Info

オンライン GNU ドキュメントシステム

Brian Fox
GNU Texinfo コミュニティ

Table of Contents

1 はじめ方 *(2004/09/16)*

この Info マニュアルの最初の部分で，Info の内部について述べています．このマニュアルの 2 番目の部分で，様々な Info の高度なコマンドについて述べ，Texinfo ファイルから生成されるものとは異なる Info の書き方を述べています．3 番目の部分は，Texinfo ファイルから Info ファイルを生成する方法の概要を説明しています．

このマニュアルはコンピュータ上で Info リーダーを使って読むように考えられています。そうすれば、読んでいる間に Info コマンドを試すことができます。 コマンドがマニュアルに書いてある通りに動作すると思うことしかできませんので、紙や HTML で読むと効果が薄くなります。このマニュアルを読んでいる際に、オンラインバージョンも試してみてください。

このマニュアルのオンライン版を見るには 2 つの方法があります。

1. シェルのコマンドラインで `info` と入力します。こうすると、Info ファイルを読むためだけに設計された Info リーダーを使うことができます。
2. コマンドラインで `emacs` と入力し、それから *`C-h i`* (*`Control-h`* に続けて *`i`*) を入力します。これにより、多くの機能を持つ Emacs の Info モードを使うことができます。

どちらの場合でも、*`mInfo`* (文字通りに) と入力し、続けて `RET` (Return か Enter キー) と入力します。これで、スクリーン上でこのマニュアルを読み指示に従い操作を行う準備ができました。

1.1 小さなスクリーンで Info を開始する *(2004/09/16)*

(In Info, you only see this section if your terminal has a small number of lines; most readers pass by it without seeing it.)

端末の画面に比較的少数の行しか無いので，はじめに特別なアドバイスが必要です．

テキスト '–All—-' が画面の右下の角近くに見えている場合，見ているテキスト全体が画面にフィットしているということです．代わりに '–Top—-' が見えている場合，フィットしていない，より多くのテキストが下にあるということです．テキストを先に進めて他の画面全体を見るため，スペースバー<SPC>を押してください．前に戻るためには 'Backspace' や 'DEL'(システムによっては，このキーは 'Delete' と印字されているかもしれません) と表示されているキーを押してください．

1.2 Info の使い方 *(2004/09/16)*

ドキュメントを読むために，プログラム Info と会話をします．

Info を使うには 2 つの方法があります。つまり、Emacs を使う方法とシェルでコマンド `info` を使って起動される単独のリーダーを使う方法です。

今，1 つの情報のノードが見ています．ノードには，あるレベルで特定のトピックを説明しているテキストが含まれています．このノードのトピックは"Info の使い方 (how to use Info)"です．モードラインでは，これが `info`のノード '`Help`' だと告げています．

ノードのトップ行は，そのヘッダです．ノードのヘッダは (今，見てください) このノードの後にある '`Next`'ノードが '`Help-P`'と呼ばれるノードだと告げています．高度な Info コマンドで，知っている名前ならどこのノードにも行けるようになります．スタンドアローンの Info リーダプログラムでは，ヘッダ行で，このノードの名前と info ファイルも表示しています．Emacs では，ヘッダ行は特殊な書体で表示されていて，それは画面からスクロールされて消えることはありません．このノードと，Info ファイルの名前は，Emacs によってヘッダ行から削除されています．

'Next'の隣に，ノードには'Previous'や'Up'または両方へのリンクがあるはずです．御覧のようにこのノードには，これらすべてがあります．

さて，'Help-P'という名前の'Next'ノードに行く時間です．

>> そこに移動するため，*n*を入力します．一文字だけです．引用符や入力後の
RET を入力しないでください．

マージンの '>>' は，実際にコマンドを試すときだということを意味します．

>> マウスがある場合で，次のノードへ行くために既に *n* の入力の練習をしている場合，同じことを "マウスで行なう" ためにマウスの右ボタンで'Next'リンクをクリックしてください．

1.3 前のノードに戻る

このノードは'Help-P'と呼ばれています．ご存知のように，'Previous'ノードは'Help'で，そこから*n*コマンドを使用してやって来ました．もう一度*n*コマンドを使用すると，次のノード'Help-^L'に行きます．

>> しかし，まだ*n*を入力しないでください．最初に*p*コマンドを試したり，'Prev'のリンク上でマウスをクリックしたりして'Previous' ノードへ行きましょう．そこに行ったら，また*n*して，ここに戻ることができます．

Emacs でこれを読んでいる場合，メニューバーの右端近くに'Info'の項目が見えるでしょう．'Info'メニューバーの項目上でマウスをクリックすると，'Next'と'Prev'を含めたコマンドのメニューが開きます (そして，まだ学習していないものもそれ以外にあります)．

この全てが，恐らく余りに簡単過ぎて無礼だと思いますが，読み捨てしないでください．事態はますます複雑になってきます．また，そのときが来たと告げられるまで新しいコマンドを試さないでください．さもないと，後から出てくる重要な警告を Info が通り過ぎてしまいます．

>> さて，ノード'Help-^L'に行くため*n*したり'Next'リンクでマウスをクリックしたりして，より多くのことを学んでください．

1.4 スペース，DEL，B，そして^L コマンド

このノードのモードラインは，現在ノード'Help-^L'にいることを，ヘッダラインは*p*で'Help-P'に戻れることを告げています．このノードのタイトルは強調され、下線がついているかもしれません．それはノードが関連するものを告げています．

これは大きなノードで，ディスプレイ画面に全てがフィットしません．画面の右下付近に'--All----'ではなく'--Top-----'という文字列が見えるので，見えていないものがもっとあることが分かります．

SPC，BACKSPACE(または DEL)[1]，そして*b*コマンドで，一度に全ての画面がフィットしなくても"動き回る"ことが可能になります．SPCで前に進み，画面の下に続くものを見ることができます．DELや BACKSPACEで後ろへ戻り，画面のトップを見ることができます (何回かスペースを入力しなければ，トップまで何もありません)．

---

[1] このマニュアルで"バックスペースまたは DEL"と呼んでいるキーは，キーボードによって異なるラベルが付いています．ENTERや RETキーの上の小さいもので，通常 Emacs 以外でカーソルの前の文字，すなわち直前に入力した文字を削除するものからキーを探してください．それには，'Backspace'または'<-'，また'DEL'のときには'Delete'のラベルが付いています．

>> さて，SPCを入力してみてください (その後で BACKSPACEを入力し
ここに戻ってください).

SPCを入力すると画面の下 2 行がページの最初に現れ，その後の行が続きます．DELや BACKSPACEで，ページの最初の 2 行を下に通常は持っていきますが，その上の行でフルスクリーンにする意味が無い場合は，全てを下に持っていくわけではありません．

これを Emacs で読んでいる場合，画面からスクロールされて消えることが無いヘッダ行が見えていることに注意してください．そのため，'Next'，'Prev'，そして 'Up'のリンクは常に見えていて，これらのリンクをマウスでクリックすることで，ノードのどこからでも，これらのリンク場所に行くことが可能なので便利です．

SPCと DELは, 現在のノードで前後に移動するだけではありません．ノード間の移動も行います。現在のノードの最初で SPC を入力すると次のノードへ、ノードの最後で DEL (あるいは BACKSPACE) を入力すると前のノードへ移動します．具体的には，Info ファイルのすべてのノードを，単一の論理的に連結しているものとしてスクロールしていきます．これで，SPCを入力するだけでマニュアルの最初から最後まで完全に読むことができるでしょう．そして、DEL (あるいは BACKSPACE) を入力することで、最後から最初に向かってマニュアルを完全に読むことができるでしょう。

この順序では，ノードのサブノードは親ノードに続いて表示されます．ノードにメニューがある場合, SPCでメニューでリストアップされているサブノードに, 一つずつ移動していきます．ノードの終りに到達すると，そのすべてのサブノードを見たことになり，SPCで次のノードか次のノードの親に移動します.

この頃のキーボードには，'PageUp'と 'PageDown'('Prior'と 'Next'かもしれませんが) というラベルの付いた二つのスクロールキーがあるものが多くなっています．これらのキーがあるキーボードでは，SPCと BACKSPACEのように，これらを使用してテキストの前後に移動することが可能です．しかし，SPCと BACKSPACEとは異なり，PAGEUPと PAGEDOWNのキーでは，現在のノードの最初と最後を越えてスクロールすることはありません.

画面がおかしい場合，*C-l*(*Control-L*のことです–CTRL を押したまま *L*や *l*を入力することです) を入力することで，Info に再表示させることが可能です.

>> *C-l*を，今入力してください.

今見ているノードの最初に戻るため，何回も<BACKSPACE>を押すことが可能です．最初に戻るため，単純に *b*を入力することも可能です.

>> 今やってみてください．(我々は，ここまで来るまでに最初の画面がいっぱい
になるように十分に冗長なことを書いたのですが，画面が大きすぎて十分で
ないかもしれません．必要なら，Emacs や Info ウィンドウを小さくしてくださ
い．) そして，SPCで戻ってきてください.

画面が縦に非常に長い場合，このノード全体が一度にフィットするかもしれません．その場合, *b*では何も起こりません．だけど、もう少し小さいウィンドウを使えば *b*による効果を確認できますよ。

さて, かなりの数のコマンドを学びました．どれかを使いたいのですが, それがどれかを忘れた場合, コマンドリストを簡潔に表示する?を入力してみてください (Emacs では, それで Info-summary コマンドを実行します)．リストを見終えたら，SPCを何回か押して消し去ってください.

>> さあ?を試してください．終りまでリストの画面全体を連続して見るた
め，SPCを押してください．そして消え去るまで SPCを複数回入
力してください.

(スタンドアローンの Indo リーダを使用している場合，ここに戻るため *C-x 0*を入力してください，すなわち–CTRLを押したまま *x*を入力し，CTRLと *x*を離し，そして *0*を押してください–これはゼロで文字"o"ではありません．)

今から，警告無しで大きなノードに出会っても，教えられなくてもSPCとBACKSPACEを使用して動き回る方法を知っていることとします．全ての端末が同じ大きさの画面を持っているわけではないので，もう警告することはできません．

>> さて，次のノードを見るため，*n*を入力したり，‘Next’リンク
でマウスをクリックしたりしてください．

1.5 Emacs info で非表示なテキスト

メニューについて述べる前に、Emacs を使って Info を読んでいるユーザのみに関連したことを述べておく必要があります。スタンドアローンの Info リーダを使って読んでいるユーザは *]* を入力してスキップできます。

Emacs ではスタンドアローン版で表示されるある種のテキストは通常表示されません。技術的に言うと、それらのテキストは‘invisibility(非表示)’ の属性を持つためです。非表示テキストは実際のテキストの一部になります。(デフォルトでは) これらのテキストはキルしてヤンクした時、印刷した時などに表示されます。他のテキストと同じように保存することもできます。こうして、そこにテキストがあることを知っておくと役立ちます。

コマンド *M-x visible-mode* を実行することで非表示のテキストを表示させることができます。visible-mode はマイナーモードです。だから、2 回使うとふたたびテキストは非表示となります。"メニュー"の下やこのノードの先頭行でこのコマンドの効果を見てみなさい。

非表示テキストがいつも表示されているのがよければ、Info-hide-note-references を nil に設定するといい。visivle mode を有効にしておくことは代替にはなりません。Emacs Info は表示されているテキストを変更するような‘display’属性 (それほど多くはありませんが) も使っているからです。visible mode は非表示のテキストのみに影響します。このチュートリアルで、‘Emacs’ の動作を説明する時には、デフォルトの Emacs における動作を説明します。

さて、*]*を入力して、コマンド *]* と *[* について学ぶことにしよう。

1.5.1 コマンド *]* と *[*

今、*n*を入力すると、このノードには次のノードが無いというエラーが表示されます。同様に *p* を入力すると、前のノードが無いというエラーメッセージが表示されます (正確なメッセージは Info リーダに依存します)。これは、*n*や *p*が同じレベルの次や前のノードへ移動するコマンドだからです。今のノードは前のノードにあったメニューに含まれています。そのため、今のノードは下位のレベルと考えられるのです。このノードは前のノードメニューにある (3 回書かれているけれども) 唯一のノードです。それゆえ、*n* や *p* で移動できる次や前のノードを持たないのです。

n を整然と入力することでマニュアルを移動していると、多くのノードを読み飛ばす危険性があります。一方、SPCを使う場合にはそのような危険性はありません。なぜなら、ノードの下部までスクロールして、さらにSPCを入力すると、レベルに関わらず続くノードに移動するからです。最初に下部までスクロールさせないで、すぐに次のノードに行きたい場合には、*]*を入力します。

同様に、BACKSPACEはレベルに関わらず、現在のノードの先頭までスクロールした後で、前にあるノードに移動できます。すぐに前のノードへ移動したい時には *[*を入力します。

例えば、*[n [*という 3 段階の操作でここに戻ってくるでしょう。逆方向であれば、*] p]*とできます。

では、*]*を入力して、次のノードでメニューについて学びましょう。

1.6 メニューと 'm' コマンド

ノード間の移動に使用する 'n'(次) と 'p'(前)、SPC、BACKSPACE、*]*、*[*コマンドだけでは，ノードは直線的な操作に制限されます．メニューで，構造的に分岐することが可能になります．メニューは，移動可能な他のノードのリストです．それは実際に，Info が解釈可能な，特殊な書式のノードのテキストの一部です．メニューのはじめは，常に '* Menu:' で始まる行で識別されます．その方法で始まっている行の場合だけ，ノードにメニューが含まれます．使用可能なメニューだけが，常に移動可能なノードです．他のノードでメニューを使用するため，最初にノードに移動する必要があります．

メニューの最初から，'*'で始まるそれぞれの行はサブトピックを示します．行は普通，サブトピックの名前を (':'に続いて) 含んでいて，ノード名はサブトピックについて説明し，追加としてサブトピックの説明があります．'*'で始まらないメニューの行は特別な意味を持ちません – それらは読み易いようになっているだけで，追加のサブトピックを宣言しているものではありません．例は以下のようになります．

```
* Foo:  Node about FOO.      This tells about FOO.
```

サブトピックの名前は Foo で，ノードは 'Node about FOO'を説明しています．行の残りは読む人への情報です．[[しかし，この行は本当のメニューアイテムではなく，それは単に，上の行が '* Menu:'で始まっていないからです．本当のメニューアイテムで '*' は行の最初にあります。これは " 通常は隠された"テキストである ': Node about FOO.'が、この例では Visivle mode がオフの時でさえ、見えているからです。"]]

他のノードに行くためメニューを使用するとき (その方法はまもなく説明します)，指定するはサブトピック名で，メニュー行の最初にあるものです．Info はメニュー行を探すためにそれを使用し，それからノード名に復元し，そのノードへ行きます．サブトピック名とノード名の両方がある理由は，ノード名はコンピュータに対して意味が多すぎ，そのため汚く見えるかもしれないためです．サブトピック名は，ユーザの指定で便利にするためだけに選択することが可能です．ノード名はユーザが指定するのに便利で，サブトピックと同じ名前のときも良くあります．このため以下のような省略があります．

```
* Foo::   This tells about FOO.
```

これはサブトピック名とノード名が同じで, 'Foo' だということを意味します．(':' は通常 Emacs では非表示になります)

>> 今，このノードのメニューを見つけるために SPCを使用し，b と
SPCで前に戻ってください．御覧のように，メニューは実際にノードに
現れます．ノードを見てもメニューが見つからない場合は，ノードにメニュー
が無く，m コマンドは利用できません．

スクリーンにメニューが表示された状態で SPCを 1 回入力すると、他のノード (メニューの最初のノード) へ移動します。その場合には、BACKSPACE で戻ります。

サブノードの 1 つに行くコマンドは *m*です。このコマンドは既に使用したコマンドとは、さらに入力を求める点が大きく異なっています。

既に知っているコマンドではそのような追加入力は必要ありませんでした。キーを入力すれば、Info はすぐに処理を済ませ、他のコマンドが使用可能になります。しかし、*m*コマンドは違い、*name of the subtopic*(サブトピックの名前) が必要になります。一度、*m*を入力すると、サブトピックの名前を読み取ろうとするのです。

スタンドアローンの Info では、今，画面の下近くのダッシュを多く含む行を探してください (スタンドアローンの Info リーダではこれが Emacs のモードラインと同じものになります)．その下にもう一行ありますが，普通は空白です (この領域は Emacs ならエコー領域になります)．空

の場合，Info は *n*や *b*や SPC や *m*といったコマンドに対する準備ができています．その行がコロンで終るテキストを含んでいる場合，Info が現在のコマンドへの入力を読み込もうとしていることを意味します．そのときは，Info が入力を必要としていて，その入力を使用しようとしているので，コマンドは動作しません．それに応答するものを入力して開始したコマンドを終了するか，コマンドをキャンセルするために *Control-g*を入力する必要があります．このようなことをやり終えると，行はまた空白になります．

メニューでサブノードに行くコマンドは *m*です．*m*を入力後，画面の下の行は ‘Menu item: ’ と表示します．行きたいサブトピック名を入力し，終りに RETを入力する必要があります．Emacs では，*m*はコマンド Info-menuを実行します．

サブトピック名を省略することが可能です．省略がユニークでない場合，最初に一致したサブトピックが選択されます．メニューには，それぞれのサブトピック名に対して，大文字で可能な限り短い省略を書いているものもあるので，必要な入力数が分かります．サブトピックを入力するとき，大文字小文字は問題ありません．アイテム名の終りや内部にスペースを入れるべきではありませんが，メニュー項目にスペースがある場合は例外です．

サブトピック名の入力を助けるため，補完機能も使用可能です．名前の一部を入力後，TABキーを入力した場合，残りの名前が手品のように埋まります – 入力からユニークなものをできるだけ続けます．

カーソルをメニューのサブトピック行に移動した場合，引数を入力する必要はありません．RETを入力するだけで，それで行のサブトピックに行きます．サブトピック行でマウスの中ボタンをクリックして移動することもできます。

練習用に与えられたメニューが以下にあります．このメニューは 1 つの場所，Help-FOO に行く方法を 3 つ提供します．

(Emacs を使っているのであれば、visible mode をオンにしなさい)

>> 今，m を入力し，何が起こるか見てください．

今，*m*コマンドの"内部"にいます．コマンドは今使用できません．次にすることは，サブトピック名を入力することです．

*Control-g*を入力することで，*m*した意図を変更することができます．

>> 今それを試して，下の行がクリアされることを確認してください．

>> もう一度 m を入力してください．

>> アイテム名 *BAR*を入力してください．まだ RETを入力しないでください．

アイテム名を入力している間，間違えた場合に一文字キャンセルするため，DEL(または BACKSPACE) キーを使用することが可能です．

>> ‘R’をキャンセルするために DELを押してください．置換するためもう一度 ‘R’を入力することが可能です．しかし，‘BA’は有効な省略なので，そうする必要はありません．

>> 今，行く準備ができました．RETを入力してください．

‘Help-FOO’ に行った後，ここに戻るはずです．

メニューのサブトピック行とそれらの間を移動するもう一つの方法は TAB を入力することです．TABを入力するごとに，次のサブトッピック行に移動します．前のサブトピック行に移動する方法は，*M-TAB*を入力する方法です – すなわち，METAキーを押すまたは押したままにし，それから TABを押します．(キーボードによっては，METAキーは ‘Alt’のラベルが付いているかもしれません．)

サブトピック行にカーソルを移動すると，RETを押すことでそのサブトッピックのノードに行きます．

端末がマウスをサポートしている場合，サブトピックに行く方法はもう一つあります．マウスポインタを，サブトピックの行の最初の '*'から短いサブトピック名の終りとなるコロン ':'の間に移動してください．サブトピックの名前が変化するのが分かるでしょう (大抵は、背景色が変化する)。そしてプラットフォームがサポートしていればマウスポインタも変化することでしょう。マウスをその場所のままにして，しばらくすると，"Mouse-2: go to that node"と告げているツールチップがポップアップされるか、同じメッセージがスクリーンの最下行に表示されます。

*Mouse-2*は，左から数えてマウスの 2 番目のボタンです – ボタンが 2 つのマウスでは右端になり，ボタンが 3 つのマウスでは真中になります．そして，マウスポインタがメニューのサブトピックにあるときに *Mouse-2*を押すと，ソノサブトッピックに移動します．

More generally, *Mouse-2* in an Info buffer finds the nearest link to another node and goes there. For example, near a cross reference it acts like *f*, in a menu it acts like *m*, on the node's header line it acts like *n*, *p*, or *u*, etc. At end of the node's text *Mouse-2* moves to the next node, or up if there's no next node. より一般的には，Info バッファで *Mouse-2*を押すと，他のノードの最も近いリンクを見つけ，そこに移動します．例えば，相互参照の近くでは *f*のように動作し，メニューでは *m*のように動作し，ノードのヘッダ行では，*n*，*p*，または *u*のように動作します．ノードテキストの終りで *Mouse-2*を押すと，次のノードに移動したり，次のノードが無い場合は上のノードに移動します．

>> より多くのコマンドを理解するために n を入力してください.

1.6.1 *u* コマンド

おめでとう!これがノード 'Help-FOO'です．それには 'Help-M'へのポインタ 'Up'があり，それは *m*コマンドでやってくる前のノードです．これは一般的な習慣です – メニューからたどり着いたノードには，メニューに戻るための 'Up'ノードがあります．メニューはツリー構造を降りていき，'Up'は上がっていきます．一方，'Previous'は普通"前に戻るのではなく，同じレベルに留まる"ために使用されします．

"Up"に対するコマンド *u*を入力して，ノード 'Help-M' に戻ること可能です (*u*で実行されるEmacs コマンドは 'Info-up' です)．それでノードの前に行きます – 読んでいたところに戻るため，何回か SPC を入力する必要があります．(Emacs に組み込まれているような Info リーダによっては，'Help-M'で読んでいたのと同じ場所に行くものもあります．)

上に行くもう一つの方法は，ヘッダ行に表示されている 'Up'ポインタを *Mouse-2*でクリックする方法です (マウスがあれば提供されています).

>> 今，'Help-M'に戻るため，*u*を入力してください.

1.7 相互参照をたどる

Info ドキュメントで，相互参照を見ることも多いでしょう．相互参照はこのように見えます．See Section 2.7.1 [Help-Cross], page 14。そのテキストは本物の利用可能な相互参照で，'Cross'という名で，ノード名 'Help-Cross'を指しています (Emacs ではノード名は隠されています。表示/非表示を切り替えるには *M-x visible-mode*とします).

相互参照をたどる方法は 2 つあります．カーソルをその上に移動し，メニューと同様に RETを押すことでできます．RETで，カーソルがある相互参照をたどります．また，*f*を入力し，相互参照の名前 (この場合は 'Cross') を引数として指定することもできます．Emacs の Info では，*f*で 'Info-follow-reference'を実行します.

*f*コマンドでは，相互参照を名前で選択するので，カーソルの位置は問題になりません．カーソルが相互参照上やその近くにある場合，*f*はデフォルトとしてそれを参照する名前として提案します．RETを入力することで，その参照先をたどります．しかし，別の参照先の名を入力した場合，*f*はその名前を持つ別の参照先をたどります．

>> *f*を入力し，*Cross*を続けて，RETを入力してください．

参照先の名前を入力するとき，入力したものを編集するためにDEL(またはBACKSPACE) を使用することが可能です．参照先をたどるときに気が変わってしまった場合は，コマンドをキャンセルするために*Control-g*を使用することも可能です．*f*コマンドでは補完も利用可能です．現在のノードでTABを入力すると，すべての相互参照を補完することが可能です．

現在のノードで全ての相互参照のリストを得るため，*f*の後で*?*を入力することが可能です．*f*はリストを出力した後でも相互参照名の指定を待っているので，実際には参照をたどるつもりが無い場合，*f*をキャンセルするため*Control-g*を入力すべきです．

>> このノードの相互参照のリストを得るため*f?*を入力してください．そして，*Control-g*を入力し，*f*を停止する方法を理解してください．

メニュー間を移動するTABキーと*M-TAB*キーで，メニュー外部の相互参照へも移動します．

時折、相互参照は別のファイル (言い換えると別の"マニュアル") や時にはリモートマシンにあるファイルにさえ (Info は Emacs と一緒に配布されているし、スタンドアローンの Info ではリモートリンクを使わないようにしているけれど) 移動できます。そういった相互参照は次のようになります。See Section “Overview of Texinfo” in *Texinfo: The GNU Documentation Format*。(このリンクを辿ったら、*l*でここへ戻る)。この括弧内にある ‘texinfo’ (スタンドアローン版で表示される) はファイル名を指しています。現在のファイルと違う場合には、相互参照にファイル名やノード名が表示されるのです。Emacs ではファイル名は (他のテキストとともに) 表示されていません。(*M-x visible-mode*を使って表示/非表示にできます)

このノードの残りは Emacs 版でのみ有効です。スタンドアローン版を使っている場合には、すぐに*n*を入力できます。

ユーザーによっては、章を移動することよりもマニュアルを移動することの方がかなり大きなことのように感じるかもしれません。こういったユーザーは、実際に他のマニュアルに移動する前にそのことを知りたがります。そして、与えられても気付かなければ、*t* (次のノードを参照のこと) のような Info コマンドはユーザを混乱させることになります。

マウスを相互参照の上に置き、その相互参照が別のマニュアルを指していれば、別のボックス (ツールチップ) やエコー領域に情報が表われ、相互参照によって移動するファイルを (括弧の中に) 表示することでしょう。このことはメニューのサブトピック名でも同じです。マウスがあって、相互参照 ‘Overview’ の上にマウスを置いておくと、何が起こるか分かるでしょう。

マウスを相互参照上に移動させることなく利用できる情報をいつでも得たい場合には、`Info-hide-note-references` を t から他の値に変更します (see Section 2.13 [Emacs Info Variables], page 15)。リモートマシンやアクセスの遅い場所への相互参照がたくさんあって、ローカルとリモートファイルへのリンクが区別できないような場合に有効でしょう。

>> *n* を入力してさらに学びましょう

1.8 中級 Info コマンド

導入の講座はほとんど終りです．中級のコマンドを学ぶために，もう少し続けてください．

ほとんどの Info ファイルには索引があり，それは実際には，メニュー以外に何もない大きなノードになっています．メニューには，索引でリストアップされているトピックごとに一つのメ

ニューになっています．ファイルのメインメニューから，*m*コマンドで索引のノードを見つけることができます．トピックを記述しているノードに移動するために，索引ノードでもう一度 *m*を使用することも可能です．

ショートカット Info コマンド 'i' もあり，それはすべてのことを行ないます．それは与えられたトピック (文字列) を索引で検索し，そのトピックに対する索引にリストアップされているノードに移動します．完全な説明は，See Section 2.2 [Info Search], page 11

別のノードに移動していて，それまでの移動段階を再追跡したい場合，*l* コマンド (*l*は"last"に対応します) でそうすることができ，そのとき一度に一つのノードを移動します．ノードからノードに移動する間，Info は特別な履歴リストに訪問したノードを記録します．*l*コマンドで，履歴リストのノードにもう一度訪れることになります．*l*コマンドが成功するたび，履歴を一ステップ戻ります．

Emacs では，*l*でコマンド `Info-last`を実行します．

>> *p p n* と入力し、それから 3 度 *l* を入力しなさい。そして、それぞれの *l* が行うことを理解するために停止しなさい。最後にはここに戻ってくるはずです。

*l*と *p*の違いに注意してください．*l*は前回_あなたが_いた場所に移動し，*p*は常にヘッダで'`Previous`'ノードとされるノード (このノードでは'`Prev`'は'`Help-Xref`'に導くものです) に移動します．

*d*コマンド (Emacs では `Info-directory`) で，ディレクトリノードにすぐに移動します．このノードは，Info に入ったときに最初に見るもので，存在する全てのノードに (直接，あるいは間接的に他のメニューを通じて) 導くメニューがあります．ディレクトリノードは，システムにインストールされている，またはされているはずの，すべてのマニュアルと Info ドキュメントをリストアップしています．

>> *d*を試し，*l*でここに戻ってください (ええ，戻ってください)．

*t*コマンドでマニュアルの'`Top`'ノードに移動します．マニュアルのメインメニューを見たい場合や，特定のトップレベルのメニュー項目を選択したい場合に便利です．*t*で実行される Emacs コマンドは `Info-top-node` です．

相互参照の上やその近くで *Mouse-2*をクリックしても，その参照先をたどっていきます．マウスポインタを相互参照に移動し，それに応答してテキストに下線が引かれたマウスポインタが変化する状態を見ることで，相互参照がマウスに反応することが分かるはずです．

>> 今 *n*を入力し，この講座の最後のノードを見てください．

より高度な Info の機能は，See Chapter 2 [Expert Info], page 10。

2 熟練者のための Info *(2004/09/21)*

この章では，様々な高度な Info コマンドを記述しています．(スタンドアローンの Info リーダを使用している場合，それ特有の追加コマンドがあり，それらは Section “GNU Info” in *GNU Info* のいくつかの章で説明されています．)

この章では，Texinfo ファイルとは異なる Info の書き方を記述しています．(しかし，Info ファイルと印刷されたマニュアル、HTML や DocBook のような書式などで生成することが可能なので，ほとんどの状況では Texinfo ファイルを書く方がいいでしょう．)See Section “Overview of Texinfo” in *Texinfo: The GNU Documentation Format*

2.1 高度な Info コマンド

より容易に動き回るれるようになる Info コマンドには以下のものがあります．

g で指名したノードに移動する

ノードの名前を知っている場合，*g*，名前，RETと入力することでそこに移動することが可能です．このため，*gTopRET*でこのファイルの ‘Top’と呼ばれるノードへ移動します．(これは *t*と同じです．Section 1.8 [Help-Int], page 8 を参照してください．) *gExpertRET*でここへ戻ってきます．Emacs では *g*でコマンド Info-goto-nodeを実行します．

*m*とは異なり，*g*では省略を使用できません．しかし，補完は可能なので，TABを入力して部分的なノード名を完全なものにすることが可能で す．

他のファイルのノードへ移動するため，ノード名の前に，カッコ内にファイル名を含めることが可能です．このため，*g(dir)TopRET*で Info ディレクトリノードへ移動し，それはファイル dir の ‘Top’ノードです．同様に，*g(emacs)TopRET*で Emacs のマニュアルのトップノードに移動します．

ノード名 ‘*’はファイル全体を指定します．そのため，*g*RET* と入力することで現在のファイルの全てを見たり，*g(FILENAME)RET* で他のファイルの全てを見ることが可能です．

1 − *9* でメニューのサブトピックを選択する

システムが要求するそれぞれの入力文字がいやな場合，コマンド *1*，*2*，*3*，*4*，. . .，*9*の使用を好むでしょう．それらは，引数と供に使用されている *m*コマンドの短いものです．*1*は現在のノードメニューの最初のものに行きます．*2*は 2 番目の項目に行く等となっています．スタンドアローンの Info リーダでは *0*で最後のメニュー項目まで移動します．これで，存在するたくさんの項目を数える必要が無くなります．Enacs では数字のキーはコマンド Info-nth-menu-itemを実行します．

画面が複数のフォントをサポートしていて，Emacs の Info モードを Info ファイルを読むために使用している場合，5 番目のメニュー項目に対する ‘*’が目立つように，色が付いていたり，下線が引かれていたりするような他の属性があり，9 番目の項目も同様になっています．これで，項目に対して使用されている番号をひと目で見ることが容易になります．

端末によっては，色も下線もサポートしていません．実際に項目を数える必要がある場合，その代わりに *m*を使用して名前を指定したり，メニュー間を素早く移動するために TABを使用した方が良いでしょう．

e で Info ドキュメントを編集可能にする

Info コマンド *e*で，Info モードから普通の Emacs 編集モードに変更し，それで現在のノードのテキストを編集することが可能になります．Info に切替えるため，*C-c C-c*と入力してください．*e* コマンドは，変数 Info-enable-editが nilでない場合のみ利用可能です．

*e*コマンドは Emacs でのみ動作し，そこではコマンド `Info-edit`を実行します．スタンドアローンの Info リーダでは Info ファイルを変数可能にすることはできないので，*e*を入力するとノードの終りに移動します．

2.2 指定した主題が述べてある Info ドキュメントを探す

内部のノード間を移動するコマンドで，マニュアル全体や大きな部分を読むことが可能になります．しかし，マニュアルの情報をできるだけ早く見つける必要があり，それを探すノードは何かを知らない場合はどうでしょう？これは，マニュアルをリファレンスとして利用していたり，プログラムの使用を開始する前にマニュアル全体を読むのが非現実的な場合に生じます．

Info には，素早く探すための強力な検索機能があります．マニュアルの索引やそのテキストを検索することが可能です．

マニュアルで記述されているものに関連しているほとんどの主題は索引にされているので，最初に索引を検索すべきでしょう．*i*コマンドで，主題を入力するように促され，索引で主題を検索します．入力した主題の索引項目が見つかる場合，索引項目が指し示しているノードに移動します．探している問題がそこに記述されているかどうかを調べるためノードを見るべきでしょう．そうでなければ，主題に一致している追加の索引項目に移動するため，*,*を一回以上入力してください．

*i*コマンドでは，文字列の一部として入力した文字列が含まれるすべての索引項目が見つかります．一致するたびに，Info は見つかった索引項目の全体をエコーエリアに表示します．索引項目全体のテキストで探しているものに関連しているかどうかを決定するのに十分な情報が得られることも多いので，そのノードを表示してみる前に，Emacs がエコーエリアに表示したものを読むことを我々は推奨します．

*i*は文字列の一部から検索するので，索引でどのように綴られているかが不確かな場合でも主題を検索することが可能です．例えば，入力の一部を補完 (complete) する (例えば，`TAB`を入力するとき) コマンドに適切なものを探したいと仮定します．"complete"，"completion"，そして"completing"について記述している索引を得たい場合，*icompletRET*と入力することが可能でしょう．

プログラムを説明している Info ドキュメントでは，プログラムが提供しているコマンド，オプション，そしてキーの組み合わせが索引にあるでしょう．コマンド，オプション，またはキーの記述を探している場合，*i*でトピックの入力を促されているとき，その名前を入力してください．例えば，*C-f*キーが行なうことの記述を読みたい場合，*iC-fRET*を入力してください．ここで，*C-f*は 3 つのリテラル文字 ‘C’，‘-’，そして *f*のことで，*C-f*に割り当てられているコマンドを実行するために Emacs 内部で入力する"Control-f"のコマンドキーではありません．

Emacs では，*i*でコマンド `Info-index`を実行します．

*s*コマンドで，ファイル全体から文字列を検索することが可能になります．それは，必要場合は次のノードに切り替えます．*s*に続き探す文字列を入力し，`RET`で終了します．同じ文字列を再度検索するため，*s*に続けて `RET`を入力します．ファイルのノードはファイルにある順番で探され，それは，メニューのツリー構造と ‘`next`’ポインタの順番に関連している必要はありません．しかし通常は，2 つの順番はそんなに異なりません．あらゆる状況で，ヘッダが見えない場合 (*s*でカーソルを文字があるところに移動しても，そこはノードの最初ではないので，これは生じます)，到着したノードを判定するため *b*を実行することが可能です．

Emacs では，*Meta-s*は *s*と同じです．これは検索コマンドとして *M-s*を使用している，他の GNU パッケージとの互換性のためです．*s* と *M-s*はどちらも Emacs でコマンド `Info-search`を実行します．

2.3 Info に新しいノードを追加する *(2004/09/21)*

新しいトピックを Info ディレクトリのリストに追加するため，以下のようにする必要があります．

1. そのトピックをドキュメントにするため，いくつかのファイルで，いくつかのノードを作成してください．
2. ディレクトリのメニューにそのトピックを書いてください．See Section 2.4 [Menus], page 13.

Usually, the way to create the nodes is with Texinfo (see Section "Overview of Texinfo" in *Texinfo: The GNU Documentation Format*); this has the advantage that you can also make a printed manual or HTML from them. You would use the '`@dircategory`' and '`@direntry`' commands to put the manual into the Info directory. However, if you want to edit an Info file manually and install it manually, here is how. 通常，ノードを作る方法として Texinfo を用います (see Section "Overview of Texinfo" in *Texinfo: The GNU Documentation Format*)．これには，印刷されたマニュアルや HTML を作成することもできるという利点もあります．マニュアルと Info ディレクトリに追加に置くために、'`@dircategory`' と '`@direntry`' を使うことになるだろう。しかし，Info ファイルを編集して、それを自分でインストールしたい場合，以下の方法があります．

新しいノードを，既存のドキュメントファイルや，新しいものに書き込むことが可能です．それは，その前に '`^_`'文字が存在し (ユーザは見えません．このノードには 1 つありますが，あなたは見ることができません)，それは，'`^_`', '`^L`'("フォームフィード")，またはファイルの終りのいずれかで終了する必要があります．[1]

ノードを開始する '`^_`'には，改行または '`^L`'と改行を続ける必要があり，その後にノードのヘッダ行を続けます．ヘッダ行には (Info で見つかるように) ノード名を与える必要があり，'`Next`', '`Previous`'，そして '`Up`'ノード (存在する場合は全て) をの名前を述べる必要があります．御覧のように，このノードの '`Up`'ノードはノード '`Expert Info`'です．'`Next`'ノードは '`Menus`'です．

キーワード *Node*，*Next*，*Previous*，そして *Up* は，あらゆる順番で，ヘッダ行のどこにでも書いてもかまいませんが，推奨される順番はこの文のものです．それぞれのキーワードには，コロン，スペース，そしてタブを続け，その後に適切な名前を続ける必要があります．名前はタブ，カンマ，または改行で終了してもかまいません．スペースでは終了しません．ノード名にはスペースを含めてもかまいません．名前の大文字小文字の違いは重要ではありません．

ノード名には 2 つの形式が有ります．現在のファイルのノードは，ノードの最初の行の '`Node:`'後に現われるもので命名されています．例えば，このノードの名前は '`Add`'です．他のファイルのノードは，このノードの '`(info)Add`'の様に，'`(`*`filename`*`)`*`node-within-file`*'で命名されています．ファイル名が"./"で始まる場合，それは現在のディレクトリと相対的になります．それ以外の場合，それはあなたのサイトの標準的な Info ディレクトリから開始して相対的なものになります．名前 '`(`*`filename`*`)Top`'は，'`(`*`filename`*`)`'で省略することが可能です．慣習で，名前 '`Top`' は，単一ファイルでの"最上位の"ノードに使用されます – その '`Up`'ノードはファイルの外部を指し示します．'`Directory`'ノードは '`(dir)`'で，それはサイトにインストールされているすべての Info ドキュメントをリストしている大きなメニューを保持しているファイル `dir`を指し示します．'`Directory`'でリストアップされているドキュメントファイルの '`Top`'ノードには，その中に '`Up: (dir)`' が有るでしょう．

---

[1] '`^L`'を新しいノードの終りに書く場合，'`^L`'ではノードを␣開始␣することが不可能なので，その後に次のものを開始する '`^_`' があることを確かめてください．また，ノードの境界をページの境界にする良い方法は，'`^_`'の␣直後に␣'`^L`'を同じように書くことです．

ノード名*は特別です．それはファイル全体を参照します．そのため，*g**は現在のファイル全体を表示します．ノード*を使用することで，ファイルをツリーのノードに組織化しない，時代遅れの形式にすることを可能にします．

ノード自身の名前を述べる ‘Node:’の名前には，Info がノードを探すとき，ファイル名が存在することを予期していないので，ファイル名を含めてはいけません．‘Next’，‘Previous’，そして ‘Up’の名前ではそれらを含めてもかまいません．このノードで，‘Up’ノードは同じファイルに有るので，それを使用する必要は有りません．

このファイルのノードには，ヘッダ行にファイル名があることに注意してください．ファイル名は Info で無視されますが，それらは，ユーザがノードを識別する助けとなるコメントとして提供されています．

2.4 How to Create Menus

2.5 メニューの作成方法 *(2004/09/21)*

Info 階層のすべてのノードには，メニューを持たせてもかまいません – それはサブノードのリストです．*m*コマンドで，端末から読み込まれたトピックに対して現在のノードのメニューを検索します．

メニューは ‘* Menu:’ で始まる行で開始します．行の残りはコメントです．行の先頭から ‘* ’ で始まる全ての行は，単一のトピックをリストアップします．トピックの名前 – このトピックを選択するため，ユーザが *m*コマンドに与える必要がある引数 – は，星型 (アスタリスク) とスペースの直後にあり，コロン，スペース，そしてタブが続き，そしてそのトピックを述べているノード名が続きます．‘Next’，‘Previous’，そして ‘Up’が続くノード名に似ているノード名は，タブ，カンマ，または改行で終端してもかまいません．それは，ピリオドで終端してもかまいません．

ノード名とトピック名が同じ場合，2 度名前を与える代わりに，省略の ‘* *name*::’を使用してもかまいません (そして，それがメニューが乱雑になるのを抑えるので，可能なときはいつでもそれを使用すべきです)．

トピック名の先頭付近がお互いに異なるようにトピック名を選択することは思慮深いことです – これで，ユーザは短い省略の入力が可能になります．長いメニューでは，最小限受容できる省略となる，それぞれの項目名の初めを大文字化することは良い考えです (長いメニューとは，5 つ以上の項目のものです)．

ノードのメニューでリストアップされているノードは，(メニューのあるノード) の"サブノード"と呼ばれ，(メニューのあるノード自身は) それら (サブノード) の"上位"になります．(サブノードは) それぞれ，上位への ‘Up:’を示すものがあるべきです．(サブメニューの) 全てを見たい人が，メニューに何度も訪れる必要がないように，全てまたはほとんどのサブノードが ‘Next’と ‘Previous’ポインタで並ぶように整列すると役に立つことも多いでしょう．

Info ディレクトリは，ノード ‘(dir)Top’の単純なメニューです – すなわち，ファイル .../info/dirのノード ‘Top’です．その他のあらゆるメニューと全く同じようなメニューに，新しい項目を書くことが可能です．Info ディレクトリは，infoと呼ばれるファイルディレクトリと同じではありません．Info のファイルの多くは，そのファイルディレクトリに存在しますが，必須ではないこともあります．そして，そのディレクトリのファイルが Info ディレクトリのノードに自動的にリストアップされないこともあります．

また，Info ノードグラフは，"階層的"だと主張されていますが，実際それは任意の方向に向いたグラフにもなり得ます．共有されている構造とポインタサイクルは完全に利用可能になっていて，表現の意味として適切な場合はそれらが使用可能です．ファイルの全てのノードが，連結さ

れている構造となる形式にする必要はありません．実際このファイルには，2 つの連結されているコンポーネントがあります．あなたはその中の 1 つにいて，それはノード ‘Top’ の下にあります．それ以外は hコマンドで行くノード ‘Help’を含んでいます．実際，ごみ収集人がいないので，サブ構造を示さなくても困った問題はありませんが，しかしこのようなサブ構造は，今まで誰もその存在を見つけることができないので，どちらかと言うと無用です．

2.6 Creating Cross References

2.7 相互参照を作成する *(2004/09/21)*

相互参照は，行の先頭に書かれている必要があるメニュー項目と異なり，テキストのどこにでも書くことが可能です．相互参照は，‘*’の代わりに ‘*note’があること以外，メニュー項目のように見えます．それは，‘)’がノード名の一部にあることも多いので，‘)’で終端することはできません．括弧で相互参照を囲みたい場合，ピリオドを最初に用いて終端してください．相互参照ポインタの 2 つの例は以下のようになります．

```
*Note details: commands.  (See *note 3: Full Proof.)
```

これらは単なる例です．それが"導く"場所は実際には存在しません！

2.7.1 The node reached by the cross reference in Info

2.7.2 Info の相互参照で到着するノード *(2004/09/21)*

これは，‘Cross’と命名されている相互参照で到着するノードです．

このノードは，相互参照で到着することを目的とした特殊なもので，ほとんどの相互参照は，Info ドキュメントの構造のどこか遠くに"属している"場所があります．そのため，このノードに来たところへ戻るための ‘Next’，‘Previous’，または ‘Up’ へのリンクがあることを期待することはできません．一般的に，l (el) コマンドがそこへ戻る唯一の方法です．

>> 相互参照でやってきたノードに戻るために lを入力してください．

2.8 Quitting Info

2.9 Info を終了する *(2004/09/21)*

Info から出てこれまでしていたことに戻るため，*Quit* するために q を入力してください．これは Emacs で Info-exitを実行します．

これが Info の使用上の基本コースの最後です．Info ドキュメントでの移動方法と，メニューや相互参照をたどる方法を既に学んできました．これで，新しいパッケージを学ぶときに新しいユーザが行なう，マニュアルの最初から最後まで読む準備ができています．

それ以外の Info コマンドは，マニュアルで何かを素早く探すときに役に立ちます– すなわち，マニュアルをチュートリアルとしてではなくリファレンスとして使用する必要があるときです．検索コマンドも同じように学んで欲しいという気持ちもあります．今そうしたい場合，この相互参照 Section 2.2 [Info Search], page 11 をたどってください．

更にもう一つのコマンドの組は，熟練ユーザに対して意味があります．Info のドキュメントのディレクトリノードを探すと見つかるはずです．それらを見つけることは，通常の方法で Info を使用するための良い練習になるでしょう．

>> Info ディレクトリノードに移動するために dを入力してください．そし

て，Info のノードと利用可能なそれ以外のヘルプを見るために，*mInfo* とリターンを入力してください．

2.10 Info ファイルのタグ表 *(2004/09/21)*

大きな Info ファイルにタグ表を与えることで，ノードへのアクセスを速くすることが可能です．プログラムのタグ表と異なり，Info ファイルのタグ表はファイル自身に存在し，Info がファイルを読み込むときはいつでも自動的に使用されます．

タグ表を作成するため，Emacs Info モードを使用しているファイルのノードへ行 き，*M-x Info-tagify*を入力してください．ファイルを保存するため，*C-x C-s*を使用する必要があります．Texinfo パッケージの一部の `makeinfo`コマンドで生成された Info ファイルは，常に最初にタグ表があります．

一度 Info ファイルにタグ表を持たせると，それが最新であることを確かめる必要があります．(Texinfo ソースを編集するのではなく)Info ファイルを直接編集し，テキストが削除される結果として，タグ表が記録している位置からノードがファイルの中で千文字以上移動される場合，Info はもはやそのノードを見つけることができないでしょう．タグ表を更新するため，再び `Info-tagify` コマンドを使用してください．

Info ファイルのタグ表はファイルの終わりにあり，以下のようになっています．

```
^_^L
Tag Table:
File: info, Node: Cross-refs^?21419
File: info,  Node: Tags^?22145
^_
End Tag Table
```

ノードごとに 1 行を含んでいて，この行が，(ノード名の直後で終る) ノードヘッダの初めの部分，‘`DEL`’文字，そしてファイル内でのノードを開始する文字の位置を含んでいることに注意してください．

2.11 Checking an Info File

2.12 Info ファイルの調査 *(2004/09/21)*

Info ファイルを作成するとき，他のノードからのポインタを作成しているときにノード名を簡単に忘れてしまうものです．ノードに対して間違った名前を書いた場合，誰かが Info を使用してポインタへの移動を試みるまで発見されません．Info ファイルの照合は，すべてのノードへのポインタを調査して，無効なポインタを報告する自動的な処理です．すべての ‘`Next`’，‘`Previous`’，そして ‘`Up`’は，すべてのメニュー項目とすべての相互参照として調査されます．更に，‘`Previous`’を示すものを持たない ‘`Next`’すべてが報告されます．他のファイルへのポインタの調査は大変遅いので，ファイル内の中のポインタのみ調査されます．しかし，それらは通常ほとんどありません．

Info ファイルを調査するため，ファイルのノードを見ている間に，Emacs Info モードで `M-x Info-validate`をしてください．

2.13 Emacs の Info モード変数 *(2004/09/21)*

以下の変数で，Emacs での Info モードの動作を修正してもかまいません．1 つまたは複数のこれらの値を，対話的に，または`~/.emacs`初期化ファイルで設定してもかまいません．See Section

“Examining and Setting Variables” in *The GNU Emacs Manual*. スタンド アローンの Info リーダプログラムには，Section “Manipulating Variables” in *GNU Info* に記述されている独自の変数セットがありま す.

`Info-directory-list`
: Info ファイルを探すディレクトリリストです．それぞれの項目は文字列 (ディレクトリ名) または `nil`(デフォルトディレクトリを試します) です．初期化されていない場合は，それを初期化するために Info はユーザ変数 `INFOPATH` を使用し，環境変数 `INFOPATH`が無い場合は，`Info-default-directory-list`を使用します.

  Emacs の Info モードとスタンドアローンの Info で，Info が検索するディレクトリのリストをカスタマイズしたい場合，両方のプログラムに適用される `INFOPATH`環境変数を設定するのが最も良い方法です.

`Info-additional-directory-list`
: Info ドキュメントファイルを検索する追加のディレクトリリストです．これらのディレクトリは，`dir`ファイルにマージするために検索されません.

`Info-fontify`
: `nil`ではない値に設定されるとき，Info ファイルの強調が利用可能です．デフォルトは `t`です．`info-node`，`info-xref`，`info-header-xref`, `info-header-node`, `info-menu-5`, `info-menu-header` と `info-title-`*n*`-face` (N はセクションレベルで，1 から 4 の間の数です)，概観をカスタマイズするため，*M-x customize-face RET face RET*と入力してくださ い – *face*はここでリストアップされているものの一つになります.

`Info-hide-note-references`
: 既に説明したように，Emacs の Info では通常メニューと相互参照に含まれるテキストの一部を隠します．このオプションを `nil` に設定することで，この機能を完全に無効にすることができます．この変数を `nil` でも `t`でもない値に設定すると，中間の振舞い，役立つすべてのテキストは表示するが限定的な用途しかないテキストは隠すようになります.

`Info-scroll-prefer-subnodes`
: `nil` でない値に設定すると，`SPC` と `BACKSPACE` (あるいは `DEL`) キーをメニューで入力すると，最後か最初までスクロールする前にサブノードへ移動します．例えば，ノードのメニューがスクリーンに表われると，次に入力する `SPC` でメニューに含まれるサブノードへ移動できます．このオプションを `nil`に設定すると，スタンドアローンの Info リーダと似た動作になります．つまり，現在のノードの最後で入力した時のみ最初のサブノードへ移動します．デフォルトは `nil`です.

`Info-enable-active-nodes`
: `nil`でない値が設定されているとき，ノードに関連付けされた Lisp コードを Info が実行すること可能にします．Lisp コードはノードが選択されたとき実行されます．実行される Lisp コードにはノードデリミタ (‘`DEL`’文字) と以下のような‘`execute: `’タグを続けるべきです.

```
^_execute: (message "This is an active node!")
```

`Info-enable-edit`
: `nil`に設定し，‘`e`’(`Info-edit`) コマンドを利用不可にします．`nil`でない値では，それが可能です．See Section 2.3 [Add], page 12

3 Texinfo ファイルから Info ファイルを作成する *(2004/09/21)*

`makeinfo`は，Texinfo ファイルを Info ファイルに変換するユーティリティです．`texinfo-format-region`と `texinfo-format-buffer`は，同じことを行なう GNU Emacs の関数です．

Texinfo ファイルの書き方を学ぶため，See Section “Overview of Texinfo” in *Texinfo: The GNU Documentation Format* を参照ください

Texinfo ファイルから Info ファイルの作成する方法を学ぶため，See Section “Creating an Info File” in *Texinfo: The GNU Documentation Format* を参照ください

Info ファイルを作成後，インストール方法を学ぶため，See Section “Installing an Info File” in *Texinfo: The GNU Documentation Format* を参照ください

索引

以下は，すべてのコマンド，変数，そしてこのドキュメントで説明しているトピックのアルファベット順のリストです.

(Index is nonexistent)