Module os

Le module os est un module très pratique permettant d'interagir au travers de fonctions avec le système d'exploitation, vos fichiers et répertoires. Vous reconnaitrez certaines des commandes habituelles UNIX.

Savoir où l'on se trouve et se déplacer :

`os.getcwd()` # renvoie le chemin du répertoire courant

`os.chdir("/home/xyz/workdir/")` # nous déplace dans le répertoire workdir (attention de bien lui donner le chemin absolu ou relatif)

Lister les fichiers et répertoires d'un répertoire

`os.listdir()` # liste les fichiers et rép du répertoire courant

`os.listdir("../UEpython/seance6")` # liste les fichiers et rép du répertoire dont le chemin relatif est ../UEpython/ (i.e., dans le répertoire UEpython qui se situe au-dessus du répertoire courant)

Créer un répertoire

`os.mkdir("seance6")` # crée un répertoire seance6 dans le répertoire courant

`os.mkdir("../UEpython/seance6")` # crée le répertoire dont le chemin relatif est ../UEpython/

Effacer un répertoire/fichier

`os.rmdir("seance6")` # efface le répertoire seance6 si il se trouve dans le répertoire courant ! Sinon, pensez à bien indiquer le chemin du répertoire à effacer

`os.remove("genome.fa")` # efface le fichier genome.fa. Encore un fois, pensez à bien indiquer le chemin du fichier à effacer et/où d'anticiper sa localisation.

Appeler les commandes du shell

`os.system("cp genome.fa ../fichiers_fasta")` # copie le fichier genome.fa dans le répertoire ../fichiers_fasta

`os.system("python compute_Distmap.py -pdb 1brs.pdb")` # très pratique pour appeler d'autres scripts par ex, conférer aussi la commande subprocess() qui gèrre plus finement les processus.

subprocess.run(["ls","-lrth"])

Manipuler les chemins de fichiers

`os.path.basename("../UEpython/seance6")` # renvoie le nom du fichier dépourvu de son chemin (même si le fichier n'existe pas !).

`os.path.dirname("../UEpython/seance6")` # inverse de basename, renvoie le chemin du fichier dépourvu de son nom de fichier (même si le fichier n'existe pas !).

`os.path.exists("../UEpython/seance6")` # renvoie True si le répertoire existe

`os.path.exists('../Seance3/1brs.pdb')` # fonctionne aussi pour les fichiers

`os.path.isdir('../Seance3/seance6')` # renvoie True si seance6 est un répertoire

`os.path.isfile('../Seance3/1brs.pdb')` # renvoie True si 1brs.pdb est un répertoire

Petit détour par le module glob

Très pratique pour manipuler les chemins. Va rechercher les chemins matchant un pattern spécifique.

`filelist = glob.glob("../Seance1/*pdb")` # filelist contient maintenant tous les fichiers terminant par l'extension *pdb

Très pratique car ensuite je peux manipuler mes fichiers comme je veux, les ouvrir, les renommer, les déplacer etc

`glob.iglob("../Seance1/*pdb")` # renvoie un itérateur, pratique si je ne veux pas garder en mémoire la liste des fichiers (fait sens lorsqu'on manipule de nombreux fichiers), on travaille alors à la volée sur chaque fichier.

```
for i in glob.iglob("../Seance2/*txt"):
     f = open(i)
     li = f.readlines()
     f.close()
     print(li)
```