



Initiation à la programmation

Don't Panic !

Staff 42 pedago@42.fr

Résumé: Ce document ne doit servir qu'en cas de nécessité absolue.

Table des matières

| | | |
|------------|---|----------|
| I | Préambule | 2 |
| II | Le Terminal | 3 |
| III | Les commandes de base en 5 mn (ou peut être un peu plus) | 4 |
| IV | Configurer votre shell | 8 |

Chapitre I

Préambule

DON'T PANIC

Tout là-bas, au fin fond des tréfonds inexplorés et mal famés du bout du bras occidental de la Galaxie, traîne un petit soleil jaunâtre et minable.

En orbite autour de celui-ci, à la distance approximative de cent cinquante millions de kilomètres, se trouve une petite planète bleu-vert dont les habitants - descendus du singe - sont primitifs au point de croire encore que les montres à quartz numériques sont une vache de chouette idée.

Cette planète a - ou plutôt, elle avait - un problème, à savoir celui-ci : la plupart de ses habitants étaient malheureux la plupart du temps. Bien des solutions avaient été suggérées mais la plupart d'entre elles faisaient largement intervenir la mise en circulation de petits bouts de papier vert, chose curieuse puisque en définitive ce n'étaient pas les bouts de papier vert qui étaient malheureux.

Et donc le problème subsistait ; des tas de gens se sentaient minables et la plupart étaient effectivement misérables - y compris les possesseurs de montres à quartz numériques.

Douglas Adams, Le guide du voyageur galactique, 1979

Chapitre II

Le Terminal

- Le terminal est votre meilleur ami. Mais aussi votre nouvel environnement de travail.
- Le terminal permet de travailler sur votre ordinateur de manière rapide et textuelle.
- Sous MacOS, le terminal utilisé se nomme iTerm. Vous pouvez le trouver en cliquant sur la loupe en haut à droite de votre écran et en tapant son nom ("iTerm"). Cliquez ensuite sur le programme pour l'ouvrir.

Chapitre III

Les commandes de base en 5 mn (ou peut être un peu plus)

- Pour créer un répertoire, la commande se nomme **mkdir**.

La commande **mkdir** suivi du nom de votre choix crée un nouveau répertoire du nom voulu.

```
$> mkdir example
$>
```

- Pour lister le contenu du répertoire courant, utilisez la commande **ls**.

```
$> ls
example
$>
```

Notez le répertoire exemple qui est bel et bien créé.

- Plusieurs options existent. Vous pouvez les trouver en lisant le manuel de la commande en question avec **man**.

```
$> man ls
$>
```

Vous ouvrez ainsi un manuel de la commande **ls** que vous pouvez quitter en appuyant sur la lettre 'q'.

- Les options se rajoutent à la suite des commandes. Ainsi, **ls -la** va adjoindre les options -a et -l à la commande **ls**.

```
$> ls -la
total 2896
drwxr-xr-x 11 froh  staff    374 Sep 24 16:34 .
drwxr-xr-x  5 froh  staff    170 Sep 24 15:33 ..
-rw-r--r--@ 1 froh  staff   6148 Sep 24 15:34 .DS_Store
drwxr-xr-x  2 froh  staff    68 Sep 24 15:59 example
$>
```

- Votre répertoire 'exemple' étant créé. Il est temps de vous déplacer dedans. L'équivalent du clic sur un répertoire est ici remplacé par la commande **cd nomdurépertoire**.

```
$> cd exemple
$>
```

Vous venez de vous déplacer dans votre répertoire 'exemple'.

- Pour vérifier où vous vous trouvez, et afficher l'arborescence de fichier. Il suffit d'utiliser la commande **pwd**

```
$> pwd
/Users/froh/piscine-HEC/example
$>
```

Bien entendu, vous n'aurez pas le même chemin que celui sur l'exemple.

- **cd** qui signifie en fait 'change directory' vous permet de vous déplacer facilement de répertoire en répertoire.

Petits détails utiles :

cd .. : Vous remonte d'un niveau dans l'arborescence. Par exemple, si vous êtes dans le sous répertoire "exemple", cela vous replace dans le répertoire qui contient "exemple". **"cd"** : Vous remet dans votre 'Home directory' qui est le répertoire de base lorsque vous ouvrez votre terminal.

- **touch** suivi du nom d'un fichier va créer un fichier vide, que vous allez pouvoir remplir avec divers éditeurs de texte.

```
$> ls -la
total 0
drwxr-xr-x  2 froh  staff   68 Sep 25 18:29 .
drwxr-xr-x  7 froh  staff  238 Sep 24 17:24 ..
$> touch test
$> ls -la
total 0
drwxr-xr-x  3 froh  staff  102 Sep 25 18:29 .
drwxr-xr-x  7 froh  staff  238 Sep 24 17:24 ..
-rw-r--r--  1 froh  staff   0 Sep 25 18:29 test
$>
```

- **rm** suivi du nom d'un fichier va supprimer le fichier. **rm -rf** force la suppression, si vous voulez, par exemple, supprimer un répertoire.

```
$> ls -la
total 0
drwxr-xr-x  3 froh  staff  102 Sep 25 18:29 .
drwxr-xr-x  7 froh  staff  238 Sep 24 17:24 ..
-rw-r--r--  1 froh  staff   0 Sep 25 18:29 test
$> rm test
$> ls -la
total 0
drwxr-xr-x  2 froh  staff   68 Sep 25 18:29 .
drwxr-xr-x  7 froh  staff  238 Sep 24 17:24 ..
$>
```

```
$> ls -la
total 408
drwxr-xr-x  7 froh  staff   238 Sep 25 18:36 .
drwxr-xr-x  5 froh  staff   170 Sep 24 15:33 ..
drwxr-xr-x  2 froh  staff    68 Sep 25 18:30 example
$> rm -rf example
$> ls -la
total 374
drwxr-xr-x  6 froh  staff   204 Sep 25 18:36 .
drwxr-xr-x  5 froh  staff   170 Sep 24 15:33 ..
$>
```

Attention, supprimer quelque chose en ligne de commande est définitif, faites donc bien attention à ce que vous tapez pour ne pas effacer votre répertoire de travail ou vos fichiers par erreur.

- **chmod** vous permet de donner des droits d'exécution ou de lecture sur un fichier. Il est suivi d'arguments permettant de donner les permissions de votre choix.

```
$> ls
hec.rb
$> ./hec.rb
zsh: permission denied: ./hec.rb
$> chmod 700 hec.rb
$> ./hec.rb
HEC
$> ls -l
total 8
-rwx-----  1 froh  staff  29 Sep 25 23:06 hec.rb
$>
```

Notez que **ls -l** vous donne des détails sur les droits appliqués à chaque fichier ou répertoire.

- **cat** est une commande vous permettant d'afficher le contenu d'un fichier. Bien entendu, s'il n'y a rien dans le fichier, vous n'affichez rien.

```
$> cat bonjour.rb
#!/usr/bin/ruby
print "bonjour !"
$>
```

- **open** est une particularité de MacOS qui vous permet de lancer certain programme.

Ainsi, **open** suivi du nom du pdf vous ouvrira le pdf.

open . (le **.** est important) vous ouvrira le répertoire courant dans le Finder (l'explorateur de fichier de votre mac).

```
$> open dontpanic.pdf
$>
$> open .
$>
```

- Enfin, les touches ctrl et C pressées simultanément, stopperont les programmes ou les commandes en cours de route.
Ce qui pourra vous être utile lors de certaines commandes malencontreuses ou boucles infinies de vos programmes.

```
$> yes "ctrl + c"  
ctrl + c  
ctrl + c  
ctrl + c  
^C  
$>
```

Vous êtes maintenant prêts à tout !

Chapitre IV

Configurer votre shell

- Votre shell peut être personnalisé à votre convenance.
- De base, vous utilisez le zsh, d'autres shells existent tels que bash par exemple.
- Les shells sont les "noyaux" au centre de votre console de commandes. Ils gèrent différemment l'affichage et l'option. Prenons comme exemple, le retour à la ligne.

```
$> zsh
$> echo -n "Salut"
Salut%
$> bash
$> echo -n "Salut"
Salut$> exit
$> exit
```

- Configurer votre shell est assez simple. Dans votre 'Home directory' (en utilisant la commande **cd**), vous pouvez éditer un fichier `.zshrc` avec l'un des éditeurs de texte par défaut (emacs, vim, nano).

Le `.zshrc` représente la configuration par défaut de votre terminal zsh.

Editer ce fichier vous permet de créer des raccourcis (aussi nommés alias), ou de customiser un peu l'apparence.

- Pour créer ou éditer ce fichier sous emacs.

emacs .zshrc et rajoutez la ligne suivante :

alias subl="/Applications/Sublime Text.app/Contents/SharedSupport/bin/subl"

Sauvegarder en appuyant sur CTRL + X puis CTRL + S

Quitter en appuyant sur CTRL + X puis CTRL + C

Vous pouvez maintenant tester votre nouvel éditeur de texte avec cette commande.

```
$> subl .
$>
```

- Un éditeur de texte graphique et déjà configuré pourra vous être très utile pour réaliser vos exercices. Très facile d'accès, Sublime Text est un éditeur complet possédant une coloration syntaxique de base. Il reconnaît le type de programme que vous écrivez et colorie les mots clés de façon à rendre le tout plus lisible.

- Quelques commandes utiles :

subl nomdufichier pour ouvrir un fichier précis.
Command + S pour sauvegarder