hr>	<hr />
<link>	<link />
<meta>	<meta />
<base>	<base />
<isindex>	<isindex />
<input>	<input />

- XHTML richiede che tutti gli elementi che in HTML possono non avere un tag di chiusura invece lo abbiano.

HTML	XHTML
<html>...	<html>...</html>
<body>...	<body>...</body>
<p>...	<p>...</p>
...	...
<dt>...	<dt>...</dt>
<dd>...	<dd>...</dd>

- In HTML, gli attributi che consistono di due lettere o numeri, senza spazi o altri simboli, non richiedono le virgolette. In XHTML, invece, tutti gli attributi devono essere compresi fra virgolette, senza eccezioni.

HTML	XHTML
rowspan=2	rowspan="2"
class=property	class="property"
width=45	width="45"
id=fig17	id="fig17"

e così via.

Cascading Style Sheets

- In HTML, alcuni attributi non hanno valore, basta la loro presenza. In XHTML, viene richiesto un valore indentico a quello del loro nome.

HTML	XHTML
checked	checked="checked"
selected	selected="selected"

- I file XHTML devono dichiarare un DOCTYPE appena prima di `<html>`, come segue:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
...
```

Questo indica il primo elemento del formato (`html`), il nome univoco dello stesso incluso il numero di versione (`-//W3C//DTD XHTML 1.1//EN`) e infine in quale locazione sul Web trovare la grammatica formale del formato dichiarato.