
Emacs

1 Principes de base

1.1 Lancement

Lancez l'éditeur de texte `emacs` en arrière plan.

- tapez `C-h` (la notation `C-<c>` représente la combinaison des touches CONTROL et `<c>`); le message suivant apparaît dans la mini-fenêtre en bas de l'écran :

`C-h (Type? for further options)`

- tapez `C-h` à nouveau, ou bien taper ?.
- tapez `C-g` pour revenir à l'état initial : `C-g` permet l'abandon du traitement en cours

1.2 Description de l'écran

Remarquez en bas de la fenêtre ce qu'on appelle le mini-buffer, qui sert à emacs pour communiquer avec vous. Lorsque vous tapez une commande, elle s'affiche dans le mini-buffer.

Au dessus se trouve, avec en gris « GNU Emacs », la « ligne de mode » (mode line en anglais). Elle donne des indications telles que le nom du fichier en cours d'édition, éventuellement le numéro de la ligne où se trouve votre curseur, etc.

1.3 Utilisation du clavier

La plupart des commandes d'Emacs utilisent les touches CONTROL ou ESC .

- la touche CONTROL s'utilise en combinaison avec un caractère `<c>` (comme pour la touche SHIFT)
- la touche ESC doit être relâchée, comme un caractère à part entière.
- notations :
 - `C-f` : CONTROL + `f`
 - `M-f` : ESC puis `f` ou ALT + `f` ; se lit *méta-f*
- **clé** : caractère ou suite de caractères *attaché* à une commande

2 Mécanismes d'aide

2.1 Description des clés, commandes et liaisons

2.1 EXERCICE Tapez `C-h` deux fois, commentez les différents messages. Parmi les options proposées, essayez le caractère `c`.

Essayez ensuite `C-h c` sur plusieurs clés : `C-a`, `M-c`, `C-x C-s`,... Quel résultat donne-t-elle pour un caractère ordinaire ?

Essayez `C-h k` sur les mêmes clés que précédemment : comment `emacs` modifie-t-il son environnement ? (ouverture d'une seconde «fenêtre» associé à un buffer spécial appelé `*Help*` ; pour revenir à la fenêtre principale, tapez `C-x 1`)

2.2 EXERCICE Essayez `C-h w` sur des noms de commandes, par exemple `find-file`, `kill-line`, `center-line`, en utilisant le mécanisme de complétion.

Toutes les commandes n'étant pas liées à une clé, il est nécessaire de pouvoir demander la description d'une commande par son nom : **C-h f**.

Essayez **C-h f** sur les mêmes commandes que précédemment.

2.3 EXERCICE (POUR CEUX QUI SONT EN AVANCE) La commande **describe-bindings** liée à la clé **C-h b** donne la liste des liaisons. Parcourir cette liste, trouvez les différentes commandes d'aide.

2.2 Description du mode

Pour connaître le mode de fonctionnement dans la fenêtre courante, utilisez la clé

C-h m

On se trouve dans le mode fondamental, qui correspond au fonctionnement par défaut.

2.4 EXERCICE Changez de mode en tapant **M-x text-mode** (pensez à utiliser le mécanisme de complétion), puis retapez **C-h m**; essayez les nouvelles commandes disponibles dans ce mode. Demandez de l'aide sur la combinaison de clés **M-o M-s**. Que peut-on en conclure sur le mécanisme d'aide?

2.3 Recherche d'une information - Pour ceux qui sont en avance

La commande **apropos**, liée à la clé **C-h a**, permet de rechercher une information à partir d'un mot clé :

C-h a
Command apropos (regexp): center

2.5 EXERCICE (POUR CEUX QUI SONT EN AVANCE) Recherchez une commande qui permet d'insérer un fichier dans le buffer.

3 Edition de fichiers

3.1 Chargement d'un fichier

Utilisez la clé **C-x C-f** *attachée* à la commande **find-file**. Pour le besoin de l'exercice recopiez dans votre **home directory** le fichier `/usr/share/doc/python3.1/README.gz`, puis décompressez-le à l'aide de la commande **gunzip**.

3.1 EXERCICE Exemple de chargement de fichier

- tapez **C-x** : *emacs* attend la suite;
- tapez **C-f** : *emacs* exécute la commande **find-file** en affichant dans la mini-fenêtre le message


```
Find file: ~/
```
- possibilité d'obtenir la liste des complétions en tapant **?** ou **TAB** : *emacs* ouvre une seconde fenêtre, nommée ***Completions*** pour afficher la liste des complétions; taper ensuite **R C-i** (complétion également possible avec **SPC**)
- tapez **RETURN** pour valider la proposition `~/README`. La *fenêtre de complétion* est refermée et le chargement du fichier s'exécute.

Lors du chargement la fenêtre ***scratch*** est remplacée par une fenêtre de nom **README** dans laquelle est affiché le contenu du fichier **README**; on dit que la fenêtre *est ouverte sur* le fichier **README**.

- Tapez un caractère : il est inséré dans le buffer. Le fichier n'a pas changé.
 - buffer différent du fichier : `***-Emacs:...`
 - **nécessité de sauvegarder les modifications.**

3.2 Sauvegarde du buffer

Il s'agit de la commande `save-buffer` liée à la clé `C-x C-s`. Lors de la sauvegarde, affichage d'un message :

Wrote /.../README

3.2 EXERCICE Après la sauvegarde, observez la ligne de mode : pourquoi les deux étoiles `**` ont-elles disparu ? Retapez `C-x C-s`. Commentez.

Tapez `C-x C-w`, puis tapez `README2` dans la mini-fenêtre. Quel est l'effet de cette commande ?

4 Commandes de base

4.1 Déplacement

- essayez `C-⟨fleche⟩` pour les différentes flèches du clavier
- `C-l` : recentrage vertical du texte en fonction du curseur et réaffichage
- Déplacez-vous dans le fichier à l'aide de `M->`, `M-<` ou `M-g g` suivi d'un numéro de ligne.

4.2 Suppression

4.1 EXERCICE Insérer, puis supprimer des caractères, des mots, des lignes, en utilisant les clés `C-d`, `M-d`, `M-DEL` et `C-k`. Essayez également :

- `M-u`, `M-l` et `M-c`
- `M-\` (entre deux mots), `M-SPC`.

4.3 Copier-coller

On peut sélectionner une portion de texte soit à la souris, soit au clavier. Pour sélectionner une région au clavier il faut d'abord déposer une marque au début du texte que l'on veut sélectionner en tapant `C-SPC`, puis il suffit de se déplacer avec les flèches. La région sélectionnée apparaît surlignée.

Pour copier cette région tapez `M-w`, pour la couper tapez `C-w`. Le texte coupé est placé dans une sorte de tampon : le *kill ring*. Pour le coller, tapez `C-y`. Il faut noter qu'il y a un historique des "copier" et "couper". Ainsi, `C-y` colle la dernière chose à avoir été copiée ou coupée, vous pouvez taper ensuite plusieurs fois `M-y` et les textes précédents dans l'historique sont collés.

4.2 EXERCICE Effectuez des copier-coller au clavier dans le buffer `README`. Testez l'historique des "copier" et "couper".

4.4 Complétion

La clé `M-/` vous permet de compléter un mot en cours de frappe, si vous l'avez déjà utilisé précédemment dans le buffer. Par exemple si vous devez taper à plusieurs reprises un même nom de fonction ou de variable, la deuxième fois, tapez simplement le début du nom puis tapez `M-/`. Emacs complètera automatiquement. Si plusieurs complétions sont possibles, tapez de nouveau `M-/` et emacs vous fera d'autres propositions.

4.3 EXERCICE Récupérez sur la page des tds le fichier `tableaux.py`. Ouvrez un nouveau fichier de nom `init.py`. Ecrivez dans ce fichier en première ligne `#!/bin/sh`¹. En utilisant le mécanisme de complétion, écrivez ensuite dans le fichier les commandes shell pour effectuer les actions suivantes :

- créer un répertoire de nom `ProjetPython`,
- copier dans ce répertoire le fichier `tableaux.py`,
- se déplacer dans le répertoire `ProjetPython`,

1. cette ligne garantit que le fichier sera exécuté par un shell de type `sh`

- lancer la commande `python3 tableaux.py` en redirigeant le résultat dans un fichier de nom `resultat.txt`,
- afficher le contenu du fichier `resultat.txt` sur la sortie standard.

Depuis la console, changez les droits d'accès du fichier `init_py` pour le rendre exécutable, puis lancez-le en tapant :

```
./init_py
```

4.5 Recherche et remplacement

- la clé `C-s` permet la recherche incrémentale d'une chaîne de caractères que l'on tape dès qu'apparaît dans le mini-buffer l'invite *I-Search* : `.` Notez que la recherche se fait au fur et à mesure que l'on tape. Chaque `C-s` ensuite va vers la prochaine occurrence de la chaîne dans le buffer (vers l'avant). La touche `Esc` arrête la recherche.
- la clé `C-r` permet une recherche en arrière dans le buffer. On tape la chaîne à rechercher dès qu'apparaît l'invite *I-Search backward* : `.`
- la combinaison de clés `C-x C-k` (ou `ESC %`) permet de remplacer différentes occurrences d'une chaîne de caractères par une autre chaîne. Tapez la chaîne à remplacer dès l'invite *Query Replace* : `,` puis tapez `RETURN`. Tapez ensuite la nouvelle chaîne dès l'invitation *With* : `.` Emacs surligne une occurrence de la chaîne dans le buffer et vous demande si oui ou non il faut la remplacer. Répondre : `SPC` ou `y` pour oui, `DEL` ou `BACKSPACE` ou `n` pour non, pour remplacer toutes les occurrences dans le buffer (sans demande de confirmation).

4.4 EXERCICE Recherchez dans le fichier `README` les occurrences de la chaîne `install`. Placez-vous à la fin du fichier et recherchez en arrière la chaîne `source`. Remplacez les occurrences du mot `versions` apparaissant dans un titre de paragraphe par `Versions`.

4.6 Le paramètre universel : `C-u` - Pour ceux qui sont en avance

4.5 EXERCICE Essayez les commandes suivantes :

```
C-u !
C-u C-u *
C-u 3 >
C-u C-u SPC
C-u C-d
C-u C-u M-u C-u 10 C-k
```

Le préfixe `C-u` permet de transmettre un paramètre, en général numérique, à la commande qui suit. La plupart du temps, il est interprété comme un facteur de répétition. peut avoir une autre signification, comme dans les commandes `C-u 0 C-k` ou `C-u 2 C-v`.

REMARQUE 2 Essayez avec des nombres négatifs :

```
C-u -3 C-e, C-u -5 C-d, C-u -10 M-c.
```

4.7 “undo”

Les clés `C-_`² ou `C-x u` permettent d'annuler l'effet de la dernière commande. Il est possible d'enchaîner plusieurs *undo* et de remonter l'historique des modifications.

4.6 EXERCICE Édition du fichier `README` :

- cherchez une commande pour aligner les paragraphes et appliquez-la à des paragraphes de texte.

2. nécessite parfois l'utilisation du “shift” en même temps que du “control”

- mettez entièrement en majuscules les titres de paragraphes.
- revenez grâce au undo à la version initiale non capitalisée du fichier.

5 Liens et commandes

5.1 EXERCICE Essayez `M-x kill-line`. Utilisez le mécanisme de complétion de commandes : tapez `M-x k C-i` (*emacs* propose des complétions), puis tapez `i C-i`, puis `l C-i` (complétions également possibles avec `SPC`).

Essayez `C-u 6 M-x kill-line`.

5.2 EXERCICE Chaque clé est liée à une commande, au moyen d'un mécanisme appelé *binding* (liaison). La commande `global-set-key` permet de créer une nouvelle liaison. Recherchez de l'information sur cette commande.

Utilisez cette commande pour créer une liaison entre la touche `F9` et la commande `compile` (la commande `compile` permet de compiler des programmes sous *emacs*, par défaut elle lance la commande shell `make`³, mais vous pouvez remplacer `make` dans le mini-buffer par `python3` ou tout autre commande de compilation).

6 Tags

Lorsqu'un programme comporte plusieurs fichiers, le but des *tags* est de trouver rapidement l'endroit où est définie une fonction.

6.1 EXERCICE Récupérez le fichier `calcul.py`, copiez le dans le répertoire `ProjetPython` et ouvrez-le avec *Emacs*. Utilisez la touche `F9`, qui permet maintenant de lancer une compilation, et remplacez `make -k` par la commande `etags *.py`. Cette commande génère un fichier `TAGS` qui contient des informations sur l'emplacement des définitions de fonctions dans vos fichiers.

Dans *Emacs*, ouvrez le fichier `calcul.py` et placez le curseur sur la première lettre de `creer_tableau_1D`. Si vous tapez la clé `M-.` vous voyez apparaître dans le minibuffer l'invite `Find tag :(default creer_tableau_1D)`, vous pouvez soit taper un nom de fonction, soit prendre la valeur par défaut `creer_tableau_1D`. Dans ce cas tapez `RETURN` (puis `RETURN` à nouveau pour lui confirmer d'utiliser le fichier `TAGS`), alors *Emacs* ouvre le fichier `tableaux.py` en se plaçant directement à l'endroit où est définie la fonction.

Si vous voulez revenir dans le buffer précédent, tapez la clé `C-x b` qui permet de changer de buffer, puis `calcul.py` (ou `RETURN` si la valeur par défaut est la bonne).

Le programme `etags` reconnaît aussi les langages `C`, `C++`, `Java`, `Common Lisp` ...

7 Fenêtres

Emacs peut afficher plusieurs «fenêtres», chacune avec un texte différent (une «fenêtre» signifie ici une zone particulière dans la fenêtre d'*Emacs*).

7.1 EXERCICE Observez le comportement des commandes `C-x 2`, `C-x o`, puis à nouveau `C-x 2`, `C-x o`, `C-x o`, `C-x 1`, `C-x 3`, `C-x 0`. Quel est l'intérêt d'ouvrir plusieurs fenêtres sur le même buffer ? Mettez en pratique en plaçant une fenêtre au début d'un buffer et une seconde à la fin du même buffer.

Essayez `C-x ^` et `C-x }`.

Utilisez `C-x 4 C-f` pour ouvrir un fichier différent dans une autre fenêtre.

8 Jeux de caractères

Il existe de nombreux jeux de caractères pour encoder ses fichiers.

3. `make` est utilisée entre autre pour compiler les programmes écrits en langage `C`, cette commande ne peut fonctionner sans l'existence d'un fichier nommé `makefile`

- 1) Regardez la liste des jeux de caractères disponible dans Emacs en utilisant la commande :
`Option -> Mule (Multilingual Environment) -> Set Coding Systems -> For Saving This Buffer -> SPC`
- 2) De manière générale essayez le plus souvent possible d'utiliser le jeu de caractères `utf-8`⁴. Sélectionnez ce jeu de caractères. Quel est le raccourci clavier permettant de sélectionner un jeu de caractères ?
- 3) Vérifiez dans la ligne d'état d'Emacs que votre fichier est bien encodé en `utf-8`.

9 Clavier QWERTY et accents - Pour ceux qui sont en avance

Sur certains clavier, comme par exemple les claviers QWERTY, vous n'avez pas directement accès aux caractères accentés tels que : é, à, etc. Sur les claviers AZERTY vous n'avez généralement pas accès directement à tous les caractères accentués comme par exemple : À, É, etc. Sous Emacs il est possible d'utiliser le jeu de caractères ISO 8859-1 même avec un clavier QWERTY.

- 1) Regardez la liste des jeux de caractères que vous pouvez associer à votre clavier via la commande :
`Option -> Mule (Multilingual Environment) -> Select Input Method...-> SPC`
Quel est le raccourci clavier permettant de sélectionner le jeu de caractères associé à votre clavier ?
- 2) Pour utiliser le jeu de caractères ISO 8859-1, vous pouvez sélectionner (via la commande `Select Input Method...`) la méthode `latin-1-alt-postfix`.
- 3) Testez cette méthode en tapant par exemple successivement sur les touche E puis '. Testez cette méthode avec de nombreux caractères accentués.
- 4) Le raccourci clavier `C-\` est un commutateur, mais à quoi sert-il ?

4. Il s'agit des caractères Unicode, cf. www.unicode.org