



Terminal

Objectifs	1
Le Terminal ?	2
Le contexte d'exécution	3
Un peu de pratique	4
Changer de répertoire	5
Afficher le contenu d'un répertoire	6
Remonter dans le répertoire parent	7
Hold on !	8
L'astuce de la touche Tab	9
Créer un nouveau dossier	10
Créer un nouveau fichier	11
Renommez un fichier	12
Déplacer un fichier	13
Copier un fichier	14
Supprimer un fichier	15
Ecrire dans un fichier et lire le contenu	16
Lire un fichier	17
RTFM !	18
Conclusion	19
Exercice	20
Création de répertoire et fichiers....	20
Copier, déplacer, supprimer...	20
Lire, écrire, supprimer encore...	20

Objectifs

- Comprendre le fonctionnement du terminal
- Lister les commandes de base du terminal



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 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"*



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 Print Working Directory
 - 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 "Change Directory"
- 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.
- Ca 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`
- Ca 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. ☐☐



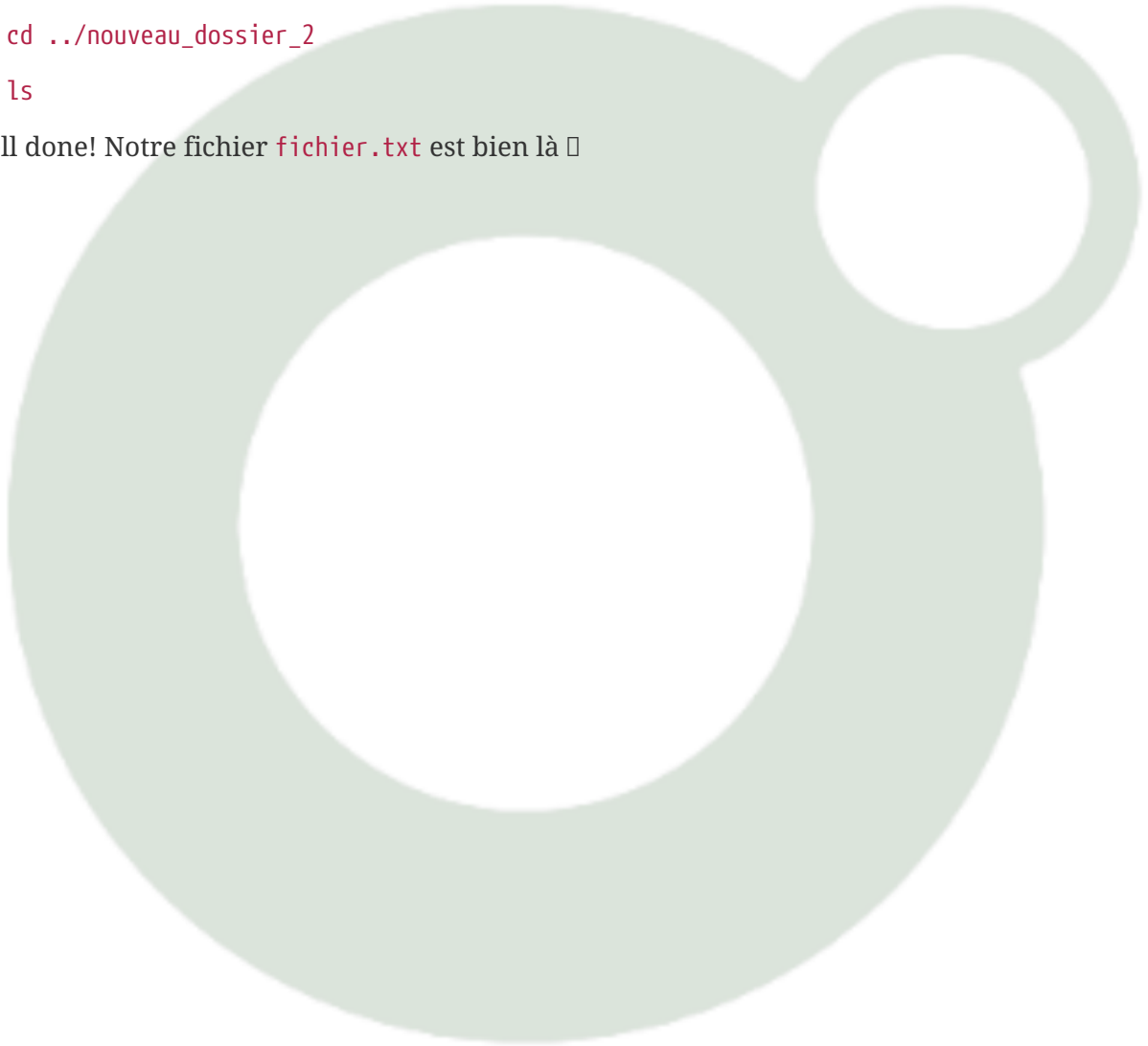
Renommez un fichier

- 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`



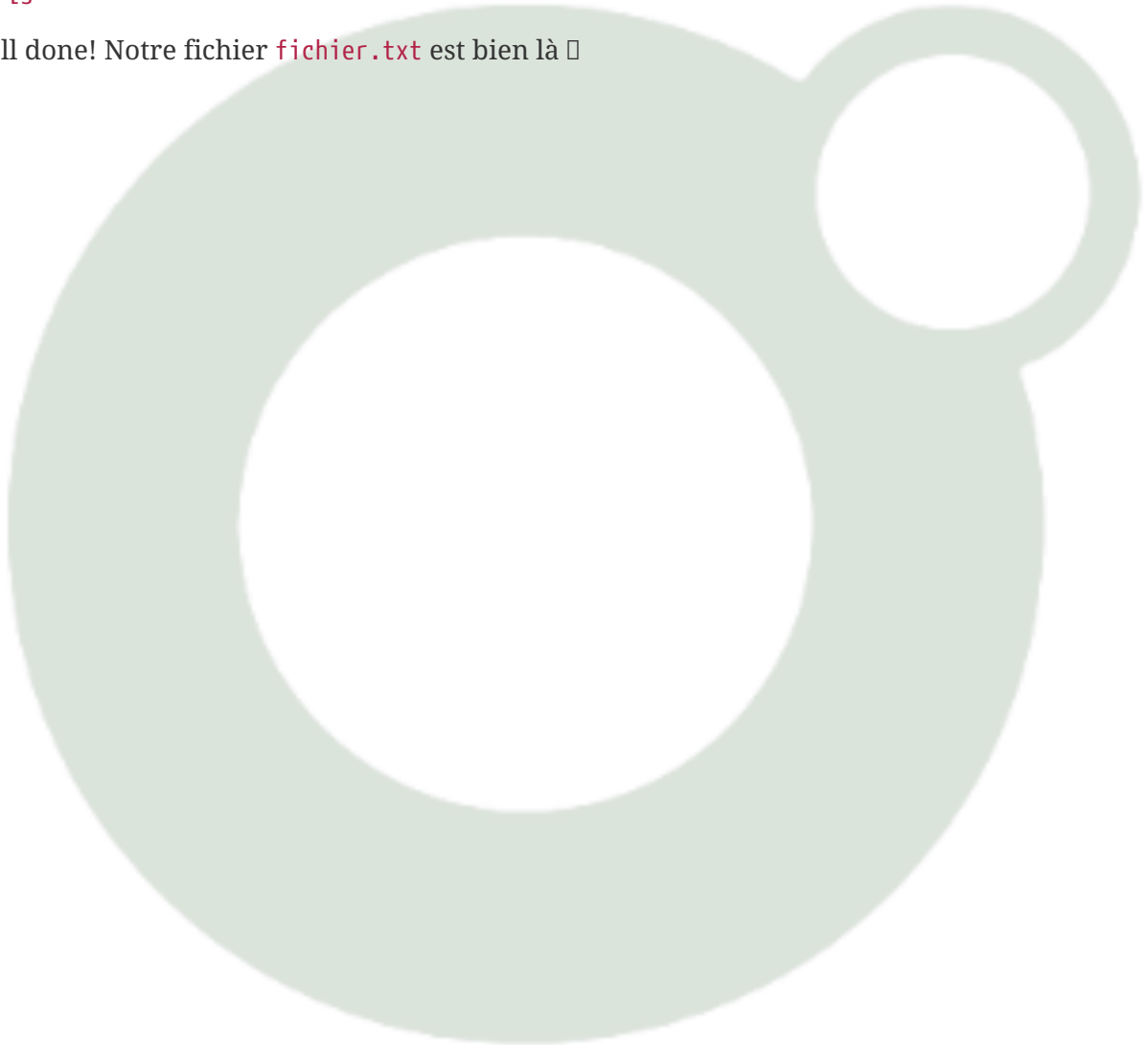
Déplacer un fichier

- 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 ? ☐
- 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. ☐



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 !



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` permet d'écrire dans un fichier.
- `cat` permet de lire le contenu d'un fichier.
- `rm` permet de supprimer un fichier.
- 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 !** ☐



Exercice

Oui mais voilà, je n'ai pas oublié... Alors une petite mise en pratique afin de récapituler l'ensemble de ce qu'on a vu ☐ ☐

Création de répertoire et fichiers...

- Créer un répertoire appelé "MonDossier" dans votre répertoire personnel.
- Aller dans le répertoire "MonDossier"
- A l'intérieur de ce dossier créer 3 fichiers : 'fichier1.txt', 'fichier2.txt', 'fichier3.txt'
- Créer un répertoire appelé "Backup" à l'intérieur de "MonDossier"

```
mkdir MonDossier // On crée le dossier
cd /MonDossier   // On se rend dans le dossier
> fichier1.txt | > fichier2.txt | > fichier3.txt // On crée les 3 fichiers à la suite.
Les commandes sont séparées par |
mkdir Backup      // On crée le dossier
```

Copier, déplacer, supprimer...

- Copier 'fichier1.txt' dans le répertoire "Backup"
- Renommer 'fichier2.txt' en 'nouveau_fichier.txt'
- Déplacer 'nouveau_fichier.txt' dans le répertoire "Backup"
- Supprimer 'fichier3.txt'

```
cp fichier1.txt Backup/fichier1.txt
mv fichier2.txt nouveau_fichier.txt
mv nouveau_fichier.txt Backup/
rm fichier3.txt
```

Lire, écrire, supprimer encore...

- Afficher le contenu du répertoire "MonDossier" pour vérifier que les fichiers et répertoires sont correctement organisés.
- Vérifier également dans le répertoire "Backup"
- Ecrire une phrase dans le fichier 'fichier1.txt'
- Afficher le contenu du 'fichier1.txt'
- Supprimer le répertoire "Backup" ainsi que son contenu en une seule commande

```
ls
```

```
cd Backup | ls  
echo "J'écris une phrase dans le fichier fichier1.txt" >> fichier1.txt  
cat fichier1.txt  
rm -r Backup
```

