

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_variabibile*

@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="comando">
<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">
```