



Le terminal

🕒 Date de création	@28 mai 2024 16:36
📁 Cours	Terminal
📁 Type	O'clock
☑ Relue ?	☑
☰ Mise en page ok ?	oui
☑ Prise de note ajoutée	☑

Le Terminal ?

- C'est la "ligne de commande"
- Différent de l'interface graphique habituelle
- On écrit ce qu'on veut lui faire faire :
 - en tapant des commandes

Le terminal est sensible à la CASS il faut donc bien faire attention à respecter les espaces, les points, mes majuscules, minuscules.

Le contexte d'exécution

Lorsqu'on ouvre un Terminal, on a quelque chose comme...

jean@machine:/var/www/html\$

- "jean" est ici le nom de l'**utilisateur**
- "machine" est ici le nom de l'**ordinateur**
- "/var/www/html" est ici le **répertoire dans lequel on est**

- "\$" est un caractère signifiant "**tout ce qui suit ce caractère sera exécuté comme une commande**"
- Il existe 2 types de chemins, en relatif ou en absolu
 - relatif : partir de la ou je suis pour aller quelque part, dans un dossier etc
 - absolu : repartir de la racine peu importe la ou je suis
 - les deux s'utilisent il faut simplement choisir le plus pratique/ rapide selon le contexte
 - ex : cd .. → remonter dans le dossier de manière relative
 - cd /home/student → remonter de manière absolue

Un peu de pratique

- On ouvre une fenêtre de Terminal.
- les devs disent "on ouvre un terminal"
- Et pour savoir dans quel répertoire on se trouve on tape la commande pwd et on l'exécute avec la touche Enter.
- pwd signifie **P**rint **W**orking **D**irectory
- Tu viens de lancer ta première commande dans le Terminal!

Changer de répertoire

- Depuis le Terminal, on veut se placer dans le dossier /var/www/html.
- Pour changer de répertoire, on tape **cd suivi du chemin**.
- Dans notre cas on tape cd /var/www/html puis touche Enter.
- cd signifie "**C**hange **D**irectory"
- Ici cd est le **nom de la commande**.
- Et /var/www/html est un **argument** fourni à la commande

Afficher le contenu d'un répertoire

- Nous sommes dans le dossier `/var/www/html`
- On aimerait lister le contenu de ce répertoire, on lance la commande `ls`
- ls signifie "**LiSt**"

Remonter dans le répertoire parent

- Nous sommes dans le dossier `/var/www/html`
- On aimerait se placer dans le répertoire parent, c'est à dire `/var/html`
- Pour se placer dans le répertoire parent, on lance la commande **cd ..**
- Tiens, et si on s'amusait à lister le contenu de ce répertoire ? On lance la commande `ls`

Hold on !

- Heuu... pourquoi on n'a pas le même résultat, avec la dernière commande `ls`
- Ahhh oui, on a changé de répertoire de travail !
- Si on est dans `/var/www/html`, `ls` nous retourne son contenu
- Et si on est dans `/var/www`, `ls` nous retourne un autre contenu !

L'astuce de la touche Tab

- Taper des lignes de commandes ça peut être fastidieux
- Heureusement on a la touche Tab du clavier (tabulation)
- Dans le Terminal, la touche **Tab** permet de *compléter*
 - Une commande
 - Un argument

- Et c'est là que le prof te montre, par exemple, comment compléter le chemin `/var/www/html`, grâce à la touche Tab de son clavier

Créer un nouveau dossier

- Vous êtes bien dans le dossier `/var/www/html` ?
- Alors tapez cette commande `mkdir nouveau_dossier_1`
 - `mkdir` signifie **MaKe DIRectory**.
- Ça a l'air de ne rien avoir fait ? C'est normal !
- Un petit `ls` pour voir le contenu peut être ?
- Ô Joie, Ô Bonheur! Notre `nouveau_dossier_1` est bien là !
- Créez donc un autre dossier `nouveau_dossier_2`, on s'en servira plus tard.

Créer un nouveau fichier

- Allons dans notre nouveau dossier `cd /nouveau_dossier_1`.
- Tapez cette commande `> fichier.txt`
- Ça a l'air de ne rien avoir fait ?... On vous la fait pas à vous !!!
- Oui, un petit `ls` nous permet de voir qu'un fichier `fichier.txt` a été créé.
- Pour créer un fichier on peut également utiliser la commande `touch`

Renommez un fichier

- Pour renommer un fichier, tapez `mv fichier.txt fileRenamed.txt`.
- Le petit `ls` nous confirme que notre fichier a correctement été renommé.
- On le rrenomme pour la forme
 - `mv fileRenamed.txt fichier.txt`

Déplacer un fichier

- Vous avez donc fichier.txt dans nouveau_dossier_2 ?
- Ah non, il est dans nouveau_dossier_1 !
- Donc si vous êtes dans nouveau_dossier_1, tapez cette commande mv fichier.txt ../nouveau_dossier_2
 - mv signifie **MoVe** et donc *cette commande peut déplacer mais aussi renommer un fichier ▲*
- Allons vérifier si notre fichier a bien été déplacé ! Comment on fait ? ħ
- Oui c'est bien ça !
 - cd ../nouveau_dossier_2
 - ls
- Well done! Notre fichier fichier.txt est bien là

Copier un fichier

- On a également la possibilité de copier un fichier.
- Tapez cette commande cp fichier.txt ../nouveau_dossier_1/fichier.txt
 - cp signifie **Copy Paste**
- Allons vérifier si notre fichier a bien été copié ! Comment on fait ?
- Oui c'est bien ça !
 - cd ../nouveau_dossier_1
 - ls
- Well done! Notre fichier fichier.txt est bien là

Supprimer un fichier

- On a également la possibilité de supprimer un fichier.
- Tapez cette commande rm fichier.txt

- rm signifie **ReMove**
- Vérifions si notre fichier a bien été supprimé ! Comment on fait ?
- Oui c'est bien ça !
 - ls
- Well done! Notre fichier fichier.txt n'est plus là, il a bien été supprimé

Ecrire dans un fichier et lire le contenu

- Tapez la commande : echo "J'écris du texte dans un fichier en ligne de commande !" > fichier.txt
- Un ls vous confirme qu'un fichier.txt a été créé.
- Pour lire le contenu, tapez la commande **cat fichier.txt**.
- Vous retrouvez le texte précédemment écrit
- Pour rajouter du texte vous pouvez taper echo "J'ajoute une deuxième ligne !" >> fichier.txt
 - Notez bien qu'on utilise >> afin de rajouter du texte. Si vous utilisez seulement > , l'ancien contenu du fichier sera écrasé. ⚠
- On refait un cat fichier.txt et on constate qu'on a bien rajouté une ligne de texte.
- Pour écrire dans un document en ligne de commande on peut également utiliser la commande nano
 - il faut bien penser à faire ctrl+X pour sortir du document
 - et taper o ou y pour valider

Lire un fichier

- Pour lire un fichier on a vu que l'on pouvait utiliser **cat**
- L'inconvénient peut être la taille dudit fichier.

- Pour les gros fichiers, on privilégiera la commande **most** ou **less** qui rend la lecture plus agréable.
- Pour ne lire que le début ou la fin d'un fichier, on utilisera **head** ou **tail**.

RTFM !

- Une expression que vous retrouverez régulièrement dans votre carrière. Au début, vous recevrez cet acronyme comme réponse à vos questions et bientôt c'est vous qui répondrez avec celui-ci.
- Toujours est-il qu'il est difficile de digérer toutes ces informations qui restent pourtant non exhaustives
- Du coup il existe la commande **man** !
 - Tapez la commande man ls
- Cette commande vous **renvoie directement sur le "manuel d'utilisation" de la commande ls**.
- Donc si vous avez un doute sur l'utilisation d'une commande, n'oubliez pas d'aller voir la doc !
- Pour avoir de l'aide sur comment fonctionne une commande on peut utiliser la commande - - help également

Autres infos

- Pour faire plusieurs commandes à la suite on peut les séparer par des ;
- Il existe une option récursive -r pour voir tout ce qu'il y a dans un dossier cela fonctionne aussi avec la commande tree

Conclusion

- **pwd** pour savoir où je suis

- `ls` pour lister le contenu de l'endroit où je suis
- `cd` permet de se déplacer dans son système de fichiers.
- `mkdir` pour créer un nouveau répertoire.
- `echo` et `nano` permet d'écrire dans un fichier.
- `cat` permet de lire le contenu d'un fichier.
- `rm` permet de supprimer un fichier.
- `rmdir` pour supprimer un dossier
- `code .` pour ouvrir VSCode dans le dossier dans lequel on est
- `mv` pour déplacer ou renommer
- `cp` pour copier coller
- `tree` pour afficher l'arborescence des fichiers et dossiers (sudo snap install tree → si on doit l'installer de nouveau)
- `touch` pour créer un nouveau fichier dans un dossier
- `rm -r nom du dossier` pour supprimer un dossier qui n'est pas vide (si on doit forcer la manip on rajoute un `f` ce qui donne `rm -rf`)
- `clear` pour effacer tout ce qu'il y a d'écrit dans le terminal
- Vous aurez l'occasion de découvrir d'autres commandes mais celles ci vont déjà vous permettre de vous déplacer dans votre machine, et de créer des dossiers et fichiers, la base !