Tutorat M1 - Bash & Python

Amélie Gruel

Septembre 2019

1 Partie 1 - Bash

Pour commencer ce tutorat, on va d'abord créer à l'aide de commandes bash un répertoire sur le Bureau, intitulé `Tutorats`. Déplace toi dans ce répertoire, puis ajoutez y un répertoire `Tutorat1`.

Ensuite, balade toi dans tes répertoires. Petit conseil : l'idéal serait d'avoir un fichier `M1`, puis un sous fichier `S1` et des sous-sous-fichiers correspondant vos différents cours !

Si tu n'as pas encore organisé ton espace de travail, crée au moins un répertoire `M1` dans les Documents, et déplace-y donc le répertoire `Tutorats` qu'on a créé précédemment dans le Bureau.

Déplace toi à l'intérieur de ce fichier, puis dans celui `Tutorat1` : une fois à l'intérieur, crée un fichier `exercice1`.

Finalement, tu as changé d'avis et tu veux apporter de la joie dans ton terminal : renomme le dossier `Tutorats` en `TutoratParLesMeilleursM2AuMonde` (si si, ça met des paillettes dans ta vie je te jure).

Maintenant que tu as aménagé ton espace de travail de manière optimale, tu vas enfin pouvoir t'attaquer à Python : modifie le nom de ton fichier `exercice1` afin de pouvoir l'utiliser en script Python.

2 Partie 2 - Python : les bases

2.1 Exercice 1

Pour commencer à coder un script Python, ouvre le fichier `exercice1.py` dans un éditeur de code.

> NB : il existe de nombreuses commandes pour ouvrir un script dans un éditeur particulier à partir du terminal. Entre autres :
>
> - `code nomdufichier` pour ouvrir le script dans Visual Studio Code
> - `atom nomdufichier &` pour ouvrir le script dans Atom
> - `emacs nomdufichier &` pour ouvrir le script sur Emacs

Question a

Rédige un script qui demande un nombre entier à l'utilisateur, puis affiche successivement les entiers allant de 1 à ce nombre avec un pas de 1.

Le terminal affichera 1, 2, 3, etc jusqu'au nombre donné.

Question b

Reprends le script précédent en indiquant pour chaque chiffre si il est pair ou impair.

Petit indice : il faut utiliser une boucle if/elif.

2.2 Exercice 2

Question a

Écris un script qui définit au hasard un entier comprit être 1 et 10, puis demande à l'utilisateur un chiffre jusqu'à ce qu'il trouve le bon. Tant que l'utilisateur ne trouve pas le bon nombre, le terminal lui demandera un nouveau chiffre. Une fois le bon chiffre trouve, le terminal affichera un message de félicitation.

Il faut utiliser une boucle while.

Question b

Reprends le script précédent, en le modifiant afin qu'après chaque essai raté le terminal indique à l'utilisateur si le chiffre a trouvé est plus petit ou plus grand.

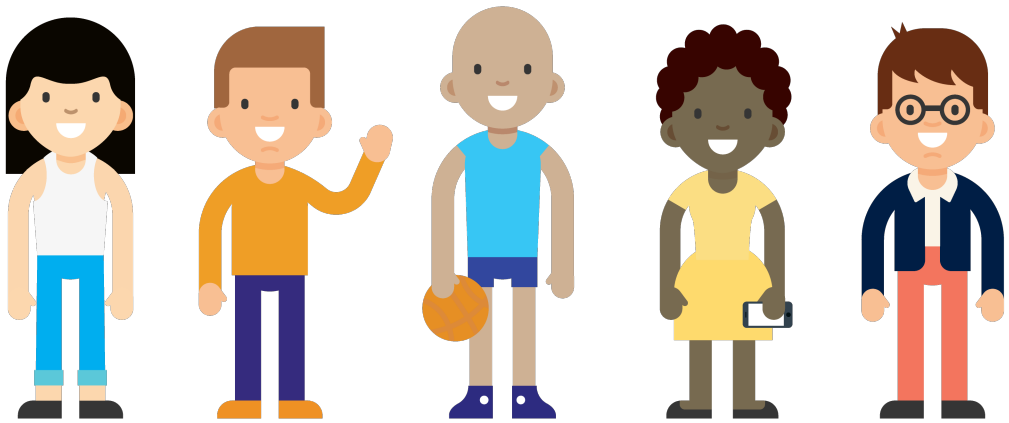
2.3 Exercice 3 - BONUS

La suite de Fibonacci est une suite récursive se comportant ainsi :

$$\begin{cases} U_{n+2} &= U_{n+1} + U_n \\ U_0 &= 0 \\ U_1 &= 1 \end{cases}$$

Crée un programme qui demande à l'utilisateur pour quel n veut-il obtenir le U_n correspondant, puis lui retourne ce résultat.

3 Partie 3 - Python : y a qui dans le master déjà ?



3.1 Exercice 1

Crée une liste nommée `master`, contenant les noms de 5 M2.

Ensuite, ajoute un nom de professeur dans cette liste.

Finalement, tu te dis que les M2 tous seuls sont quand même plus cools, du coup enlève le professeur de ta liste ! Attention, fais le sans récrire entièrement ta liste.

Enfin, affiche le contenu de ta liste sur ton terminal ainsi que sa taille.

Chacune de ces actions doivent avoir lieu dans le même script, sur des lignes successives.

3.2 Exercice 2

Maintenant, on aimerait recenser tous les membres du Master, qu'ils soient en M1, M2, M3 et même les profs !

Mais que faudrait-il utiliser afin de stocker le nom des membres, et y associer leur statut (M1, M2, M3, prof)?

Utilise cette structure de données, que l'on nommera `master_complet`, afin de recenser au moins 6 personnes.

Ajoute y une personne après l'avoir initialisé.

Ensuite, affiche le statut de la personne de ton choix.

Enlève une personne de ton choix.

Maintenant, affiche le contenu de ta structure.

Enfin, affiche uniquement les élèves et professeurs, puis uniquement les statuts possibles en utilisant une méthode associée à ta structure de données.

Comme tout à l'heure, chacune de ces actions doivent avoir lieu dans le même script, sur des lignes de successives.

3.3 Exercice 3 - BONUS

L'utilisateur aimerait pouvoir recenser les membres de son Master à sa guise, en choisissant lui-même au fur et à mesure les actions qu'il veut effectuer. Rédige donc un programme contenant un menu, à l'aide duquel l'utilisateur pourra choisir entre :

1. créer un dictionnaire vide
2. y ajouter un membre du Master et son statut
3. enlever un membre de son choix
4. afficher le contenu du dictionnaire. Chaque nom sera affichée sur une ligne successive, avec le statut correspondant indiqué à la suite entre parenthèses.
5. obtenir le statut d'un certain membre. Le terminal devra alors demander à l'utilisateur d'entrer un nom, puis retournera le statut correspondant.
6. obtenir le nom des tous les membres ayant un certain statut. Le terminal devra alors demander à l'utilisateur d'entrer un statut, puis retournera tous les membres correspondant.
7. quitter le menu

Une fois que tu en es à ce point, préviens un M2 afin qu'il t'explique comment faire un menu.

4 Partie 4 - Python : un petit tour dans le CREMI

Ca y est, les premiers cours dans le CREMI ont eu lieu, tu commences enfin à te repérer à l'intérieur. Mais catastrophe !! il est 8h12, tu es à la bonne salle mais tu viens de recevoir un mail disant que finalement le cours a lieu 2 étages plus haut !

Il faut que tu te dépêches d'aller jusque là bas. Représente donc le CREMI à l'aide d'une liste de liste. Cette liste à deux dimensions génère une grille 10x10, dans laquelle tu devras te déplacer. Ta position initiale en x=5 et y=5 sera indiquée par un caractère (tel que "X").

Crée une fonction te permettant d'afficher le contenu de ta grille sur ton terminal. Une manière possible de l'afficher serait comme dans la figure 1.

```
(base) amelie@L-E7-SAMMY:~/Documents/M2/Tutorats$ python2 PetitTourDansLeCREMI.py
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 X 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

Figure 1: Terminal affichant la grille représentant le CREMI. Les 0 indiquent les cases dans lesquelles tu peux te déplacer, le X la position de ton joueur. Ici il est en position x=5 et y=5.

Une fois que cela est fait, implémente une fonction te demandant dans quelle direction tu veux te diriger, puis effectuant ce déplacement. A la fin de chaque déplacement, la nouvelle grille devra être affichée sur ton terminal (tu peux pour cela appeler la fonction précédent).

Attention ! Tu ne peux pas te déplacer en dehors de la grille.

5 Partie 5 - Python : il était une fois

Télécharge le fichier "partie5.txt", qui contient le script d'un chef d'oeuvre du cinéma américain. Avant qu'un personnage parle, son nom est indiqué ainsi : `"> NOMPERSO"`, sur une seule ligne. Enregistre ces noms dans une liste, afin de connaître quels sont les personnages prenant part au dialogue. Attention à ne pas en mettre en double !

On veut maintenant savoir qui est le personnage principal : pour cela, crée une fonction qui parcourt le fichier ligne par ligne et compte le nombre de fois où chacun des personnages trouvés ci-dessus parle. Le personnage principal sera donc celui dont le nom apparaît le plus souvent.

Pour faire cela, utilise un dictionnaire : la clé sera le nom du personnage, et la valeur sera augmentée de 1 à chaque mot lu correspondant à la clé.

Enfin, tu ne veux pas avoir à lire le dictionnaire afin de trouver qui parle le plus, mais tu veux que ton programme le fasse pour toi ! Rajoute donc une fonction à ton script afin qu'il identifie quel personnage parle le plus (c'est-à-dire quel personnage a la plus grande valeur associée dans le dictionnaire).

Maintenant que tu sais qui parle le plus, enregistre le contenu du dictionnaire obtenu précédemment dans un fichier `blablabla.txt`. Les personnages doivent apparaître selon le nombre de fois où chacun parle : celui qui parle le plus est sur la première ligne, et celui parlant le moins en dernier.