TD COLD #7

Python

Le but de l'exercice est de réaliser un script python qui s'execute en ligne de commande, comme une des autres commandes du shell. L'intéret est de montrer qu'il est simple de compléter les commandes que vous connaissez simplement en écrivant vos propres scripts.

Moulinette

Une moulinette est un petit script python qui s'execute comme une commande. L'idée est de taper le nom du script dans une ligne de commande, comme on taperait le nom d'une commande, et que le shell execute le script.

Pour cela, il y a plusieurs points à vérifier :
- le script doit avoir les droits d'execution ;
- le script doit être dans un répertoire inscrit dans le PATH ;
- le script doit commencer par « #! » suivi du chemin vers l'interpréteur python

Une bonne pratique pour un script Python est d'avoir tout son code dans des fonctions, et de terminer le script par l'appel de la fonction principale, encapsulé par : `if __name__ == "__main__":`

`__name__` est une var globale fournie par Python.
Si le fichier est inclu dans un autre fichier (*import*), `__name__` a pour nom le nom de l'import, sinon, si le fichier est lancé comme un script, `__name__` a pour nom `__main__`. Cette ligne de code permet donc de rendre le fichier à la fois executable en tant que script, mais aussi importable et donc réutilisable.

Outils python

Python est intrinsèquement prévu pour faire ce genre de scripts. D'ailleurs, Linux tourne grâce à certains de ces scripts, et c'est la raison pour laquelle il ne faut jamais désinstaller la distribution Python livrée de base avec votre Linux.
Un certain nombres de méthodes et d'outils python sont très utiles pour ce genre de taches. Vous en trouverez la liste ci-jointe, à vous de chercher la documentation correspondante pour savoir comment les utiliser.

os.walk()

Permet de se déplacer récursivement dans un répertoire et de lister le contenu.

os.path.join()

Recrée un chemin à partir de composants du chemin, en insérant les bons séparaturs (« / » sous Linux).

os.path.getsize()

Retourne la taille du fichier passé en paramètre.

os.path.splitext()

Retourne l'extension du chemin passé en paramètre.

open()

Ouvre un fichier, sur lequel il est possible de faire une boucle « for » pour récupérer chacune des lignes.

.strip()

Méthode d'une chaine de caractère qui supprime les espaces à gauche et à droite.

.startswith()

Méthode d'une chaine de caractère qui retourne vrai si la chaine commence par la sous-chaine donnée en paramètre.

.split()

Méthode d'une chaine de caractère qui sépare la chaine en sous-chaines, par défaut en utilisant les espaces comme délimiteurs.

argparse

Module standard de Python qui prend en charge le parsing de la ligne de commande, c'est à dire le fait de reconnaitre les options et paramètres passés au script.