**Exercices 03**

**Mise en situation**

Créez un nouveau fichier python(**.py**), dans votre environnement, pour chacune des questions. Ensuite, assurez-vous de respecter les bonnes pratiques.

* Ajoutez un Docstring
* Avoir des commentaires
* Le code est bien indenté
* Vos variables sont bien nommées
* Le code est bien espacé
* La longueur des lignes sont adéquates
* Aucun mot réservé n'a été utilisé

Aussi, n’oubliez pas d’ajouter en commentaire votre : Nom, date et description du programme.

**Question 01 – Gestion d’une bibliothèque**

Créez un programme de gestion de bibliothèque de livres avec des fonctions pour ajouter, emprunter, retourner et lister les livres.

* Faites une fonction main pour le menu qui permet d’appeler les 4 fonctions.
* Vous devez utiliser une liste (variable globale) pour stocker vos livres et un dictionnaire pour stocker le titre et l’auteur de chaque livre (variable locale).
* Lorsque vous ajoutez un livre :
  + Elle ne retourne aucune donnée
  + Demandez à l’utilisateur le titre et l’auteur du livre.
  + Ajouter une 3e valeur booléenne au livre dont la clé est la disponibilité.
* Lorsque vous listez un livre :
  + Elle retourne le nombre de livres.
  + Affichez chaque livre selon un des 2 formats suivants:
    - TITRE par AUTEUR – Disponible
    - TITRE par AUTEUR – Emprunté
  + De retour dans le main du programme, affichez le total de livres dans la bibliothèque.
* Lorsque vous empruntez un livre :
  + Elle retourne si le livre a été trouvé ou non.
  + Demandez le titre du livre à l’utilisateur.
  + De retour dans le main du programme, si le livre existe et qu’il est disponible, affichez un message que le livre est emprunté et identifiez le livre comme non disponible. Sinon, affichez un message qu’il ne soit pas trouvé ou pas disponible.
* Lorsque vous retournez un livre :
  + Elle retourne le texte en message
  + Demandez le titre du livre à l’utilisateur.
  + Si le livre existe et qu’il est non disponible, affichez un message que le livre est retourné et identifiez le livre comme disponible. Sinon, affichez un message qu’il ne soit pas trouvé ou pas emprunté.

**02 – Jeu du pendu**

Créez une version simplifiée du jeu du pendu. Écrivez des fonctions pour choisir un mot aléatoire, afficher les lettres devinées et vérifier si le mot a été deviné.

* Faites une fonction main pour l’exécution du jeu et l’appel des fonctions.
* Créez une liste de mot (Variable globale) entrée à la main.
* À l’aide d’une fonction, utilisez le module ‘random’ pour déterminer un mot aléatoire à faire trouver.
* Créez une fonction qui permet de faire afficher les lettres devinées et les lettres qui restent à trouver (avec le caractère ‘\_’)
* Créez une fonction qui permet de vérifier si le mot a été trouvé.
* Votre jeu doit rouler jusqu’à tant que le mot soit complètement trouvé, peu importe le nombre d’essais.

**03 – Gestionnaire de hockey**

Créez une nouvelle version de votre gestionnaire de hockey en créant des fonctions et un module. Toutes les fonctions doivent faire partie d’un nouveau fichier que vous allez devoir importer dans votre programme principal. Voici quelques pistes sur les fonctions que vous devriez avoir.

* Demander le nom du joueur
* Vérification de l’existence du joueur
* Vérification s’il y’a des joueurs dans l’équipe
* Vérification des points du joueur pour déterminer qui en a le plus et le moins