

# GNU Emacs Reference Card

(for version 20)

## Emacs を開始する

GNU Emacs 20 を開始するにはその名前を入力する: `emacs`  
譜の名前を指定して編集する時は下記譜の操作 `Files` を参照

## Emacs を終了する

Emacs を中断する (X の時はアイコン化) `C-z`  
Emacs を完全に終了する `C-x C-c`

## 譜の操作 Files

譜を読み込む (read) `C-x C-f`  
譜を disk に書き出す (save) `C-x C-s`  
全ての譜を保存する `C-x s`  
別の譜の内容を作業領域に読み込む (insert) `C-x i`  
本当に欲しい譜の内容と取替える `C-x C-v`  
作業領域 (buffer) を指定した譜に書き出す `C-x C-w`  
版管理 (RCS) のための持出し・登録 `C-x C-q`

## 説明を読む Getting Help

`C-h` (か `F1`) を入力してその説明に従う。初めて使うのなら、  
`C-h t` と入力して、説明入門編 (tutorial) を開始する

説明の窓を消す `C-x 1`  
説明の次の窓に行く `C-M-v`  
文字列に一致した説明を表示 (apropos) `C-h a`  
一文字入力すると起動する関数を表示する `C-h c`  
関数についての説明 `C-h f`  
そのモードに設定されていることを表示 `C-h m`

## 問題からの復旧 Error Recovery

入力・あるいは実行を中断 (abort) `C-g`  
システム障害で失った譜を取戻す `M-x recover-file`  
変更を元に戻す (undo) `C-x u` or `C-_`  
作業領域を編集開始時の状態に戻す `M-x revert-buffer`  
乱れた画面を再描画 `C-l`

## 順次検索 Incremental Search

前方検索 `C-s`  
後方検索 `C-r`  
正規表現で検索 `C-M-s`  
正規表現で逆方向検索 `C-M-r`  
直前検索一致文字列を選択 prev `M-p`  
次に見つかる検索一致文字列を選択 next `M-n`  
順次検索を終了 `RET`  
最終入力文字による移動を取消 `DEL`  
検索を中止 `C-g`

検索を繰返すには `C-s` か `C-r` を入力するとその向きに続ける。  
検索中に `C-g` を入力すると最後の検索だけを中止する

## 移動 Motion

【移動の単位】  
文字 `C-b` 後方 `C-f` 前方  
単語 `M-b` `M-f`  
行 `C-p` `C-n`  
行頭・行末 `C-a` `C-e`  
文章 `M-a` `M-e`  
段落 `M-{` `M-}`  
頁 `C-x [` `C-x ]`  
S 式 `C-M-b` `C-M-f`  
関数 `C-M-a` `C-M-e`  
作業領域の最初・最後 `M-<` `M->`  
次の画面に移動 `C-v`  
前の画面に移動 `M-v`  
左に移動 `C-x <`  
右に移動 `C-x >`  
現在行を画面の中央に表示するように移動 `C-u C-l`

## 切取と消却 Killing and Deleting

【切取・消却の対象】  
文字 (切取でなく消却) `DEL` `C-d`  
単語 `M-DEL` `M-d`  
行 (端まで) `M-O C-k` `C-k`  
文 `C-x DEL` `M-k`  
S 式 `M-- C-M-k` `C-M-k`  
範囲 (region = 印と点の間) を切取 `C-w`  
範囲を切取保存 (kill ring) に複写 `M-w`  
文字まで切取 `M-z` 文字  
最後に切取ったものを貼付 `C-y`  
直前の貼付を一つ前の切取保存と取替 `M-y`

## 印を付ける Marking

ここに印 mark を付ける `C-@` or `C-SPC`  
印と点 (point = 現失印位置の直前) を交換 `C-x C-x`  
印を引数の数の語だけ減らす `M-@`  
段落に印を付ける `M-h`  
頁に印を付ける `C-x C-p`  
S 式に印を付ける `C-M-@`  
関数に印を付ける `C-M-h`  
作業領域全体に印を付ける `C-x h`

## 対話置換 (Query Replace)

対話形式での文字列置換を開始 `M-%`  
正規表現を使って同じく置換 `M-x query-replace-regexp`  
【その後の入力として有効な鍵は】  
一つ置換える。次に行く `SPC`  
一つ置換える。とどまる `,`  
置換えないで次に行く `DEL`  
残りを全て置換えてしまう `!`  
一つ前の置換を元に戻す `~`  
対話置換 (Query Replace) から抜ける `RET`  
再帰編集に入る。(抜けるには `C-M-c`) `C-r`

## 複数窓 Multiple Windows

(左右に二つ書いてある場合、右側は他の枠に作用する)

他の窓を全て消す `C-x 1`  
上下に二つの窓を開ける `C-x 2` `C-x 5 2`  
この窓を消す `C-x 0` `C-x 5 0`  
左右に二つの窓を開ける `C-x 3`  
他の窓の中を上下に移動する scroll `C-M-v`  
もう一つ窓に移動 `C-x o` `C-x 5 o`  
他の窓の作業領域を選択する `C-x 4 b` `C-x 5 b`  
他の窓に作業領域を表示する `C-x 4 C-o` `C-x 5 C-o`  
他の窓で譜を開ける find-file `C-x 4 f` `C-x 5 f`  
他の窓で譜を書込無で開ける read `C-x 4 r` `C-x 5 r`  
他の窓で Dired を開ける `C-x 4 d` `C-x 5 d`  
他の窓で名札 tag を探す `C-x 4 .` `C-x 5 .`  
窓を上下に大きくする `C-x ^`  
窓の左右をせまくする `C-x {`  
窓の左右を広くする `C-x }`

## 整形 Formatting

現在行を段落化する (モードによる) `TAB`  
範囲を段落化する (モードによる) `C-M-\`  
S 式を段落化する (モードによる) `C-M-q`  
範囲を引数だけ必ず段落化する `C-x TAB`  
現在の点の後に改行を挿入 `C-o`  
行の残りを一つの行に下に移動 `C-M-o`  
点の上下の空白行を削除 `C-x C-o`  
直前行 (引数付で次行) とつないで一行に `M-^`  
点の両側の全ての空白を削除 `M-\`  
点にちょうど一つの空白を挿入 `M-SPC`  
段落を整形 `M-q`  
整形幅 fill column を設定 `C-x f`  
各行の左端に付ける文字列 prefix を設定 `C-x .`  
顔 face を設定 `M-g`

## 大小文字変換 Case Change

単語を大文字にする `M-u`  
単語を小文字にする `M-l`  
単語の最初の文字を大文字にする `M-c`  
範囲を大文字にする `C-x C-u`  
範囲を小文字にする `C-x C-l`

## 入力窓 The Minibuffer

入力窓では次の鍵が使用出来る:  
可能な限り補完する `TAB`  
一語だけ補完する `SPC`  
補完した上で実行する `RET`  
補完の候補を表示 `?`  
直前の入力を取戻す `M-p`  
何を入力するか、或いは初期値を取戻す `M-n`  
正規表現で履歴を後方検索 `M-r`  
正規表現で履歴を前方検索 `M-s`  
操作を中止 `C-g`  
`C-x ESC ESC` と入力すると、直前の操作を呼出して編集 (変更) 出来る。`F10` を使えば、メニュー棒を表示する

# GNU Emacs Reference Card

## 作業領域 Buffers

他の作業領域を(名前で)呼出す C-x b  
 全ての作業領域名を表示する C-x C-b  
 いまの作業領域を捨てる C-x k

## 前後の入替 Transposing

文字を入替える C-t  
 単語を入替える M-t  
 行を入替える C-x C-t  
 S式を入替える C-M-t

## 綴りの確認 Spelling Check

現在の単語を調べる M-\$  
 範囲にある全ての単語を調べる M-x ispell-region  
 作業領域の全ての単語を調べる M-x ispell-buffer

## 名札 Tags

名札の定義を探す M-.  
 次の名札を探す C-u M-.  
 新しい名札定義の譜を指定 M-x visit-tags-table  
 名札表の全ての譜を正規表現検索 M-x tags-search  
 全ての譜で対話置換を実行 M-x tags-query-replace  
 名札検索や対話置換を続ける M-,

## シェル Shells

シェル操作を実行する M-!  
 シェル操作を範囲に対して実行 M-|  
 範囲をシェル操作で篩にかける filter C-u M-|  
 シェル窓 \*shell\* を開け対話を開始する M-x shell

## 長方形 Rectangles

長方形をレジスタに写す C-x r r  
 長方形を切取る C-x r k  
 長方形を貼付ける C-x r y  
 長方形を挿入(現在文字を右にずらす) C-x r o  
 長方形領域を空白にする C-x r c  
 各行に(固定)文字列を付ける C-x r t

## 省略 Abbrevs

全域省略を追加 C-x a g  
 モード限定省略を追加 C-x a l  
 この省略に全域定義を追加 C-x a i g  
 この省略にモード限定定義を追加 C-x a i l  
 明示的に省略を展開 C-x a e  
 直前の単語を動的に展開 M-/

## 正規表現 Regular Expressions

改行を除く全ての一字に一致 . (dot)  
 無か一つ以上の繰返に一致 \*  
 一つ以上の繰返に一致 +  
 無か一つだけの繰返 ?  
 正規表現で使う文字 c をそのまま使う時 \c  
 「または」("or") 演算子 \|  
 グループ化、算術かっこ \( ... \)  
 n 番目のグループに一致した文字列と同じもの \n  
 単語の区切 \b  
 単語の区切でない \B  
 【一致するもの】 開始に一致 終了に一致  
 行 ~ \$  
 単語 \< >  
 作業領域 \'  
 【文字の種類】 それらに一致 それら以外に一致  
 明示的に指定 [...] [^ ... ]  
 単語形式 \w \W  
 c という形式の文字 \sc \Sc

## 国際文字集 International Character Sets

第一言語を指定 M-x set-language-environment  
 全ての入力方法を表示 M-x list-input-methods  
 入力方法の入切 C-\  
 次の操作の coding system を指定 C-x RET c  
 全ての coding systems を表示 M-x list-coding-systems  
 好みの coding system を選択 M-x prefer-coding-system

## 情報 Info

情報 (Info) を開始する C-h i  
 指定した関数・変数の情報を探す C-h C-i  
 【節 node の中での移動】  
 先に進む SPC  
 前に戻る DEL  
 節の最初に行く . (dot)  
 【節と節を行き来する】  
 次の節 next n  
 前の節 previous p  
 一つ上に戻る up u  
 項目を名前を選ぶ menu m  
 n(1-9) 番目の項目を選ぶ n  
 相互参照 cross reference に従う(1で戻る) f  
 最後に見た節に戻る last l  
 案内節 directory node に戻る d  
 名前指定して任意節に移動 go g  
 【その他】  
 説明入門編 tutorial を開始 h  
 情報 Info を終了 q  
 節を正規表現で検索 M-s

## レジスタ Registers

範囲をレジスタに保存 C-x r s  
 レジスタ内容を作業領域に挿入 C-x r i  
 点の値(位置)をレジスタに保存 C-x r SPC  
 レジスタに保存してある値(位置)に移動 C-x r j

## 鍵盤手続 Keyboard Macros

鍵盤手続きの登録を開始 C-x (  
 鍵盤手続きの登録を終了 C-x )  
 最後に登録した鍵盤手続を実行 C-x e  
 最後に登録した鍵盤手続に追加 C-u C-x (  
 最後の登録に名前を付ける M-x name-last-kbd-macro  
 Lisp 定義形式で作業領域に挿入 M-x insert-kbd-macro

## Emacs Lisp の取扱い操作

点の前までの S 式を評価する C-x C-e  
 現 defun を評価する C-M-x  
 範囲を評価する M-x eval-region  
 入力窓から読んで評価する M-:  
 標準システム directory から譜を読む M-x load-library

## 簡単な個人環境設定 Simple Customization

変数と顔 faces の設定 M-x customize  
 【全体に有効な鍵の割当てを Emacs Lisp で設定する例】  
 (global-set-key "\C-cg" 'goto-line)  
 (global-set-key "\M-#" 'query-replace-regexp)

## 操作を用意する Writing Commands

(defun 操作名 (引数)  
 "説明"  
 (interactive "型")  
 本体)  
 【例】  
 (defun this-line-to-top-of-window (line)  
 "行を画面の一番上に表示するように移動。  
 引数付の時は引数の行に点を設定"  
 (interactive "P")  
 (recenter (if (null line)  
 0  
 (prefix-numeric-value line))))  
 ここで、interactive の型とはどのように引数を取るかを表  
 わす。C-h f interactive とすれば詳しい説明が得られる。

Copyright © 1997 Free Software Foundation, Inc.  
 v2.2 for GNU Emacs version 20, June 1997  
 designed by Stephen Gildea

Permission is granted to make and distribute copies of this card provided the  
 copyright notice and this permission notice are preserved on all copies.

For copies of the GNU Emacs manual, write to the Free Software Foundation,  
 Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA