

# Conventions pour pouvoir travailler ensemble

## Notations

Dans les divers documents du cours, nous utiliserons les notations suivantes.

## Substitutions

Les chaînes de caractère entre chevrons < > doivent être remplacées par leur signification. Lors de la substitution, les chevrons doivent être supprimés.

Exemple: si votre adresse mail est `alice@example.com`, alors la commande

```
$ git config user.email "<email_address>"
```

sera traduite en :

```
$ git config user.email "alice@example.com"
```

## Options

Les chaînes de caractère entre crochets [ ] sont facultatives (options).

Exemple: la commande

```
$ wc [-l] config
```

sera traduite en `wc config` ou en `wc -l config` selon l'information que vous souhaitez avoir sur le contenu du fichier `config`.

## Commandes shell

Les commandes shell peuvent être exécutées par l'utilisateur `root` ou un user non privilégié.

- les commandes introduites par `#` doivent être évaluées par l'utilisateur `root`.
- les commandes introduites par `$` doivent être évaluées par un utilisateur non privilégié.

Ces caractères ne font pas partie de la commande, ils servent à signaler quel utilisateur doit lancer la commande. C'est en général le dernier caractère du prompt du shell de l'utilisateur concerné.

Pour lancer une commande `<commande>` en tant que `root`, soit vous passez `root` avec la commande `su` puis vous tapez la commande `<commande>`, soit (si `sudo` est installé et que vous faites partie du groupe `sudo`) tapez la commande `sudo <commande>`

L'intérêt de faire commencer les commandes à taper en tant que `root` par un `#` est que le shell interprète tout ce qui suit le caractère `#` comme un commentaire. Ainsi, les personnes qui seraient tentées de copier/coller brutalement une ligne potentiellement dangereuse pour le système (ou dont la souris glisserait en sélectionnant une ligne de trop) seront protégées puisque `# <commande>` sera interprété comme un commentaire par le shell qui n'exécutera pas la commande.

Exemples:

```
# apt update
```

indique que la commande `apt update` doit être exécutée par l'utilisateur `root`.

```
$ less /etc/passwd
```

indique que la commande `less /etc/passwd` doit être exécutée par un user non-privilegié (en général l'user avec lequel vous vous êtes connecté·e).

## Traductions

La plupart de la documentation concernant l'administration système (sur le web ou les pages du manuel) est en anglais et plusieurs termes sont à connaître. Dans le cours, certains termes seront traduits du français vers l'anglais, la traduction sera mise entre parenthèses et précédés de **en:**, les termes anglais traduits en français seront précédés de **fr:**.

Exemple: « Poussez (en: push) vos changements vers le dépôt remote (fr: distant) »

## Partager ses messages d'erreur

Les messages d'erreur sont très utiles pour comprendre pourquoi quelque chose ne fonctionne pas. Il est donc important de pouvoir les partager de sorte à faciliter leur utilisation par la personne destinataire :

- ne faites **pas de copie d'écran** de votre terminal (sauf si on vous le demande explicitement). Préférez le texte. En effet, une façon de comprendre ce qui ne marche pas est de copier le message d'erreur dans un moteur de recherche. Avec une copie d'écran, vous imposez à votre interlocut·rice de recopier à la main lettre à lettre le message d'erreur.
- copiez **l'intégralité de la commande tapée et du message d'erreur qui suit**, ne le tronquez pas, ça permet à votre interlocut·rice d'avoir un peu plus de contexte.

## Utiliser mattermost

### Garder le fil

Les messages de mattermost sont rangés de façon hiérarchique (imaginez ce qui se passerait si tous les messages de toutes les personnes de l'université se retrouvaient à la suite les uns des autres).

Ainsi, mattermost est structuré sous forme d'une arborescence de profondeur 4, correspondant aux 4 colonnes de la page web (on en voit que 3 par défaut), de gauche à droite (la racine est à gauche) :

1. Les équipes, comme `BU` ou `12-info-2022-2023`
2. Les salons (en: channel), comme `sysadmin` ou `Centre-ville` qui sont rattachés à une équipe
3. Les fils (en: threads), qui sont rattachés à un salon. Un fil correspond à une discussion.
4. Les messages (en: posts), qui sont rattachés à un fil

Lorsque vous écrivez quelque chose dans un salon mattermost, mattermost crée un nouveau fil. Pour continuer le fil d'une discussion existante, il ne faut pas juste écrire en dessous (ça créerait un nouveau fil), il faut cliquer sur la flèche grise courbée **Répondre**. À ce moment apparaît une 4e colonne tout à droite qui liste tous les messages du fil. Tant que vous écrivez dans la 4e colonne tout à droite, vous restez dans le fil. Si vous écrivez dans la 3e colonne, vous créez un nouveau fil (sur un nouveau sujet).

Si une personne quitte son fil et en crée un nouveau alors qu'il s'agit de la même discussion, nous pouvons gentiment lui demander de garder son fil, de sorte à ne pas tout mélanger. Ainsi, la personne peut recopier son dernier message dans le fil précédent et supprimer le nouveau fil.

Quel est l'intérêt ? Il arrive souvent que plusieurs étudiant·es soient confronté·es aux mêmes problèmes. Le fait que les discussions soient ainsi rangées par fil, permet aux autres étudiant·es de le consulter.

Structurer les discussions en fils permet aussi de garder le contexte dans le temps. Imaginez plein de discussions en parallèle sans structuration en fils, il faudrait remonter pour se souvenir de quoi on parle.

Certains fils peuvent contenir des centaines de messages et s'étaler sur plusieurs jours (lorsqu'un·e étudiant·e a crashé sa machine par exemple), ainsi, avoir rangé tous les messages dans le même fil permet de suivre la discussion au long cours.

## Formater les blocs de code

Comme expliqué dans le paragraphe « Partager ses messages d'erreur », il est recommandé de partager vos commandes et messages d'erreur sous forme de texte.

Mattermost utilise le langage de formattage de texte `markdown`. Consultez le lien **Aide** en bas à droite de la fenêtre mattermost.

En particulier, si vous entourez un mot avec des étoiles, il sera mis en italique. Le problème lorsqu'on veut partager du code est que si ce code contient des étoiles, elles disparaîtront et le code sera illisible.

Pour que markdown n'interprète pas un bloc de code, il suffit de mettre trois accents graves ou « apostrophes inversées » (en: backquote) avant et après le bloc. Un backquote s'obtient avec la combinaison de touches **AltGr+7**. Il est même possible de spécifier le langage de programmation utilisé (dans notre cas `bash`) pour bénéficier de la coloration syntaxique.

Si plusieurs commandes sont mises à la suite, laissez une ligne vide avant chaque nouvelle commande.