

# Texinfo: lo standard della documentazione GNU



Introduzione a Texinfo .....	1365
Esempio introduttivo .....	1366
Logica fondamentale di Texinfo .....	1369
Struttura di un documento Texinfo .....	1373
Indici .....	1381
Aspetto del testo e ambienti speciali .....	1388
Elenchi e tabelle .....	1391
Modifica dello stile TeX .....	1396
Localizzazione .....	1396
Riferimenti .....	1397
Texinfo: libro e ipertesto .....	1399
Sequenza dei nodi secondo Texinfo .....	1402
Definizione automatica della sequenza dei nodi e problemi relativi .....	1404
Limitazioni originali della struttura a nodi .....	1406
Riferimenti ipertestuali e limitazioni verbali .....	1407
Altri tipi di riferimento .....	1411
Riepilogo dei comandi relativi a nodi, ancore e riferimenti	1413
Sgmltexi: installazione e utilizzo .....	1417
Installazione di Sgmltexi .....	1418

Come si usa il programma frontale .....	1420
Riferimenti .....	1425
Sgmltexi: struttura .....	1427
Struttura generale per un sorgente Sgmltexi .....	1428
Scomposizione del documento, nodi e menù Info .....	1451
Codifica .....	1454
Sgmltexi: contenuti .....	1457
Paragrafi .....	1458
Indici e riferimenti incrociati .....	1458
Delimitazione di parole e di frasi .....	1463
Delimitazione di blocchi di testo .....	1465
Elenchi e tabelle .....	1468
Inserzioni .....	1472
Definizioni .....	1473
Codice condizionato e codice letterale in base alla composizione .....	1478
Corrispondenza tra Texinfo e Sgmltexi .....	1485

# Introduzione a Texinfo



Esempio introduttivo .....	1366
Logica fondamentale di Texinfo .....	1369
Scomposizione del documento .....	1370
Inserimento di simboli speciali .....	1371
Struttura di un documento Texinfo .....	1373
Capitoli e sezioni .....	1376
Nodi .....	1378
Menù di riferimenti .....	1380
Indici .....	1381
Indice generale .....	1382
Indici analitici .....	1383
Definizione di indici aggiuntivi .....	1386
Fusione di indici .....	1387
Aspetto del testo e ambienti speciali .....	1388
Elenchi e tabelle .....	1391
Elenco puntato e numerato .....	1391
Elenco descrittivo .....	1393
Tabelle vere e proprie .....	1394
Modifica dello stile TeX .....	1396
Localizzazione .....	1396
Riferimenti .....	1397

Texinfo è un sistema di composizione ideato per la documentazione del progetto GNU, allo scopo di permettere la produzione di documenti ipertestuali in formato Info e di documenti stampati, attraverso il sistema di composizione TeX, a partire da un sorgente unico. Attualmente è disponibile anche la possibilità di comporre in HTML, cosa che completa il sistema Texinfo e lo rende uno strumento essenziale, ma anche molto valido.

A seconda di come è organizzata la propria distribuzione GNU, gli script che compongono il sistema Texinfo potrebbero far parte di un pacchetto indipendente, oppure essere inseriti direttamente all'interno della distribuzione teTeX (LaTeX).

Emacs permette di gestire in modo automatico molte particolarità del sorgente Texinfo, facilitando così il lavoro dell'utilizzatore. In questo capitolo si vuole mostrare solo l'essenziale di Texinfo, pertanto, tutta la parte che riguarderebbe la gestione di Emacs viene ignorata. Questo e altri particolari possono essere approfonditi nella documentazione originale di Texinfo.

## Esempio introduttivo

«

Di solito, il modo migliore per cominciare a comprendere il funzionamento di un sistema di composizione, è quello di partire da un esempio, per avere modo di vedere subito come comporlo in pratica.

```
\input texinfo @c -*-texinfo-*-
@c %***start of header
@setfilename esempio.info
@settitle Introduzione a Texinfo
@c %***end of header

@setchapternewpage odd
```

```

@ifinfo
Questo è un esempio molto breve di un documento scritto
utilizzando il sistema Texinfo.

Copyright @copyright{} 1999 Tizio Tizi
@end ifinfo

@titlepage
@sp 10
@comment Per il titolo viene utilizzato un corpo molto grande.
@center @titlefont{Titolo di esempio}

@c I due comandi seguenti iniziano la pagina del copyright.
@page
@vskip 0pt plus 1filll
Copyright @copyright{} 1999 Tizio Tizi
@end titlepage

@node Top, Suddivisione del documento, , (dir)
@comment nodo-attuale, nodo-successivo, nodo-precedente, nodo-superiore

@menu
* Suddivisione del documento::      Il primo capitolo di questo esempio
                                     molto breve.
* Paragrafi::                       Il secondo capitolo.
* Indice analitico::                L'indice analitico.
@end menu

@node Suddivisione del documento, Paragrafi, Top, Top
@comment nodo-attuale, nodo-successivo, nodo-precedente, nodo-superiore
@chapter Suddivisione del documento con Texinfo
@cindex suddivisione
@cindex capitolo
@cindex sezione
@cindex sottosezione

Un documento scritto in Texinfo è organizzato in capitoli, che possono
essere suddivisi in sezioni, sottosezioni e sotto-sottosezioni:

@enumerate
@item
capitolo -- @code{@@chapter};
@item
sezione -- @code{@@section};

```

```

@item
sottosezione -- @code{@@subsection};
@item
sotto-sottosezione -- @code{@@subsubsection};
@end enumerate

@node   Paragrafi, Indice analitico, Suddivisione del documento, Top
@comment nodo-attuale, nodo-successivo, nodo-precedente, nodo-superiore
@chapter Paragrafi in un sorgente Texinfo
@cindex paragrafo
@cindex testo

Il testo normale di un documento scritto in Texinfo è suddiviso in
paragrafi senza l'indicazione esplicita di alcun comando speciale.
Di conseguenza, basta inserire una riga vuota nel sorgente, per
produrre la separazione tra un paragrafo e il successivo.

@node   Indice analitico, , Paragrafi, Top
@comment nodo-attuale, nodo-successivo, nodo-precedente, nodo-superiore
@unnumbered Indice analitico

@printindex cp

@contents
@bye

```

Si suppone di avere nominato il file di questo sorgente ‘`esempio.texinfo`’. Di seguito vengono mostrati i comandi necessari alla composizione per generare un file Info, un risultato in HTML (in due modi differenti), un file PostScript e un file PDF.

```
$ makeinfo esempio.texinfo [Invio]
```

```
$ makeinfo --html esempio.texinfo [Invio]
```

```
$ texi2html esempio.texinfo [Invio]
```

```
$ texi2dvi esempio.texinfo ; dvips -t a4 -o esempio.ps  
esempio.dvi [Invio]
```

```
$ texi2dvi --pdf esempio.texinfo [Invio]
```

Nel primo caso viene generato il file Info ‘esempio.info’; nel secondo e nel terzo si ottiene il file ‘esempio.html’ (affiancato eventualmente da un file contenente l’indice generale); nel quarto caso si ottiene il file ‘esempio.ps’; nell’ultimo si ottiene il file ‘esempio.pdf’.

## Logica fondamentale di Texinfo

Texinfo è TeX a cui è stato applicato uno stile speciale, per cui il simbolo ‘@’ sostituisce la barra obliqua inversa (‘\’). Questo si ottiene attraverso uno stile contenuto nel file ‘texinfo.tex’, che viene incluso opportunamente con il comando TeX iniziale: «

```
\input texinfo
```

Da quel punto in poi, la barra obliqua inversa ha valore letterale. Sempre allo scopo di ridurre al minimo i simboli che hanno significati speciali, i commenti si indicano attraverso un comando apposito: ‘@c’, oppure ‘@comment’. A questo proposito, si può osservare che la prima riga mostrata nell’esempio introduttivo, contiene proprio un commento, subito dopo la dichiarazione dell’inclusione dello stile per Texinfo:

```
\input texinfo @c --texinfo--
```

Si tratta di una stringa convenzionale, che è bene utilizzare anche se non è strettamente necessaria alla composizione di un sorgente Texinfo, perché riguarda Emacs, permettendogli di identificare il file e di qualificarlo per quello che è.

Una volta chiarita la natura TeX di un sorgente Texinfo, si può comprendere il comportamento generale del sistema, nel momento in cui

la composizione viene fatta per arrivare alla stampa. In particolare, si può intendere il modo in cui vengono considerati gli spazi, che vengono eliminati quando sembrano superflui, così come si può intendere il motivo per cui basta separare i blocchi di testo con una o più righe vuote (o bianche), per ottenere la separazione in paragrafi. Tuttavia, le cose cambiano quando la composizione avviene in modo da generare un file Info: in questo caso gli spazi aggiuntivi contano e anche le righe vuote superflue possono essere prese in considerazione.

Nonostante la sua natura TeX, Texinfo è orientato alla generazione di un ipertesto consultabile attraverso un terminale a caratteri; pertanto, è su questo punto che si fondano le sue caratteristiche e le sue limitazioni.

## Scomposizione del documento

«

Un documento Texinfo è articolato in due modi distinti, che devono avvenire simultaneamente. Da una parte si trova l'articolazione del testo nel modo più adatto a un libro, con i suoi capitoli e le sezioni a livelli diversi (come fa LaTeX), dall'altra parte c'è un ipertesto organizzato a grafo (un reticolo di collegamenti uniti assieme da dei nodi), dove i nodi sono i vari blocchi di informazioni.

Texinfo non pone limitazioni particolari all'uso dei nodi, tuttavia il buon senso richiede che siano usati in modo compatibile con la struttura «cartacea» del documento. In generale, ogni capitolo deve avere un nodo corrispondente, mentre le sezioni potrebbero averlo se ciò è opportuno, e lo stesso vale per le sottosezioni. Nell'esempio introduttivo, prima della dichiarazione del primo capitolo, si vede l'indicazione del nodo relativo:

```
@node      Suddivisione del documento, Paragrafi, Top, Top
@comment  nodo-attuale, nodo-successivo, nodo-precedente, nodo-superiore
@chapter  Suddivisione del documento con Texinfo
```

Dal momento che la composizione in formati finali diversi genera risultati differenti, c'è poi l'esigenza di poter distinguere il testo che deve essere usato per una o l'altra composizione. Per questo si possono circoscrivere delle porzioni di testo tra i comandi '@ifinfo' '@end ifinfo', '@iftex' '@end iftex', e '@ifhtml' '@end ifhtml'. Nell'esempio introduttivo si vede proprio l'uso di questi comandi per inserire del testo che viene utilizzato solo nella composizione in formato Info:

```
@ifinfo
Questo è un esempio molto breve di un documento scritto
utilizzando il sistema Texinfo.

Copyright @copyright{} 1999 Tizio Tizi
@end ifinfo
```

## Inserimento di simboli speciali

Il linguaggio di composizione utilizzato da Texinfo, attribuisce un significato speciale al simbolo '@' e alle parentesi graffe. Per indicare questi caratteri in modo letterale, basta farli precedere da un altro '@'. In questo senso, la sequenza '@*carattere*' rappresenta spesso la richiesta esplicita di fare riferimento al carattere in modo letterale. La tabella u87.6 elenca alcune di queste sequenze di escape.

Tabella u87.6. Comandi per rappresentare alcuni simboli speciali.

Comando	Descrizione
@@	Un solo simbolo '@'.

Comando	Descrizione
@{	Una parentesi graffa aperta.
@}	Una parentesi graffa chiusa.
@<SP>	Uno spazio non interrompibile.

In modo simile si possono definire delle lettere speciali, se queste non sono disponibili attraverso la tastiera. La tabella u87.7 mostra i comandi utili per rappresentare le vocali accentate italiane. Tuttavia, è opportuno osservare che non è sempre conveniente l'uso di questi comandi, se si ritiene di poter usare una codifica migliore dell'ASCII tradizionale. Infatti, se si usa un comando come '@`a' si rischia poi di vedere 'a`' nella composizione Info, cosa che non succede se si utilizza la codifica ISO 8859-1.

Tabella u87.7. Comandi per la rappresentazione delle vocali accentate nella lingua italiana.

Comando	Risultato	Comando	Risultato
@`a	à	@`A	À
@`e	è	@`E	È
@'e	é	@'E	É
@`i	ì	@`I	Ì
@`o	ò	@`O	Ò
@`u	ù	@`U	Ù

# Struttura di un documento Texinfo



In un sorgente Texinfo, prima di arrivare alla scomposizione del testo in capitoli e nodi, c'è una parte iniziale che merita un po' di attenzione. La prima dichiarazione in assoluto è quella dell'inserimento dello stile 'texinfo.tex', come è già stato mostrato, quindi si incontrano le dichiarazioni '@setfilename' e '@settitle' che costituiscono l'intestazione del sorgente:

```
\input texinfo @c --texinfo--
@c ***start of header
@setfilename esempio.info
@settitle Introduzione a Texinfo
@c ***end of header
```

La prima di queste due dichiarazioni serve a definire il nome del file Info finale, che in caso di necessità potrebbe anche essere scomposto in più file, dove quello indicato rappresenta così solo il file di partenza; la seconda dichiara il titolo del documento in breve. Texinfo richiede che le dichiarazioni dell'intestazione siano racchiuse tra due commenti ben definiti, come si vede dall'esempio. È importante che siano riprodotti nello stesso modo che è stato mostrato:

```
@c ***start of header
...
@c ***end of header
```

La parte successiva all'intestazione, viene usata per utilizzare dei comandi specifici per la composizione TeX, come nel caso di '@setchapternewpage' che permette di definire se i capitoli debbano iniziare in una pagina nuova e se questa debba essere una pagina dispari, oppure se ciò sia indifferente:

Comando	Descrizione
@setchapternewpage on	fa in modo che ogni capitolo inizi a pagina nuova;

Comando	Descrizione
<code>@setchapternewpage off</code>	fa in modo che i capitoli possano iniziare nella stessa pagina in cui finiscono quelli precedenti;
<code>@setchapternewpage odd</code>	fa in modo che ogni capitolo inizi in una nuova pagina dispari.

Dopo questo genere di definizioni, si passa normalmente alla presentazione formale del documento, annotando le informazioni legali e, nel caso particolare della composizione in TeX, specificando anche l'aspetto della prima pagina, quella del titolo. Nel caso di un documento Info non ha molta importanza la preparazione di una facciata introduttiva; in effetti, questa non esiste (come è possibile vedere in seguito a proposito del nodo iniziale). In questo senso, non c'è nemmeno il posto per le informazioni sul copyright, che vengono comunque inserite, delimitandole tra i comandi '`@ifinfo`' e '`@end ifinfo`', allo scopo che queste siano effettivamente annotate all'inizio nel file Info, anche se in una zona che poi non viene consultata attraverso la navigazione ipertestuale.

```
@ifinfo
Questo è un esempio molto breve di un documento scritto
utilizzando il sistema Texinfo.

Copyright @copyright{} 1999 Tizio Tizi
@end ifinfo
```

Alla fine di questa parte introduttiva del sorgente Texinfo, appare generalmente un blocco delimitato dai comandi '`@titlepage`' e '`@end titlepage`', che riguardano esclusivamente la composizione con TeX, con lo scopo di definire le pagine iniziali dal titolo alle informazioni sul copyright.

```

@titlepage
@sp 10
@comment Per il titolo viene utilizzato un corpo molto grande.
@center @titlefont{Titolo di esempio}

@c I due comandi seguenti iniziano la pagina del copyright.
@page
@vskip 0pt plus 1filll
Copyright @copyright{} 1999 Tizio Tizi
@end titlepage

```

I comandi che si vedono nell'esempio dovrebbero essere abbastanza intuitivi, dal momento che si tratta praticamente di TeX:

Comando	Descrizione
<code>@sp <i>n</i></code>	richiede uno spazio verticale di <i>n</i> righe;
<code>@center @titlefont{ <i>titolo</i></code>	centra il titolo che viene stampato utilizzando un carattere adatto, pensato appositamente per questo ( <code>@titlefont</code> );
<code>@page</code>	esegue un salto pagina;
<code>@vskip 0pt plus 1filll</code>	porta il testo successivo alla base della pagina (si tratta di un comando TeX, dove la parola <code>'1filll'</code> è scritta correttamente con tre «l»).

Il manuale di Texinfo propone anche un'altra forma, in cui si utilizzano i comandi `'@title'`, `'@subtitle'` e `'@author'`. Il loro significato è intuitivo e l'esempio seguente dovrebbe chiarirne l'uso: si osservi in particolare la presenza di due sottotitoli e di due autori.

```
@titlepage
@title Titolo di esempio
@subtitle Primo sottotitolo
@subtitle Secondo sottotitolo
@author Tizio Tizi
@author Caio Cai
@page
@vskip 0pt plus 1filll
Copyright @copyright{} 1999 Tizio Tizi, Caio Cai
@end titlepage
```

Nell'esempio introduttivo che è stato mostrato, non appare circoscritto alcun pezzo riservato alla composizione in HTML. In effetti, **'texi2html'** ignora l'inizio del sorgente Texinfo, a parte l'intestazione, dalla quale ottiene il titolo del documento e il nome del file HTML principale che deve generare.

Al termine di un documento Texinfo, deve essere usato il comando **'@bye'** per concludere esplicitamente la composizione.

## Capitoli e sezioni

«

Prima di affrontare il problema che riguarda la scomposizione del documento in nodi, vale la pena di vedere come avviene la scomposizione in capitoli e sezioni, dal momento che è qualcosa di più semplice, trattandosi di un concetto comune a molti altri sistemi di composizione. Semplificando le cose, si può affermare che un documento Texinfo è suddiviso in capitoli, che possono essere suddivisi a loro volta in sezioni, sottosezioni e sotto-sottosezioni. Tuttavia, esistono diversi tipi di «capitoli» e di «sezioni»; la tabella u87.15 riepiloga questi comandi.

Tabella u87.15. Comandi per la suddivisione del testo in base al risultato stampato.

Comando	Descrizione
<code>@chapter</code>	Capitolo normale con numerazione.
<code>@section</code>	Sezione normale con numerazione.
<code>@subsection</code>	Sottosezione normale con numerazione.
<code>@subsubsection</code>	Sotto-sottosezione normale con numerazione.
<code>@unnumbered</code>	Capitolo senza numerazione.
<code>@unnumberedsec</code>	Sezione senza numerazione.
<code>@unnumberedsubsec</code>	Sottosezione senza numerazione.
<code>@unnumberedsubsubsec</code>	Sotto-sottosezione senza numerazione.
<code>@appendix</code>	Capitolo numerato in modo letterale -- appendice.
<code>@appendixsec</code>	Sezione di un'appendice.
<code>@appendixsubsec</code>	Sottosezione di un'appendice.
<code>@appendixsubsubsec</code>	Sotto-sottosezione di un'appendice.
<code>@majorheading</code>	Titolo importante senza numerazione e senza salto pagina.
<code>@chapterheading</code>	Capitolo senza numerazione e senza salto pagina.
<code>@heading</code>	Sezione senza numerazione.

Comando	Descrizione
<code>@subheading</code>	Sottosezione senza numerazione.
<code>@subsubheading</code>	Sotto-sottosezione senza numerazione.

In particolare, la suddivisione che fa capo al capitolo di tipo ‘**@unnumbered**’, riguarda generalmente gli indici, o le introduzioni, mentre la suddivisione ‘**@...heading**’, permette di realizzare dei documenti in forma di relazione.

L’esempio introduttivo mostra in che modo si definisce l’inizio di un capitolo, utilizzando il comando ‘**@chapter**’ seguito dal titolo, senza bisogno che questo sia delimitato in qualche modo. La stessa cosa varrebbe per le sezioni e per le altre classificazioni inferiori.

<code>@chapter</code> Suddivisione del documento con Texinfo
--

## Nodi



I nodi di Texinfo sono le unità di informazioni raggiungibili attraverso una navigazione ipertestuale. A differenza della struttura di capitoli e sezioni, i nodi sono unità non divisibili, quindi non esistono dei sotto-nodi. Questo fatto crea una sovrapposizione imperfetta tra la struttura a nodi e la struttura di capitoli e sezioni.

Il nodo viene dichiarato attraverso il comando ‘**@node**’ e la sua estensione va da quel punto fino alla dichiarazione del nodo successivo. Convenzionalmente, si dichiarano i nodi subito prima di un capitolo, o di una sezione (o anche di una sottosezione, ecc.); in questo modo, la struttura a nodi ha una qualche corrispondenza con la struttura cartacea del documento. A questo proposito, è impor-

tante stabilire l'estensione dei nodi: per cominciare potrebbe essere conveniente avere nodi contenenti un capitolo intero; in questo caso si dichiarerebbero solo in corrispondenza di questi. Nel capitolo [u88](#) viene trattato meglio il problema dell'abbinamento dei nodi con la struttura del documento, in particolare per gli automatismi che vengono offerti da Texinfo.

La struttura ipertestuale prevede un nodo di partenza obbligatorio, denominato '**Top**', all'interno del quale si trova normalmente un elenco di riferimenti, paragonabile a un indice generale, e una serie di altri nodi definiti dall'autore, con nomi liberi.

```
@node Top, Suddivisione del documento, , (dir)
```

L'ipertesto Info, prevede l'aggregazione dei vari documenti scritti per questo sistema, attraverso un nodo precedente a quello '**Top**': si tratta del nodo '**(dir)**', corrispondente al file contenente l'indice iniziale di tutti i documenti Info installati effettivamente nel proprio sistema.

La dichiarazione di un nodo implica l'indicazione del suo nome, seguito da tre riferimenti ad altrettanti nodi: il nodo successivo, secondo un ordine ideale stabilito dall'autore; il nodo precedente; il nodo superiore.

```
@node nome_del_nodo, nodo_successivo, nodo_precedente, nodo_superiore
```

In pratica, questa struttura prevede una sequenza di nodi, stabilita in qualche modo, assieme al riferimento di un nodo di livello superiore. Il primo nodo in assoluto è '**(dir)**', mentre il primo nodo del documento è '**Top**'. Volendo utilizzare una struttura di nodi corrispondenti ai capitoli, senza suddivisioni ulteriori, si avrebbe una

sola sequenza di nodi dal primo all'ultimo capitolo, dove per tutti l'unico nodo superiore sarebbe **'Top'**. Se invece si volesse realizzare una suddivisione maggiore, è ragionevole che i nodi contenuti in un capitolo formino una sequenza, dove il nodo superiore potrebbe essere quello iniziale del capitolo stesso.

Osservando la dichiarazione del nodo **'Top'** dell'esempio, si può vedere che il nodo precedente non è stato indicato, mentre il nodo superiore è **'(dir)'**. Infatti, non esiste un nodo precedente a **'Top'**, mentre al di sopra di quello c'è solo l'indice generale, corrispondente al nome convenzionale **'(dir)'**.

## Menù di riferimenti



La composizione in formato Info può generare automaticamente l'indice analitico, ma non l'indice generale. Per ottenere una sorta di indice generale, occorre indicare manualmente i riferimenti da qualche parte, di solito nel nodo **'Top'**. Nel caso dell'esempio introduttivo, sono stati indicati i riferimenti ai capitoli e all'indice analitico:

```
@menu
* Suddivisione del documento::      Il primo capitolo di questo esempio
                                     molto breve.
* Paragrafi::                       Il secondo capitolo.
* Indice analitico::                L'indice analitico.
@end menu
```

Come si vede, tra i comandi **'@menu'** e **'@end menu'**, sono stati indicati i nomi dei nodi da raggiungere, seguiti da una descrizione. Le voci di questi menù possono essere più articolate, ma in generale, se possibile, conviene mantenere questa forma elementare:

\* *nome\_nodo\_da\_raggiungere* : : *descrizione*

Chi scrive un documento in Texinfo può anche fare a meno di preoccuparsi di questi menù, ma dovrebbe inserire almeno le voci che fanno riferimento ai nodi dell'indice analitico. Chi utilizza Emacs può anche ottenere la preparazione di questo menù automaticamente; per apprenderne il modo può consultare la documentazione originale su Texinfo.

## Indici

Gli indici e i riferimenti di qualunque tipo siano, si ottengono attraverso l'indicazione di un'etichetta (da una parte) e di un riferimento che punta all'etichetta. Gli indici in particolare, sono una raccolta di riferimenti realizzata in modo automatico.

Tabella u87.19. Comandi per la gestione di indici generali o analitici.

Comando	Descrizione
@contents	Inserisce l'indice generale completo.
@shortcontents	Inserisce un indice generale ridotto.
@summarycontents	Inserisce un indice generale ridotto.
@cindex <i>voce</i>	Inserisce una voce nell'indice analitico normale.
@kindex <i>voce</i>	Inserisce una voce nell'indice dei comandi da tastiera.
@pindex <i>voce</i>	Inserisce una voce nell'indice dei programmi.

Comando	Descrizione
<code>@findex voce</code>	Inserisce una voce nell'indice delle funzioni.
<code>@vindex voce</code>	Inserisce una voce nell'indice delle variabili.
<code>@tindex voce</code>	Inserisce una voce nell'indice dei tipi di dati.
<code>@defindex xy</code>	Crea l'indice <i>xy</i> .
<code>@xyindex voce</code>	Inserisce una voce nell'indice <i>xy</i> .
<code>@synindex da a</code>	Trasferisce le voci di un indice in un altro.
<code>@syncodeindex da a</code>	Come '@synindex' usando un carattere dattilografico.
<code>@printindex xy</code>	Inserisce l'indice corrispondente alla sigla <i>xy</i> .

## Indice generale

«

L'indice generale viene realizzato solo nella composizione che si avvale di TeX, oltre che in quella per il formato HTML. Ciò avviene in modo automatico, indicando semplicemente il punto in cui questo deve apparire: l'inizio dei capitoli e delle classificazioni inferiori viene annotato nell'indice generale. Come accennato in precedenza, questo meccanismo non riguarda la composizione nel formato Info, per cui si utilizza un menù di riferimenti che fa le funzioni di indice generale.

L'indice generale può essere collocato all'inizio o alla fine del documento, in tal caso dopo gli indici analitici eventuali. Con il comando '**@contents**' si richiede la sua realizzazione in corrispondenza del

comando stesso. È importante osservare che questo comando crea anche il titolo necessario, al contrario di ciò che avviene con l'indice analitico, come viene descritto tra poco.

Vale la pena di annotare il fatto che sono disponibili altri due comandi per ottenere la realizzazione di indici analitici meno dettagliati. Si tratta di '@shortcontents' e '@summarycontents'. La differenza tra i due sta solo nel titolo utilizzato per introdurli.

## Indici analitici

Per quanto riguarda l'indice analitico, per ottenerlo è necessario indicare nel testo delle etichette apposite, con le quali si ottiene l'inserimento delle voci relative. Questo viene fatto più o meno come avviene in altri sistemi di composizione, ma con Texinfo occorre tenere conto di alcune particolarità che derivano dalla sua specializzazione ipertestuale. Prima di tutto, si deve considerare che secondo la politica di Texinfo, le etichette riferite a voci da inserire nell'indice analitico devono essere uniche. In questo modo si semplifica una serie di problemi nella navigazione di un documento Info, che non può essere ambigua. A questo proposito, la documentazione originale di Texinfo suggerisce di utilizzare voci descrittive piuttosto dettagliate. Nel momento in cui si devono usare voci descrittive, si può porre anche il problema del modo in cui il lettore va a cercarle nell'indice, per cui, se ci sono più modi per indicare lo stesso concetto, è meglio inserire più etichette alternative per l'indice analitico. Texinfo è in grado di gestire diversi indici analitici specifici:

- un indice normale per i concetti che vengono affrontati;

- un indice dei comandi da tastiera (combinazioni di tasti con funzionalità particolari nell'ambito di ciò che viene descritto);
- un indice dei programmi (nomi dei programmi eseguibili);
- un indice delle funzioni;
- un indice delle variabili;
- un indice dei tipi di dati.

I primi due tipi di indici dovrebbero avere un significato evidente; per gli altri, si fa riferimento a ciò che riguarda i linguaggi di programmazione.

Si intuisce che non è tecnicamente necessario utilizzare tutti questi indici. In generale, ci si potrebbe limitare all'inserimento delle voci nell'indice analitico normale. Tuttavia, se si vuole realizzare un documento in Texinfo, seguendo le convenzioni, è bene fare uso dell'indice giusto per ogni cosa. Questo permette in seguito l'aggregazione con altri documenti che hanno seguito le stesse convenzioni.

La sintassi per la dichiarazione di una voce nei vari indici analitici di Texinfo, può essere fatta secondo uno degli schemi seguenti, che rappresentano nell'ordine i tipi di indice descritti sopra:

```
@cindex voce_da_inserire_nell'indice_analitico_normale
```

```
@kindex voce_da_inserire_nell'indice_analitico_dei_comandi_da_tastiera
```

`@pindex` *voce\_da\_inserire\_nell'indice\_analitico\_dei\_programmi*

`@findex` *voce\_da\_inserire\_nell'indice\_analitico\_delle\_funzioni*

`@vindex` *voce\_da\_inserire\_nell'indice\_analitico\_delle\_variabili*

`@tindex` *voce\_da\_inserire\_nell'indice\_analitico\_dei\_tipi\_di\_dati*

A ogni tipo di indice è abbinata una sigla, il cui utilizzo viene descritto tra poco. La tabella u87.20 elenca queste sigle.

Tabella u87.20. Sigle dei vari tipi di indice analitico di Texinfo.

Sigla	Indice
cp	Indice analitico normale.
ky	Indice analitico dei comandi da tastiera.
pg	Indice analitico dei programmi.
fn	Indice analitico delle funzioni.
vr	Indice analitico delle variabili.
tp	Indice analitico dei tipi di dati.

Nell'esempio introduttivo è stato mostrato l'uso del comando '`@cindex`' per inserire alcune voci nell'indice analitico normale:

```
@chapter Suddivisione del documento con Texinfo
@cindex suddivisione
@cindex capitolo
@cindex sezione
@cindex sottosezione
```

La voce che si inserisce nell'indice analitico, può essere anche composta da più parole, separate normalmente da uno spazio, senza bisogno di utilizzare dei delimitatori di alcun tipo: l'interpretazione corretta del comando è garantita dal fatto che questo deve stare da solo in una riga.

L'inserimento nel documento finale di un indice analitico, viene richiesto in modo esplicito, attraverso il comando '@**printindex**', seguito dalla sigla corrispondente all'indice desiderato. In generale, questo viene fatto all'interno di un capitolo non numerato, come è stato mostrato nell'esempio introduttivo, dove si vede la richiesta di creazione di un indice generale normale.

```
@unnumbered Indice analitico

@printindex cp
```

## Definizione di indici aggiuntivi

«

Dovrebbe essere ormai chiaro che ogni indice va usato per il suo scopo, altrimenti diventa impossibile la fusione di più documenti in un libro unico. Tuttavia, di fronte a questa filosofia di Texinfo che distingue tra gli indici, ci si può trovare di fronte all'esigenza di crearne degli altri. Questo si può fare semplicemente, attribuendo a questi indici particolari una sigla di due sole lettere.

Per fare un esempio, si potrebbe desiderare l'introduzione di un indice specifico per raccogliere gli elementi SGML di un DTD. Volendo chiamare questo indice con la sigla '**m1**', si dichiara l'utilizzo di

un tale indice nell'intestazione del sorgente Texinfo con il comando **'defindex'**:

```
\input texinfo @c --texinfo--
@c %**start of header
@setfilename esempio.info
@settitle Introduzione a Texinfo
@defindex ml
@c %**end of header
```

La sintassi di questo comando non prevede altro, per cui non viene mostrata formalmente. Per inserire una voce in un indice definito in questo modo, si usa il comando **'xyindex'**, dove **xy** sono le due lettere che lo definiscono.

```
@chapter Gli elementi interni al testo
@mlindex em
@mlindex strong
```

L'esempio mostra l'inserimento delle voci **'em'** e **'strong'**.

Per ottenere l'inserimento dell'indice si procede come avviene già per gli indici già previsti:

```
@printindex ml
```

## Fusione di indici

La suddivisione dettagliata delle voci da inserire nell'indice analitico è una cosa ragionevole solo in quanto resta possibile la fusione di più gruppi assieme, in un indice unico. Ciò si ottiene con i comandi **'@synindex'** e **'@syncodeindex'** che hanno la stessa sintassi:

```
@synindex sigla_indice_di_origine sigla_indice_di_destinazione
```

```
@syncodeindex sigla_indice_di_origine sigla_indice_di_destinazione
```

Questi comandi si possono inserire solo all'inizio del documento, preferibilmente nell'intestazione, come si vede nell'esempio seguente:

```
@c %**start of header
@setfilename sgmltexi.info
@settitle Sgmltexi
...
@syncodeindex ml cp
@syncodeindex vr cp
@c %**end of header
```

In questo caso, si fa in modo di riversare le voci dell'indice '**ml**' e '**vr**' nell'indice standard ('**cp**').

I due comandi si distinguono perché nel primo caso le voci vengono trasferite in modo normale, mentre nel secondo queste vengono trasformate per renderle nella destinazione in modo dattilografico (praticamente avvolte nel comando '@code{ }').

## Aspetto del testo e ambienti speciali



Texinfo prevede una serie di comandi per delimitare parti di testo riferite a oggetti particolari; nello stesso modo, ci sono altri comandi per definire delle forme di enfaticizzazione, senza stabilire le caratteristiche di ciò che si indica. Le tabelle u87.27 e u87.28 elencano questi comandi, quando sono riferiti a parole e frasi inserite nel corpo normale.

Tabella u87.27. Comandi per delimitare oggetti particolari nel testo.

Comando	Descrizione
@code{ <i>testo</i> }	Esempio letterale di un pezzo di programma.
@kbd{ <i>testo</i> }	Digitazione dalla tastiera.

Comando	Descrizione
<code>@key{<i>testo</i>}</code>	Nome convenzionale di un tasto.
<code>@samp{<i>testo</i>}</code>	Esempio letterale di una sequenza di caratteri.
<code>@var{<i>testo</i>}</code>	Variabile metasintattica.
<code>@url{<i>testo</i>}</code>	Indirizzo URI.
<code>@file{<i>testo</i>}</code>	Nome di un file.
<code>@email{<i>testo</i>}</code>	Indirizzo di posta elettronica.
<code>@dfn{<i>testo</i>}</code>	Una definizione introdotta per la prima volta.
<code>@cite{<i>testo</i>}</code>	Riferimento bibliografico (titolo di un libro).

Tabella u87.28. Comandi per l'enfatizzazione generica.

Comando	Descrizione
<code>@emph{<i>testo</i>}</code>	Enfatizzazione normale.
<code>@strong{<i>testo</i>}</code>	Enfatizzazione più evidente.
<code>@sc{<i>testo</i>}</code>	Maiuscoletto.
<code>@r{<i>testo</i>}</code>	Carattere tondo normale.

La tabella u87.30 mostra l'elenco dei comandi riferiti ad ambienti particolari, usati per mostrare esempi, per le citazioni, oltre che per altre forme di evidenziamento del testo. I comandi in questione si

usano da soli su una riga; hanno un'apertura e una chiusura, secondo la forma '@comando' e '@end comando'. Per esempio, nel caso della citazione da mettere in evidenza, si può fare come nell'esempio seguente:

```
@quotation
Questa è una citazione.
@end quotation
```

Tabella u87.30. Comandi per delimitare blocchi di testo con funzioni specifiche.

Comando	Descrizione
@quotation	Testo citato.
@example	Codice, comandi e simili.
@smallexample	Come '@example', ma più piccolo.
@lisp	Codice LISP.
@smalllisp	Come '@lisp', ma più piccolo.
@display	Testo illustrativo senza uno scopo specifico.
@smalldisplay	Come '@display', ma più piccolo.
@format	Testo illustrativo, rispettando le interruzioni di riga.
@smallformat	Come '@format', ma più piccolo.
@cartouche	Disegna un riquadro arrotondato attorno al testo.

Anche se ciò non riguarda precisamente l'argomento di questa sezio-

ne, vale la pena di mostrare brevemente come si dichiara una nota a piè pagina:

```
@footnote{testo}
```

## Elenchi e tabelle

Gli elenchi vengono realizzati in Texinfo, più o meno come avviene con altri linguaggi di composizione. Dal momento che si deve poter arrivare al formato Info, le tabelle che vengono gestite sono molto simili a degli elenchi, pertanto è corretto trattare i due argomenti assieme.

### Elenco puntato e numerato

L'elenco puntato viene delimitato dai comandi '@**itemize**' e '@**end itemize**'. Gli elementi dell'elenco vengono introdotti dal comando '@**item**'. Il simbolo usato per segnalare l'inizio dei vari elementi dell'elenco, viene dichiarato esplicitamente attraverso un argomento del comando '@**itemize**', '@**bullet**' o '@**minus**', che si riferiscono rispettivamente a un pallino (o un asterisco) e a un trattino. L'esempio seguente mostra un elenco con due voci principali, dove la prima si scompone in altre due voci inferiori; le voci principali sono introdotte da un pallino, mentre quelle inferiori sono evidenziate da un trattino.

```
@itemize @bullet
@item
primo elemento principale;

@itemize @minus
@item
primo sottoelemento;
```

```
@item
secondo sottoelemento;
@end itemize

@item
secondo elemento principale.
@end itemize
```

Gli spazi tra le voci sono opportuni, per ottenere un buon risultato nella composizione in formato Info, mentre per la composizione attraverso TeX, la cosa è indifferente.

L'elenco numerato è simile a quello puntato e si distingue solo perché è racchiuso tra i comandi '@**enumerate**' e '@**end enumerate**'. In particolare, in questo caso, al posto di definire il tipo di pallino da utilizzare, è possibile specificare il primo valore da utilizzare nell'elenco: se si tratta di un numero, quello diviene il primo valore di un elenco numerato vero e proprio; se si tratta di una lettera, si ottiene di un elenco numerato in modo letterale, a partire da quella lettera. L'esempio seguente mostra un elenco numerato che parte dal numero zero (quando di solito partirebbe da uno).

```
@enumerate 0
@item
primo elemento;

@item
secondo elemento;

@item
terzo elemento.
@end enumerate
```

## Elenco descrittivo



L'elenco descrittivo è quello che per ogni punto mostra una parola o una frase a cui associa una descrizione. Per Texinfo, gli elenchi descrittivi sono delle tabelle a due colonne. L'ambiente viene delimitato dai comandi '@table' e '@end table', dove in particolare, il primo riceve un argomento che specifica il tipo di composizione da utilizzare per la prima colonna di questa specie di tabella. Per esempio,

```
@table @file
@item /etc/passwd
il file degli utenti;

@item /etc/group
il file dei gruppi.
@end @table
```

fa in modo che «/etc/passwd» e «/etc/group» vengano delimitati automaticamente con il comando '@file', mentre il resto, cioè la loro descrizione, viene lasciata con il carattere normale del testo.

Si può osservare che in questo caso il comando '@item' ha un argomento, che rappresenta la voce descrittiva dell'elenco. In situazioni particolari, può essere necessario indicare due voci assieme per la stessa descrizione; per questo esiste il comando '@itemx' che può essere usato subito dopo un comando '@item' normale.

```
@table @file
@item /etc/passwd
@itemx /etc/group
i file degli utenti e dei gruppi;

@item /etc/printcap
il file di configurazione del sistema di stampa.
@end @table
```

Esistono due varianti al comando '@table': si tratta di '@ftable'

e `@vtable`. Il loro funzionamento è identico a `@table`, con l'unica aggiunta che le voci indicate come argomento dei comandi `@item` o `itemx` vengono inserite automaticamente nell'indice delle funzioni e delle variabili, rispettivamente.

```
@vtable @code
@item PATH
contiene l'elenco dei percorsi per i file eseguibili;

@item HOME
contiene l'indicazione della directory personale abbinata
all'utente attuale.
@end @vtable
```

## Tabelle vere e proprie

«

Le tabelle vere e proprie di Texinfo sono delimitate attraverso i comandi `@multitable` e `@end multitable`. Il comando di apertura richiede l'indicazione di altre informazioni che permettono di determinare l'ampiezza delle varie colonne. A questo proposito può essere usato il comando `@columnfractions`, oppure degli esempi di testo:

```
@multitable @columnfractions frazione...
...
@end multitable
```

```
@multitable {testo_di_esempio}...
...
@end multitable
```

Il testo di esempio va racchiuso tra parentesi graffe, che quindi fanno parte del comando. Le frazioni sono valori decimali, la cui somma

complessiva dovrebbe dare l'unità o un valore inferiore.

```
@multitable @columnfractions .2 .3 .5
<synellipsis>
@end multitable
```

L'esempio mostra il caso si una tabella che prevede tre colonne, dove la prima occupa un'ampiezza pari al 20 % del totale, la seconda il 30 % e l'ultima il restante 50 %. Se non si desiderano indicare questi valori percentuali si può usare l'altro metodo, come nell'esempio seguente:

```
@multitable {bla bla} {bla bla bla} {bla bla bla bla bla}
...
@end multitable
```

Le righe di queste tabelle sono introdotte dal comando '@item', a cui segue il testo della prima colonna. Il testo delle colonne successive viene introdotto da uno o più comandi '@tab'. L'esempio seguente mostra una tabella con tre colonne:

```
@multitable @columnfractions .25 .25 .5
@item colore
@tab valore
@tab annotazioni
@item nero
@tab 0
@tab il colore iniziale
@item marrone
@tab 1
@tab
@item rosso
@tab 2
@tab
...
@item bianco
@tab 9
@tab il colore finale
@end multitable
```

## Modifica dello stile TeX



Lo stile standard predisposto per la composizione attraverso TeX potrebbe richiedere delle modifiche per qualche ragione. In generale, non è il caso di modificare il file che rappresenta lo stile standard, dal momento che è sufficiente farsene una copia da tenere assieme al sorgente Texinfo: quando si procede alla composizione, il file di stile ‘`texinfo.tex`’ che si trova nella directory corrente, ha la precedenza.

Nell’estratto seguente vengono mostrate le righe utili che possono essere modificate per ottenere una traduzione dei termini che vengono inseriti automaticamente:

```
% Definizioni in italiano.
\ifx\putwordAppendix\undefined \gdef\putwordAppendix{Appendice}\fi
\ifx\putwordChapter\undefined \gdef\putwordChapter{Capitolo}\fi
\ifx\putwordfile\undefined \gdef\putwordfile{file}\fi
\ifx\putwordInfo\undefined \gdef\putwordInfo{Info}\fi
\ifx\putwordMethodon\undefined \gdef\putwordMethodon{Method on}\fi
\ifx\putwordon\undefined \gdef\putwordon{on}\fi
\ifx\putwordpage\undefined \gdef\putwordpage{pagina}\fi
\ifx\putwordsection\undefined \gdef\putwordsection{sezione}\fi
\ifx\putwordSection\undefined \gdef\putwordSection{Sezione}\fi
\ifx\putwordsee\undefined \gdef\putwordsee{vedere}\fi
\ifx\putwordSee\undefined \gdef\putwordSee{Vedere}\fi
\ifx\putwordShortContents\undefined \gdef\putwordShortContents{Indice breve}\fi
\ifx\putwordTableofContents\undefined \gdef\putwordTableofContents{Indice generale}\fi
```

## Localizzazione



Il lavoro di nazionalizzazione del sistema Texinfo è ancora in corso. Recentemente è stato introdotto il comando ‘`@documentlanguage`’ con il quale si ottiene la conversione automatica dei termini che vengono inseriti automaticamente in fase di composizione. Il comando riceve un argomento corrispondente alla sigla della lingua a cui si

vuole fare riferimento, espressa secondo lo standard ISO 639 (sezione 13.3). Il comando si colloca preferibilmente nell'intestazione del sorgente. L'esempio seguente mostra la selezione della lingua italiana.

```
@c %**start of header
@setfilename prova.info
@settitle Prova
...
@documentlanguage it
@c %**end of header
```

## Riferimenti

- Robert J. Chassel, Richard Stallman, *GNU Texinfo*, Free Software Foundation, Inc.

<http://www.gnu.org/software/texinfo/manual/texinfo/>



# Texinfo: libro e ipertesto

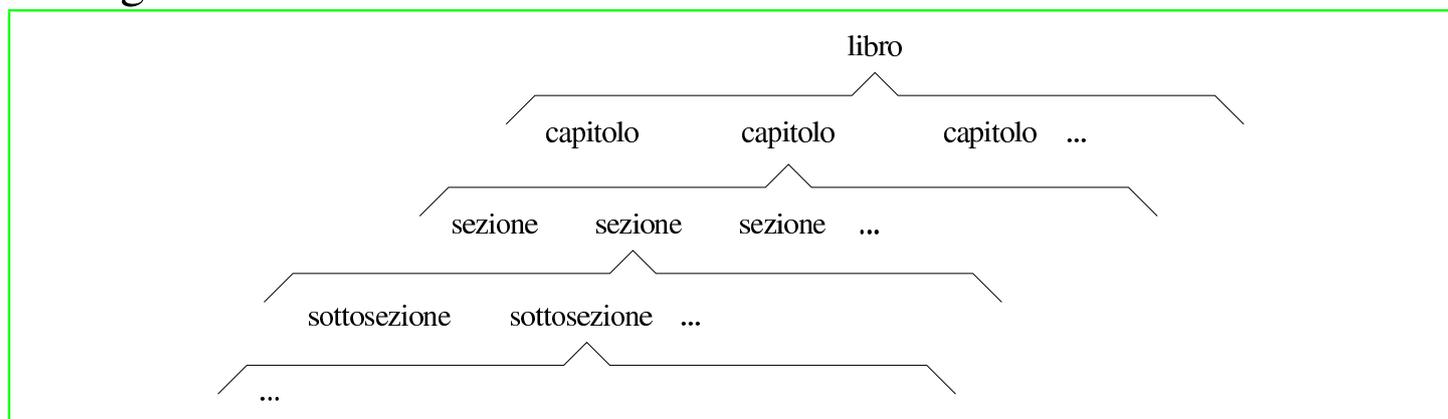


Sequenza dei nodi secondo Texinfo .....	1402
Definizione automatica della sequenza dei nodi e problemi relativi 1404	
Limitazioni originali della struttura a nodi .....	1406
Riferimenti ipertestuali e limitazioni verbali .....	1407
Altri tipi di riferimento .....	1411
Riepilogo dei comandi relativi a nodi, ancore e riferimenti ..	1413

Nel capitolo introduttivo è già stato affrontato il problema della gestione dei nodi in un documento Texinfo, ma alcuni aspetti sono stati solo sfiorati. Texinfo non può essere considerato pensando esclusivamente a uno dei risultati di composizione finale che possono essere generati, altrimenti si perde di vista la logica complessiva. In generale si sovrappongono due esigenze: il documento cartaceo da sfogliare e il documento elettronico da attraversare in modo ipertestuale.

Il documento cartaceo, ovvero il libro, ha una struttura ad albero che deriva dalla tradizione. Semplificando molto le cose si può rappresentare come nella figura u88.1, dove si vede che tutto viene suddiviso in capitoli, che possono eventualmente essere suddivisi ulteriormente in segmenti di livello inferiore (le sezioni).

Figura u88.1. Struttura ad albero di un libro.



Un libro particolarmente corposo potrebbe anche raggruppare assieme i capitoli in parti; mentre un'opera potrebbe anche essere suddivisa in raggruppamenti ancora più grandi, che di solito corrispondono ai volumi, ovvero ai tomi.

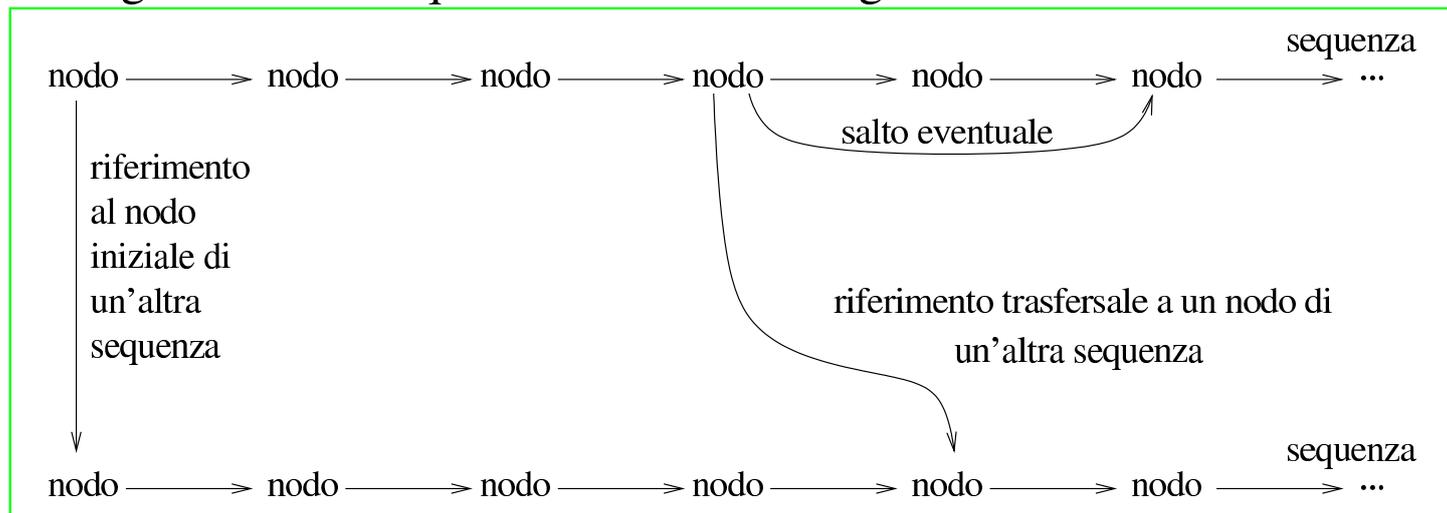
La suddivisione ad albero mostrata nella figura, non basta a descrivere la struttura di un libro. Infatti, occorre considerare che i capitoli, se suddivisi in sezioni, non sono composti semplicemente dalla somma di queste sezioni, in quanto, prima di tali suddivisioni introducono il problema, che poi viene descritto in modo particolareggiato. In pratica, è come se ogni capitolo suddiviso in sezioni contenesse una sezione fantasma iniziale. Lo stesso ragionamento vale per le sezioni che si articolano in sottosezioni e così via con le classificazioni inferiori.

Lo stesso discorso può valere per la classificazione in parti e in tomi, anche se in questi casi, le informazioni che precedono i capitoli, o le parti, tendono a non avere la stessa valenza.

Il documento elettronico ipertestuale è composto da blocchi di informazioni che Texinfo definisce opportunamente come nodi. Generalmente questi nodi sono indivisibili, come avviene con Texinfo, ma tendono a essere raggruppati in sequenze ideali, oltre che preve-

dere la possibilità di saltare ad altri nodi (e quindi ad altre sequenze potenziali) attraverso riferimenti trasversali.

Figura u88.2. Sequenze di nodi e collegamenti trasversali.



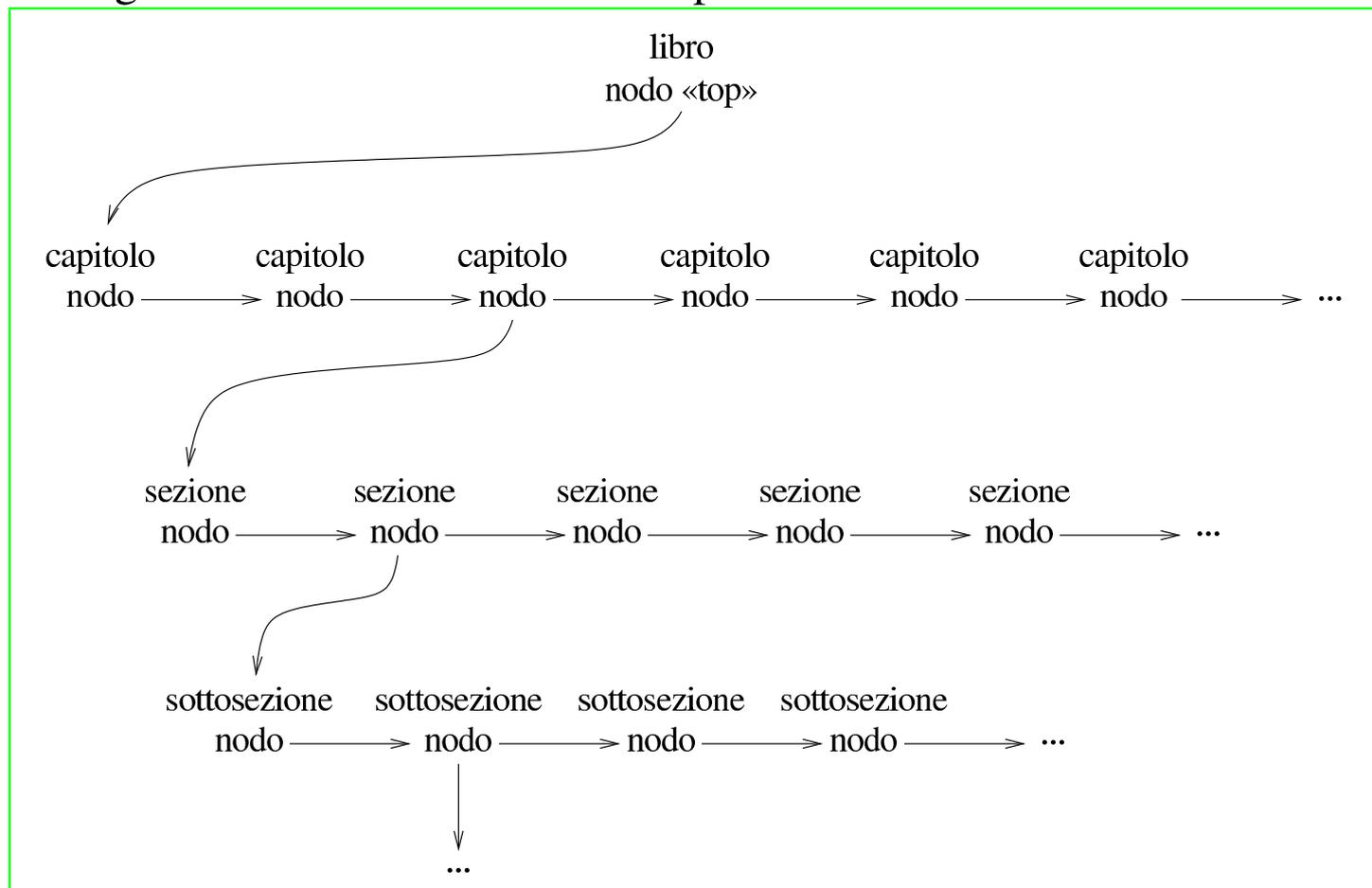
L'ipertesto puro è un documento in cui le informazioni sono raggiunte con un ordine che viene deciso durante la lettura, dove non c'è l'esigenza di leggere tutto e non ci si preoccupa se volontariamente o inavvertitamente si salta qualche nodo che compone l'ipertesto complessivo. In generale è proprio questo il problema: un ipertesto non dispone necessariamente di un percorso predefinito di lettura.

Nel momento in cui si intende realizzare un documento unico, simultaneamente libro e ipertesto, si deve giungere in qualche modo a un compromesso. Texinfo consente di realizzare un ipertesto estremamente complesso, oppure un libro tradizionale; se però si vogliono fare entrambe le cose, di solito conviene realizzare l'ipertesto secondo la struttura stessa del libro.

Texinfo propone una sequenza predefinita, che viene generata automaticamente quando i nodi vengono dichiarati subito prima delle suddivisioni tradizionali del documento (capitoli, sezioni, ecc.), senza specificare la sequenza a cui appartengono. In generale si forma-

no delle sequenze gerarchiche di questi nodi, dove si può passare ai livelli inferiori solo attraverso dei salti aggiuntivi, come si vede nella figura u88.3.

Figura u88.3. Struttura dei nodi predefinita secondo Texinfo.



I salti aggiuntivi che permettono di raggiungere una sequenza inferiore di nodi vengono raggruppati convenzionalmente in un menù finale di riferimenti.

## Sequenza dei nodi secondo Texinfo

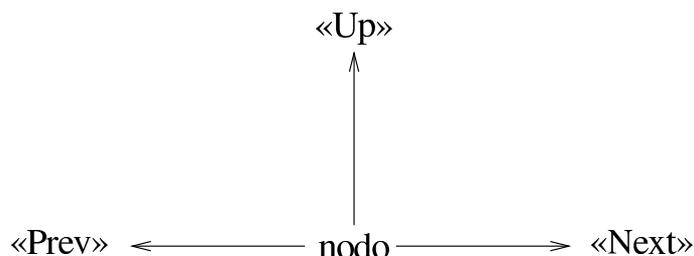
«

Secondo Texinfo, i nodi hanno tre riferimenti che servono a comporre le sequenze e a legare tali sequenze in una dipendenza gerarchica. Si tratta di:

- **'Next'** il riferimento al nodo successivo nella sequenza;

- ‘**Prev**’ il riferimento al nodo precedente nella sequenza;
- ‘**Up**’ il riferimento al nodo gerarchicamente precedente.

Figura u88.4. Riferimenti standard che si diramano a partire da un nodo di Texinfo.



Il primo nodo di un documento Texinfo deve essere denominato ‘**Top**’, che corrisponde idealmente a tutto ciò che precede il contenuto vero e proprio di un libro (subito dopo la copertina fino alla prefazione esclusa).

Per riprodurre lo schema gerarchico predefinito a cui si accennava in precedenza, dove si distinguono delle sequenze di nodi distinte a livelli diversi, è necessario inserire alla fine del testo di questi nodi un menù di riferimenti ai nodi di una sequenza inferiore. In generale, il nodo ‘**Top**’ prevede l’inserimento di un menù di riferimenti ai nodi della sequenza principale, corrispondente in pratica ai capitoli di un libro; se i capitoli si articolano in strutture inferiori, anche i nodi relativi devono disporre di un menù che faccia riferimento alle sezioni, continuando così fino all’ultimo livello che si deve raggiungere.

Tuttavia, resta il fatto che tutto questo non sia indispensabile, dal momento che le sequenze dei nodi potrebbero essere determinate in modo arbitrario, senza rispettare la struttura tipica di un libro; senza avere così la necessità di definire tutti questi menù.

È proprio questa idea legata alla presenza di sequenze di nodi separate gerarchicamente che impone la presenza dei menù e di conseguenza impone l'esistenza del nodo 'Top'. A questo punto, è il caso di osservare che il nodo 'Top' non può appartenere a una sequenza di altri nodi allo stesso suo livello; per questo, viene inserito normalmente nella stessa sequenza dei capitoli.

## Definizione automatica della sequenza dei nodi e problemi relativi

«

Texinfo prevede una struttura predefinita dei nodi che lo compongono, compatibile con la struttura di un libro, nel modo che è già stato mostrato in precedenza (figura u88.3). Per raggiungere questo risultato, nel sorgente Texinfo si indicano i nodi specificando solo il nome (senza stabilire relazioni con il nodo precedente, quello successivo e quello superiore). Tuttavia, questo non basta, perché se dal nodo si deve articolare una sequenza inferiore gerarchicamente, è necessario predisporre anche il menù relativo. L'esempio seguente rappresenta un nodo corrispondente a un capitolo, estratto dallo stesso sorgente dalla documentazione di Texinfo:

```
@node Overview
@chapter Overview of Texinfo

@dfn{Texinfo} is a documentation system that uses a single source file
to produce both online information and printed output. This means that
instead of writing two different documents, one for the online
information and the other for a printed work, you need write only one
document. Therefore, when the work is revised, you need revise only
that one document.

@menu
* Reporting Bugs::          Submitting effective bug reports.
* Using Texinfo::          Create printed or online output.
* Info Files::             What is an Info file?
```

* Printed Books::	Characteristics of a printed book or manual.
* Formatting Commands::	@@-commands are used for formatting.
* Conventions::	General rules for writing a Texinfo file.
* Comments::	Writing comments and ignored text in general.
* Minimum::	What a Texinfo file must have.
* Six Parts::	Usually, a Texinfo file has six parts.
* Short Sample::	A short sample Texinfo file.
* Acknowledgements and History::	Contributors and genesis.
@end menu	

Il capitolo si articola in diverse sezioni e tutte devono essere elencate nel menù che si vede. Questo fatto può essere sentito come una limitazione, che bene o male costringe l'autore a curarsi della realizzazione di questi riferimenti ipertestuali. Oltre a questo, è il caso di considerare il modo in cui si presenta il documento quando viene fatta la composizione in forma Info: quando si accede al nodo del capitolo, si vedono solo quelle poche righe iniziali, mentre per entrare nelle sezioni successive occorre passare per la selezione del menù.

Il problema della predisposizione di questi menù si può risolvere utilizzando Emacs, attraverso alcuni comandi specifici della modalità Texinfo. Tuttavia, questa non è la soluzione definitiva, dal momento che si costringe a utilizzare Emacs, mentre chi non vuole farlo resta costretto ad arrangiarsi a mano.

Il problema dei capitoli spezzati in nodi separati è più serio. In effetti la suddivisione fatta attraverso le sequenze gerarchiche di nodi è perfettamente logica; tuttavia, nel momento in cui ci si accinge a leggere un documento del genere, sarebbe forse più logico scorrere il capitolo verticalmente per raggiungere le sue classificazioni inferiori come se si trattasse di un'unica pagina. Purtroppo, il sistema Info non consente di avere dei sottonodi, ovvero dei riferimenti a posi-

zioni intermedie di un nodo, come invece avviene con l'HTML. Per risolvere in pratica questa limitazione bisogna limitarsi ad attribuire i nodi ai capitoli, tenendo presente che in questo modo non è possibile indicare nel testo dei riferimenti diretti a classificazioni inferiori. Tuttavia, recentemente è stato introdotto il comando '@**anchor**' con cui si ottiene l'inserimento di un'etichetta raggiungibile come se si trattasse di un nodo, smussando così il problema dei nodi.

## Limitazioni originali della struttura a nodi

«

Originariamente i nodi di Texinfo rappresentavano gli unici oggetti che potevano essere raggiunti attraverso riferimenti ipertestuali. In pratica, per fare riferimento a un capitolo o a una sezione, occorreva definire il nodo relativo per poi poter utilizzare comandi della serie '@...**ref**'.

È già stato visto che in generale conviene definire dei nodi in corrispondenza di tutti i capitoli, in modo da creare una sequenza definita a partire dal menù collocato nel nodo '**Top**'. Tuttavia, se c'è la necessità di fare riferimento a una sezione particolare di un certo capitolo, diventerebbe necessario dichiarare anche lì un nodo. Ma non basta definire un nodo in una sezione lasciando stare le altre sezioni, perché si creerebbe disordine nell'insieme; in pratica, se si definisce un nodo per una sezione di un certo capitolo, diventa indispensabile definire i nodi per le altre sezioni dello stesso capitolo, avendo poi cura di predisporre il menù necessario.

A questo problema si è posto rimedio aggiungendo il comando '@**anchor**{ }' che ha lo scopo di collocare un'ancora, ovvero un'etichetta raggiungibile attraverso riferimenti ipertestuali, senza dichiarare implicitamente l'inizio di un nodo in quella posizione.

```
@anchor{nome_ancora}
```

I nomi delle ancore appartengono allo stesso dominio dei nomi dei nodi, per cui non si devono creare dei conflitti nella scelta dei nomi. Inoltre, quando si fa riferimento a un'ancora, nella composizione Info si ottiene normalmente di raggiungere l'inizio del nodo in cui si trova.

Anche la gestione degli indici analitici è condizionata dalla struttura a nodi di Texinfo. In generale non è indispensabile che la voce da collocare in un indice analitico si trovi all'inizio di un nodo; quando però dall'indice analitico si vuole raggiungere il testo in cui questa è stata dichiarata, si arriva in realtà all'inizio del nodo in cui questa si trova. Se la voce si trova all'interno di una piccola sottosezione, mentre l'unico nodo disponibile è quello che fa capo al capitolo, si raggiunge l'inizio del capitolo. Naturalmente, questo vale per la navigazione di un documento che è stato composto in formato Info, mentre nelle altre forme di composizione il problema scompare o viene attenuato.

## Riferimenti ipertestuali e limitazioni verbali

Texinfo nasce come un sistema di composizione per documentazione scritta in lingua inglese. Attualmente il lavoro attorno a Texinfo si rivolge anche verso le esigenze delle altre lingue, selezionabili attraverso il comando '**@document language**', ma questo lavoro non è ancora completo nel momento in cui si scrivono queste note.

Texinfo dispone di quattro comandi diversi per i riferimenti ipertestuali, il cui scopo è quello di adattarsi alle esigenze del contesto.

Ma in questo caso, il contesto è prevalentemente di tipo «verbale». Vale la pena di descrivere brevemente questi quattro comandi, mostrando le conseguenze pratiche del loro utilizzo. Qui non vengono mostrate tutte le varianti perché ciò richiederebbe un capitolo apposito, mentre la documentazione originale è molto chiara a questo proposito.<sup>1</sup>

Vengono considerati i comportamenti confrontando solo la composizione Info e quella stampata (DVI, PostScript e PDF), perché l'HTML non ha ancora una sistemazione definitiva.

Questi comandi ricevono più argomenti distinti in base all'uso di una virgola di separazione. Per questa ragione la virgola non può essere usata all'interno di un argomento. Si tratta evidentemente di una limitazione importante da tenere in considerazione.

```
@xref{nodo, titolo_per_info, titolo_o_argomento, file_info, titolo_del_documento_s
```

Il comando '**@xref{}**' consente di ottenere dei riferimenti ipertestuali molto descrittivi. In generale è obbligatoria l'indicazione del nodo (il primo argomento), mentre il resto può essere omissivo. Dopo l'indicazione del nodo, alcuni argomenti successivi riguardano esclusivamente la composizione Info, mentre gli altri solo la composizione stampata (e simili). Nella composizione Info viene indicato il nome del nodo; se fornito appare il titolo specifico per la composizione Info ed eventualmente anche il nome del file Info esterno in cui cercarlo. Nella composizione per la stampa si ha l'indicazione

del titolo dell'argomento e se non viene fornita questa indicazione ci si limita a mostrare il nome del nodo stesso; se poi il riferimento è interno al documento viene aggiunta l'indicazione della pagina, altrimenti diventa necessario fornire il titolo del documento esterno che così appare al posto del numero della pagina.

L'uso più semplice di '@xref{}' è quello in cui si indica solo il nodo, come nell'esempio seguente:

```
Bla bla bla. @xref{Din don dan}. Bla bla bla...
```

Il risultato nell'ipertesto Info è:

```
Bla bla bla. *Note Din don dan::. Bla bla bla...
```

mentre con la composizione per la stampa l'aspetto è molto diverso:

```
Bla bla bla. See Chapter 3 [Din don dan], page 22. Bla bla bla...
```

Tanto per cominciare si può comprendere che si tratta di un riferimento che può essere collocato solo all'inizio di una frase (di un testo inglese), dal momento che la prima parola, '**See**', ha l'iniziale maiuscola. Eventualmente non è detto che il riferimento debba concludersi con un punto fermo come avviene nell'esempio, ma la frase che continua è comunque condizionata dal modo in cui viene rappresentato tale riferimento.

```
@ref{nodo, titolo_per_info, titolo_o_argomento, file_info, titolo_del_documento_sta
```

Il comando '@ref{}' si comporta in modo analogo a '@xref{}' con la differenza che nella composizione per la stampa non viene generata la parola '**See**' iniziale. Ciò consente di collocare il riferimento alla fine di una frase, oppure, con l'accortezza necessaria, anche in

mezzo.

```
@pxref{nodo, titolo_per_info, titolo_o_argomento, file_info, titolo_del_documento}
```

Il comando ‘**@pxref{}**’ è il più strano per chi non scrive utilizzando la lingua inglese. La lettera «p» sta per «parentheses», cioè parentesi, quelle all’interno delle quali dovrebbe essere collocato. A differenza del comando ‘**@xref{}**’, la composizione per la stampa inizia con ‘**see**’ (iniziale minuscola), mentre la composizione Info aggiunge un punto fermo.

```
@inforef{nodo, titolo, file_info}
```

Il comando ‘**@inforef{}**’ serve a fare riferimento a un file Info esterno, per il quale non si vuole o non si può fare riferimento a un’analogia versione stampata. Mentre nella composizione Info il risultato è uguale a quanto è già stato visto per ‘**@xref{}**’, nella composizione stampata viene fatto esplicito riferimento a un file Info. Si può ottenere una cosa simile a quella seguente:

```
Bla bla bla. See Info file ‘miofile’, node ‘Din don dan’. Bla bla bla...
```

Si intende che il riferimento sia fatto per essere collocato esattamente all’inizio di un periodo, dato il fatto che anche qui la parola ‘**See**’ ha l’iniziale maiuscola.

Tabella u88.10. Comandi per i riferimenti incrociati.

Comando	Descrizione o risultato
@anchor{ <i>etichetta</i> }	Inserisce un ancora nel testo.

Comando	Descrizione o risultato
<code>@xref</code>	«See...»
<code>@ref</code>	Come ‘ <b>@xref</b> ’, ma senza «See».
<code>@pxref</code>	«see...»
<code>@inforef</code>	«See Info file...».

## Altri tipi di riferimento

Texinfo dispone di altri tipi di riferimento che però risultano indolori dal punto di vista della lingua utilizzata per scrivere il proprio documento.

```
@uref{uri, descrizione, testo_sostitutivo_dell'indirizzo }
```

Il comando ‘**@uref{ }**’ consente di annotare un indirizzo URI secondo modalità differenti: se si indica solo il primo argomento, viene mostrato in ogni tipo di composizione; se appare anche il secondo argomento, vengono mostrate entrambe le cose, la descrizione e l’indirizzo, tranne nel caso della composizione HTML, in cui l’indirizzo non viene più mostrato; se si indica il terzo argomento (il secondo diventa superfluo), non si vuole mostrare l’indirizzo URI, mentre nella composizione HTML viene comunque attivato il riferimento. Si osservino gli esempi seguenti.

```
Bla bla bla @uref{http://www.dinkel.brot.dg/} bla bla bla...
```

Questo genera l’inserimento dell’indirizzo nel testo senza delimitazioni, in ogni tipo di composizione.

```
Bla bla bla @uref{http://www.dinkel.brot.dg/, Titolo} bla bla bla...
```

In questo modo, la composizione per la stampa e quella per Info generano un risultato del tipo:

```
Bla bla bla Titolo (http://www.dinkel.brot.dg/) bla bla bla...
```

Invece, nella composizione HTML l'indirizzo URI scompare dalla vista, nel modo seguente:

```
Bla bla bla <a href="http://www.dinkel.brot.dg/">Titolo</a> bla bla bla...
```

Infine, l'esempio seguente mostra l'uso del terzo argomento (si noti l'uso della coppia di virgole per segnalare l'assenza del secondo argomento):

```
Bla bla bla @uref{http://www.dinkel.brot.dg/, , Titolo} bla bla bla...
```

Nella composizione stampata e in quella Info si perde completamente l'informazione dell'indirizzo URI:

```
Bla bla bla Titolo bla bla bla...
```

Nella composizione HTML l'indirizzo URI rimane nascosto alla vista, come è già stato visto in precedenza:

```
Bla bla bla <a href="http://www.dinkel.brot.dg/">Titolo</a> bla bla bla...
```

A fianco di '@uref{}' si pone anche un comando specifico per l'annotazione di indirizzi di posta elettronica:

```
@email{indirizzo, descrizione}
```

Il comando '@email{}' si comporta in pratica come '@uref{}', con la differenza che il terzo argomento non esiste, per cui si mostra sempre l'indirizzo, che eventualmente viene preceduto dalla sua

descrizione. Nel caso della composizione in HTML, viene generato un riferimento ipertestuale del tipo ‘**mailto:**’.

Esiste un altro modo di indicare un riferimento a un indirizzo URI. Si tratta del comando ‘**@url{}**’, che serve solo a mostrare tale indirizzo, senza generare nel formato HTML alcun riferimento:

```
@url{uri}
```

L’indirizzo URI viene mostrato senza delimitazioni in ogni tipo di composizione. In generale può essere conveniente utilizzare questo comando al posto di ‘**uref{}**’ quando si indica un indirizzo ipotetico o un indirizzo che non è più valido (al quale non sarebbe opportuno puntare con un riferimento ipertestuale).

## Riepilogo dei comandi relativi a nodi, ancore e riferimenti



- ```
@node nome_del_nodo, nodo_successivo, nodo_precedente, nodo_superiore
```

Definizione di un nodo, da collocare subito prima di un capitolo, una sezione, o di un’altra classificazione analoga. Il comando occupa una riga.

- ```
@menu  
[testo_descrittivo]  
voce_del_menus  
...  
@end menu
```

Si tratta della definizione di un menù da collocare alla fine del testo di un nodo, per raggiungere una sequenza di nodi di livello inferiore. Le voci del menù possono avere due forme alternative:

\* [ *(file\_info)* ] *nome\_nodo* :: *titolo\_o\_argomento*

\* *nome\_della\_voce* : [ *(file\_info)* ] *nome\_nodo* . *titolo\_o\_argomento*

In generale, la seconda forma è usata molto poco.

- @anchor{*nome\_ancora*}

Definizione di un'ancora, ovvero un'etichetta a cui poter fare riferimento attraverso comandi '@...ref'. I nomi delle ancore e i nomi dei nodi appartengono allo stesso dominio.

- @xref{*nodo*, *titolo\_per\_info*, *titolo\_o\_argomento*, *file\_info*, *titolo\_del\_documento*}

@ref{*nodo*, *titolo\_per\_info*, *titolo\_o\_argomento*, *file\_info*, *titolo\_del\_documento*}

@pxref{*nodo*, *titolo\_per\_info*, *titolo\_o\_argomento*, *file\_info*, *titolo\_del\_documento*}

Tre tipi complementari di riferimento a un nodo dello stesso documento o di un documento esterno, realizzato sempre con Texinfo. Nel primo caso il riferimento va posto all'inizio di un periodo;

nel secondo può stare all'interno o alla fine di una frase; nel terzo caso deve essere collocato tra parentesi.

- `@inforef{nodo, titolo, file_info}`

Riferimento a un nodo esterno di un documento disponibile solo in forma Info. Il riferimento va posto all'inizio di un periodo.

- `@uref{uri, descrizione, testo_sostitutivo_dell'indirizzo}`

`@email{indirizzo, descrizione}`

Riferimento a un URI generico o a un indirizzo di posta elettronica, per il quale la composizione HTML genera un riferimento.

- `@url{uri}`

Annotazione pura e semplice di un indirizzo URI senza creare alcun riferimento nella composizione HTML.

<sup>1</sup> È il caso di ricordare che le parentesi graffe fanno parte dei comandi di Texinfo.



# Sgmltexi: installazione e utilizzo



Installazione di Sgmltexi .....	1418
Gettext .....	1419
Dipendenze .....	1420
Come si usa il programma frontale .....	1420
Riferimenti .....	1425

Sgmltexi <sup>1</sup> è un DTD e un sistema frontale per la composizione in Texinfo a partire da un formato SGML. L'idea alla base di Sgmltexi è quella di avere la possibilità di scrivere un documento Texinfo attraverso la semplificazione e la guida che può dare un sistema SGML.

All'interno di Sgmltexi, la gestione dei nodi di Texinfo può avvenire in modo automatico e trasparente, generando un menù Info unico nel nodo 'Top'. I nomi dei nodi, quando sono generati automaticamente, usano stringhe del tipo «cap 1», «app A»,...

Sgmltexi ha uno schema preciso: ci possono essere una o più introduzioni iniziali; nella parte centrale c'è un corpo che può essere scomposto in vario modo; ci possono essere delle appendici; al termine possono apparire degli indici analitici. Il corpo è organizzato in capitoli, che possono essere raggruppati in parti ed eventualmente anche in tomi. In tal modo, si possono gestire facilmente anche documenti di grandi dimensioni.

Sgmltexi è un progetto che non viene più sviluppato, in considerazione del fatto che la documentazione GNU tende a migrare verso Docbook, pur garantendo la compatibilità con Texinfo.

## Installazione di Sgmltexi

«

Sgmltexi è composto da due eseguibili Perl: `'sgmltexi'` e `'sgmltexi-sp2texi'`. Questi due file devono essere collocati in una directory in cui possono essere avviati senza bisogno di indicare il percorso; in pratica in una directory elencata all'interno della variabile di ambiente `'PATH'`.

Evidentemente, è necessario l'interprete Perl; precisamente questi programmi cercano il file `"/usr/bin/perl"`. Se il proprio sistema operativo è organizzato diversamente, è necessario intervenire modificando la prima riga dei due eseguibili:

```
#!/usr/bin/perl  
...
```

Sgmltexi si aspetta di trovare alcuni file:

- `"/etc/sgmltexi/sgmltexi.cat"`  
il catalogo SGML di Sgmltexi;
- `"/etc/sgmltexi/sgmltexi.dcl"`  
la dichiarazione SGML di Sgmltexi;
- `"/etc/sgmltexi/sgmltexi.dtd"`  
il DTD di Sgmltexi;
- `"/etc/sgmltexi/entities/"`  
la directory contenente i file delle entità SGML standard.

Tutti questi file dovrebbero trovarsi esattamente dove previsto; in alternativa si devono realizzare almeno dei collegamenti per ricreare i percorsi stabiliti.

## Gettext

I messaggi di Sgmltexi possono essere tradotti. Per installare i file PO già esistenti è necessario compilarli come nell'esempio seguente:

```
$ msgfmt -vvvv -o sgmltexi.mo it.po [Invio]
```

In questo esempio, il file 'it.po' viene compilato generando il file 'sgmltexi.mo'. Questo file può essere collocato in '/usr/share/locale/it/LC\_MESSAGES/', o in un'altra posizione analoga in base agli standard del proprio sistema operativo.

Se non è disponibile il modulo Perl-gettext, che serve a Sgmltexi per accedere alle traduzioni, è possibile eliminare il suo utilizzo e simulare la funzione di Gettext. In pratica si commentano le istruzioni seguenti:

```
# Non si vuole usare gettext.  
#use POSIX;  
#use Locale::gettext;  
#setlocale (LC_MESSAGES, "");  
#textdomain ("sgmltexi");
```

Inoltre, si tolgono i commenti dalla dichiarazione della funzione fittizia '**gettext ()**', come si vede qui:

```
sub gettext  
{  
    return $_[0];  
}
```

## Dipendenze

«

È il caso di riepilogare le dipendenze di Sgmltexi da altri applicativi:

- Perl

dal momento che si tratta di un programma scritto in Perl, deve essere presente l'interprete relativo;

- SP o Jade

per l'analisi SGML occorre il programma `'nsgmls'` che fa parte del pacchetto SP o anche del pacchetto Jade;

- Perl-gettext

per accedere ai messaggi tradotti del programma, è necessario il modulo Perl-gettext, salva la possibilità di escluderne l'utilizzo come è già stato mostrato;

- TeX e Texinfo

per arrivare a una composizione finale è necessario ovviamente disporre di Texinfo, che potrebbe già essere integrato nella propria distribuzione TeX (di solito si tratta di teTeX).

## Come si usa il programma frontale

«

Una volta preparato il sorgente in formato Sgmltexi, bisogna utilizzare il programma `'sgmltexi'` per controllare l'elaborazione SGML e gli altri applicativi di composizione di Texinfo.

Di solito, la cosa migliore per iniziare lo studio di un sistema di composizione, è partire da un esempio banale, funzionante, che consenta di apprendere l'uso elementare degli strumenti relativi.

```

<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN">
<sgmltexi lang="it">
<head>
  <admin>
    <setfilename content="esempio.info">
    <settitle content="Esempio">
  </admin>
  <titlepage>
    <title>Esempio</title>
    <subtitle>Un esempio per un documento in formato Sgmltexi</subtitle>
    <abstract>
      <p>Questo è solo un esempio di un documento scritto
        utilizzando Sgmltexi.</p>
    </abstract>
    <author>Pinco Pallino &lt;ppinco@dinkel.brot.dg&gt;</author>
    <legal>
      <copyright>Copyright &copy; 2000 Pinco Pallino</copyright>
      <license>
        <p>Permission is granted to copy, distribute and/or
          modify this document under the terms of the GNU Free
          Documentation License, Version 1.1 or any later version
          published by the Free Software Foundation; with no
          Invariant Sections, with no Front-Cover Texts, and with
          no Back-Cover Texts. A copy of the license is included
          in the section entitled "GNU Free Documentation
          License".</p>
      </license>
    </legal>
  </titlepage>
  <contents>
</head>
<body>
<h1>Esempio generale</h1>

<p>Questo è l'esempio tipico di un capitolo di Sgmltexi...</p>

<p>Non c'è molto da scrivere in questo caso...</p>

</body>
</sgmltexi>

```

Supponendo di avere installato correttamente Sgmltexi (e anche Te-

xinfo), supponendo inoltre che il file si chiami ‘prova.sgml’, si possono usare i comandi seguenti:

- `$ sgmltexi --sgml-check prova.sgml [Invio]`

per verificare la correttezza formale dell’SGML;

- `$ sgmltexi --texi prova.sgml [Invio]`

per ottenere semplicemente il file ‘prova.texinfo’, in formato Texinfo;

- `$ sgmltexi --info prova.sgml [Invio]`

per ottenere il file ‘prova.info’, in formato Info;

- `$ sgmltexi --dvi prova.sgml [Invio]`

per ottenere il file ‘prova.dvi’, in formato DVI;

- `$ sgmltexi --ps prova.sgml [Invio]`

per ottenere il file ‘prova.ps’, in formato PostScript;

- `$ sgmltexi --pdf prova.sgml [Invio]`

per ottenere il file ‘prova.pdf’, in formato PDF;

- `$ sgmltexi --html prova.sgml [Invio]`

per ottenere il file ‘prova.html’, in formato HTML;

- `$ sgmltexi --xml prova.sgml [Invio]`

per ottenere il file ‘prova.xml’, in formato XML di Texinfo;

- `$ sgmltexi --docbook prova.sgml [Invio]`

per ottenere il file ‘prova.xml’, in formato XML di Docbook.

La sintassi di ‘**sgmltexi**’ è quella che si vede nello schema seguente:

```
sgmltexi [opzioni] sorgente_sgml
```

In generale, è bene che il nome del file sorgente in formato Sgmltexi abbia l’estensione standard ‘.sgml’.

Opzione	Descrizione
--help	Mostra una guida sintetica e termina di funzionare.
--version	Mostra le informazioni sulla versione e termina di funzionare.
--force	Quando il contesto lo consente, forza le situazioni. Può essere utile in particolare per la composizione in formato Info e in formato HTML, per passare la stessa opzione al programma ‘ <b>makeinfo</b> ’.
--number-sections	Numera le sezioni quando ciò non è previsto in modo normale.
--clean	Elimina i file intermedi che non servono, abbinati al nome del sorgente.
--verbose	Mostra più informazioni durante l’elaborazione.
--input-encoding= <i>codifica</i>	Stabilisce la codifica del file in ingresso, tenendo conto che sono ammissibili solo le parole chiave ‘ <b>ISO-8859-n</b> ’, dove <i>n</i> va da 1 a 10.
--sgml-include= <i>entità_parametrica</i> --include= <i>entità_parametrica</i>	Assegna la parola chiave ‘ <b>INCLUDE</b> ’ all’entità parametrica SGML indicata. Questo serve ad abilitare l’inclusione di porzioni di sorgente SGML che sono controllate in questo modo.

Opzione	Descrizione
<pre>--paper=<i>formato_composizione</i></pre>	<p>Serve a definire in qualche modo il formato finale stampato di composizione. Sono disponibili le parole chiave seguenti: <b>'letter'</b>, <b>'a4'</b>, <b>'a4wide'</b>, <b>'a4latex'</b> e <b>'small'</b>.</p>
<pre>--setchapternewpage={ on↵ ↵   off   odd }</pre>	<p>Definisce l'inizio dei capitoli nella composizione per la stampa, ignorando il marcatore <b>'&lt;setchapternewpage content="..."&gt;'</b> all'interno del sorgente del documento.</p>
<pre>--footnotestyle={end↵ ↵   separate }</pre>	<p>Definisce la collocazione delle note a piè pagina, ignorando il marcatore <b>'&lt;footnotestyle content="..."&gt;'</b> all'interno del sorgente.</p>
<pre>--headings={on   off   single↵ ↵   double   singleafter↵ ↵   doubleafter }</pre>	<p>Attiva o disattiva le intestazioni, specificando eventualmente l'uso di intestazioni uguali o distinte. Questa opzione fa ignorare il marcatore <b>'&lt;headings content="..."&gt;'</b> all'interno del sorgente del documento.</p>
<pre>--sgml-syntax --sgml-check</pre>	<p>Controlla la correttezza formale del sorgente SGML, mostrando gli errori trovati.</p>
<pre>--sp</pre>	<p>Genera un risultato «post-SP», nel senso che restituisce soltanto quanto ottenuto dall'analizzatore SGML, a scopo diagnostico.</p>
<pre>--texi --texinfo</pre>	<p>Genera un sorgente Texinfo.</p>

Opzione	Descrizione
<code>--dvi</code>	Compone generando un risultato in formato DVI.
<code>--ps</code> <code>--postscript</code>	Compone generando un formato in PostScript.
<code>--pdf</code>	Compone generando un formato PDF.
<code>--info</code>	Genera un risultato in formato Info.
<code>--text</code>	Genera un risultato in formato testo puro.
<code>--html</code>	Genera un risultato in formato HTML.
<code>--xml</code>	Genera un risultato in formato XML di Texinfo.
<code>--docbook</code>	Genera un risultato in formato XML Docbook.

- `$ sgmltexi --sgml-syntax prova.sgml [Invio]`

Analizza la validità formale del sorgente ‘`prova.sgml`’.

- `$ sgmltexi --ps prova.sgml [Invio]`

Genera un risultato in formato PostScript attraverso l’aiuto di ‘`texi2dvi`’ e di ‘`dvips`’.

## Riferimenti

- Daniele Giacomini, *Sgmltexi*  
[extra/sgmltexi/](#)



- Gaetano Paolone, *Sgmltexi*, pacchetto GNU/Linux Debian  
<http://packages.debian.org/sgmltexi>  
<http://ftp.debian.org/debian/pool/main/s/sgmltexi/>

<sup>1</sup> **Sgmltexi** GNU GPL

# Sgmltexi: struttura



Struttura generale per un sorgente Sgmltexi .....	1428
Intestazione .....	1435
Informazioni amministrative .....	1436
Pagine iniziali .....	1440
Indice generale .....	1444
Nodi e menù Info iniziale .....	1445
Introduzione .....	1446
Corpo .....	1447
Appendice .....	1449
Indici analitici .....	1450
Scomposizione del documento, nodi e menù Info .....	1451
Numerazione o meno dei titoli .....	1453
Codifica .....	1454
Entità standard e non standard .....	1455

Sgmltexi impone uno schema preciso al documento, in base alle consuetudini dei documenti stampati. Questo capitolo descrive brevemente tale struttura.

# Struttura generale per un sorgente Sgmltexi



Il sorgente Sgmltexi tipico inizia così:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN">
```

Naturalmente, potrebbe essere conveniente la definizione iniziale di alcune entità generali, come si vede nell'esempio seguente:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN">
[
<!ENTITY EDITION "2000.05.20">
...
...
]>
```

Tutto il documento viene racchiuso all'interno dell'elemento **'sgmltexi'**, rispettando una certa struttura: deve esserci un elemento **'head'**, ci può essere un elemento **'intro'**, ci deve essere un elemento **'body'**, infine ci può essere un elemento **'appendix'**. Lo spazio successivo all'elemento **'appendix'** può essere occupato da alcuni indici analitici (cosa che viene descritta meglio in seguito).

```
<sgmltexi>
<head>
...
</head>
<intro>
...
</intro>
<body>
...
</body>
<appendix>
...
</appendix>
</sgmltexi>
```

L'elemento **'sgmltexi'** ha tre attributi: **'lang'**, **'charset'**, **'spacing'**. Attraverso l'attributo **'lang'** si definisce il linguaggio

in cui è scritto il documento, richiamando implicitamente una configurazione particolare all'interno di Texinfo. Questo linguaggio si indica assegnando una sigla corrispondente allo standard ISO 639 (sezione 13.3), come si vede nell'esempio seguente:

```
<sgmltexi lang="it">
```

L'attributo **'charset'** permette di indicare il valore da assegnare al comando **'@documentencoding'** di Texinfo. L'uso di questo attributo viene oscurato dall'opzione **'--input-encoding'**, se questa viene usata. Infatti, tale opzione implica un'elaborazione del sorgente per cui si genera un file Texinfo in formato ISO 646 (ASCII tradizionale), cosa che fa perdere di significato al comando **'@documentencoding'**.

La composizione di un sorgente Texinfo dà risultati differenti a seconda dei casi, per cui alle volte può essere conveniente scrivere usando comandi come **'@`a'** («à»), mentre altre volte conviene scrivere usando una codifica ISO 8859-*n*, annotando questo nel comando **'@documentencoding'**. Probabilmente, è prevista la sistemazione di questo problema nelle prossime versioni di Texinfo; per ora l'ambivalenza di Sgmltexi può aiutare in tal senso.

L'attributo **'spacing'** dovrebbe essere superfluo, dal momento che serve a definire la spaziatura alla fine del punto fermo. Questo comportamento dovrebbe essere definito automaticamente in base alla scelta del linguaggio. Questo attributo consente quindi di forzare la situazione, imponendo una spaziatura non conforme allo standard. I valori che si possono assegnare sono: **'normal'**,

‘**french**’ e ‘**uniform**’. Assegnando ‘**french**’, oppure ‘**uniform**’, si ottiene in pratica la stessa cosa che si otterrebbe con il comando ‘**@frenchspacing**’ di Texinfo. L’esempio seguente rappresenta ciò che potrebbe essere conveniente in un testo italiano:

```
<sgmltexi lang="it" charset="ISO-8859-1" spacing="uniform">
```

Tabella u90.6. Elementi SGML che compongono la struttura generale.

Elemento o attributo	Contenuto	Descrizione
sgmltexi		Contenitore del documento.
<i>lang</i>	Attributo	Sigla ISO 639 del linguaggio.
<i>charset</i>	Attributo	Codifica nella forma ‘ <b>ISO-8859-n</b> ’.
<i>spacing</i>	Attributo	‘ <b>normal</b> ’, ‘ <b>french</b> ’ e ‘ <b>uniform</b> ’.
head		Intestazione del documento.
admin		Informazioni amministrative.
setfilename	Vuoto	Inserisce il comando ‘ <b>@setfilename</b> ’.
<i>content</i>	Attributo	Il nome del primo file Info da generare.
settitle	Vuoto	Inserisce il comando ‘ <b>@settitle</b> ’.
<i>content</i>	Attributo	Titolo.
setchapternewpage	Vuoto	Inserisce il comando ‘ <b>@setchapternewpage</b> ’.
<i>content</i>	Attributo	Separazione dei capitoli: ‘ <b>on</b> ’, ‘ <b>off</b> ’, ‘ <b>odd</b> ’.

Elemento o attributo	Contenuto	Descrizione
footnotestyle	Vuoto	Inserisce il comando '@footnotestyle'.
<i>content</i>	Attributo	Piè pagina: 'end', 'separate', 'empty'.
headings	Vuoto	Inserisce il comando '@headings'.
<i>content</i>	Attributo	Intestazioni: 'on', 'off', 'single',
		'double', 'singleafter', 'doubleafter'.
defindex	Vuoto	Inserisce il comando '@defindex'.
<i>name</i>	Attributo	Sigla di due lettere dell'indice analitico.
defcodeindex	Vuoto	Inserisce il comando '@defcodeindex'.
<i>name</i>	Attributo	Sigla di due lettere dell'indice analitico.
synindex	Vuoto	Inserisce il comando '@synindex'.
<i>from</i>	Attributo	L'indice di origine: una sigla di due lettere.
<i>to</i>	Attributo	L'indice di destinazione: una sigla di due lettere.
syncodeindex		Inserisce il comando '@syncodeindex'.
<i>from</i>	Attributo	L'indice di origine: una sigla di due lettere.
<i>to</i>	Attributo	Destinazione in cui deve apparire in dattilografico.

Elemento o attributo	Contenuto	Descrizione
infodir	Vuoto	Comando '@direntry' in modo automatico.
infodir	#PCDATA	Comando '@direntry' con un contenuto letterale.
titlepage		Informazioni delle prime pagine.
title	%inline;	Inserisce il comando '@title'.
subtitle	%inline;	Inserisce il comando '@subtitle'.
abstract	%block;	Descrizione del contenuto del documento.
author	%inline;	Inserisce il comando '@author'.
frontcovertext	%block;	Testo da inserire in copertina.
tpextra	%block;	Testo aggiuntivo nelle prime pagine.
legal		Informazioni legali alla base della seconda pagina.
copyright	%inline;	Una riga di copyright.
publishnote	%block;	Note da mostrare prima della licenza.
license	%block;	Condizioni con cui è rilasciato il documento.
coverart	%block;	Note sulla copertina, da mostrare dopo la licenza.
dedications	%block;	Pagina delle dediche.
contents	Vuoto	Indice generale standard.
shortcontents	Vuoto	Indice generale ridotto.
summarycontents	Vuoto	Indice generale ridotto.

Elemento o attributo	Contenuto	Descrizione
menu	Vuoto	Inserisce un menù Info automatico.
topnode	Vuoto	Specifica il nodo iniziale.
<i>next</i>	Attributo	Riferimento al nodo successivo.
<i>prev</i>	Attributo	Riferimento al nodo precedente.
<i>up</i>	Attributo	Riferimento al nodo superiore.
menu		Inserisce un menù Info manuale.
detailmenu	#PCDATA	Dettaglio nel menù Info.
intro		Delimita i capitoli che compongono l'introduzione.
h1		Titolo di un capitolo introduttivo.
h2		Titolo di una sezione introduttiva.
h3		Titolo di una sottosezione introduttiva.
h4		Titolo di una sotto-sottosezione introduttiva.
body		Delimita il corpo del documento.
tomeheading		Titolo di un tomo.
partheadng		Titolo di una parte.
h1		Titolo di un capitolo.
h2		Titolo di una sezione.
h3		Titolo di una sottosezione.

Elemento o attributo	Contenuto	Descrizione
h4		Titolo di una sotto-sottosezione.
appendix		Delimita i capitoli che compongono l'appendice.
h1		Titolo di un'appendice.
h2		Titolo di una sezione di appendice.
h3		Titolo di una sottosezione di appendice.
h4		Titolo di una sotto-sottosezione di appendice.
indexheading		Titolo di un indice analitico.
printindex	Vuoto	Inserisce un indice analitico particolare.
<i>name</i>	Attributo	Sigla dell'indice analitico da inserire.
<i>titolo_generico</i>		I titoli hanno degli attributi in comune.
<i>id</i>	Attributo	Ancora per i riferimenti ipertestuali.
<i>node</i>	Attributo	Definizione manuale del nodo.
<i>menu</i>	Attributo	Titolo che appare nel menù.
<i>next</i>	Attributo	Definizione manuale del prossimo nodo.
<i>prev</i>	Attributo	Definizione manuale del nodo precedente.
<i>up</i>	Attributo	Definizione manuale del nodo superiore.

Elemento o attributo	Contenuto	Descrizione
<i>titolo_h</i>		Dal capitolo in giù c'è un attributo aggiuntivo.
<i>type</i>	Attributo	Numerato, non numerato o intestazione semplice: <b>'numbered'</b> , <b>'unnumbered'</b> , <b>'heading'</b> .

## Intestazione

L'elemento **'head'** è il più complicato. È necessario per definire molte informazioni che riguardano il documento. Segue un esempio abbastanza completo, che si riferisce alla documentazione ipotetica dello stesso Sgmltexi.

```
<head>
  <admin>
    <setfilename content="sgmltexi.info">
    <setttitle content="Sgmltexi">
    <setchapternewpage content="odd">
    <defindex name="sg">
    <syncodeindex from="sg" to="cp">
    <infodir cat="Texinfo documentation system">
  </admin>
  <titlepage>
    <title>Sgmltexi</title>
    <subtitle>An alternative way to write Texinfo
documentation</subtitle>
    <subtitle>This edition is for Sgmltexi
&EDITION; (alpha) for Texinfo 4.0</subtitle>
    <abstract>
      <p>Sgmltexi is an SGML system (DTD and tools) to
make Texinfo documentation using SGML...</p>
      ...
    </abstract>
    <author>Daniele Giacomini &lt;daniele@swlibero.org&gt;</author>
```

```

<legal>
  <copyright>Copyright &copy; 2000 ...</copyright>
  <publishnote>
    <p>Published by...</p>
  </publishnote>
  <license>
    <p>Permission is granted to make and distribute
      verbatim copies of this manual...</p>
    ...
  </license>
  <coverart>
    <p>Cover art by ...</p>
  </coverart>
</legal>
</titlepage>
<shortcontents>
<contents>
</head>

```

Guardando l'esempio, si possono riconoscere alcuni elementi importanti: **'admin'**, usato per alcune informazioni amministrative, e **'titlepage'**.

## Informazioni amministrative



L'elemento **'admin'** viene usato per indicare al suo interno alcune informazioni che vanno prevalentemente nell'intestazione del documento Texinfo finale, oppure subito dopo. I componenti di questo ambiente non hanno un ordine preciso, nel sorgente SGML, in quanto poi vengono riordinati prima della composizione in Texinfo. Nel seguito vengono elencati e descritti gli elementi che possono apparire all'interno di **'admin'**.

- **'setfilename'**

Si tratta di un elemento vuoto, utilizzato per definire il nome del file Info finale, attraverso il comando **'@setfilename'** di Texin-

fo. Si usa con l'attributo **'content'** a cui si assegna il nome di questo file.

```
<setfilename content="sgmltexi.info">
```

L'esempio mostra il caso in cui si definisce il nome **'sgmltexi.info'**. Si può vedere che non serve il marcatore di chiusura.

- **'settitle'**

Si tratta di un elemento vuoto, utilizzato per definire il titolo per la composizione in formato Info, attraverso il comando **'@settitle'** di Texinfo. Si usa con l'attributo **'content'** a cui si assegna questo titolo.

```
<settitle content="Sgmltexi">
```

L'esempio mostra il caso in cui si definisce il nome **'Sgmltexi'**. Si può vedere che non serve il marcatore di chiusura.

- **'setchapternewpage'**

Si tratta di un elemento vuoto, non essenziale, utilizzato per definire il comando corrispondente di Texinfo: **'@setchapternewpage'**. Si assegna una parola chiave all'attributo **'content'**, tra **'on'**, **'off'** e **'odd'**.

```
<setchapternewpage content="on">
```

L'esempio mostra la richiesta esplicita di iniziare ogni capitolo in una pagina nuova.

Il programma frontale di Sgmltexi, **'sgmltexi'**, accetta un'opzione con lo stesso nome (**'--setchapternewpage={on|off|odd}'**) che prevale su quanto stabilito nel sorgente SGML in questo modo.

- **'footnotestyle'**

Si tratta di un elemento vuoto, non essenziale, utilizzato per definire il comando corrispondente di Texinfo: ‘**@footnotestyle**’. Si assegna una parola chiave all’attributo ‘**content**’, che può essere ‘**end**’ o ‘**separate**’.

```
<footnotestyle content="end">
```

L’esempio mostra la richiesta esplicita di inserire i piè pagina alla fine della pagina a cui si riferiscono.

Il programma frontale di Sgmltexi accetta un’opzione con lo stesso nome (‘**--footnotestyle={end|separate}**’) che prevale su quanto stabilito nel sorgente SGML in questo modo.

- ‘**headings**’

Si tratta di un elemento vuoto, non essenziale, utilizzato per definire il comando corrispondente di Texinfo: ‘**@headings**’. Si assegna una parola chiave all’attributo ‘**content**’, che può essere: ‘**on**’, ‘**off**’, ‘**single**’, ‘**double**’, ‘**singleafter**’, ‘**doubleafter**’.

```
<headings content="on">
```

L’esempio mostra la richiesta esplicita di mostrare le intestazioni.

Il programma frontale di Sgmltexi accetta un’opzione con lo stesso nome, a cui si assegnano le stesse parole chiave (‘**--headings=impostazione**’), che prevale su quanto stabilito nel sorgente SGML in questo modo.

- ‘**defindex**’, ‘**defcodeindex**’

Si tratta di elementi vuoti, non essenziali, utilizzati per definire i comandi corrispondenti di Texinfo: ‘**@defindex**’ e ‘**@defcodeindex**’. Si assegna un nome composto da due lettere all’attributo ‘**name**’, per definire un indice analitico aggiuntivo;

in particolare, utilizzando l'elemento '**defcodeindex**' si ottiene la creazione di un indice analitico composto da voci riprodotte in dattilografico.

```
<defindex name="sg">
```

L'esempio mostra la definizione dell'indice analitico normale, identificato dalla sigla '**sg**'.

Naturalmente, si possono inserire più elementi '**defindex**' e '**defcodeindex**', quanti sono gli indici specifici che si vogliono dichiarare.

- '**synindex**', '**syncodeindex**'

Questi due elementi vuoti, vengono usati per copiare le voci di un indice analitico all'interno di un altro, come fanno i comandi corrispondenti di Texinfo: '@**synindex**' e '@**syncodeindex**'. Questi due elementi richiedono l'indicazione di due attributi, '**from**' e '**to**', a cui si assegna rispettivamente la sigla dell'indice analitico di partenza e quella dell'indice di destinazione. Si osservi l'esempio:

```
<syncodeindex from="fn" to="cp">
```

In questo caso, si trasferiscono tutte le voci dell'indice '**fn**' (quello delle funzioni) nell'indice '**cp**' (l'indice analitico standard). In particolare, dal momento che si tratta di '**syncodeindex**', le voci che vengono trasferite sono poi rese in modo dattilografico (con il comando '@**code**').

- '**infodir**'

Questo elemento viene usato per definire una voce da inserire nell'elenco principale Info, quando il file relativo viene installato con il comando '**install-info**'. L'elemento contiene l'attribu-

to ‘**cat**’ a cui si assegna la categoria, come si fa con il comando ‘**@dircategory**’ di Texinfo.

```
<infodir cat="Texinfo documentation system">
```

L’elemento ‘**infodir**’ può essere vuoto, come appena mostrato nell’esempio, ottenendo così l’inserimento di una sola riga nel corpo del comando ‘**@direntry**’ di Texinfo, utilizzando le informazioni già conosciute: il nome del file Info e il titolo del documento. Se si vuole fare a mano, è possibile inserire queste informazioni all’interno dell’elemento, come nell’esempio seguente:

```
<infodir cat="Sistema di documentazione Sgmltexi">
* Sgmltexi: (sgmltexi).           Il mio bel manuale di Sgmltexi
* Introduzione: (sgmltexi)Intro 1.  Introduzione al sistema Sgmltexi
</infodir>
```

## Pagine iniziali

«

L’elemento ‘**titlepage**’ viene utilizzato per circoscrivere le informazioni che appaiono nelle primissime pagine del documento. L’ordine degli elementi contenuti è importante e gli errori vengono segnalati dal sistema di analisi SGML.

- ‘**title**’

L’elemento ‘**title**’ serve a contenere il titolo del documento nella sua forma stampata. Si traduce in Texinfo nel comando ‘**@title**’. Il suo utilizzo è molto semplice, come si vede dall’esempio seguente:

```
<title>Sgmltexi</title>
```

- ‘**subtitle**’

Questo elemento permette l'indicazione di un sottotitolo. Non è obbligatorio e può essere usato più volte per indicare più sottotitoli successivi.

```
<subtitle>An alternate way to write Texinfo documentation</subtitle>
```

- **'abstract'**

L'elemento **'abstract'** è facoltativo e si può usare una volta sola. Serve a racchiudere dei blocchi di testo, per esempio elementi **'p'**, che descrivono in breve il contenuto del documento. Il contenuto di questo elemento viene utilizzato nella composizione Info, inserendolo nella parte iniziale del nodo **'top'**.

```
<abstract>
  <p>Sgmltexi is an SGML system (DTD and tools) to
  make Texinfo documentation using SGML...</p>
  ...
  <p>...</p>
</abstract>
```

- **'author'**

Questo elemento, che deve essere indicato almeno una volta e può ripetersi a piacere, serve a contenere il nominativo di uno degli autori del documento. In Texinfo si traduce nel comando **'@author'**.

```
<author>Tizio Tizi &lt;tizio@dinkel.brot.dg&gt;</author>
<author>Caio Cai &lt;caio@dinkel.brot.dg&gt;</author>
```

L'esempio mostra anche l'inclusione dell'indirizzo di posta elettronica, che comunque non sarebbe necessario.

- **'frontcovertext'**

Questo elemento facoltativo, permette di inserire dei blocchi di testo all'interno della copertina.

- **'tpextra'**

Questo elemento facoltativo, può essere usato in diverse situazioni all'interno delle pagine iniziali. Il suo scopo è quello di delimitare dei blocchi di testo che non hanno trovato una classificazione specifica.

Per la precisione, questo elemento può apparire subito prima e subito dopo dell'elemento **'legal'**, inoltre, se viene usato l'elemento **'dedications'**, può essere aggiunto subito dopo di questo.

- **'legal'**

L'elemento **'legal'** si articola a sua volta in altri elementi più dettagliati, allo scopo di descrivere tutto ciò che rappresenta gli aspetti legali del documento: il copyright, la nota sui diritti (concessi o esclusi), oltre ad altre informazioni amministrative legate all'edizione.

- **'copyright'**

Questo elemento serve a contenere l'indicazione relativa ai diritti di autore. Se nel tempo si sono succeduti diversi proprietari, l'elemento **'copyright'** può essere indicato più volte, in base alla necessità (in base a quanto concordato). Si osservi l'esempio seguente:

```
<copyright>Copyright &copy; 1987-1999 Tizio Tizi</copyright>  
<copyright>Copyright &copy; 2000 Caio Cai</copyright>
```

- **'publishnote'**

L'elemento **'publishnote'**, facoltativo, permette l'inclusione di blocchi di testo il cui scopo è quello di inserire informazioni relative alla pubblicazione. Si può usare in modo simile a quanto si vede nell'esempio seguente:

```
<publishnote>
  <p>Published by...</p>
  <p>...</p>
</publishnote>
```

## – ‘**license**’

L’elemento ‘**license**’ è fatto per contenere blocchi di testo che descrivono le condizioni con le quali è rilasciato il documento, che solitamente si rifanno a una licenza allegata da qualche parte (eventualmente in un’appendice).

```
<license>
  <p>Permission is granted to copy, distribute and/or
  modify this document under the terms of the GNU Free
  Documentation License, Version 1.1 or any later version
  published by the Free Software Foundation; with no
  Invariant Sections, with no Front-Cover Texts, and with
  no Back-Cover Texts. A copy of the license is included
  in the section entitled "GNU Free Documentation
  License".</p>
</license>
```

## – ‘**coverart**’

L’elemento ‘**coverart**’, facoltativo, consente di scrivere una nota su chi sia l’ideatore della copertina. In generale, se si usa Sgmltexi non ha senso preoccuparsi di una cosa del genere, dal momento che tutto viene guidato dallo schema SGML del DTD. Tuttavia, esiste la possibilità di fare questa annotazione ugualmente.

```
<coverart>
  <p>Cover art by ...</p>
</coverart>
```

L’elemento ‘**legal**’ può essere usato anche in modo più semplice, se la struttura prevista non soddisfa le esigenze reali. In pratica, al posto degli elementi appena descritti, può contenere dei semplici blocchi di testo, come nell’esempio seguente:

```

<legal>
  <p>Copyright &copy; 2000 ...</p>

  <p>Published by...</p>

  <p>Permission is granted to make and distribute
  verbatim copies of this manual...</p>

  <p>Cover art by ...</p>
</legal>

```

- **‘dedications’**

Dopo l’elemento **‘legal’**, l’elemento **‘dedications’** consente di elencare le dediche del documento. Queste appaiono esclusivamente nella composizione stampata, in una pagina apposita. L’elemento **‘dedications’** è predisposto per l’inserimento di blocchi di testo di qualunque genere.

```

<dedications>
  <flushright>Ad Anna,<br>la mia amata.</flushright>
</dedications>

```

## Indice generale



Dopo l’elemento **‘titlepage’** è possibile collocare uno o più indici generali, più o meno dettagliati.

- **‘contents’**

L’elemento **‘content’**, vuoto, richiede l’inserimento di un indice generale dettagliato. Si traduce in pratica nel comando **‘@content’** di Texinfo.

- **‘shortcontents’, ‘summarycontents’**

Questi due elementi, vuoti, servono a includere rispettivamente i comandi **‘@shortcontent’** e **‘@summarycontent’** di Texinfo. Lo scopo è quello di ottenere un tipo di indice generale ridotto. Se

si usa questo tipo di indice, si include solo uno dei due elementi in questione.

## Nodi e menù Info iniziale

In mancanza di indicazioni, Sgmltexi gestisce da solo i collegamenti riferiti al nodo **‘Top’**, oltre a un menù unico per Info, collocato nello stesso nodo iniziale.

Volendo è possibile dichiarare espressamente il nodo **‘Top’**, attraverso l’elemento **‘topnode’**, che si usa vuoto con tre eventuali attributi: **‘next’**, **‘prev’** e **‘up’**. L’elemento **‘topnode’** si colloca, eventualmente, subito dopo gli indici generali.

```
<topnode next="intro" prev="Top" up="(dir)">
```

Dopo l’elemento **‘topnode’**, è possibile specificare il menù iniziale in modo dettagliato, attraverso l’elemento **‘menu’**. L’esempio seguente mostra un caso abbastanza articolato, benché abbreviato, in cui si vede anche l’inclusione dell’elemento **‘detailmenu’**:

```

<menu>
* Copying::                Your rights.
* Overview::              Texinfo in brief.
...
* Structuring::          How to create chapters, sections, subsections,
                        appendices, and other parts.
* Nodes::                How to write nodes.
...

<detailmenu>

  --- The Detailed Node Listing ---

Overview of Texinfo

* Reporting Bugs::        Submitting effective bug reports.
* Using Texinfo::        Create printed or online output.
* Info Files::           What is an Info file?
...
</detailmenu>
</menu>

```

Naturalmente, non si tratta di elementi indispensabili, ma solo utili se si desidera avere il controllo della gestione dei nodi del documento che si ottiene.

## Introduzione



Dopo l'elemento **'head'** ci può essere l'elemento **'intro'**, il cui scopo è quello di definire uno spazio in cui i capitoli assumono il ruolo di sezioni introduttive, non numerate. Nell'ambito di questo spazio, i «capitoli» sono delimitati nello stesso modo utilizzato nel corpo del documento (l'elemento **'body'**) e nelle appendici (l'elemento **'appendix'**).

```
<intro>
<h1>Introduction to Sgmltexi</h1>

<p>Sgmltexi is a DTD with tools to get Texinfo...</p>

<p>Sgmltexi manage Texinfo nodes automatically,...</p>

</intro>
```

## Corpo

Il corpo del documento è contenuto nell'elemento '**body**', che si colloca dopo l'elemento '**head**' e dopo l'elemento '**intro**' eventuale.

Il corpo può essere suddiviso in capitoli, oppure in parti, o anche in tomi, a seconda della dimensione del progetto di documentazione che si intende avviare. Lo spazio del tomo, della parte, del capitolo, o di una classificazione inferiore, non è delimitato esplicitamente, in quanto appare soltanto la dichiarazione del titolo, all'interno di un elemento che cambia a seconda del livello gerarchico. In pratica, il titolo di un tomo è racchiuso nell'elemento '**tomeheading**', mentre quello di una parte è inserito nell'elemento '**partheading**'.

I capitoli e le classificazioni inferiori hanno titoli delimitati da elementi analoghi a quelli dell'HTML: '**h1**', '**h2**', '**h3**' e '**h4**'. Questa classificazione, a partire da '**h1**' in giù, riguarda nello stesso modo l'introduzione e l'appendice.

```
<body>
<parthead>Networking</parthead>

<h1>IP protocol history</h1>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<h2>ISO-OSI model</h2>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<h1>IPv4 and IPv6</h1>

<p>Bla bla bla...</p>

<p>...</p>

</body>
```

Ogni elemento che racchiude un titolo consente l’inserimento dell’attributo ‘**id**’, il cui scopo è quello di definire una stringa di identificazione, da usare come obiettivo per i riferimenti incrociati.

```
<h1 id="ip history">IP protocol history</h1>
```

È importante rammentare che, a causa di una limitazione progettuale di Texinfo, queste etichette per i riferimenti ipertestuali non possono contenere la virgola.

Ogni elemento che racchiude un titolo consente l’inserimento degli attributi ‘**node**’ e ‘**menu**’, con i quali è possibile stabilire il nome del nodo relativo e la descrizione che deve apparire nel menù (purché questo sia generato automaticamente). In mancanza di queste indi-

cazioni, vengono generati dei nomi in modo automatico, mentre si usa il titolo come descrizione del nodo.

```
<h1 node="IPv4" menu="La storia del protocollo IP">Storia di IPv4</h1>
```

Ogni elemento che racchiude un titolo consente l'inserimento dell'attributo **'numbered'**, a cui si possono assegnare esclusivamente le parole chiave **'on'** oppure **'off'**. In condizioni normali, l'attributo contiene la parola chiave **'on'**, che implica la numerazione dei titoli, salvo il caso dell'introduzione. Assegnando esplicitamente la parola chiave **'off'** si ottiene un titolo non numerato in un contesto che non lo prevederebbe.

```
<h1 numbered="off">Riconoscimenti</h1>
```

Ogni elemento che racchiude un titolo consente l'inserimento degli attributi **'next'**, **'prev'** e **'up'**. Con questi si può alterare la catena di scorrimento dei nodi, specificandoli manualmente. In generale dovrebbe essere preferibile lasciare fare a Sgmltexi.

## Appendice

Dopo il corpo del documento, delimitato dall'elemento **'body'**, può apparire l'appendice, contenuta nell'elemento **'appendix'**. Al suo interno si possono inserire dei «capitoli», introdotti da un titolo contenuto in un elemento **'h1'**, che vengono trattati correttamente come appendici. Dopo i titoli delimitati da **'h1'**, sono ammissibili naturalmente anche segmenti di livello inferiore.

```

<appendix>
<h1>GNU Free Documentation License</h1>

<p indent="off"><strong>GNU Free Documentation License</strong></p>

<p indent="off">Version 1.1, March 2000</p>

<format>
Copyright &copy; 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
</format>
...
...
</appendix>

```

## Indici analitici



Dopo il corpo e dopo il blocco delle appendici, è possibile inserire uno o più indici analitici. Questi si dichiarano con un titolo, attraverso l'elemento '**indexheading**' e con il riferimento al tipo di indice che si vuole esattamente, con l'elemento vuoto '**printindex**'. Si osservi l'esempio seguente in cui si inseriscono due indici: quello delle funzioni (la sigla '**fn**') e quello standard (la sigla '**cp**').

```

<indexheading>Index of functions</indexheading>
<printindex name="fn">
<indexheading>Concept index</indexheading>
<printindex name="cp">

```

Come si vede dall'esempio, l'elemento '**printindex**' ha l'attributo '**name**', a cui si assegna la sigla corrispondente all'indice che si vuole inserire.

## Scomposizione del documento, nodi e menù Info

Per scrivere della documentazione di qualità, secondo i canoni di Texinfo, è necessario gestire direttamente i nodi e i menù. Con Sgmltexi si possono dimenticare i nodi e i menù, ma il risultato in formato Info potrebbe soffrirne. Tuttavia, come in parte è già stato mostrato, è possibile scegliere diversi livelli di automatismo in questa gestione.

Gli elementi usati per delimitare le intestazioni, da ‘h1’ a ‘h4’, possono incorporare gli attributi ‘node’ e ‘menu’. Ciò prevale sulla determinazione automatica relativa. Si osservi l’esempio:

```
<h1 id="ip history" node="history" menu="History of IP protocol">  
IP protocol history</h1>
```

In questo caso, si ottiene l’inserimento della riga seguente nel menù relativo:

```
* history::          History of IP protocol
```

I due attributi, ‘node’ e ‘menu’, possono essere usati in modo indipendente: l’attributo che non viene usato, viene sostituito in modo automatico.

Avendo accesso ai nodi, è possibile farvi riferimento per dei riferimenti incrociati, senza bisogno di usare l’attributo ‘id’.

Come già descritto in precedenza, Sgmltexi crea automaticamente il nodo ‘**Top**’ iniziale. Il menù relativo può essere definito esplicitamente e in tal caso tutti i nodi e tutte le descrizioni relative devono essere inseriti manualmente.

Inserendo l'elemento **'menu'** alla fine del testo di un capitolo, o di una sezione inferiore, si ottiene l'aggiunta di un menù Info in corrispondenza di quel punto. Si osservi l'esempio:

```
<h1>IP protocol history</h1>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<menu>

<h2>ISO-OSI model</h2>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<h2>More information</h2>

<p>Bla bla bla...</p>

<p>...</p>
```

In questo caso, si ottiene l'inserzione di un menù, gestito automaticamente, prima delle sezioni di livello **'h2'**. Volendo, si può indicare il menù in modo preciso, come si vede di seguito:

```
<menu>
* IP layer::          IP ISO-OSI layer model
* more on IP::       More details on IP
</menu>
```

Quando un menù viene descritto in questo modo, i nomi dei nodi devono essere identici a quelli dichiarati negli elementi delle intestazioni. In pratica, scrivendo un menù in modo manuale, anche i nodi devono essere dichiarati esattamente, come si vede qui:

```

<h1>IP protocol history</h1>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<menu>
* IP layer::          IP ISO-OSI layer model
* more on IP::       More details on IP
</menu>

<h2 node="IP layer">ISO-OSI model</h2>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<h2 node="more on IP">More information</h2>

<p>Bla bla bla...</p>

<p>...</p>

```

È evidente, in questa situazione, che l'attributo **'menu'**, il cui scopo sarebbe quello di controllare la descrizione del nodo nel menù, non può essere preso in considerazione in questo caso.

## Numerazione o meno dei titoli

Texinfo consente di inserire dei titoli riferiti a capitoli o sezioni inferiori, con o senza numerazione. Inoltre, consente anche di dichiarare dei titoli che non devono apparire nell'indice generale. Per controllare questa possibilità con Sgmltexi, si può utilizzare l'attributo **'type'** che riguarda tutti gli elementi **'h $n$ '**:

```

<h $n$  type="{numbered | unnumbered | heading}">titolo</h $n$ >

```

In mancanza dell'indicazione dell'attributo, è come se gli fosse stata assegnata la parola chiave **'numbered'**, con la quale i titoli del corpo e delle appendici sono numerati (con numeri o lettere rispettivamente). Utilizzando la parola chiave **'numbered'** si ottiene l'inserimento di un titolo non numerato (nel caso dell'introduzione è sempre senza numerazione); con la parola chiave **'heading'** si ottiene un titolo non numerato e anche non segnalato nell'indice generale (in questo senso può essere utile anche nell'introduzione).

## Codifica

«

Sgmltexi ha una gestione incompleta per le codifiche ISO 8859-*n*. È incompleta perché Texinfo non è in grado di riprodurre tutti i caratteri. Ci sono due modi per definire l'uso di una codifica particolare con Sgmltexi: l'opzione **'--input-encoding'** e l'attributo **'charset'** all'interno dell'elemento **'sgmltexi'**.

La scelta genera risultati differenti. L'opzione **'--input-encoding'** genera una trasformazione dei caratteri in entità SGML, che successivamente sono tradotte in codice Texinfo. In questo modo, il codice Texinfo che si ottiene è sicuramente in ASCII puro (ISO 646), dove le entità che non hanno alcuna corrispondenza in Texinfo. vengono mostrate come **'[ETH ]'**, tanto per fare un esempio. L'uso dell'attributo **'charset'** si traduce semplicemente nel comando **'@documentencoding'**; in certe situazioni, il risultato della composizione può essere buono o meno. A seconda del risultato migliore che si riesce a ottenere, si può scegliere un modo invece dell'altro.

Una buona strategia può essere l'uso dell'attributo **'charset'** in ogni caso, aggiungendo l'opzione **'--input-encoding'** quando

Texinfo non genera una composizione piacevole (di solito quando si genera un formato per la stampa).

## Entità standard e non standard

Il DTD di Sgmltexi include tutte le entità standard ISO 8879. Tuttavia, non tutte le entità sono gestibili da Texinfo; pertanto, quando si usa un'entità non gestibile, viene mostrata nella composizione finale come racchiusa tra parentesi quadre, per esempio come '[ETH ]'.

Sgmltexi mette a disposizione qualche entità non standard, necessaria per mantenere la compatibilità con Texinfo. Queste entità speciali sono elencate nella tabella u90.41.

Tabella u90.41. Entità non standard.

Macro SGML	Comando Texinfo	Descrizione
<code>&amp;dots;</code>	<code>@dots{ }</code>	Tre puntini.
<code>&amp;enddots;</code>	<code>@enddots{ }</code>	Quattro puntini.
<code>&amp;TeX;</code>	<code>@TeX{ }</code>	Il nome «TeX»
<code>&amp;result;</code>	<code>@result{ }</code>	
<code>&amp;expansion;</code>	<code>@expansion{ }</code>	
<code>&amp;print;</code>	<code>@print{ }</code>	
<code>&amp;error;</code>	<code>@error{ }</code>	
<code>&amp;point;</code>	<code>@point{ }</code>	

Macro SGML	Comando Texinfo	Descrizione
<code>&amp;today;</code>	<code>@today{ }</code>	
<code>&amp;esexcl;</code>	<code>@!</code>	Punto esclamativo alla fine di una frase.
<code>&amp;esperiod;</code>	<code>@.</code>	Punto fermo alla fine di una frase.
<code>&amp;nes;</code>	<code>@:</code>	Frase che non si conclude.
<code>&amp;esquest;</code>	<code>@?</code>	Punto interrogativo alla fine di una frase.

# Sgmltexi: contenuti



Paragrafi .....	1458
Indici e riferimenti incrociati .....	1458
Delimitazione di parole e di frasi .....	1463
Delimitazione di blocchi di testo .....	1465
Elenchi e tabelle .....	1468
Inserzioni .....	1472
Definizioni .....	1473
Codice condizionato e codice letterale in base alla composizione 1478	
Problemi .....	1481

Dopo la struttura generale, il sorgente Sgmltexi si articola generalmente in elementi che possono essere classificati sommariamente in blocchi e in testo interno a un blocco. Nei DTD comuni si utilizzano frequentemente le entità parametriche ‘%**block;**’ e ‘%**inline;**’, per definire questi due grandi raggruppamenti. Nel DTD di Sgmltexi si usa la stessa convenzione e in questo senso vanno interpretate tali sigle nelle tabelle riassuntive.

A titolo di esempio, un blocco è qualcosa di simile a un paragrafo, un elenco, una tabella; un elemento interno alla riga è fatto per contenere del testo, eventualmente assieme a delle enfatizzazioni di qualche genere. Di solito, anche se questo fatto non può valere in generale, un elemento interno alla riga è fatto per contenere testo

o altri elementi dello stesso genere; al contrario, un elemento che costituisce un blocco, può contenere altri blocchi, oppure del testo interno alla riga.

Il DTD di Sgmltexi non prevede elementi che possano contenere testo interno alla riga o blocchi a scelta, come accade invece nell'HTML.

## Paragrafi

«

I blocchi di testo più comuni sono dei paragrafi, delimitati dall'elemento '**p**', il quale può apparire con un rientro iniziale o meno, a seconda dell'uso dell'attributo '**indent**'. I paragrafi, compresi quelli centrati che si ottengono con l'elemento '**center**', contengono testo o altri elementi interni alla riga.

Tabella u91.1. Paragrafi con Sgmltexi.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
p	%inline;	Blocco di testo elementare, riconducibile al paragrafo.
<i>indent</i>	Attributo	Rientro prima riga: ' <b>on</b> ', ' <b>off</b> '.
center	%inline;	Blocco di testo centrato: '@ <b>center</b> '.

## Indici e riferimenti incrociati

«

Sgmltexi mette a disposizione diversi elementi il cui scopo è quello di permettere delle inserzioni per generare degli indici o

dei riferimenti incrociati, riproducendo i comandi equivalenti di Texinfo.

Le voci degli indici analitici vengono inserite attraverso un gruppo di elementi vuoti: ‘**cindex**’, ‘**findex**’, ‘**vindex**’, ‘**kindex**’, ‘**pindex**’, ‘**tindex**’ e ‘**userindex**’. Tutti questi elementi hanno lo stesso attributo ‘**entry**’, che serve a specificare la voce da inserire nell’indice relativo. In particolare, l’elemento ‘**userindex**’ ha in più l’attributo ‘**name**’ per specificare l’indice al quale si vuole fare riferimento.

Questi elementi possono essere usati solo dopo la dichiarazione di una sezione (un titolo di qualunque tipo, dal tomo in giù), ma prima del testo normale che ne seguirebbe. Per esempio così:

```
<h1>IP protocol history</h1>
<cindex entry="IP protocol">
<cindex entry="history">

<p>Bla bla bla...</p>
```

La tabella u91.3 riassume brevemente l’uso di questi elementi.

Tabella u91.3. Voci degli indici analitici.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
<b>cindex</b>	Vuoto	Voce dell’indice analitico normale.
<i>entry</i>	Attributo	Voce da inserire nell’indice.
<b>findex</b>	Vuoto	Voce dell’indice analitico delle funzioni.
<i>entry</i>	Attributo	Voce da inserire nell’indice.
<b>vindex</b>	Vuoto	Voce dell’indice analitico delle variabili.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
<i>entry</i>	Attributo	Voce da inserire nell'indice.
kindex	Vuoto	Voce dell'indice analitico dei tasti premuti.
<i>entry</i>	Attributo	Voce da inserire nell'indice.
pindex	Vuoto	Voce dell'indice analitico dei programmi.
<i>entry</i>	Attributo	Voce da inserire nell'indice.
tindex	Vuoto	Voce dell'indice analitico dei tipi di dati.
<i>entry</i>	Attributo	Voce da inserire nell'indice.
userindex	Vuoto	Voce di un indice analitico definito dall'utilizzatore.
<i>entry</i>	Attributo	Voce da inserire nell'indice.
<i>name</i>	Attributo	Sigla identificativa dell'indice definito dall'utente.
printindex	Vuoto	Inserisce l'elenco delle voci dell'indice specificato.
<i>name</i>	Attributo	Sigla identificativa dell'indice.

Ogni indice analitico si distingue in base a una sigla di due lettere. Gli indici analitici già previsti da Texinfo hanno una sigla fissa, mentre tutte le altre combinazioni possono essere usate per gli indici stabiliti dall'utilizzatore. La tabella u91.4 riassume le sigle degli indici standard, la cui conoscenza è necessaria per poter usare correttamente l'elemento '**printindex**' allo scopo di riprodurre l'elenco

dell'indice relativo.

Tabella u91.4. Sigle identificative degli indici analitici standard.

Sigla	Descrizione
cp	Indice analitico normale.
ky	Indice analitico dell'uso della tastiera.
pg	Indice analitico dei programmi.
fn	Indice analitico delle funzioni.
vr	Indice analitico delle variabili.
tp	Indice analitico dei tipi di dati.

Gli elementi utilizzati per realizzare dei riferimenti incrociati sono vuoti e sono sempre interni alla riga di testo. Tutte le informazioni necessarie sono passate attraverso attributi. Dal momento che questi elementi rispecchiano fedelmente i comandi equivalenti di Texinfo, viene mostrata solo la tabella u91.5, senza entrare nel dettaglio del significato di ognuno di loro.

Tabella u91.5. Riferimenti incrociati.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
anchor	Vuoto	Comando '@anchor' di Texinfo.
<i>id</i>	Attributo	Stringa di identificazione dell'ancora.
xref	Vuoto	Comando '@xref' di Texinfo.
<i>id</i>	Attributo	Nodo o ancora a cui si fa riferimento.
<i>name</i>	Attributo	Nome del riferimento.
<i>title</i>	Attributo	Titolo o argomento a cui si fa riferimento.
<i>info</i>	Attributo	Nome del file Info.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
<i>ptitle</i>	Attributo	Titolo dell'edizione stampata.
ref	Vuoto	Comando '@ref' di Texinfo.
<i>id</i>	Attributo	Nodo o ancora a cui si fa riferimento.
<i>name</i>	Attributo	Nome del riferimento.
<i>title</i>	Attributo	Titolo o argomento a cui si fa riferimento.
<i>info</i>	Attributo	Nome del file Info.
<i>ptitle</i>	Attributo	Titolo dell'edizione stampata.
pxref	Vuoto	Comando '@pxref' di Texinfo.
<i>id</i>	Attributo	Nodo o ancora a cui si fa riferimento.
<i>name</i>	Attributo	Nome del riferimento.
<i>title</i>	Attributo	Titolo o argomento a cui si fa riferimento.
<i>info</i>	Attributo	Nome del file Info.
<i>ptitle</i>	Attributo	Titolo dell'edizione stampata.
inforef	Vuoto	Comando '@inforef' di Texinfo.
<i>id</i>	Attributo	Nodo o ancora a cui si fa riferimento.
<i>name</i>	Attributo	Nome del riferimento.
<i>info</i>	Attributo	Nome del file Info.
uref	Vuoto	Comando '@uref' di Texinfo.
<i>uri</i>	Attributo	Indirizzo URI a cui si fa riferimento.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
<i>name</i>	Attributo	Nome del riferimento.
<i>replace</i>	Attributo	Testo di rimpiazzo da mostrare.
email	Vuoto	Comando '@email' di Texinfo.
<i>email</i>	Attributo	Indirizzo di posta elettronica.
<i>name</i>	Attributo	Titolo o descrizione dell'indirizzo.

In particolare, è opportuno osservare che l'attributo 'id' degli elementi 'hn', 'partheadng' e 'tomeheadng', è un'ancora a cui possono puntare tutti i vari tipi di riferimenti incrociati disponibili (tranne 'uref' e 'email' che puntano a degli URI).

L'esempio seguente mostra come usare l'elemento 'pxref' in modo molto semplice:

```
<p>Sgmltexi crea automaticamente il nodo Top. Come già spiegato in precedenza, (<pxref id="top node menu">), il menù può essere...</p>
```

## Delimitazione di parole e di frasi

Un certo numero di elementi serve a delimitare parole o frasi, per qualche motivo. Il DTD di Sgmltexi è molto permissivo, in modo tale che ogni elemento di questi può contenere qualunque altro elemento interno alla riga di testo. Ciò è stato fatto per assicurare la massima compatibilità con Texinfo, ma in futuro potrebbero essere poste delle piccole limitazioni.

La tabella u91.7 elenca questi elementi, assieme a 'kbdinputstyle', che si usa per specificare lo stile di

rappresentazione del contenuto dell'elemento **'kbd'**.

Tabella u91.7. Delimitazione di parole e frasi.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
code	<code>%inline;</code>	Comando <code>'@code'</code> di Texinfo.
kbd	<code>%inline;</code>	Comando <code>'@kbd'</code> di Texinfo.
kbdinputstyle	Vuoto	Comando <code>'@kbdinputstyle'</code> di Texinfo.
<i>style</i>	Attributo	Stile: <code>'code'</code> , <code>'example'</code> , <code>'distinct'</code> .
key	<code>%inline;</code>	Comando <code>'@key'</code> di Texinfo.
samp	<code>%inline;</code>	Comando <code>'@samp'</code> di Texinfo.
var	<code>%inline;</code>	Comando <code>'@var'</code> di Texinfo.
env	<code>%inline;</code>	Comando <code>'@env'</code> di Texinfo.
file	<code>%inline;</code>	Comando <code>'@file'</code> di Texinfo.
command	<code>%inline;</code>	Comando <code>'@command'</code> di Texinfo.
option	<code>%inline;</code>	Comando <code>'@option'</code> di Texinfo.
dfn	<code>%inline;</code>	Comando <code>'@dfn'</code> di Texinfo.
cite	<code>%inline;</code>	Comando <code>'@cite'</code> di Texinfo.
acronym	<code>%inline;</code>	Comando <code>'@acronym'</code> di Texinfo.
url	<code>%inline;</code>	Comando <code>'@url'</code> di Texinfo.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
emph	%inline;	Comando '@emph' di Texinfo.
strong	%inline;	Comando '@strong' di Texinfo.
sc	%inline;	Comando '@sc' di Texinfo.
roman	%inline;	Comando '@r' di Texinfo.
italic	%inline;	Comando '@i' di Texinfo.
bold	%inline;	Comando '@b' di Texinfo.
typewriter	%inline;	Comando '@t' di Texinfo.

Viene mostrato un esempio molto semplice dell'uso dell'elemento '**strong**':

```
<p><strong>Pinco Pallino</strong> è un uomo molto vecchio...</p>

<p><strong>Tizio Tizi</strong> ha studiato tecnologia delle
comunicazioni...</p>
```

## Delimitazione di blocchi di testo

Alcuni elementi servono a delimitare blocchi di testo, o un tipo particolare di testo interno alle righe. Il DTD di Sgmltexi è molto permissivo per assicurare la massima compatibilità con Texinfo, ma in futuro potrebbero essere poste delle piccole limitazioni.

La tabella u91.9 elenca questi elementi, assieme a '**pre**', che permette di inserire del testo preformattato, e a '**exdent**', utilizzato all'interno di '**pre**' per ottenere delle righe che sporgono verso l'esterno.

Tabella u91.9. Delimitazione di blocchi di testo.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
exdent	%inline;	Comando '@exdent' di Texinfo.
pre	%inline;	Testo preformattato.
quotation	%block;	Comando '@quotation' di Texinfo.
display	%block; o 'pre'	Comando '@display' di Texinfo.
smalldisplay	%block; o 'pre'	Comando '@smalldisplay' di Texinfo.
example	%block; o 'pre'	Comando '@example' di Texinfo.
smallexample	%block; o 'pre'	Comando '@smallexample' di Texinfo.
flushleft	%inline;	Comando '@flushleft' di Texinfo.
flushright	%inline;	Comando '@flushright' di Texinfo.
lisp	%block; o 'pre'	Comando '@lisp' di Texinfo.
smalllisp	%block; o 'pre'	Comando '@smalllisp' di Texinfo.
cartouche	%block; o 'pre'	Comando '@cartouche' di Texinfo.
format	%block; o 'pre'	Comando '@format' di Texinfo.
smallformat	%block; o 'pre'	Comando '@smallformat' di Texinfo.
texinfo		Codice Texinfo incorporato.

In generale, l'uso di questi elementi è molto semplice, come si può vedere in questo caso:

```
<example>
<p>Bla bla bla...</p>
<p>Bla bla bla...</p>
</object>
```

L'esempio seguente, invece, mostra l'uso dell'elemento **'pre'**, allo scopo di incorporare del testo preformattato, pur continuando a espandere le macro SGML:

```
<example>
<pre>
#!/usr/bin/perl
while ($line = &gt;STDIN&gt;)
{
    chomp $line;
    print ("$line\r\n");
}
</pre>
</object>
```

In aggiunta, si può delimitare il contenuto dell'elemento **'pre'** per poterlo scrivere in modo letterale:

```
<example>
<pre>
<![CDATA[
#!/usr/bin/perl
while ($line = <STDIN>)
{
    chomp $line;
    print ("$line\r\n");
}
]]>
</pre>
</example>
```

## Elenchi e tabelle



Elenchi e tabelle, sono blocchi di testo. La gestione di Texinfo per ciò che riguarda queste strutture, è abbastanza speciale. Qui viene riassunto tutto nella tabella u91.13, che però richiede la conoscenza dei comandi di Texinfo corrispondenti.

Tabella u91.13. Elenchi e tabelle.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
itemize	' <b>item</b> ', ' <b>itemx</b> ', %block;	Comando '@ <b>itemize</b> ' di Texinfo.
<i>mark</i>	Attributo	Segno usato al posto del pallino iniziale.
enumerate	' <b>item</b> ', ' <b>itemx</b> ', %block;	Comando '@ <b>enumerate</b> ' di Texinfo.
<i>start</i>	Attributo	Valore iniziale dell'elenco numerato.
table	' <b>item</b> ', ' <b>itemx</b> ', %block;	Comando '@ <b>table</b> ' di Texinfo.
<i>emphasis</i>	Attributo	Enfasi della colonna descrittiva: ' <b>asis</b> ', ' <b>code</b> ', ' <b>samp</b> ', ' <b>var</b> ', ' <b>kbd</b> ', ' <b>file</b> '.
vtable	' <b>item</b> ', ' <b>itemx</b> ', %block;	Comando '@ <b>vtable</b> ' di Texinfo.
<i>emphasis</i>	Attributo	Enfasi della colonna delle variabili: ' <b>asis</b> ', ' <b>code</b> ', ' <b>samp</b> ', ' <b>var</b> ', ' <b>kbd</b> ', ' <b>file</b> '.
ftable	' <b>item</b> ', ' <b>itemx</b> ', %block;	Comando '@ <b>ftable</b> ' di Texinfo.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
<i>emphasis</i>	Attributo	Enfasi della colonna delle funzioni: ‘asis’, ‘code’, ‘samp’, ‘var’, ‘kbd’, ‘file’.
item	%inline; o vuoto.	Comando ‘@item’ di Texinfo.
itemx	%inline; o vuoto.	Comando ‘@itemx’ di Texinfo.
multitable		Comando ‘@multitable’ di Texinfo.
columnfraction	‘.n’.	Colonna larga 0,n volte lo spazio totale.
columnexample	Testo puro.	Colonna larga tanto quanto l’esempio.
raw	%inline;, ‘tab’.	Riga di una tabella.
tab	Vuoto	Separatore tra una colonna e la successiva.

Vengono mostrati alcuni esempi, a cominciare da un elenco non numerato:

```
<itemize mark="#">
<item>
  <p>Primo elemento dell'elenco.</p>
<item>
  <p>Secondo elemento.</p>
</itemize>
```

In questo caso, si ottiene un elenco puntato di due sole voci, dove al posto del pallino usuale, appare il simbolo ‘#’. Sostituendo l’elemento ‘**itemize**’ con ‘**enumerate**’, si ottiene un elenco numerato:

```
<enumerate start="3">
<item>
  <p>Primo elemento dell'elenco.</p>
<item>
  <p>Secondo elemento.</p>
</enumerate>
```

In questo caso, si fa in modo che il primo dei due elementi abbia il numero tre. L'elenco descrittivo si ottiene attraverso l'elemento **'table'**, dove gli elementi **'item'** contengono le voci relative. Si osservi l'esempio:

```
<table emphasis="code">
<item>ls</item>
<itemx>dir</itemx>
  <p>Elenco del contenuto della directory.</p>
<item>cd</item>
  <p>Cambia directory.</p>
</table>
```

Si intende così che l'elemento **'itemx'** serve quando un elemento dell'elenco è composto da più di una voce.

Le tabelle, intese come quelle a cui si è abituati di solito, sono gestite attraverso l'elemento **'multitable'**. Questo, prima dell'indicazione delle righe che compongono la tabella, richiede di specificare quante sono le colonne e quanto larghe devono essere. Per questo, all'inizio occorre utilizzare una serie di elementi **'columnfraction'**, oppure **'columnexample'**, attraverso i quali si specificano proprio queste larghezze (in percentuale o attraverso un testo di esempio). L'esempio seguente mostra il caso di una tabella le cui colonne sono state definite in modo percentuale:

```

<multitable>
<columnfraction>.30</columnfraction>
<columnfraction>.70</columnfraction>
<raw><strong>Parametro LOC</strong>
<tab><strong>Posizione corrispondente</strong>
</raw>
<raw>h
<tab>posizione attuale
</raw>
<raw>t
<tab>superiore
</raw>
<raw>b
<tab>inferiore
</raw>
<raw>p
<tab>pagina
</raw>
</multitable>

```

In alternativa, dato che la larghezza delle colonne dipende proprio dai titoli, si potrebbe fare così:

```

<multitable>
<columnexample>Parametro LOC</columnexample>
<columnexample>Posizione corrispondente</columnexample>
<raw><strong>Parametro LOC</strong>
<tab><strong>Posizione corrispondente</strong>
</raw>
<raw>h
<tab>posizione attuale
</raw>
<raw>t
<tab>superiore
</raw>
<raw>b
<tab>inferiore
</raw>
<raw>p
<tab>pagina
</raw>
</multitable>

```

In entrambi i casi, lo scopo è quello di ottenere uno specchietto si-

mile a quello che segue. Si osservi che non ci sono didascalie e nemmeno esiste la possibilità di collocare dinamicamente la tabella.

Parametro LOC	Posizione corrispondente
h	posizione attuale
t	superiore
b	inferiore
p	pagina

## Inserzioni

«

Alcuni elementi sono difficilmente classificabili in gruppi particolari. Qui, vengono distinti in due raggruppamenti: quelli interni alle righe e quelli che rappresentano dei blocchi. A questi corrispondono le tabelle u91.20 e u91.21.

Tabella u91.20. Inserzioni interne alle righe.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
dmn	#PCDATA	Comando '@dmn' di Texinfo.
math	#PCDATA	Comando '@math' di Texinfo.
footnote	%inline;	Comando '@footnote' di Texinfo.
image	Vuoto	Comando '@image' di Texinfo.
<i>name</i>	Attributo	Nome del file da inserire, senza estensione.
<i>width</i>	Attributo	Ampiezza dell'immagine.
<i>height</i>	Attributo	Altezza dell'immagine.
whole	%inline;	Comando '@w' (previene l'interruzione di riga).
br	Vuoto	Comando '@*' (interruzione di riga).

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
dh	Vuoto	Comando '@-' (separazione facoltativa).
hyphenation	Vuoto;	Comando '@hyphenation' di Texinfo.
<i>words</i>	Attributo	Elenco di parole separate in sillabe.

Tabella u91.21. Inserzione di blocchi.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
sp	Vuoto	Comando '@sp' di Texinfo.
<i>lines</i>	Attributo	Quantità di righe da saltare (un numero intero).
page	Vuoto	Comando '@page' di Texinfo.
group	%block;	Comando '@group' di Texinfo.
need	Vuoto	Comando '@need' di Texinfo.
<i>mils</i>	Attributo	Millesimi di pollice richiesti.

## Definizioni

Texinfo prevede un grande numero di comandi per la descrizione di definizioni di vario genere. Queste «definizioni» vanno intese generalmente come dei modelli sintattici. È un po' difficile comprendere bene quando usare questa o quella forma di definizione; per cui occorre studiare la documentazione di Texinfo.

Tutte le forme di definizione si dichiarano attraverso un elemento provvisto di diversi attributi. Questo elemento contiene generalmente la descrizione del modello, in una serie di blocchi di testo, ma in particolare potrebbe contenere la descrizione degli argomenti, all'interno dell'elemento **'args'**, comune a tutte le definizioni che ne hanno.

Tabella u91.22. Definizioni.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
args	%inline;	Argomenti di una definizione.
deffn	'args', %block;	Comando '@deffn' di Texinfo.
<i>cat</i>	Attributo	Categoria della funzione.
<i>name</i>	Attributo	Nome della funzione.
deffnx	Vuoto	Comando '@deffnx'. Attributi come 'deffn'.
defun	'args', %block;	Comando '@defun' di Texinfo.
<i>name</i>	Attributo	Nome della funzione.
defunx	Vuoto	Comando '@defunx'. Attributi come 'defun'.
defmac	'args', %block;	Comando '@defmac' di Texinfo.
<i>name</i>	Attributo	Nome della macro.
defmacx	Vuoto	Comando '@defmacx'. Attributi come 'defmac'.
defspec	'args', %block;	Comando '@defspec' di Texinfo.
<i>name</i>	Attributo	Nome di uno <i>special form</i> .
defspecx	Vuoto	Comando '@defspecx'. Attributi come 'defspec'.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
defvr	%block;	Comando '@defvr' di Texinfo.
<i>cat</i>	Attributo	Categoria della variabile.
<i>name</i>	Attributo	Nome della variabile.
defvrx	Vuoto	Comando '@defvrx'. Attributi come 'defvr'.
defvar	%block;	Comando '@defvar' di Texinfo.
<i>name</i>	Attributo	Nome della variabile.
defvarx	Vuoto	Comando '@defvarx'. Attributi come 'defvar'.
defopt	%block;	Comando '@defopt' di Texinfo.
<i>name</i>	Attributo	Nome dell'opzione.
defoptx	Vuoto	Comando '@defoptx'. Attributi come 'defopt'.
deftypefn	'args', %block;	Comando '@deftypefn' di Texinfo.
<i>cat</i>	Attributo	Categoria.
<i>type</i>	Attributo	Tipo di dati.
<i>name</i>	Attributo	Nome.
deftypefnx	Vuoto	Comando '@deftypefnx'. Attributi come 'deftypefn'.
deftypefun	'args', %block;	Comando '@deftypefun' di Texinfo.
<i>type</i>	Attributo	Tipo di dati.
<i>name</i>	Attributo	Nome.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
<code>deftypefunx</code>	Vuoto	Comando <code>'@deftypefunx'</code> . Attributi come <code>'deftypefun'</code> .
<code>deftypevr</code>	<code>%block;</code>	Comando <code>'@deftypevr'</code> di Texinfo.
<i>cat</i>	Attributo	Categoria.
<i>type</i>	Attributo	Tipo di dati.
<i>name</i>	Attributo	Nome.
<code>deftypevr<sub>x</sub></code>	Vuoto	Comando <code>'@deftypevr<sub>x</sub>'</code> . Attributi come <code>'deftypevr'</code> .
<code>deftypevar</code>	<code>%block;</code>	Comando <code>'@deftypevar'</code> di Texinfo.
<i>type</i>	Attributo	Tipo di dati.
<i>name</i>	Attributo	Nome.
<code>deftypevar<sub>x</sub></code>	Vuoto	Comando <code>'@deftypevar<sub>x</sub>'</code> . Attributi come <code>'deftypevar'</code> .
<code>args</code>	<code>%inline;</code>	Argomenti di una definizione.
<code>defcv</code>	<code>%block;</code>	Comando <code>'@defcv'</code> di Texinfo.
<i>cat</i>	Attributo	Categoria.
<i>class</i>	Attributo	Classe.
<i>name</i>	Attributo	Nome.
<code>defcv<sub>x</sub></code>	Vuoto	Comando <code>'@defcv<sub>x</sub>'</code> . Attributi come <code>'defcv'</code> .
<code>defivar</code>	<code>%block;</code>	Comando <code>'@defivar'</code> di Texinfo.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
<i>class</i>	Attributo	Classe.
<i>name</i>	Attributo	Nome.
defivarx	Vuoto	Comando '@defivarx'. Attributi come 'defivar'.
deftypeivar	%block;	Comando '@deftypeivar' di Texinfo.
<i>class</i>	Attributo	Classe.
<i>type</i>	Attributo	Tipo.
<i>name</i>	Attributo	Nome.
deftypeivarx	Vuoto	Comando '@deftypeivarx'. Attributi come 'deftypeivar'.
defop	'args', %block;	Comando '@defop' di Texinfo.
<i>cat</i>	Attributo	Categoria.
<i>class</i>	Attributo	Classe.
<i>name</i>	Attributo	Nome.
defopx	Vuoto	Comando '@defopx'. Attributi come 'defop'.
defmethod	'args', %block;	Comando '@defmethod' di Texinfo.
<i>class</i>	Attributo	Classe.
<i>name</i>	Attributo	Nome.
defmethodx	Vuoto	Comando '@defmethodx'. Attributi come 'defmethod'.
deftypemethod	'args', %block;	Comando '@deftypemethod' di Texinfo.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
<i>class</i>	Attributo	Classe.
<i>type</i>	Attributo	Tipo.
<i>name</i>	Attributo	Nome.
deftypemethodx	Vuoto	Comando '@deftypemethodx'. Attributi come 'deftypemethod'.
deftp	'args', %block;	Comando '@deftp' di Texinfo.
<i>cat</i>	Attributo	Categoria.
<i>name</i>	Attributo	Nome.
deftpx	Vuoto	Comando '@deftpx'. Attributi come 'deftp'.

Ecco un esempio molto semplice:

```
<deffn cat="Command" name="sgmltexi">
  <args>[<var>options</var>]... <var>sgml_source</var></args>

  <p>This is the front-end for the SGML to Texinfo system.</p>
</deffn>
```

La composizione in formato Info genera il risultato seguente:

```
- Command: sgmltexi [OPTIONS]... SGML_SOURCE
  This is the front-end for the SGML to Texinfo system.
```

## Codice condizionato e codice letterale in base alla composizione

«

Texinfo ha la possibilità di selezionare del codice in dipendenza del tipo di composizione finale. In SGML si possono fare cose simili attraverso le sezioni marcate, ma non si tratta della stessa cosa. Per

questa ragione, Sgmltexi include alcuni elementi speciali corrispondenti ai comandi che servono a Texinfo per selezionare il codice, consentendo anche di inserire pezzi di codice letterale.

Tabella u91.25. Codice condizionato e codice letterale in base alla composizione.

Elemento o attributo	Contenuto	Descrizione o corrispondenza con Texinfo
ifinfo	<code>%inline;</code>	<code>'@ifinfo'</code> ... <code>'@end ifinfo'</code>
ifinfoblock	<code>%block;</code>	<code>'@ifinfo'</code> ... <code>'@end ifinfo'</code>
iftex	<code>%inline;</code>	<code>'@iftex'</code> ... <code>'@end iftex'</code>
iftexblock	<code>%block;</code>	<code>'@iftex'</code> ... <code>'@end iftex'</code>
ifhtml	<code>%inline;</code>	<code>'@ifhtml'</code> ... <code>'@end ifhtml'</code>
ifhtmlblock	<code>%block;</code>	<code>'@ifhtml'</code> ... <code>'@end ifhtml'</code>
ifnotinfo	<code>%inline;</code>	<code>'@ifnotinfo'</code> ... <code>'@end ifnotinfo'</code>
ifnotinfoblock	<code>%block;</code>	<code>'@ifnotinfo'</code> ... <code>'@end ifnotinfo'</code>
ifnottex	<code>%inline;</code>	<code>'@ifnottex'</code> ... <code>'@end ifnottex'</code>
ifnottexblock	<code>%block;</code>	<code>'@ifnottex'</code> ... <code>'@end ifnottex'</code>
ifnohtml	<code>%inline;</code>	<code>'@ifnohtml'</code> ... <code>'@end ifnohtml'</code>
ifnohtmlblock	<code>%block;</code>	<code>'@ifnohtml'</code> ... <code>'@end ifnohtml'</code>
tex	<code>#PCDATA</code>	<code>'@tex'</code> ... <code>'@end tex'</code>
html	<code>#PCDATA</code>	<code>'@html'</code> ... <code>'@end html'</code>
texinfo	<code>#PCDATA</code>	Codice Texinfo.

È importante osservare che `'ifinfo'`, `'iftex'`, `'ifhtml'`, `'ifnotinfo'`, `'ifnottex'` e `'ifnohtml'`, sono elementi interni alla riga di testo, che contengono lo stesso genere di cosa. Al contrario, `'ifinfoblock'`, `'iftexblock'`, `'ifhtmlblock'`, `'ifnotinfoblock'`, `'ifnottexblock'` e `'ifnohtmlblock'`, sono blocchi che contengono altri blocchi. Questa distinzione è necessaria per evitare problemi nella definizione del documento SGML (nel DTD).

In particolare, gli elementi `'tex'`, `'html'` e `'texinfo'`, sono fatti per contenere testo letterale solitamente racchiuso tra `'<![CDATA['` e `']>'`.

L'elemento `'texinfo'` non ha un comando equivalente in Texinfo, perché rappresenta del codice Texinfo. Si osservi l'esempio seguente:

```
<p>The letter <texinfo>@ubaraccent{o}</texinfo> is a special...</p>
```

Usando questo elemento, potrebbe essere necessario forzare l'interpretazione letterale anche da parte dell'SGML. In tal caso, il contenuto dell'elemento può essere racchiuso come si vede qui:

```
<p>The letter <texinfo><![CDATA[@ubaraccent{o}]]></texinfo> is a...
```

Il caso particolare dell'esempio non mostra una situazione in cui sia indispensabile l'interpretazione SGML letterale, tuttavia questo è il modo quando succede tale circostanza.

Viene mostrato un altro esempio nell'uso di codice letterale specifico per il tipo di composizione. L'intenzione è quella di mostrare un'espressione matematica molto semplice:  $123 + 10^{-1}$ .

```
<p><tex><![CDATA[ $123+10^{-1}$ ]]></tex>  
<html><![CDATA[123+10<sup>-1</sup>]]></html>  
<ifinfo>123+10^-1</ifinfo>  
= 12.3</p>
```

Si potrebbe notare una sorta di incoerenza nell'uso degli elementi letterali, assieme a **'ifinfo'**, il cui scopo è solo quello di essere preso in considerazione quando la composizione produce il formato Info. Il fatto è che gli altri due elementi letterali, oltre che contenere codice letterale per il tipo rispettivo di composizione, sono implicitamente elementi condizionali. Dal momento che la composizione Info non può prevedere una codifica letterale speciale, l'unico modo per integrare le varie parti è quello di usare **'ifinfo'** per rappresentare in qualche modo l'espressione, anche in questo caso.

## Problemi

Texinfo, come TeX e \*roff, distingue i blocchi di testo in quanto separati da una o più righe vuote. In tal modo, la distinzione tra blocchi di testo e testo interno alle righe, è solo una questione di spazio verticale. Per esempio, il pezzo seguente di un sorgente Texinfo, mostra tre ambienti del tipo **'@ifcomposizione'**, che sono parte dello stesso blocco di testo, ovvero lo stesso paragrafo.

```
La composizione attuale è
@iftex
TeX
@end iftex
@ifhtml
HTML
@end ifhtml
@ifinfo
Info
@end ifinfo
e si può vedere che...
```

In una situazione differente, questi ambienti possono diventare blocchi isolati di testo, come si vede qui:

```
La composizione attuale è:

@iftex
TeX
@end iftex

@ifhtml
HTML
@end ifhtml

@ifinfo
Info
@end ifinfo

Si può vedere che...
```

Con un sistema SGML, questa confusione di ruoli non è desiderabile, oltre che essere difficile da realizzare. Questo è il motivo per cui Sgmltexi distingue tra ‘*@ifcomposizione*’ o ‘*@ifnotcomposizione*’, e ‘*@ifcomposizioneblock*’ o ‘*@ifnotcomposizioneblock*’.

Sgmltexi cerca di mantenere le interruzioni di riga contenute all’interno del sorgente SGML, ma per questo ci sono delle conseguenze nell’uso degli ambienti condizionali, del tipo interno alle righe. Ciò dipende dal fatto che necessariamente occorre aggiungere delle in-

terruzioni aggiuntive. Si supponga di voler scrivere qualcosa come ciò che segue:

```
<p>La composizione attuale
è <iftex>TeX</iftex><ifhtml>HTML</ifhtml><ifinfo>Info</ifinfo>, per cui
si sa cosa comporta questo fatto.</p>
```

Ci si aspetta che i marcatori di apertura e di chiusura vengano rimpiazzati aggiungendo anche le interruzioni di riga appropriate. Ma se fosse così, il risultato sarebbe quello seguente, in cui ciò che prima era testo interno alla riga diventa invece un blocco separato:

```
La composizione attuale
è
@iftex
TeX
@end iftex

@ifhtml
HTML
@end ifhtml

@ifinfo
Info
@end ifinfo
, per cui
si sa cosa comporta questo fatto.</p>
```

Per risolvere il problema, questi elementi intesi come ambienti condizionali interni alle righe, non introducono alcuna interruzione iniziale o finale che sia; rimane compito dell'autore il preoccuparsi di questo problema. Per questo, il sorgente di Sgmltexi deve essere scritto come si vede nell'esempio seguente, considerando anche che non c'è alcun modo di mettere la virgola dopo il nome del tipo di composizione.

```
<p>La composizione attuale è  
<iftex>TeX</iftex>  
<ifhtml>HTML</ifhtml>  
<ifinfo>Info</ifinfo>  
per cui si sa cosa comporta questo fatto.</p>
```

Lo stesso problema appare con gli elementi **'tex'** e **'html'**, ma in tal caso non c'è bisogno di qualificarne il contenuto, che si intende sempre come testo interno alle righe.

```
<p>  
<tex>  
$$ \chi^2 = \sum_{i=1}^N  
    \left (y_i - (a + b x_i)  
    \over \sigma_i\right)^2 $$  
</tex>  
</p>
```

Utilizzando un sistema SGML, l'inserzione di codice letterale per il tipo di composizione particolare che si utilizza, è da considerarsi come l'**ultima risorsa**. In altri termini, se sono necessari tali espedienti, è evidente che l'SGML è la scelta sbagliata per scrivere la propria documentazione.

# Corrispondenza tra Texinfo e Sgmltexi



In questo capitolo conclusivo della parte dedicata a Sgmltexi, si riepiloga brevemente l'uso di questo sistema di composizione, attraverso la comparazione con Texinfo. In questo modo, si può comprendere cosa di Texinfo non è disponibile con Sgmltexi.

Si osservi che nei modelli sintattici, le parentesi graffe hanno significato letterale, facendo parte dei comandi di Texinfo.

## @spazio\_bianco

```
&emsp;
```

## @!

```
&esexcl;
```

*End sentence exclamation mark*

## @"x

## @'x

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione '`--input-encoding`', o con l'attributo '`charset`' dell'elemento '`sgmltexi`'.

@\*

```
<br>
```

@, {x}

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione '`--input-encoding`', o con l'attributo '`charset`' dell'elemento '`sgmltexi`'.

@-

```
<dh>
```

@.

```
&esperiod;
```

*End sentence period*

@:

```
&nes;
```

*Not ending sentence*

@=x

Non disponibile.

@?

```
&esquest;
```

*End of sentence question mark*

@@

```
@
```

@^

@`

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione '`--input-encoding`', o con l'attributo '`charset`' dell'elemento '`sgmltexi`'.

@{

```
{
```

@}

```
}
```

@~

@AA{ }

@aa{ }

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione '`--input-encoding`', o con l'attributo '`charset`' dell'elemento '`sgmltexi`'.

@**acronym**{ *abbreviazione* }

```
<acronym>abbreviazione</acronym>
```

**@AE{ }**

**@ae{ }**

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione '`--input-encoding`', o con l'attributo '`charset`' dell'elemento '`sgmltexi`'.

**@afivepaper**

**@afourpaper**

**@afourlatex**

**@afourwide**

In sostituzione di questi comandi, si possono usare le opzioni della riga di comando: '`--paper=a5`', '`--paper=a4`', '`--paper=a4latex`', '`--paper=a4wide`'.

**@alias *nuovo=esistente***

Non disponibile. Probabilmente si può rimediare inserendo il comando all'interno dell'elemento '`texinfo`'.

**@anchor{ *nome* }**

```
<anchor id="nome">
```

**@appendix *titolo***

**@appendixsec *titolo***

**@appendixsection *titolo***

**@appendixsubsec *titolo***

**@appendixsubsection *titolo***

**@appendixsubsubsec** *titolo*

**@appendixsubsubsection** *titolo*

Le appendici si ottengono nell'ambito dell'elemento 'appendix'.

**@asis**

La parola 'asis' è usata come argomento dell'attributo 'emphasis' degli elementi 'table', 'vtable' e 'ftable'.

**@author** *autore*

```
<author>autore</author>
```

**@b{testo }**

```
<bold>testo</bold>
```

**@bullet{ }**

```
&bull;
```

**@bye**

```
</sgmltexi>
```

**@c** *commento*

**@comment** *commento*

Non è disponibile un elemento equivalente, dal momento che l'SGML offre un suo sistema per annotare i commenti. Se necessario, questo comando può essere incluso all'interno di un elemento 'texinfo'.

## @cartouche

```
<cartouche>  
blocco_di_testo  
|  
|  
</cartouche>
```

## @center *testo*

```
<center>testo</center>
```

Non si può usare nel titolo del documento.

## @centerchap *titolo*

Non disponibile.

## @chapheading *titolo*

```
<h1 type="heading">titolo</h1>
```

## @chapter *titolo*

```
<h1>titolo</h1>
```

## @cindex *voce*

```
<cindex entry="voce">
```

## @cite{*riferimento*}

```
<cite>riferimento</cite>
```

## **@clear** *indicatore*

Non disponibile. Eventualmente può essere usato all'interno dell'elemento `'texinfo'`.

## **@code** { *testo* }

```
<code>sample</code>
```

## **@command** { *nome\_comando* }

```
<command>nome_comando</command>
```

## **@contents**

```
<contents>
```

## **@copyright** { }

```
&copy;
```

## **@defcodeindex** *nome\_indice*

```
<defcodeindex>nome_indice</defcodeindex>
```

## **@defcv** *categoria classe nome*

## **@defcvx** *categoria classe nome*

```
<defcv cat="categoria" class="classe" name="nome">
  [<defcvx cat="categoria" class="classe" name="nome">] ...
  ...
  ...
</defcv>
```

## **@deffn** *categoria nome argomento...*

## **@deffnx** *categoria nome argomento...*

```
<deffn cat="categoria" name="nome">
  <args>argomento...</args>
  [<deffnx cat="categoria" name="nome">
    <args>argomento...</args>] ...
  ...
  ...
</deffn>
```

## **@defindex** *nome\_indice*

```
<defindex>nome_indice</defindex>
```

## **@definfoenclose** *nuovo\_comando prima dopo*

Non disponibile. Eventualmente può essere usato all'interno dell'elemento **'texinfo'**.

## **@defivar** *classe nome\_variabile\_di\_istanza*

## **@defivarx** *classe nome\_variabile\_di\_istanza*

```
<defivar class="classe" name="nome_variabile_di_istanza">
  [<defivarx class="classe" name="nome_variabile_di_istanza">] ...
  ...
  ...
</defivar>
```

## **@defmac** *nome\_macro argomento...*

## **@defmacx** *nome\_macro argomento...*

```
<defmac name="nome_macro">
  <args>argomento...</args>
  [<defmacx name="nome_macro">
    <args>argomento...</args>] ...
  ...
  ...
</defmac>
```

## **@defmethod** *classe nome\_metodo argomento...*

## **@defmethodx** *classe nome\_metodo argomento...*

```
<defmethod class="classe" name="nome_metodo">
  <args>argomento...</args>
  [<defmethod class="classe" name="nome_metodo">
    <args>argomento...</args>] ...
  ...
  ...
</defmethod>
```

**@defop** *categoria classe nome argomento...*

**@defopx** *categoria classe nome argomento...*

```
<defop cat="categoria" class="classe" name="nome">  
  <args>argomento...</args>  
  [<defopx cat="categoria" class="classe" name="nome">  
    <args>argomento...</args>] ...  
  ...  
  ...  
</defop>
```

**@defopt** *nome\_opzione*

**@defoptx** *nome\_opzione*

```
<defopt name="nome_opzione">  
  [<defoptx name="nome_opzione">]  
  ...  
  ...  
</defopt>
```

**@defspec** *nome argomento...*

**@defspecx** *nome argomento...*

```
<defspec name="nome">
  <args>argomento...</args>
  [ <defspecx name="nome">
    <args>argomento...</args> ] ...
  ...
  ...
</defspec>
```

**@defftp** *categoria nome attributo...*

**@defftpX** *categoria nome attributo...*

```
<defftp cat="categoria" name="nome">
  <args>attributo...</args>
  [ <defftpX cat="categoria" name="nome">
    <args>attributo...</args> ] ...
  ...
  ...
</defftp>
```

**@deftypefn** *classificazione tipo\_dati nome argomento...*

**@deftypefn** *classificazione tipo\_dati nome argomento...*

```
<deftypefn cat="classificazione" type="tipo_dati" name="nome">
  <args>argomento...</args>
  [<deftypefnx cat="classificazione" type="tipo_dati" name="nome">
    <args>argomento...</args>] ...
  ...
  ...
</deftypefn>
```

**@deftypefun** *tipo\_dati nome\_funzione argomento...*

**@deftypefunx** *tipo\_dati nome\_funzione argomento...*

```
<deftypefun type="tipo_dati" name="nome_funzione">
  <args>argomento...</args>
  [<deftypefunx type="tipo_dati" name="nome_funzione">
    <args>argomento...</args>] ...
  ...
  ...
</deftypefun>
```

**@deftypeivar** *classe tipo\_dati nome\_variabile*

**@deftypeivarx** *classe tipo\_dati nome\_variabile*

```
<deftypeivar class="classe" type="tipo_dati" name="nome_variabile">
  [<deftypeivarx class="classe" type="tipo_dati" name="nome_variabile">
  ] ...
  ...
  ...
</deftypeivar>
```

**@deftypemethod** *classe tipo\_dati nome\_metodo argomento...*

**@deftypemethodx** *classe tipo\_dati nome\_metodo argomento...*

```
<deftypemethod class="classe" type="tipo_dati" name="nome_metodo">
  <args>argomento...</args>
  [<deftypemethodx class="classe" type="tipo_dati" name="nome_metodo">
    <args>argomento...</args>] ...
  ...
  ...
</deftypemethod>
```

**@deftypeop** *categoria classe tipo\_dati nome argomento...*

**@deftypeopx** *categoria classe tipo\_dati nome argomento...*

```
<deftypeop cat="categoria" class="classe" type="tipo_dati" name="nome">
  <args>argomento...</args>
  [<deftypeopx cat="categoria" class="classe" type="tipo_dati" name="nome">
    <args>argomento...</args>] ...
  ...
  ...
</deftypeop>
```

**@deftypevar** *tipo\_dati nome\_variabile*

**@deftypevarx** *tipo\_dati nome\_variabile*

```
<deftypevar type="tipo_dati" name="nome_variabile">
  [<deftypevarx type="tipo_dati" name="nome_variabile">] ...
  ...
  ...
</deftypevar>
```

**@deftypevr** *classificazione tipo\_dati nome\_variabile*

**@deftypevr x** *classificazione tipo\_dati nome\_variabile*

```
<deftypevr class="classificazione" type="tipo_dati" name="nome_variabile">
  [<deftypevr x class="classificazione" type="tipo_dati" name="nome_variabile
  ] ...
  ...
  ...
</deftypevr>
```

**@defun** *nome\_funzione argomento...*

**@defunx** *nome\_funzione argomento...*

```
<defun name="nome_funzione">
  <args>argomento...</args>
  [<defunx name="nome_funzione">
  <args>argomento...</args>] ...
  ...
  ...
</defun>
```

**@defvar** *nome\_variabile*

## **@defvarx *nome\_variabile***

```
<defvar name="nome_variabile">
  [<defvarx name="nome_variabile">] ...
  ...
  ...
</defvar>
```

## **@defvr *categoria nome\_variabile***

## **@defvr x *categoria nome\_variabile***

```
<defvr cat="categoria" name="nome_variabile">
  [<defvr x cat="categoria" name="nome_variabile">] ...
  ...
  ...
</defvr>
```

## **@detailmenu**

```
<menu>
  ...
  ...
<detailmenu>
  ...
  ...
</detailmenu>
</menu>
```

## @dfn{ *termine* }

```
<dfn>termine</dfn>
```

## @dircategory *dirpart*

### @direntry

```
<infodir cat="dirpart">  
  ...  
  ...  
</infodir>
```

## @display

```
<display>  
  blocco_di_testo  
  ...  
  ...  
</display>
```

## @dmn{ *dimensione* }

```
<dmn>dimensione</dmn>
```

## @documentdescription *descrizione* @end documentdescription

```
<documentdescription content="descrizione">
```

## @documentencoding *codifica*

```
<sgmltexi charset="codifica">
```

Definisce la codifica del sorgente Texinfo che viene generato, stabilendo implicitamente che lo stesso sorgente SGML è realizzato nello stesso modo. Viene oscurato dall'opzione `'--input-encoding'`, che prende la precedenza generando un sorgente Texinfo in formato ISO 646 puro (ASCII a 7 bit).

### **@documentlanguage *cc***

```
<sgmltexi lang="cc">
```

### **@dotaccent { *c* }**

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione `'--input-encoding'`, o con l'attributo `'charset'` dell'elemento `'sgmltexi'`.

### **@dots { }**

```
&dots;
```

### **@email { *indirizzo* , *testo\_mostrato* }**

```
<email email="indirizzo" name="testo_mostrato">
```

### **@emph { *testo* }**

```
<emph>testo</emph>
```

### **@env { *variabile\_di\_ambiente* }**

```
<env>variabile_di_ambiente</env>
```

## **@enddots{ }**

```
&enddots;
```

## **@enumerate [ *numero\_o\_lettera* ]**

```
<enumerate [start="numero_o_lettera" ] >  
<item>  
    ...  
    ...  
<item>  
    ...  
    ...  
</enumerate>
```

## **@equiv{ }**

```
&equiv;
```

## **@error{ }**

```
&error;
```

## **@evenfooting**

## **@evenheading**

## **@everyfooting**

## **@everyheading**

Non disponibile. Eventualmente può essere usato all'interno dell'elemento **'texinfo'**.

## @example

```
<example>
  bloco_di_testo
  ...
  ...
</example>
```

### Preformattato:

```
<example>
<pre>
  riga_di_testo
  ...
  ...
</pre>
</example>
```

### Letterale:

```
<example>
<pre>
<![CDATA[
  riga_di_testo
  ...
  ...
]]>
</pre>
</example>
```

## **@exampleindent**

Non disponibile. Eventualmente può essere usato all'interno dell'elemento `'texinfo'`.

## **@exlamdown**

```
&iexcl;
```

## **@exdent**

```
<pre>  
  ...  
<exdent>testo_sporgente</exdent>  
  ...  
</pre>
```

## **@expansion{ }**

```
&expansion;
```

## **@file{nome\_file }**

```
<file>nome_file</file>
```

## **@finalout**

Non gestibile, in quanto il sorgente Texinfo che viene generato contiene sempre questo comando.

## **@findex voce**

```
<findex entry="voce">
```

## @flushleft

```
<flushleft>testo</flushleft>
```

## @flushright

```
<flushright>testo</flushright>
```

## @footnote{*testo\_del\_piè\_pagina*}

```
<footnote>testo_del_piè_pagina</footnote>
```

## @footnotestyle *stile*

```
<footnotestyle content="stile">
```

In alternativa si può usare l'opzione '**--footnotestyle=stile**' della riga di comando, la quale prevale.

## @format

```
<format>  
<pre>  
    ...  
    ...  
</pre>  
</format>
```

Letterale:

```
<format>
<pre>
<![CDATA[
    ...
    ...
]]>
</pre>
</format>
```

## @frenchspacing

```
<sgmltexi spacing="french">
```

## @ftable *comando\_di\_composizione*

```
<ftable emphasis="comando">
<item>voce_descrittiva</item>
[<itemx>voce_descrittiva</itemx>] ...
    blocco_di_testo...
    ...
    ...
<item>voce_descrittiva</item>
[<itemx>voce_descrittiva</itemx>] ...
    blocco_di_testo...
    ...
</ftable>
```

## @group

```
<group>blocco_di_testo</group>
```

## @H{c}

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione '`--input-encoding`', o con l'attributo '`charset`' dell'elemento '`sgmltexi`'.

## @heading *titolo*

```
<h2 type="heading">titolo</h2>
```

## @headings on

## @headings off

## @headings single

## @headings double

```
<headings content="on">  
<headings content="off">  
<headings content="single">  
<headings content="double">
```

In alternativa si può usare l'opzione '`--headings`' della riga di comando, la quale prevale su queste direttive:

```
--headings=on
--headings=off
--headings=single
--headings=double
```

## @html

```
<html>codice_html</html>
```

## @hyphenation{*parole\_separate\_in\_sillabe*}

```
<hyphenation words="parole_separate_in_sillabe">
```

## @i{*testo*}

```
<italic>testo</italic>
```

## @ifclear *opzione*

Non disponibile. Eventualmente può essere usato all'interno dell'elemento **'texinfo'**.

## @ifhtml

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifhtml>testo_interno_alle_righe</ifhtml>
```

```
<ifhtmlblock>
    blocco_di_testo
    ...
</ifhtmlblock>
```

L'SGML dà la possibilità di usare le sezioni marcate. Queste possono essere controllate da Sgmltexi attraverso l'opzione '**--sgml-include**' della riga di comando. Per esempio, il sorgente SGML potrebbe essere simile al pezzo seguente:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN"
[
<!ENTITY % HTML          "IGNORE">
<!ENTITY % INFO          "IGNORE">
<!ENTITY % TEX           "IGNORE">
...
]>
<sgmltexi>
...
<![%HT;[
    <p>Here it is some text that is meant to appear only inside
    the HTML typesetting.</p>
]]>
<![%INFO;[
    <p>Here it is some other text that is meant to appear only
    inside the Info typesetting.</p>
]]>
<![%TEX;[
    <p>This text is meant to appear only inside the TeX
    typesetting.</p>
]]>
...
</sgmltexi>
```

Quindi, quando si genera la composizione in HTML, si deve utilizzare l'opzione '**--sgml-include=HTML**':

```
$ sgmltexi --sgml-include=HTML --html mio_file.sgml [Invio]
```

Per la composizione nel formato Info, si deve usare l'opzione '`--sgml-include=INFO`':

```
$ sgmltexi --sgml-include=INFO --info mio_file.sgml [Invio]
```

Nello stesso modo, per la composizione attraverso TeX si deve usare l'opzione '`--sgml-include=TEX`':

```
$ sgmltexi --sgml-include=TEX --tex mio_file.sgml [Invio]
```

## @ifinfo

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifinfo>testo_interno_alle_righe</ifinfo>
```

```
<ifinfoblock>  
    blocco_di_testo  
    ...  
</ifinfoblock>
```

L'SGML dà la possibilità di usare le sezioni marcate, come è già stato mostrato a proposito del comando '@ifhtml'.

## @ifnohtml

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifnohtml>testo_interno_alle_righe</ifnohtml>
```

```
<ifnothtmlblock>
    blocco_di_testo
    ...
</ifnothtmlblock>
```

L'SGML dà la possibilità di usare le sezioni marcate. Queste possono essere controllate da Sgmltexi attraverso l'opzione '**--sgml-include**' della riga di comando. Per esempio, il sorgente SGML potrebbe essere simile al pezzo seguente:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN"
[
<!ENTITY % NOTHTML          "IGNORE">
<!ENTITY % NOTINFO          "IGNORE">
<!ENTITY % NOTTEX           "IGNORE">
...
...
]>
<sgmltexi>
...
...
<![%NOTHTML; [
    <p>Here it is some text that is meant to appear only outside
    the HTML typesetting.</p>
]]>
<![%NOTINFO; [
    <p>Here it is some other text that is meant to appear only
    outside the Info typesetting.</p>
]]>
<![%NOTTEX; [
    <p>This text is meant to appear only outside the TeX
    typesetting.</p>
]]>
...
...
</sgmltexi>
```

Quindi, quando si genera la composizione in HTML, si devono utilizzare le opzioni '**--sgml-include=NOTINFO**' e

**'--sgml-include=NOTTEX':**

```
$ sgmltexi --sgml-include=NOTINFO --sgml-include=NOTTEX ↵  
↵--html mio_file.sgml [Invio]
```

Per la composizione nel formato Info, si devono utilizzare le opzioni **'--sgml-include=NOTHTML'** e **'--sgml-include=NOTTEX'**:

```
$ sgmltexi --sgml-include=NOTHTML --sgml-include=NOTTEX ↵  
↵--info mio_file.sgml [Invio]
```

Nello stesso modo, per la composizione attraverso TeX si devono utilizzare le opzioni **'--sgml-include=NOTHTML'** e **'--sgml-include=NOTINFO'**:

```
$ sgmltexi --sgml-include=NOTHTML --sgml-include=NOTINFO ↵  
↵--tex mio_file.sgml [Invio]
```

## **@ifnotinfo**

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifnotinfo>testo_interno_alle_righe</ifnotinfo>
```

```
<ifnotinfoblock>  
  blocco_di_testo  
  ...  
</ifnotinfoblock>
```

L'SGML dà la possibilità di usare le sezioni marcate, come è già stato mostrato a proposito del comando **'@ifnohtml'**.

## **@ifnotplaintext**

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifnotplaintext>testo_interno_alle_righe</ifnotplaintext>
```

```
<ifnotplaintextblock>  
  blocco_di_testo  
  ...  
</ifnotplaintextblock>
```

L'SGML dà la possibilità di usare le sezioni marcate, come è già stato mostrato a proposito del comando '@ifnohtml'.

### @ifnottex

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifnottex>testo_interno_alle_righe</ifnottex>
```

```
<ifnottexblock>  
  blocco_di_testo  
  ...  
</ifnottexblock>
```

L'SGML dà la possibilità di usare le sezioni marcate, come è già stato mostrato a proposito del comando '@ifnohtml'.

### @ifnotxml

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifnotxml>testo_interno_alle_righe</ifnotxml>
```

```
<ifnotxmlblock>  
    blocco_di_testo  
    ...  
</ifnotxmlblock>
```

L'SGML dà la possibilità di usare le sezioni marcate, come è già stato mostrato a proposito del comando '@ifnohtml'.

### @ifplaintext

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifplaintext>testo_interno_alle_righe</ifplaintext>
```

```
<ifplaintextblock>  
    blocco_di_testo  
    ...  
</ifplaintextblock>
```

L'SGML dà la possibilità di usare le sezioni marcate, come è già stato mostrato a proposito del comando '@ifnohtml'.

### @ifset *flag*

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

### @iftex

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<iftex>testo_interno_alle_righe</iftex>
```

```
<iftexblock>
    blocco_di_testo
    ...
</iftexblock>
```

L'SGML dà la possibilità di usare le sezioni marcate, come è già stato mostrato a proposito del comando '@ifhtml'.

### @ifxml

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifxml>testo_interno_alle_righe</ifxml>
```

```
<ifxmlblock>
    blocco_di_testo
    ...
</ifxmlblock>
```

L'SGML dà la possibilità di usare le sezioni marcate, come è già stato mostrato a proposito del comando '@ifhtml'.

### @ignore

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'. Se non è necessario inserire commenti nel file Texinfo che viene generato, si possono usare i commenti secondo l'SGML:

```
<!--  
    commento  
    ...  
    ...  
-->
```

**@image**{*nome\_file*, [*ampiezza*], [*altezza*] }

```
<image name="nome_file" width="ampiezza" height="altezza">
```

**@include**

Non disponibile. Eventualmente può essere usato all'interno dell'elemento **'texinfo'**. L'SGML offre un meccanismo alternativo:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN"  
[  
<!ENTITY GPL SYSTEM "licenses/gpl.sgml">  
<!ENTITY BSD SYSTEM "licenses/bsd.sgml">  
...  
...  
<sgmltexi>  
...  
...  
<appendix>  
&GPL;  
&BSD;  
...  
...  
</appendix>  
...  
</sgmltexi>
```

come si può vedere dall'esempio, l'inserzione nel testo di **'licenses/gpl.sgml'** e di **'licenses/bsd.sgml'** avviene

attraverso l'uso delle macro SGML `&GPL;` e `&BSD;`.

Se è necessario includere un file Texinfo, si può fare come si vede nell'esempio seguente:

```
<![CDATA[
<p><texinfo>
@include example.texi
</texinfo></p>
]]>
```

È necessario tenere a mente che l'elemento `texinfo` è di tipo interno alle righe di testo. Ecco perché nell'esempio è contenuto in un elemento `p`.

`@inforef{nome_nodo, [voce], nome_file_info}`

```
<inforef id="nome_nodo" name="voce" info="nome_file_info">
```

`\input file_macro`

Non è possibile inserire macro aggiuntive all'inizio del documento, oltre a quella predefinita che imposta la sintassi Texinfo.

`@item`

Questo comando di Texinfo viene usato in contesti molto diversi. All'interno di Sgmltexi non esiste un modo unico per utilizzarlo, per cui conviene vedere piuttosto la descrizione dei comandi `@table`, `@ftable`, `@vtable`, `@itemize`, `@enumerate` e `@multitable`.

## **@itemize [ *marcatore\_iniziale* ]**

```
<itemize [mark="marcatore_iniziale" ] >
<item>
    ...
    ...
<item>
    ...
    ...
</itemize>
```

## **@itemx**

Questo comando di Texinfo viene usato in contesti molto diversi. All'interno di Sgmltexi non esiste un modo unico per utilizzarlo, per cui conviene vedere piuttosto la descrizione dei comandi '@table', '@ftable' e '@vtable'.

## **@kbd{ *tasti\_premuti* }**

```
<kbd>tasti_premuti</kbd>
```

## **@kbdinputstyle *stile***

```
<kbdstyle style="stile">
```

## **@key{ *nome\_tasto* }**

```
<key>nome_tasto</key>
```

## @kindex voce

```
<kindex entry="voce">
```

## @L{}

```
&Lstrok;
```

## @l{}

```
&lstrok;
```

## @lisp

```
<lisp>  
  blocco_di_testo  
  ...  
  ...  
</lisp>
```

## Preformattato:

```
<lisp>  
<pre>  
  riga_di_testo  
  ...  
  ...  
</pre>  
</lisp>
```

## Letterale:

```
<lisp>
<pre>
<![CDATA[
    riga_di_testo
    ...
    ...
]]>
</pre>
</lisp>
```

### **@lowersections**

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

### **@macro nome\_macro {parametri}**

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

### **@majorheading titolo**

Non disponibile attualmente.

### **@math{ espressione\_matematica }**

```
<math>espressione_matematica</math>
```

### **@menu**

```
<menu> [menù_info</menu>]
```

## @minus{ }

```
&minus;
```

## @multitable *larghezza\_delle\_colonne*

```
<multitable>
<columnfraction>frazione_larghezza_complessiva</columnfraction>...
<raw>cella [<tab>cella] ...</raw>...
...
</multitable>
```

```
<multitable>
<columnexample>testo_di_esempio</columnexample>...
<raw>cella [<tab>cella] ...</raw>...
...
</multitable>
```

## @need *n*

```
<need mils="n">
```

## @node *nome*, *successivo*, *precedente*, *superiore*

La gestione manuale dei nodi di Texinfo avviene come si vede nello schema seguente, dove ci si limita a stabilire il nome del nodo in questione:

```
<hn node="nome">titolo</hn>
```

Se è necessario un controllo completo sui nodi, si possono stabilire anche gli altri dati, come nello schema seguente:

```
<hn node="nome" next="successivo" prev="precedente" up="superiore">titolo
```

Sgmltexi non fa alcun controllo di validità per quanto riguarda l'inserzione manuale dei nodi.

### **@noindent**

```
<p indent="off">
```

### **@novalidate**

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

### **o{ }**

```
&Oslash;
```

### **o{ }**

```
&oslash;
```

### **@oddfooting**

### **@oddheading**

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

## **@option { *opzione* }**

```
<option>opzione</option>
```

## **@page**

```
<page>
```

## **@pagesizes [ *ampiezza* ] [ , *altezza* ]**

Non disponibile.

## **@paragraphindent *rientro***

Non disponibile.

## **@pindex *voce***

```
<pindex entry="voce">
```

## **@point { }**

```
&point;
```

## **@pounds { }**

```
&pound;
```

## **@print { }**

```
&print;
```

## @printindex *nome\_indice*

```
<printindex name="nome_indice">
```

## @pxref{*nome\_nodo*, [*voce*], [*argomento\_o\_titolo*], [*file\_info*], [*manual*] }

```
<pxref id="nome_nodo" name="voce" title="argomento_o_titolo" info="file_info"  
  ptitle="manual">
```

## @questiondown{ }

```
&iquest;
```

## @quotation

```
<quotation>  
  testo_interno_alle_righe  
  ...  
  ...  
</quotation>
```

## @r{*testo*}

```
<roman>testo</roman>
```

## @raisesections

Non disponibile.

**@ref**{*nome\_nodo*, [*voce*], [*argomento\_o\_titolo*], [*file\_info*], [*manuale*] }

```
<ref id="nome_nodo" name="voce" title="argomento_o_titolo" info="file_info" ptitle="manuale">
```

**@refill**

Non disponibile.

**@result**{ }

```
&result;
```

**@ringaccent**{*c*}

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione '`--input-encoding`', o con l'attributo '`charset`' dell'elemento '`sgmltexi`'.

**@samp**{*testo*}

```
<samp>testo</samp>
```

**@sc**{*testo*}

```
<sc>testo</sc>
```

**@section** *titolo*

```
<h2>titolo</h2>
```

## **@set** *flag string*

Non disponibile. Eventualmente può essere usato all'interno dell'elemento `'texinfo'`.

## **@setchapternewpage** *on*

## **@setchapternewpage** *off*

## **@setchapternewpage** *odd*

Si può usare l'elemento `'setchapternewpage'`, come negli schemi seguenti:

```
<setchapternewpage content="on">  
<setchapternewpage content="off">  
<setchapternewpage content="odd">
```

In alternativa si può utilizzare l'opzione `'--setchapternewpage'`, nella riga di comando:

```
--setchapternewpage=on  
--setchapternewpage=off  
--setchapternewpage=odd
```

## **@setcontentsaftertitlepage**

Non disponibile.

## **@setfilename** *nome\_file\_info*

```
<setfilename content="nome_file_info">
```

## **@setshortcontentsaftertitlepage**

Non disponibile.

## **@settitle** *titolo*

```
<settitle content="titolo">
```

## **@shortcontents**

```
<shortcontents>
```

## **@shorttitlepage** *title*

Non disponibile.

## **@smallbook**

Si usa per questo l'opzione: `--paper=small`.

## **@smalldisplay**

```
<smalldisplay>  
  blocco_di_testo  
  ...  
  ...  
</smalldisplay>
```

## **@smallexample**

```
<smallexample>  
  blocco_di_testo  
  ...  
  ...  
</smallexample>
```

Preformattato:

```
<smallexample>
<pre>
  riga_di_testo
  ...
  ...
</pre>
</smallexample>
```

## Letterale:

```
<smallexample>
<pre>
<![CDATA[
  riga_di_testo
  ...
  ...
]]>
</pre>
</smallexample>
```

## @smallformat

```
<smallformat>
<pre>
  ...
  ...
</pre>
</smallformat>
```

## Letterale:

```
<smallformat>
<pre>
<![CDATA[
    ...
    ...
]]>
</pre>
</smallformat>
```

## @smallisp

```
<smallisp>
    blocco_di_testo
    ...
    ...
</smallisp>
```

## Preformattato:

```
<smallisp>
<pre>
    riga_di_testo
    ...
    ...
</pre>
</smallisp>
```

## Letterale:

```
<smalllisp>
<pre>
<![CDATA[
    riga_di_testo
    ...
    ...
]]>
</pre>
</smalllisp>
```

## @sp *n*

```
<sp lines="n">
```

## @ss{ }

```
&szlig;
```

## @strong{ *testo* }

```
<strong>testo</strong>
```

## @subheading *titolo*

```
<h3 type="heading">titolo</h3>
```

## @subsection *titolo*

```
<h3>titolo</h3>
```

## **@subsubheading** *titolo*

```
<h4 type="heading">titolo</h4>
```

## **@subsubsection** *titolo*

```
<h4>titolo</h4>
```

## **@subtitle** *sottotitolo*

```
<subtitle>sottotitolo</subtitle>
```

## **@summarycontents**

```
<summarycontents>
```

## **@syncodeindex** *indice\_di\_origine* *indice\_di\_destinazione*

```
<syncodeindex from="indice_di_origine" to="indice_di_destinazione">
```

## **@synindex** *indice\_di\_origine* *indice\_di\_destinazione*

```
<synindex from="indice_di_origine" to="indice_di_destinazione">
```

## **@t** { *testo* }

```
<typewriter>testo</typewriter>
```

## **@tab**

Si veda la descrizione di ‘**@multitable**’.

## **@table comando\_di\_composizione**

```
<table emphasis="commando">
<item>voce_descrittiva</item>
[<itemx>voce_descrittiva</itemx>] ...
    blocco_di_testo...
    ...
    ...
<item>voce_descrittiva</item>
[<itemx>voce_descrittiva</itemx>] ...
    blocco_di_testo...
    ...
    ...
</table>
```

## **@TeX{ }**

```
&TeX;
```

## **@tex**

```
<tex>pezzo_di_sorgente_tex</tex>
```

## **@thischapter**

## **@thischaptername**

## **@thisfile**

## **@thispage**

## **@thistitle**

Non disponibile. Eventualmente può essere usato all'interno dell'elemento **'texinfo'**.

## **@tie{}**

```
&nbsp;
```

## **@tieaccent{cc}**

Non disponibile. Eventualmente può essere usato all'interno dell'elemento `'texinfo'`.

## **@tindex voce**

```
<tindex entry="voce">
```

## **@title titolo**

```
<title>titolo</title>
```

## **@titlefont{testo}**

Non disponibile.

## **@titlepage**

Non disponibile. Si veda come è organizzata la struttura di Sgmltexi.

## **@today**

```
&today;
```

## **@top**

Viene generato automaticamente.

## **@u{c}**

## **@ubaraccent{c}**

## **@udotaccent {c}**

Non disponibile. Eventualmente può essere usato all'interno dell'elemento **'texinfo'**.

## **@unnumbered *titolo***

```
<h1 type="unnumbered">titolo</h1>
```

## **@unnumberedsec *titolo***

```
<h2 type="unnumbered">titolo</h2>
```

## **@unnumberedsubsec *titolo***

```
<h3 type="unnumbered">titolo</h3>
```

## **@unnumberedsubsubsec *titolo***

```
<h4 type="unnumbered">titolo</h4>
```

## **@uref {url, [*testo\_mostrato*], [*rimpiazzo*] }**

```
<uref uri="url" name="testo_mostrato" replace="rimpiazzo ">
```

## **@url {url}**

```
<url>url</url>
```

## **@v {c}**

Non disponibile. Eventualmente può essere usato all'interno dell'elemento **'texinfo'**.

### **@value** { *indicatore* }

Non disponibile. Eventualmente può essere usato all'interno dell'elemento 'texinfo'.

### **@var** { *metavariabile* }

```
<var>metavariabile</var>
```

### **@verb** { *x* *testo\_letterale* *x* }

```
<verb char="x"><[CDATA[testo_letterale]]></verb>
```

### **@verbatim** *testo\_letterale* **@end verbatim**

```
<verbatim>  
<[CDATA[  
testo_letterale  
]]>  
</verbatim>
```

### **@verbatiminclude** *file*

```
<verbatiminclude file="file">
```

### **@vindex** *voce*

```
<vindex entry="voce">
```

### **@vskip** *ammontare\_dell'avanzamento*

Non disponibile. Eventualmente può essere usato all'interno dell'elemento 'texinfo'.

## @vtable *comando\_di\_composizione*

```
<vtable emphasis="comando">
<item>voce_descrittiva</item>
[<itemx>voce_descrittiva</itemx>] ...
    blocco_di_testo...
    ...
    ...
<item>voce_descrittiva</item>
[<itemx>voce_descrittiva</itemx>] ...
    blocco_di_testo...
    ...
    ...
</vtable>
```

## @w{ *testo* }

```
<whole>testo</whole>
```

## @xref{ *nome\_nodo* , [ *voce* ] , [ *argomento\_o\_titolo* ] , [ *file\_info* ] , [ *manuale* ] }

```
<xref id="nome_nodo" name="voce" title="titolo_o_argomento" info="file"
ptitle="manuale">
```