



# Initiation à la programmation

## Shell

Staff 42 [pedago@42.fr](mailto:pedago@42.fr)

*Résumé: Ce document est le sujet de la journée d'initiation au shell.*

# Table des matières

I	Consignes	2
II	Procédure de rendu : Git	3
III	Préambule	5
IV	Exercice 00 : Pré-requis	6
V	Exercice 01 : 42	7
VI	Exercice 02 : testDay00	8
VII	Exercice 03 : Owi, encore...	9
VIII	Exercice 04 : Parse it baby !	10
IX	Exercice 05 : count_files	11
X	Exercice 06 : IP	12
XI	Exercice 07 : diff	13
XII	Exercice 08 : clean	15
XIII	Exercice 09 : Skip	16
XIV	Exercice 10 : midLS	17

# Chapitre I

## Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices.
- Vous serez corrigés par les autres participants du programme d'initiation. C'est le "peer-correcting" de la pédagogie 42 !
- Vous ne devez laisser dans votre répertoire aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Votre manuel de référence s'appelle `Google / man / Internet / ....`
- Lisez attentivement les exemples. Les exercices pourraient bien requérir des choses qui y sont précisées, et non dans le sujet...

# Chapitre II

## Procédure de rendu : Git

- Votre rendu est collecté via un serveur distant. Cela signifie que vous devrez envoyer sur ce serveur votre repertoire de rendu avec vos exercices suivant une arborescence précise.
- Le logiciel utilisé pour ce faire s'appelle **git**.
- Git permet de faire beaucoup de choses, vous trouverez ci-dessous les principales commandes :
- **Git clone** vous permet de **cloner** un repertoire présent sur un serveur directement dans votre repertoire courant. Cela signifie que tous les fichiers présents que vous y aviez mis depuis un autre poste se retrouveront sur votre ordinateur et pourront être modifiés et réenvoyés avec les commandes git suivantes. Cette commande est aussi importante quand vous souhaitez vérifier ce que vous avez "pushé" sur le serveur, n'hésitez pas à en abuser.

```
?> git clone vogsphere@vogsphere.42.fr:piscine/truc/machin/creme
Cloning into 'piscine'[...]
?>
```

- **Git add** vous permet d'ajouter à une liste de fichiers surveillés un ou plusieurs fichiers/repertoires. Ils seront ajoutés en l'état ce qui veut dire que si vous les modifiez par la suite, il faudra les ajouter à nouveau pour mettre à jour la liste.

```
?> git add file directory
?>
```

- **Git commit -m** vous permet de "fixer" votre liste de fichiers surveillés afin de préparer leur envoi sur le serveur. Le message est obligatoire et permet d'avoir un historique des créations et modifications que vous avez effectué au fur et à mesure de votre travail.

```
?> git commit -m "Ajout des exercices X Y Z"
[master (root-commit) 4e8b2aa] Ajout des exercices X Y Z
3 file changed, 500 insertions(+)
create mode 100755 X Y Z
?>
```

- **Git push origin master**, la commande ultime, celle qui vous permet d'envoyer sur le serveur vos exercices qui y seront collectés par la Moulinette. Vérifiez bien son retour, le moindre message d'erreur signifie que vous n'avez rien pu envoyer et que par conséquent votre répertoire sur le serveur est vide (ce qui serait dommage pour votre note). Si vous avez le moindre doute, contactez un membre du staff 42.

```
?> git push origin master
[...]  
X files written.  
?>
```



L'adresse git de votre dépôt se trouve sur l'intranet et dans le fichier config du répertoire `.git` de votre dossier de rendu, c'est avec elle que vous pourrez "git clone".

# Chapitre III

## Préambule

Voici les paroles du générique de Nicky Larson :

Une ombre file dans la nuit  
C'est un assassin qui s'enfuit  
Et comme un démon il sourit  
Son crime restera impuni  
Une voiture qui surgit  
Un coup de frein, des pneus qui crient  
Un coup de feu qui retentit  
La justice s'appelle Nicky

[Refrain]  
Dans la chaleur  
De la nuit  
Le mal est toujours puni  
Aucun danger ne l'impressionne  
Les coups durs il les affectionne  
Et la justice le passionne  
Nicky Larson ne craint personne  
Lorsque les coups de feu résonnent  
Comme un éclair il tourbillonne  
Surtout si la fille est mignonne  
Nicky Larson ne craint personne


Comme un chasseur il suit sa proie  
Pour que la justice et le droit  
Triomphent, il est prêt à donner  
Toute sa vie sans hésiter  
Quand sa silhouette apparaît  
Les méchants se mettent à trembler  
Ils savent qu'ils ne pourront jamais  
Echapper à ce justicier

[Refrain]

Ce sujet n'a, malheureusement, rien à voir avec Nicky Larson.

# Chapitre IV

## Exercice 00 : Pré-requis

	Exercice : 00
Prérequis	
Remarques : n/a	

- S'inscrire sur l'intra à toutes les activités de la semaine.
- Ecrire un mail à `gargamel@42.fr`.
- Ce mail doit avoir pour objet :

`[Monsieur le grand gourou mondial][login][je vous flatte un peu]`

- Un petit mot sympa dans le contenu du mail, avec au moins les mots suivants :


`42; autographe; maitre du monde; beau; fort; idole; grosse; manivelle`



Évidemment "login", c'est votre login.

# Chapitre V

## Exercice 01 : 42

	Exercice : 01
Les vrais savent afficher un 42	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : 42	
Fonctions Autorisées : Aucune	
Remarques : n/a	

- Créez un fichier 42 qui, lorsque l'on fait un `cat` dessus, affiche "42" suivi d'un retour à la ligne.

```
?> cat 42
42
?>
```




Attention, `man` = manuel. Faites "`man cat`" dans votre shell.  
Jetez aussi un oeil à `man less`, `man more`, `man tail`.



# Chapitre VI

## Exercice 02 : testDay00

	Exercice : 02
C'est quoi des attributs ?	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : <b>testDay00.tar</b>	
Fonctions Autorisées : Aucune	
Remarques : n/a	

- Créer le fichier **testDay00** dans votre répertoire de rendu.
- De sorte à imiter l'exemple suivant (à part "total 1") :

```
?> ls -l
total 1
-r--r-xr-x  1 login  wheel  40 Jun  1 23:42 testDay00
?>
```

- Une fois l'exercice résolu, vous exécuterez la commande ci-dessous pour créer le fichier :

```
?> tar -cf testDay00.tar testDay00
?>
```




Faites `man touch`, `man rm`, `man chmod`, `man mv`, `man tar`. Oui, c'est horrible.



Les colonnes "login" et "wheel" seront remplacés respectivement par votre login et votre groupe.

# Chapitre VII

## Exercice 03 : Owi, encore...

	Exercice : 03
	Owi, encore...
	Dossier de rendu : <i>ex03/</i>
	Fichiers à rendre : <b>exo3.sh</b>
	Fonctions Autorisées : Aucune
	Remarques : n/a

- Créer un fichier `exo3.sh` contenant une commande permettant de lister tous les fichiers de plus de 100Ko et finissant par l'extension `.docx`. Ce fichier doit pouvoir s'exécuter comme ci-dessous.

```
?> ./exo3.sh
./quentin_hidden_tape.docx
./how_to_train_your_quentin_2.docx
./pictures/quentin_leaked.docx
?>
```




Les noms de fichier ci-dessus ne sont pas représentatifs et ont juste une valeur d'exemple.



Faites man find.

# Chapitre VIII

## Exercice 04 : Parse it baby !

	Exercice : 04
Let's parse it !	
Dossier de rendu : <i>ex04/</i>	
Fichiers à rendre : <code>parse_it_baby.sh</code>	
Fonctions Autorisées : Aucune	
Remarques : n/a	

- Écrire une ligne de commande qui ira lire un fichier .csv comprenant des informations sous la forme `NOM;Prénom;email;ÉCOLE` et qui en extraira le mail. Vous ne récupérerez que la partie avant le @ de l'adresse email.
- Vous trouverez le fichier .csv de référence en annexe sur l'intranet 42.
- La sortie attendue :

```
?> ./parse_it_baby.sh | cat -e
quentin.bollache$
laurie.mezard$
andre.aubin$
[...]
mathieu.mahe$
frederic.grati$
?>
```




Faites man cut.



La commande "`ls -la | wc -l`" se fait en deux temps, la seconde partie utilise le résultat de la première.  
Google shell pipe pour en savoir plus.

# Chapitre IX

## Exercice 05 : count\_files

	Exercice : 05
	count_files.sh
	Dossier de rendu : <i>ex05/</i>
	Fichiers à rendre : <b>count_files.sh</b>
	Fonctions Autorisées : Aucune
	Remarques : n/a

- Écrire une ligne de commande qui affiche le nombre de fichiers réguliers et de répertoires dans le répertoire courant et tous ses sous-répertoires, y compris le "." du répertoire de départ.
- Exemple de sortie :


```
?> ./count_files.sh | cat -e
42$
?>
```



Faites man wc.

# Chapitre X

## Exercice 06 : IP

	Exercice : 06
IP.sh	
Dossier de rendu : <i>ex06/</i>	
Fichiers à rendre : <b>IP.sh</b>	
Fonctions Autorisées : <b>Aucune</b>	
Remarques : <b>n/a</b>	

- Écrire une ligne de commande qui affiche l'adresse IP sur le réseau interne de votre machine. Cette adresse sera suivie d'un retour à la ligne.

```
?> ./IP.sh | cat -e
10.43.1.224$
?>
```




Faites man ifconfig, man grep, man awk, man cut.



La commande "awk 'FNR == N print'" vous permettra de n'afficher que la Nième ligne d'un résultat.

# Chapitre XI

## Exercice 07 : diff

	Exercice : 07
Dossier de rendu : <i>ex07/</i>	
Fichiers à rendre : <i>ab.diff</i>	
Fonctions Autorisées : Aucune	
Remarques : n/a	

- Créer les fichiers a et b tels que :

```
?> cat -e a
STARWARS$
Episode IV, A NEW HOPE It is a period of civil war.$
$
Rebel spaceships, striking from a hidden base, have won their first victory against the evil
Galactic Empire.$
During the battle, Rebel spies managed to steal secret plans to the Empire's ultimate weapon,
the DEATH STAR,$
an armored space station with enough power to destroy an entire planet.$
$
Pursued by the Empire's sinister agents, Princess Leia races home aboard her starship,
custodian of the stolen plans that can save her people and restore freedom to the galaxy...$
$
```

```
?> cat -e b
STARWARS$
Episode IV, A NEW HOPE It is a period of civil war.$
$
Rebel spaceship, striking from a hidden base, have won their first victory against the evil
Galactic Empire.$
During the battle, Rebel spies managed to steal secret plans to the Empire's ultimate weapon,
the DEATH STAR,$
an armored space station with enough power to destroy an entire planett.$
$
Pursued by the Empire's sinisters agents, Princess Leia races home aboard her starship,
custodian of the stolen plans that can save her people and restore freedom to the galaxy...$
$
```

- Vous rendrez un fichier *ab.diff* constitué du résultat de la commande *diff* entre a et b.




Faites man diff, google redirection shell.



Attention à l'option "-e" de cat : les retours à la ligne de l'exemple n'en sont pas tous, fiez-vous aux "\$".

# Chapitre XII

## Exercice 08 : clean

	Exercice : 08
Dossier de rendu : <i>ex08/</i>	
Fichiers à rendre : <b>clean</b>	
Fonctions Autorisées : <b>Aucune</b>	
Remarques : <b>n/a</b>	

- Placez dans un fichier **clean** une ligne de commande qui va rechercher, à partir du répertoire courant et dans tous ses sous-répertoires, les fichiers se terminant par ~, ou commençant et se terminant par #.
- La ligne de commande affichera et effacera les fichiers trouvés.




Faites man find, cherchez "exec" et "delete".



# Chapitre XIII

## Exercice 09 : Skip

	Exercice : 09
skip.sh	
Dossier de rendu : <i>ex09/</i>	
Fichiers à rendre : <b>skip.sh</b>	
Fonctions Autorisées : <b>Aucune</b>	
Remarques : <b>n/a</b>	


- Écrire une ligne de commande qui affiche un `ls -l` une ligne sur deux, à partir de la première.



Faites man awk.

# Chapitre XIV

## Exercice 10 : midLS

	Exercice : 10
	midLS
	Dossier de rendu : <i>ex10/</i>
	Fichiers à rendre : <b>midLS</b>
	Fonctions Autorisées : <b>Aucune</b>
	Remarques : <b>n/a</b>

- Placez dans un fichier **midLS** la ligne de commande à taper pour lister les fichiers et les répertoires du répertoire courant, mais pas les fichiers cachés, ni "." ni ".." (rien ne commençant par un point), séparés par des virgules, triés par date de création et de manière à ce que les répertoires soient suivis d'un caractère slash.



RTFM !



Ce qui n'est pas demandé n'est pas à faire !