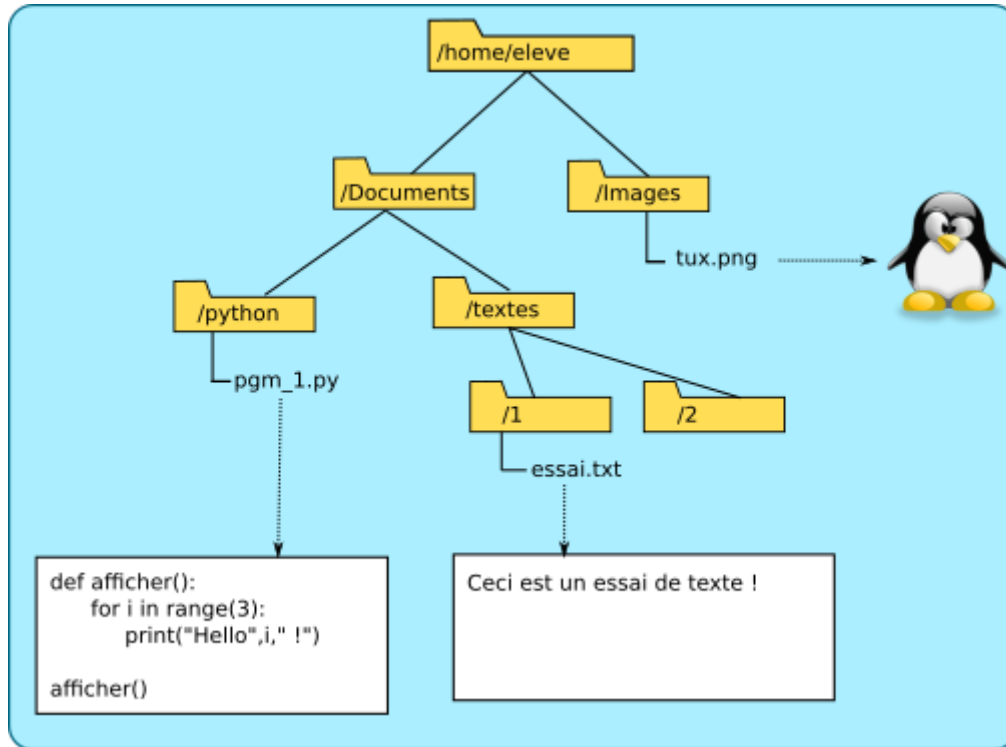


# TP : Système d'exploitation GNU/Linux

---

## -1- Organiser ses fichiers et dossiers

On considère l'arborescence de dossiers et fichiers suivante :



- 1) Vérifier que cette arborescence est bien présente sur la machine. S'ils manquent des fichiers et des dossiers, les ajouter.
- 2) Donner les chemins absolus des 3 fichiers `essai.txt`, `pgm_1.py` et `tux.png`.

## -2- Le terminal linux

Linux peut être utilisé sans aucune interface graphique ce qui en fait sa force. L'utilisation de GNU/Linux se fait alors en ligne de commandes. Le **terminal** ou **console** est une application qui permet de saisir ces lignes de commandes.

- 1) Ouvrir une nouvelle fenêtre de "Terminal". Dans quel dossier de l'arborescence êtes-vous ?
- 2) Quelles sont les commandes linux qui permettent :
  - de lister le contenu d'un dossier :
  - de changer de dossier :
  - de créer un nouveau dossier :
  - de déplacer un fichier dans un autre dossier :
- 3) Déplacez-vous dans le dossier `python` puis exécutez le programme en python `pgm_1.py`. On doit obtenir un affichage !

### -3- Afficher les processus

La commande `ps` affiche les processus utilisateur liés à la console.

Avec certaines options, la commande renvoie des informations plus ciblées. Par exemple :

- l'option `-f` affiche plus d'informations sur les processus liés au terminal (console).
  - l'option `-u` suivi du nom utilisateur affiche les processus de l'utilisateur.
  - l'option `-e` affiche tous les processus de tous les utilisateurs.
  - l'option `-o` suivi d'arguments affiche les informations pour chaque processus et pour chaque argument passé en paramètres.
- 1) Afficher les processus utilisateur avec la commande `ps` puis avec la commande `ps -f`.
  - 2) Afficher les processus utilisateur avec la commande `ps -u nom utilisateur`.
  - 3) Reprendre la commande précédente en y ajoutant l'option `-f` pour avoir plus d'informations.
  - 4) Afficher les processus utilisateur avec les arguments `pid`, `ppid` et `command`.

### -4- La gestion des processus

- 1) Lancer un navigateur et observer dans les processus sa présence.
  - a) Combien de processus enfants ont été créés ?
  - b) Quel est le PID du processus parent.
- 2) Il est possible de terminer (tuer) un processus avec la commande `kill` suivi du numéro de processus. Mettre fin en ligne de commande au processus associé à votre navigateur web.
- 3) La console permet de lancer des programmes en tâche de fond ou en arrière plan et d'avoir encore accès à la console.
  - a) Écrivez un programme en Python qui boucle infiniment puis exécutez-le dans le terminal.
  - b) Est-il possible d'afficher les processus utilisateur ? Pourquoi ? Comment reprendre la main ?
  - c) Pour éviter le problème précédent, on peut lancer le processus en arrière plan. Relancez votre programme Python en ajoutant `&` en fin de commande.
  - d) Afficher les processus liés à la console. Afficher les processus avec leur durée d'exécution et vérifier que votre programme python s'exécute bien en arrière plan.
  - e) Arrêter votre programme Python avec la commande `kill`.
- 4) Python dispose du module `os` qui permet d'obtenir des informations sur le système d'exploitation. Par exemple, il est possible de récupérer le `pid` du programme exécuté avec la méthode `getpid()`
  - a) Ajouter dans votre programme Python une variable `pid` qui a pour valeur le `pid` du programme Python en cours d'exécution. On veillera à afficher cette variable.
  - b) Lance votre programme en arrière plan et notez son PID.
  - c) Mettez fin à l'exécution de ce programme en utilisant son PID.