Fiche de référence GNU Emacs

(pour la version 21)

Démarrer Emacs

Pour démarrer GNU Emacs 21, taper simplement son nom : `emacs`

Pour charger un fichier à éditer, voir Fichiers, ci-dessous.

Quitter Emacs

suspend Emacs (sous X, l'iconifie)	`C-z`
quitte définitivement Emacs	`C-x C-c`

Fichiers

charge un fichier dans Emacs	`C-x C-f`
sauve un fichier sur disque	`C-x C-s`
sauve **tous** les fichiers	`C-x s`
insère un autre fichier dans ce tampon	`C-x i`
remplace ce fichier par un autre	`C-x C-v`
écrit le tampon dans un fichier spécifié	`C-x C-w` *(write)*
checkin/checkout du contrôle de version	`C-x C-q`

Obtenir de l'aide

Le système d'aide est simple. Taper `C-h` (ou `F1`) et suivre les instructions. Pour les débutants, taper `C-h t` pour un **tutoriel**.

supprime la fenêtre d'aide	`C-x 1`
fait défiler la fenêtre d'aide	`C-M-v`
apropos : recherche de commande par chaîne	`C-h a`
montre la fonction lancée par une clé	`C-h c`
décrit une fonction	`C-h f`
informations sur les modes courants	`C-h m`

Gestion des erreurs

annule une commande en cours	`C-g`
récupère fichier après plantage système	`M-x recover-file`
annule un changement	`C-x u` or `C-_`
ramène un tampon à son contenu d'origine	`M-x revert-buffer`
redessine l'écran	`C-l`

Recherche incrémentale

cherche en avant	`C-s` *(search)*
cherche en arrière	`C-r` *(reverse s.)*
cherche sur expression rationnelle	`C-M-s`
cherche sur expression rationnelle en arrière	`C-M-r`
chaîne de recherche précédente	`M-p`
chaîne de recherche suivante	`M-n` *(next)*
fin de recherche incrémentale	`RET`
annule l'effet du dernier caractère	`DEL`
annule la recherche courante	`C-g`

Ré-utiliser `C-s` ou `C-r` pour répéter la recherche. Si Emacs cherche toujours, `C-g` annule seulement ce qui n'est pas déjà fait.

Mouvement

unité de mouvement	**en arrière**	**en avant**
caractère	`C-b`	`C-f` *(backward...)*
mot	`M-b`	`M-f` *(...forward)*
ligne	`C-p`	`C-n` *(previous/next)*
aller au début (ou fin) de ligne	`C-a`	`C-e` *(/end)*
phrase	`M-a`	`M-e`
paragraphe	`M-{`	`M-}`
page	`C-x [`	`C-x ]`
sexp	`C-M-b`	`C-M-f`
fonction	`C-M-a`	`C-M-e`
aller en début (ou fin) de tampon	`M-<`	`M->`

écran suivant	`C-v`
écran précédent	`M-v`
fait défiler vers la gauche	`C-x <`
fait défiler vers la droite	`C-x >`
place la ligne courante au centre de l'écran	`C-u C-l`

Détruire et effacer

entité à détruire	**en arrière**	**en avant**
caractère (effacer, pas détruire)	`DEL`	`C-d` *(delete)*
mot	`M-DEL`	`M-d`
ligne (jusqu'à fin de)	`M-0 C-k`	`C-k` *(kill)*
phrase	`C-x DEL`	`M-k`
sexp	`M-- C-M-k`	`C-M-k`

détruit la **région**	`C-w`
copie la région dans le kill ring	`M-w`
détruit jusqu'à la prochaine occurence de *car*	`M-z` *car*
ramène la dernière chose détruite	`C-y` *(yank)*
remplace le dernier *yank* avec le dernier détruit	`M-y`

La marque

place la marque ici	`C-@` ou `C-SPC`
échange le point et la marque	`C-x C-x`
place la marque *arg* **mots** plus loin	`M-@`
marque le **paragraphe**	`M-h`
marque la **page**	`C-x C-p`
marque la **sexp**	`C-M-@`
marque la **fonction**	`C-M-h`
marque tout le **tampon**	`C-x h`

Remplacement

remplacement interactif d'une chaîne	`M-%`
avec les expressions rationnelles	`M-x query-replace-regexp`

Les réponses valides en mode query-replace sont

remplace celui-ci, passe au suivant	`SPC`
remplace celui-ci, ne bouge pas	`,`
saute au suivant sans remplacer	`DEL`
remplace tous les suivants	`!`
retourne au *match* précédent	`^`
quitte query-replace	`RET`
édition récursive (`C-M-c` pour sortir)	`C-r`

Fenêtres multiples

Quand il y a deux commandes, la 2ième est pour "l'autre trame".

efface toutes les autres fenêtres	`C-x 1`	
coupe la fenêtre horizontalement	`C-x 2`	`C-x 5 2`
efface cette fenêtre	`C-x 0`	`C-x 5 0`
coupe la fenêtre verticalement	`C-x 3`	
fait défiler l'autre fenêtre	`C-M-v`	
place le curseur dans une autre fenêtre	`C-x o`	`C-x 5 o`
sélectionne un tampon dans une autre fenêtre	`C-x 4 b`	`C-x 5 b`
affiche un tampon dans une autre fenêtre	`C-x 4 C-o`	`C-x 5 C-o`
lit un fichier dans un autre fenêtre	`C-x 4 f`	`C-x 5 f`
lit un fichier en lecture seule dans...	`C-x 4 r`	`C-x 5 r`
lance Dired dans une autre fenêtre	`C-x 4 d`	`C-x 5 d`
trouve un tag dans une autre fenêtre	`C-x 4 .`	`C-x 5 .`
agrandit la fenêtre	`C-x ^`	
réduit la fenêtre	`C-x {`	
élargit la fenêtre	`C-x }`	

Formattage

indente la **ligne** courante (dépend du mode)	`TAB`
indente la **région** (dépend du mode)	`C-M-\`
indente la **sexp** (dépend du mode)	`C-M-q`
indente la région de *arg* colonnes	`C-x TAB`
insère une fin de ligne après le point	`C-o`
descend le reste de la ligne	`C-M-o`
efface les lignes blanches autour du point	`C-x C-o`
joint la ligne avec la précédente (si arg, suivante)	`M-^`
efface tous les blancs autour du point	`M-\`
place exactement un blanc au point	`M-SPC`
formatte le paragraphe	`M-q`
fixe la colonne de formattage	`C-x f`
fixe le préfixe de chaque ligne	`C-x .`
fixe la "face"	`M-g`

Changement de casse

place le mot en majuscules	`M-u` *(uppercase)*
place le mot en minuscules	`M-l` *(lowercase)*
place le mot en capitales	`M-c`
place la région en majuscules	`C-x C-u`
place la région en minuscules	`C-x C-l`

Le minitampon

Les touches/clés suivantes sont définies dans le minitampon.

complète autant que possible	`TAB`
complète un seul mot	`SPC`
complète et exécute	`RET`
montre les complétions possibles	`?`
ramène l'entrée précédente du minitampon	`M-p`
ramène l'entrée suivante du minitampon	`M-n` *(next)*
cherche une exp. rat. en avant dans l'hist.	`M-s` *(search)*
cherche une exp. rat. en arrière dans l'hist.	`M-r` *(reverse s.)*
annule la commande	`C-g`

Taper `C-x ESC ESC` pour éditer et répéter la dernière commande qui a utilisé le minitampon. Taper `F10` pour utiliser la barre de menus depuis le minitampon.

Fiche de référence GNU Emacs

Tampons

sélectionne un autre tampon	`C-x b` *(buffer)*
liste tous les tampons	`C-x C-b`
détruit un tampon	`C-x k` *(kill)*

Transposition

transpose des **caractères**	`C-t`
transpose des **mots**	`M-t`
transpose des **lignes**	`C-x C-t`
transpose des **sexps**	`C-M-t`

Vérification orthographique

vérifie l'orthographe du mot courant	`M-$`
... de tous les mots de la région	`M-x ispell-region`
... de tout le tampon	`M-x ispell-buffer`

Tags

trouve un tag (une définition)	`M-.`
trouve la prochaine occurrence du tag	`C-u M-.`
spécifie un nouveau fichier de tags	`M-x visit-tags-table`
recherche exp. rat. sur tous les fichiers	`M-x tags-search`
lance query-replace sur tous les fichiers	`M-x tags-query-replace`
continue le dernier tags-search ou query-replace	`M-,`

Shells

exécute une commande shell	`M-!`
lance une commande shell sur la région	`M-|`
filtre la région par une commande shell	`C-u M-|`
démarre un shell dans la fenêtre `*shell*`	`M-x shell`

Rectangles

copie le rectangle dans un registre	`C-x r r`
détruit le rectangle	`C-x r k` *(kill)*
ramène le rectangle	`C-x r y` *(yank)*
ouvre un rectangle en décalant le texte à droite	`C-x r o`
vide le rectangle	`C-x r c` *(clear)*
préfixe chaque ligne avec une chaîne	`C-x r t`

Abbrevs

ajoute une abbrev globale	`C-x a g`
ajoute une abbrev locale au mode	`C-x a l`
ajoute l'expansion globale pour cette abbrev	`C-x a i g`
ajoute l'exp. locale au mode pour cette abbrev	`C-x a i l`
dilate *(expand)* explicitement cette abbrev	`C-x a e` *(expand)*
dilate dynamiquement le mot précédent	`M-/`

Expressions rationnelles

tout caractère sauf fin de ligne	`.` (point)
zéro ou plus répétition(s)	`*`
une ou plus répétition(s)	`+`
zéro ou une répétition	`?`
quote un caractère spécial d'exp. rat. *c*	`\`*c*
alternative ("ou")	`|`
groupement	`\( ... \)`
même texte que le *n*ième groupe	`\`*n*
sur coupure de mot *(break)*	`\b`
pas sur coupure de mot	`\B`

entité	**en début**	**en fin**
ligne	`^`	`$`
mot	`\<`	`\>`
tampon	``\` ``	`\'`
classe de caractères	**ceux-là**	**les autres**
ensemble explicite	`[ ... ]`	`[^ ... ]`
caractère de mot	`\w`	`\W`
caractère de syntaxe *c*	`\s`*c*	`\S`*c*

Caractères internationaux

spécifie le langage principal	`M-x set-language-environment`
montre toutes les méthodes d'entrée	`M-x list-input-methods`
valide/invalide une méthode d'entrée	`C-\`
fixe le système de codage pour la proch. commande	`C-x RET c`
montre tous les systèmes de codage	`M-x list-coding-systems`
choisit le système de codage préféré	`M-x prefer-coding-system`

Info

entre dans le lecteur de documentation Info	`C-h i`
trouve une variable ou fonction dans Info	`C-h C-i`

Déplacement dans un nœud :

fait défiler en avant	`SPC`
fait défiler en arrière	`DEL`
début de nœud	`.` (point)

Déplacement entre les nœuds :

nœud **suivant**	`n` *(next)*
nœud **précédent**	`p`
remonte	`u` *(up)*
sélectionne une entrée par son nom	`m`
sélectionne la *n*ième entrée (1–9)	*n*
suit le renvoi (retour avec `l`)	`f` *(follow)*
retour au dernier nœud visité	`l` *(last)*
retour au nœud répertoire	`d` *(directory)*
va à un nœud par son nom	`g` *(goto)*

Autres :

lance le **tutoriel** Info	`h` *(help)*
quitte Info	`q`
cherche le nœud contenant une exp. rat.	`M-s` *(search)*

Registres

sauve la région dans un registre	`C-x r s`
insère le contenu du registre dans le tampon	`C-x r i`
sauve la valeur du point dans un registre	`C-x r SPC`
saute au point sauvé dans un registre	`C-x r j` *(jump)*

Macros clavier

commence à définir une macro	`C-x (`
termine la définition de la macro	`C-x )`
exécute la dernière macro définie	`C-x e`
ajoute à la dernière macro	`C-u C-x (`
nomme la dernière macro	`M-x name-last-kbd-macro`
insère la définition Lisp dans le tampon	`M-x insert-kbd-macro`

Commandes concernant l'Emacs Lisp

évalue la **sexp** précédant le point	`C-x C-e`
évalue la **defun** courante	`C-M-x`
évalue la **région**	`M-x eval-region`
lit et évalue le minitampon	`M-:`
lit depuis un répertoire système standard	`M-x load-library`

Personnalisation simple

Personnalise les variables et les "faces"	`M-x customize`

Assignation globale de clés en Emacs Lisp (exemples) :

```
(global-set-key "\C-cg" 'goto-line)
(global-set-key "\M-#" 'query-replace-regexp)
```

Écrire des commandes

```
(defun nom-commande (args)
  "documentation" (interactive "patron")
  corps)
```

Un exemple :

```
(defun cette-ligne-en-haut-de-la-fenetre (ligne)
  "Positionne la ligne courante en haut de la fenetre.
Avec ARG, place le point sur la ligne."
  (interactive "P")
  (recenter (if (null line)
                0
              (prefix-numeric-value line))))
```

La chaîne `interactive` spécifie comment obtenir les arguments interactivement. Taper `C-h f interactive` pour plus de détails.

v 2.2 pour GNU Emacs version 21, juin 1997
conçu par Stephen Gildea
traduit par Christophe Deleuze, décembre 2002