

Accueil > Cours > Reprenez le contrôle à l'aide de Linux ! > Entrer une commande

Reprenez le contrôle à l'aide de Linux !

🕒 30 heures 📶 Facile

Mis à jour le 29/06/2021



Le contenu de ce cours n'est plus à jour

Nous avons archivé ce cours et n'actualiserons plus son contenu.

Accédez au contenu le plus récent en découvrant ce cours :



SYSTÈMES & RÉSEAUX

Initiez-vous à Linux

📶 Easy 🕒 8 heures

Dans ce cours débutant, découvrez Linux : un système d'exploitation gratuit et fascinant qui vous donnera un contrôle sans précédent sur votre ordinateur ! Créé par des passionnés d'informatique, Linux est un vecteur important de la philosophie du libre et l'alternative parfaite à Windows ou macOS.

VOIR LE NOUVEAU COURS

Entrer une commande

Chers amis, le grand jour est enfin arrivé ! Vous allez rentrer votre première commande en console !

Ça va, pas trop stressés ?

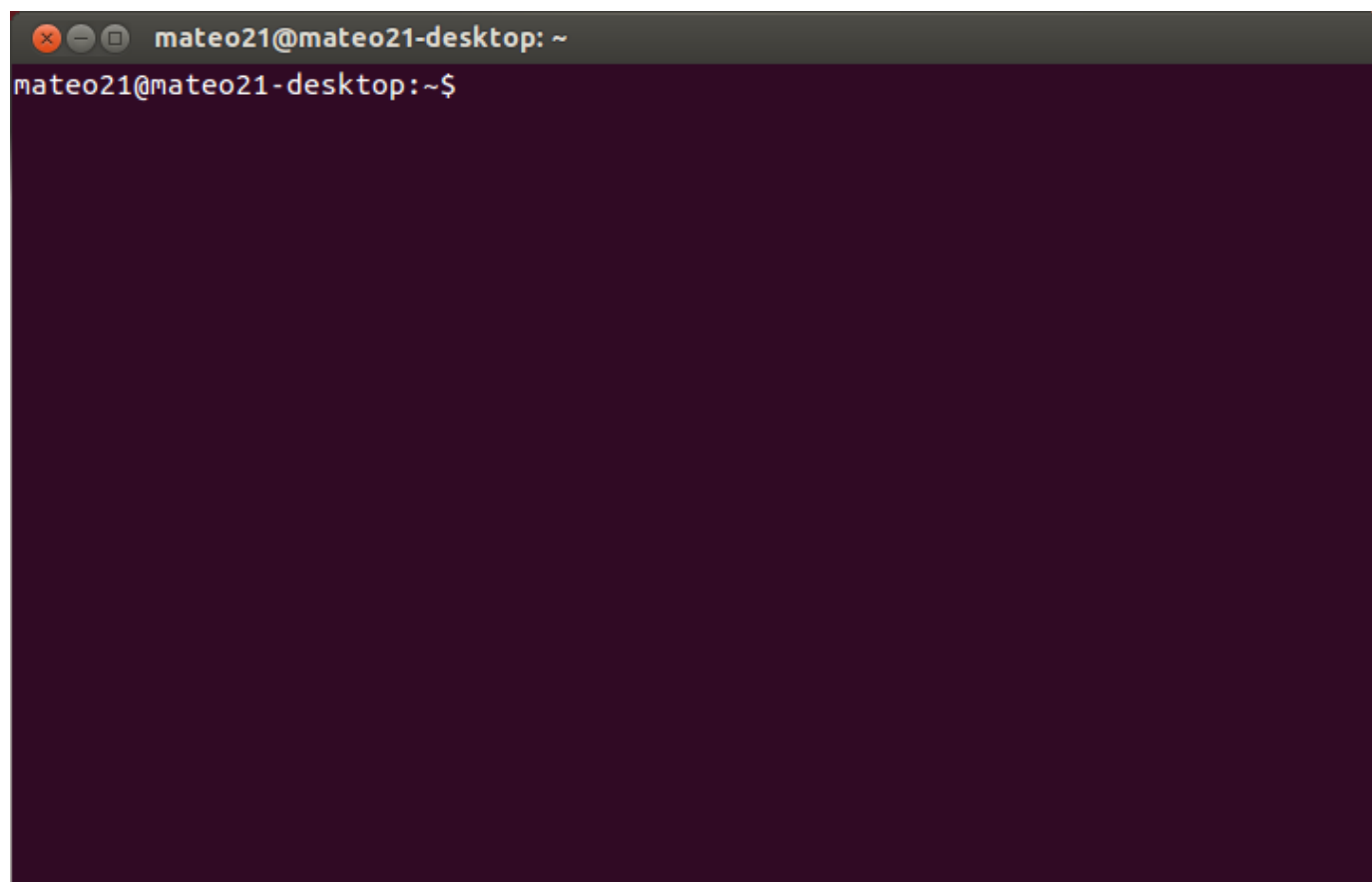
Je vous rassure, nous allons commencer par des choses très simples pour nous familiariser avec la console. Nous allons vraiment voir le B.A.-BA, le guide de survie élémentaire en quelque sorte.

L'invite de commandes



Je suppose à partir de maintenant que vous avez ouvert une console. Si vous ne savez pas faire, c'est que vous n'avez pas lu le chapitre précédent.

Le mieux, comme je vous l'ai dit, est d'ouvrir une console dans le mode graphique. Le programme **Konsole** sous KDE ou **Terminal** sous Unity fera donc très bien l'affaire (figure suivante).



Terminal de Unity

À partir de maintenant, je vais vous présenter le texte affiché en console dans des encadrés comme celui-ci :

```
mateo21@mateo21-desktop:~$
```

Ça, c'est ce que vous voyez à l'écran. Vous n'avez encore rien écrit, mais l'ordinateur vous dit bonjour à sa manière (bon, O.K., j'avoue que c'est une façon très spéciale de dire bonjour).

Ce que vous voyez là est ce qu'on appelle l'**invite de commandes**. C'est un message qui vous **invite** à rentrer une **commande** en vous donnant par la même occasion une foule d'informations. Cette invite s'affiche avant chaque commande que vous tapez.

Bien : décortiquons cette invite de commandes parce qu'elle est très intéressante.

- **mateo21** : le premier élément est votre pseudonyme. C'est le pseudo sous lequel vous êtes loggés. En effet, rappelez-vous : on peut créer plusieurs comptes utilisateurs sous Linux. Il est en général conseillé d'en générer un par personne susceptible d'utiliser l'ordinateur (un pour chaque membre de la famille, par exemple). Nous verrons plus tard comment rajouter des comptes utilisateurs.
- **@** : ce symbole n'indique rien de particulier. C'est le symbole « at » qui signifie « chez ». Si on lit l'invite de gauche à droite, on doit donc comprendre « mateo21 chez ».
- **mateo21-desktop** : ça, c'est le nom de l'ordinateur sur lequel vous êtes en train de travailler. Dans mon cas il s'appelle **mateo21-desktop**, mais j'aurais pu lui attribuer n'importe quel nom lors de l'installation.

Par exemple, on a l'habitude de donner le nom d'un membre des Simpson à chacun des serveurs du Site du Zéro : Lisa, Bart, Itchy, Scratchy... Cela permet de savoir de quelle machine on parle quand on dit « Ouh là, Bart est surchargé, il faudrait voir quel est le programme qui ralentit tout ». Si vous suivez toujours, la ligne d'invite de commandes se lit donc « mateo21 chez mateo21-desktop ». En d'autres termes, je suis identifié en tant que **mateo21** sur la machine **mateo21-desktop**.

- **::** : à nouveau, ce symbole ne veut rien dire de spécial, c'est un séparateur.
- **~** : ça, c'est le dossier dans lequel vous vous trouvez actuellement. Vous pouvez naviguer de dossier en dossier dans la console et il est très utile qu'on vous rappelle systématiquement où vous vous trouvez avant chaque commande.

Pour information, le symbole **~** signifie que vous êtes dans votre dossier personnel, ce qu'on appelle le « home » sous Linux ; c'est l'équivalent du dossier « Mes documents » de Windows. Nous étudierons plus en détail le fonctionnement des dossiers sous Linux dans le prochain chapitre.

- **\$** : ce dernier symbole est très important ; il indique votre niveau d'autorisation sur la machine. Il peut prendre deux formes différentes :
 - **\$** : signifie que vous êtes en train d'utiliser un compte utilisateur « normal », avec des

droits limités (il ne peut pas modifier les fichiers système les plus importants). Mon compte `mateo21` est donc un compte normal avec des droits limités ;

- `#` : signifie que vous êtes en mode superutilisateur, c'est-à-dire que vous êtes connectés sous le pseudonyme « root ». Le root est l'utilisateur maître qui a le droit de tout faire sur sa machine (même de la détruire !). Nous verrons le mode root plus en détail plus tard ; pour l'instant nous restons dans un compte utilisateur limité, ainsi nous ne risquons pas de faire de bêtise.

Comme vous le voyez, une fois qu'on parle la même langue que l'invite de commandes, on comprend ce qu'elle veut dire !

« Bonjour et bienvenue, vous êtes `mateo21` sur la machine `mateo21-desktop` . Vous vous trouvez actuellement dans votre dossier `home` et possédez des droits utilisateur limités. La température extérieure est de. »



Comme un peu tout sous Linux, l'invite de commandes est totalement paramétrable. Vous pouvez la raccourcir si vous trouvez qu'elle est trop longue, ou la rallonger si vous trouvez qu'elle ne donne pas assez d'informations. Vous pouvez en théorie mettre vraiment tout ce que vous voulez dans l'invite, comme par exemple l'heure actuelle (par contre, pour la température extérieure il faudra repasser).

Nous verrons comment changer cela lorsque vous aurez appris à vous servir d'un éditeur de texte !

Commandes et paramètres



On travaille dans la console en tapant ce qu'on appelle des **commandes**. Ces dernières étant nombreuses, vous ne pourrez jamais toutes les connaître... et ce n'est pas le but : le but, c'est que vous sachiez vous servir par cœur de la plupart des commandes « courantes » et, pour les moins courantes, que vous soyez capables d'apprendre à vous en servir en lisant leur **manuel d'utilisation**.

Le manuel d'utilisation est **la véritable bible de tous les linuxiens**. Vous verrez rapidement qu'ils ne jurent que par ça. Pourquoi ? Parce que c'est tout simplement un outil de référence, là où l'on peut trouver la réponse à TOUTES ses questions pour peu qu'on sache lire le manuel et qu'on prenne la peine de le faire. Un chapitre entier vous apprendra à lire le manuel : c'est vraiment très important.



Pour information, il est courant de voir un linuxien dire « RTFM » aux débutants qui posent des questions qu'il estime « simples ». RTFM est l'abréviation de « Read the fucking manual! », ce qui veut dire grosso modo « Lis le p*** de manuel ! ». Ce n'est pas vraiment

une insulte en fait, mais plutôt une sorte de... soupir d'exaspération.

Pour éviter qu'on vous dise ça un jour, je vous conseille donc de bien ouvrir vos oreilles quand je vous expliquerai comment utiliser le manuel. ;-)

Une commande simple

Bon : trêve de bavardages, on va rentrer une commande ! Par exemple, tapez `date` puis appuyez sur la touche `Entrée` du clavier.

Le résultat devrait ressembler à cela :

```
mateo21@mateo21-desktop:~$ date
lundi 20 septembre 2010, 15:39:51 (UTC+0200)
```

La première ligne contient l'invite de commandes suivie de la commande que j'ai tapée.

La seconde ligne est la réponse de l'ordinateur à cette commande.

Je suppose que vous avez deviné comme des grands ce que l'on vient de faire : on a demandé quelles étaient la date et l'heure !

Vous en voulez encore ? O.K., alors essayons une toute autre commande : tapez `ls`. C'est l'abréviation de « list », qui signifie « **lister les fichiers et dossiers du répertoire actuel** ».

```
mateo21@mateo21-desktop:~$ ls
Desktop Examples Images
```

Cela signifie que le répertoire actuel est constitué de trois dossiers : `Desktop`, `Examples` et `Images`. En général, le système colore les éléments pour que l'on puisse distinguer facilement les dossiers des fichiers.

Si vous n'avez aucune réponse, c'est que vous êtes dans un dossier qui ne contient ni fichier ni dossier.

Voilà, c'est aussi simple que cela. Une commande est constituée d'un mot et ne contient aucun espace. Dans des cas très simples comme ceux que l'on vient de voir, il suffit juste de taper la commande pour avoir une réponse ; mais dans la quasi-totalité des cas on peut (et parfois on DOIT) rentrer des options, qu'on appelle **paramètres**.

Les paramètres

Les paramètres sont des options que l'on écrit à la suite de la commande. La commande et les paramètres sont séparés par un espace, comme ceci :

```
mateo21@mateo21-desktop:~$ commande parametres
```

Les paramètres peuvent eux-mêmes contenir des espaces, des lettres, des chiffres... un peu de tout, en fait. Il n'y a pas de règle véritable sur la forme des paramètres, mais heureusement les programmeurs ont adopté une sorte de « convention » pour que l'on puisse reconnaître les différents types de paramètres.

Les paramètres courts (une lettre)

Les paramètres les plus courants sont constitués d'une seule lettre précédée d'un tiret. Par exemple :

```
commande -d
```

Si on doit donner plusieurs paramètres, on peut faire comme ceci :

```
commande -d -a -U -h
```

Ou, plus court :

```
commande -daUh
```



Attention à la casse des paramètres (majuscules / minuscules) ! Si vous écrivez `-u`, cela n'a en général pas du tout le même sens que `-U` !

Faisons un essai avec la commande `ls` et rajoutons-lui le paramètre « `a` » (en minuscule) :

```
mateo21@mateo21-desktop:~$ ls -a
.          .gconfd          .mozilla-thunderbird
..         .gimp-2.2       .nautilus
.bash_history .gksu.lock       .profile
.bash_logout .gnome           .recently-used
.bashrc      .gnome2          .recently-used.xbel
.config      .gnome2_private  .ssh
Desktop      .gstreamer-0.10  .sudo_as_admin_successful
.dmrc        .gtkrc-1.2-gnome2 .themes
.esd_auth    .ICEauthority    .thumbnails
.evolution   .icons           .Trash
Examples     .lessgst         tutos
.face        .local           .update-manager-core
.fontconfig  .macromedia      .update-notifier
.gaim        .metacity        .Xauthority
.gconf       .mozilla         .xsession-errors
```

Cela affiche tout le contenu du dossier, même les fichiers cachés.

Un « fichier caché » sous Linux est un fichier qui commence par un point. Normalement, si vous vous trouvez dans votre répertoire `home`, vous devriez avoir une bonne flopée de fichiers cachés. Ce sont en général des fichiers de configuration de programmes.

Les paramètres longs (plusieurs lettres)

Les paramètres constitués de plusieurs lettres sont précédés de deux tirets, comme ceci :

```
commande --parametre
```

Cette fois, pas le choix : si vous voulez mettre plusieurs paramètres longs, il faudra ajouter un espace entre chacun d'eux :

```
commande --parametre1 --parametre2
```

On peut aussi combiner les paramètres longs et les paramètres courts dans une commande :

```
commande -daUh --autreparametre
```



Il y a parfois deux écritures possibles pour un paramètre de commande : une version courte et une version longue. Cela permet de vous laisser le choix selon que vous préférez l'une ou l'autre.

Notez que c'est la commande qui décide des paramètres qu'elle accepte : il arrive parfois que certaines ne proposent pas le choix entre une version courte et une longue.

Testons cela sur la commande `ls` avec le paramètre `--all`, qui signifie « tout » en anglais :

```
mateo21@mateo21-desktop:~$ ls --all
.          .gconfd          .mozilla-thunderbird
..         .gimp-2.2       .nautilus
.bash_history .gksu.lock       .profile
.bash_logout .gnome           .recently-used
.bashrc      .gnome2          .recently-used.xbel
.config      .gnome2_private  .ssh
Desktop     .gstreamer-0.10  .sudo_as_admin_successful
.dmrc       .gtkrc-1.2-gnome2 .themes
.esd_auth   .ICEauthority    .thumbnails
.evolution  .icons           .Trash
Examples    .lessht          tutos
.face       .local           .update-manager-core
.fontconfig .macromedia      .update-notifier
```

```
.gaim      .metacity  .Xauthority  
.gconf     .mozilla   .xsession-errors
```

Comme vous le voyez, `--all` est un synonyme de `-a`. Ceci illustre ce que je vous disais à l'instant, comme quoi parfois une commande propose deux façons d'utiliser un paramètre : une courte et une longue.

Les valeurs des paramètres

Certains paramètres nécessitent que vous les complétiez avec une valeur. Cela fonctionne différemment selon que vous travaillez avec un paramètre long ou avec un paramètre court.

Avec un paramètre court :

```
commande -p 14
```

... cela indique que l'on associe la valeur `14` au paramètre `p`. Avec ce genre de technique, on peut par exemple faire comprendre à l'ordinateur : « **Je veux voir la liste de tous les fichiers de plus de 14 Mo** ».

Si c'est un paramètre long, on fait en général comme ceci :

```
commande --parametre=14
```

Le résultat sera le même, il est juste plus lisible mais aussi plus long à écrire.

Les autres paramètres

Je vous l'ai dit : il n'y a pas de règle absolue au niveau des paramètres et vous en rencontrerez sûrement qui fonctionnent différemment. Heureusement, les « conventions » que je viens de vous donner sont valables dans la grande majorité des cas, ce qui devrait vous permettre de vous repérer.

Certains paramètres sont donc un peu différents et dépendent vraiment des commandes. Par exemple avec `ls`, si on ajoute le nom d'un dossier (ou sous-dossier), cela affichera le contenu de ce dossier au lieu du contenu du dossier courant :

```
mateo21@mateo21-desktop:~$ ls Examples
Experience ubuntu.ogg      logo-Ubuntu.png           oo-payment-schedule.ods
fables_01_01_aesop.spx    oo-about-these-files.odt  oo-presenting-kubuntu.odp
gimp-ubuntu-splash.xcf   oo-about-ubuntu-ru.rtf    oo-presenting-ubuntu.odp
kubuntu-leaflet.png      oo-cd-cover.odg           oo-trig.xls
logo-Edubuntu.png        oo-derivatives.doc        oo-welcome.odt
logo-Kubuntu.png         oo-maxwell.odt            ubuntu Sax.ogg
```


Retrouver une commande



Linux propose tellement de commandes différentes qu'il est facile de s'y perdre et d'en oublier une. Personnellement, ça m'arrive très régulièrement, mais ce n'est heureusement pas un drame. En effet, Linux vous propose toute une série de façons de retrouver une commande que vous avez oubliée.

Autocomplétion de commande

Le premier « truc » à connaître, c'est l'autocomplétion de commande. Prenons la commande `date` par exemple : vous êtes un peu tête en l'air et vous ne savez plus comment elle s'écrit. Par contre, vous êtes sûrs des premières lettres de la commande.

Lister les commandes correspondantes

Tapez juste « `da` » dans la console, puis tapez deux fois sur la touche **Tabulation** située à gauche de votre clavier. Le résultat sera le suivant :

```
mateo21@mateo21-desktop:~$ da
dash date
mateo21@mateo21-desktop:~$ da
```

En tapant deux fois sur **Tabulation**, vous avez demandé à l'ordinateur la liste des commandes qui commencent par « `da` ». On vous a répondu « `dash` » et « `date` ». Il y a donc deux commandes qui commencent par « `da` », et vous venez de retrouver celle que vous cherchiez, c'est-à-dire « `date` ».

Bien sympathique, l'ordinateur a réécrit l'invite de commandes en dessous ainsi que le début de la commande que vous aviez tapée. Vous n'avez plus qu'à compléter avec les lettres « `te` » qui manquent et à taper **Entrée**, et ce sera bon. :-)

L'autocomplétion

Plus sympa encore, s'il n'y a qu'un seul résultat correspondant à votre recherche, l'ordinateur complètera avec les lettres qui manquent et vous n'aurez plus qu'à taper sur **Entrée** !

Par exemple, il n'y a qu'une commande qui commence par « `dat` ». Tapez donc `dat` dans la console, puis appuyez une seule fois sur **Tabulation**. La commande se complète comme par magie.

Trop de commaaaaandes !

Parfois, il y a trop de commandes correspondant à votre recherche. Faites un essai un peu brutal :

ne rentrez aucun début de commande et faites deux fois **Tab** (**Tabulation**).



Sous certaines distributions (Ubuntu 14.04 LTS et LinuxMint 17.2, par exemple), cette manipulation ne fonctionne pas. Il faut au moins entrer un caractère pour avoir une réponse.

Cela demande de faire la liste de toutes les commandes disponibles sur votre ordinateur.

```
mateo21@mateo21-desktop:~$  
Display all 2173 possibilities? (y or n)
```

Sauvage, n'est-ce pas ?

Il y a 2 173 commandes disponibles sur mon ordinateur. Plus j'installerai de programmes, plus j'aurai de commandes utilisables. N'espérez donc pas toutes les connaître, de nouveaux programmes sortent tous les jours.

À cette question, vous pouvez répondre « **y** » (*yes*) et la liste s'affichera page par page. Quelques raccourcis à connaître quand une liste s'affiche page par page :

- tapez **Espace** pour passer à la page suivante ;
- tapez **Entrée** pour aller à la ligne suivante ;
- tapez **q** pour arrêter la liste.

Si vous répondez « **n** » (*no*), il ne se passera rien ; c'est dans le cas où vous vous diriez « Ouh là, 2 173 possibilités : autant chercher une aiguille dans une botte de foin... je vais peut-être affiner ma recherche ».

L'historique des commandes

On a très souvent besoin de retrouver une commande que l'on a tapée il y a cinq minutes (ou même cinq secondes). Parfois c'est parce qu'on a oublié la commande, mais c'est souvent aussi parce qu'on a comme moi un énooorme poil dans la main et qu'on a vraiment la flemme de réécrire nous-mêmes la commande en entier.

Ce raccourci vaut de l'or : appuyez sur la flèche directionnelle **Haut** (figure suivante) ; vous verrez apparaître la dernière commande que vous avez tapée.

Si vous appuyez de nouveau sur la flèche directionnelle **Haut**, vous verrez l'avant-dernière commande, puis l'avant-avant-dernière, etc.



Flèche directionnelle Haut

Si vous appuyez sur la flèche directionnelle **Bas** (figure suivante), vous reviendrez aux commandes les plus récentes.



Flèche directionnelle Bas

C'est ainsi que je peux successivement retrouver les commandes que je viens de taper, dans l'ordre inverse :

- `ls --all ;`
- `ls -a ;`
- `ls ;`
- `date ;`
- etc.

Si vous voulez « remonter » très loin en arrière dans l'historique de vos commandes, pas la peine de taper cent fois sur la flèche directionnelle **Haut** comme des forcenés.

Il existe la commande **history** qui vous rappelle l'historique des commandes :

```
152 date
153 ls
154 ls -a
155 ls --all
156 history
```

La dernière commande tapée sera toujours **history** , forcément.

Vous remarquerez que les commandes sont numérotées : ainsi, on peut savoir que **date** est la 152ème commande que j'ai tapée dans le terminal.

Ctrl + R : rechercher une commande tapée avec quelques lettres

Dans le cas où la flèche directionnelle **Haut** et la commande **history** ne suffiraient pas à retrouver une vieille commande que vous avez tapée, il y a un raccourci super utile : **Ctrl + R** . Appuyez donc sur les touches **Ctrl** et **R** en même temps et l'ordinateur se mettra en mode « recherche d'une commande tapée » (« R » comme **R**echerche).

Là, vous pouvez taper n'importe quelle suite de lettres correspondant à une vieille commande. Par

exemple, faites `Ctrl + R` puis tapez « `all` ». Linux retrouve la commande `ls --all` qui contenait justement le mot « `all` ». Vous n'avez plus qu'à taper `Entrée` pour relancer la commande ! :-)

```
(reverse-i-search)`all': ls --all
```

Si ce n'est pas la commande que vous cherchiez, appuyez à nouveau sur `Ctrl + R` pour remonter dans la liste des commandes contenant « `all` ».

Ça a peut-être l'air bête sur une commande comme ça, mais certaines sont vraiment très longues et c'est un vrai bonheur de ne pas avoir à les réécrire en entier !

Quelques raccourcis clavier pratiques



On ne dirait pas comme ça, mais la console de Linux propose une quantité incroyable de raccourcis clavier. Ce sont des raccourcis qu'on ne peut pas deviner, qu'on a un peu de mal à retenir au début, mais quand on les connaît... waouh ! On devient un peu comme Neo dans *Matrix* en fait, on va très vite.

Ces raccourcis ne sont pas intuitifs, mais ça vaut vraiment le coup de les retenir. Les premiers temps vous reviendrez certainement souvent ici pour les consulter (et encore, si vous vous faites un petit pense-bête, ça ne devrait pas arriver), mais petit à petit vous les connaîtrez par cœur !

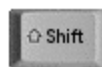
Commençons par quelques raccourcis généraux qu'il vous faut connaître.



+

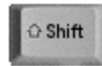


+



+





+



- **Ctrl + L** : efface le contenu de la console. Utile pour faire un peu de ménage quand votre console est encombrée, ou quand votre boss passe derrière vous et que vous n'aimeriez pas qu'il voie ce que vous étiez en train de faire.
À noter qu'il existe aussi une commande, **clear**, qui fait exactement la même chose.
- **Ctrl + D** : envoie le message **EOF** (fin de fichier) à la console. Si vous tapez ce raccourci dans une ligne de commande vide (c'est-à-dire sans avoir écrit un début de commande au préalable), cela fermera la console en cours.
À noter qu'il existe aussi la commande **exit** qui a le même effet.
- **Shift + PgUp** : vous permet de « remonter » dans les messages envoyés par la console. En mode graphique, la molette de la souris accomplit aussi très bien cette action.
La touche **Page Up** est généralement représentée sur votre clavier par une flèche directionnelle **Haut** barrée de plusieurs petites lignes horizontales.
- **Shift + PgDown** : pareil, mais pour redescendre.

Les raccourcis suivants sont utiles lorsque vous êtes en train d'écrire une longue commande.



+



+



+



- **Ctrl + A** : ramène le curseur au début de la commande. La touche **Origine** a le même effet (elle est située à côté de la touche **Fin** et représentée par une flèche pointant en haut à gauche).
- **Ctrl + E** : ramène le curseur à la fin de la ligne de commandes. La touche **Fin** a le même effet.
- **Ctrl + U** : supprime tout ce qui se trouve à gauche du curseur. Si celui-ci est situé à la fin de la ligne, cette dernière sera donc supprimée.
- **Ctrl + K** : supprime tout ce qui se trouve à droite du curseur. S'il est situé au début de la ligne, celle-ci sera donc totalement supprimée.
- **Ctrl + W** : supprime le premier mot situé à gauche du curseur. Un « mot » est séparé par des espaces ; on s'en sert en général pour supprimer le paramètre situé à gauche du curseur.
- **Ctrl + Y** : si vous avez supprimé du texte avec une des commandes **Ctrl + U** , **Ctrl + K** ou **Ctrl + W** qu'on vient de voir, alors le raccourci **Ctrl + Y** « collera » le texte que vous venez de supprimer. C'est donc un peu comme un couper-coller.

Nous allons arrêter là la liste pour le moment. Ça vous fait déjà pas mal de choses à retenir.

Il existe en fait beaucoup d'autres raccourcis clavier, mais je vous les montrerai au fur et à mesure, quand nous en aurons besoin.

Même si c'est un peu fastidieux, je vous conseille de vous entraîner à les retenir par cœur. Vous serez vraiment beaucoup plus efficaces lorsque vous les connaîtrez !

En résumé

- La console affiche une invite de commandes au début de la ligne. Cette invite rappelle votre nom d'utilisateur, le nom de la machine ainsi que le dossier dans lequel vous vous trouvez.
- On rentre des **commandes** dans la console pour demander à l'ordinateur d'exécuter des actions.
- Chaque commande peut être complétée de **paramètres** qui agissent comme des options pour modifier l'action de la commande.
- Les paramètres sont généralement constitués d'une lettre précédée d'un tiret (`-a`) ou de plusieurs lettres précédées de deux tirets (`--all`).
- Après avoir saisi les premières lettres d'une commande, on peut compléter son nom à l'aide de la touche `Tabulation` .
- On peut retrouver les commandes précédentes à l'aide des flèches directionnelles `Haut` et `Bas` ou encore effectuer une recherche parmi les commandes précédentes avec `Ctrl + R` .
- Il existe de nombreux autres raccourcis clavier qu'il est recommandé de connaître pour pouvoir profiter pleinement de la console.

INDIQUER QUE CE CHAPITRE N'EST PAS TERMINÉ

< LA CONSOLE, ÇA SE MANGE ?

LA STRUCTURE DES DOSSIERS ET FICHIERS >

Le professeur



Mathieu Nebra

Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms :o)

Découvrez aussi ce cours en...



Livre



PDF

OPENCLASSROOMS

Qui sommes-nous ?

Financements

Expérience de formation

Forum

Blog 

Presse 

OPPORTUNITÉS

Nous rejoindre 

Devenir mentor 

Devenir coach carrière 

AIDE



FAQ

POUR LES ENTREPRISES

Former et recruter

EN PLUS

Boutique 

Mentions légales

Conditions générales d'utilisation

[Politique de protection des données personnelles](#)

[Cookies](#)

[Accessibilité](#)

 Français ▼

