Serie des Exercices:
Les dictionnaires et les liste

- ## Exercice 1

  Écrivez une fonction en Python qui prend en entrée une chaîne de caractères et renvoie un dictionnaire contenant le nombre d'occurrences de chaque mot dans la chaîne. Considérez que les mots sont séparés par des espaces

  Exemple :

  ‘‘est avec et est deferent ’ ’ => dict={est :2 , et :1 , avec :1 , déférente :1}

- ## Exercice 2

  Écrivez une fonction en Python qui prend en entrée une liste de nombres et renvoie un dictionnaire contenant le nombre d'occurrences de chaque nombre pair dans la liste

- ## Exercice 3

  Écrivez une fonction en Python qui prend en entrée le chemin vers un fichier texte et renvoie un dictionnaire mettant en correspondance chaque caractère apparaissant dans le fichier avec le nombre de ses occurrences dans le fichier.

- ## Exercice 4

  Écrivez une fonction en Python qui prend en entrée le chemin vers un fichier texte contenant des notes d'étudiants. Chaque ligne du fichier contient deux champs séparés par des points- virgules : le numéro d'étudiant et la note obtenue. La fonction doit retourner un dictionnaire dont les clés sont les numéros d’etudiants et les valeurs sont les notes correspondantes.
  Exemple de fichier "notes_etudiants.txt" :
  213615200;15
  213565488;12
  214665555;18

- ## Exercice 5

  Écrivez une fonction qui prend en entrée une liste de chaînes de caractères, où chaque chaîne représente une date au format "jour/mois/année". La fonction doit retourner une liste de tuples, où chaque tuple contient trois éléments : le jour, le mois et l'année sous forme d'entiers. On pourrait tester la fonction avec la liste suivante : liste_dates = ["12/05/2022", "25/09/2023", "08/11/2021"]

  Le résultat attendu serait : [(12, 5, 2022), (25, 9, 2023), (8, 11, 2021)]

- ## Exercice 6

  Créez un tuple contenant des informations sur une personne (nom, âge, ville, etc.). Écrivez un programme qui décompose ce tuple et l'affiche sous forme de phrases.

- ## Exercice 7

  Créez un tuple avec des éléments de différents types (entiers, chaînes de caractères, flottants). Affichez le type de chaque élément dans le tuple.

- ## Exercice 8

  Écrivez une fonction qui prend en entrée une liste de chaînes de caractères représentant des adresses e-mail. La fonction doit retourner un dictionnaire où les clés sont les noms d'utilisateur extraits des adresses e-mail, et les valeurs sont les domaines correspondants. Par exemple, pour la liste ["john.doe@example.com", "alice.smith@gmail.com", "bob.jones@yahoo.com"], le dictionnaire résultant serait {"john.doe": "example.com", "alice.smith": "gmail.com", "bob.jones": "yahoo.com"}.

- ## Exercice 9

  Écrivez une fonction qui prend en entrée une liste de tuples, chaque tuple représentant une personne avec son nom et son âge. La fonction doit retourner un dictionnaire où les clés sont les noms des personnes et les valeurs sont leurs âges correspondants. Par exemple, pour la liste [("Alice", 25), ("Bob", 30), ("Charlie", 22)], le dictionnaire résultant serait {"Alice": 25, "Bob": 30, "Charlie": 22}.

- ## Exercice 10

  Écrivez une fonction qui prend en entrée une chaîne de caractères représentant une liste de courses, où chaque élément de la liste est séparé par une virgule. La fonction doit retourner un dictionnaire où les clés sont les articles de la liste de courses et les valeurs sont la quantité de chaque article à acheter. Par exemple, pour la chaîne "pommes,bananes,poires,pommes,oranges,bananes", le dictionnaire résultant serait {"pommes": 2, "bananes": 2, "poires": 1, "oranges": 1}.

- ## Exercice 11

  Écrivez une fonction qui prend en entrée une liste de chaînes de caractères représentant des noms de fichiers. La fonction doit retourner un dictionnaire où les clés sont les extensions de fichiers (la partie après le dernier point dans le nom de fichier) et les valeurs sont les noms de fichiers correspondants. Par exemple, pour la liste ["document1.txt", "image.png", "script.js", "document2.txt"], le dictionnaire résultant serait {"txt": ["document1.txt", "document2.txt"], "png": ["image.png"], "js": ["script.js"]}.

- ## Exercice 12

  Écrivez une fonction qui prend en entrée une liste de nombres entiers. La fonction doit retourner un dictionnaire où les clés sont les nombres de la liste et les valeurs sont des chaînes indiquant si le nombre est pair ou impair. Par exemple, pour la liste [2, 5, 8, 11, 14], le dictionnaire résultant serait {2: "pair", 5: "impair", 8: "pair", 11: "impair", 14: "pair"}.

- ## Exercice 13

Écrivez une fonction qui prend en entrée une liste de chaînes de caractères représentant des liens URL. La fonction doit retourner un dictionnaire où les clés sont les noms des sites extraits des liens, et les valeurs sont les types de domaines correspondants (par exemple, ".com", ".org", ".net"). Par exemple, pour la liste ["https://www.example.com", "https://www.wikipedia.org", "https://www.python.org"], le dictionnaire résultant serait {"example": ".com", "wikipedia": ".org", "python": ".org"}.

- ## Exercice 14

Écrivez une fonction qui prend en entrée une liste de chaînes de caractères représentant des numéros de téléphone. La fonction doit retourner un dictionnaire où les clés sont les noms des personnes associées aux numéros, et les valeurs sont les numéros de téléphone au format international. Par exemple, pour la liste ["John Doe: +123456789", "Alice Smith: +987654321", "Bob Jones: +555555555"], le dictionnaire résultant serait {"John Doe": "+12345678

- ## Exercice 15 :Gestion des Mots-clés

Écrivez une fonction analyser_mots_cles qui prend en entrée une chaîne de caractères représentant une liste de mots-clés séparés par des virgules. La fonction doit retourner un dictionnaire où les clés sont les mots-clés, et les valeurs sont le nombre de fois où chaque mot-clé apparaît. Ignorez la casse des mots-clés.

- ## Exercice 16: Comptage des Voyelles

Écrivez une fonction compter_voyelles qui prend en entrée une chaîne de caractères et retourne un dictionnaire où les clés sont les voyelles (a, e, i, o, u) et les valeurs sont le nombre de fois où chaque voyelle apparaît dans la chaîne.

- ## Exercice 17: Gestion des Palindromes

Écrivez une fonction verifier_palindrome qui prend en entrée une liste de mots et retourne un dictionnaire où les clés sont les mots de la liste, et les valeurs sont des booléens indiquant si chaque mot est un palindrome ou non.

- ## Exercice 18: Gestion des Heures

Écrivez une fonction convertir_heures qui prend en entrée une liste de chaînes de caractères représentant des heures au format "hh:mm". La fonction doit retourner un dictionnaire où les clés sont les heures, et les valeurs sont des booléens indiquant si chaque heure est le matin (avant midi) ou l'après-midi (midi inclus et après).

- **Exercice 19: Comparaison de Phrases**

Écrivez une fonction comparer_phrases qui prend en entrée deux phrases et retourne un dictionnaire où les clés sont les mots communs aux deux phrases, et les valeurs sont des tuples indiquant la position de chaque mot dans chaque phrase.

- **Exercice 20 :**

Une bibliothèque souhaite informatiser la gestion de ses livres. On suppose que les informations sur les livres sont stockées dans un fichier "livres.txt" sous la forme suivante :

101,Harry Potter,JK Rowling,Fantasy,500
202,To Kill a Mockingbird,Harper Lee,Fiction,350
303,The Great Gatsby,F. Scott Fitzgerald,Classic,200
404,The Hobbit,J.R.R. Tolkien,Fantasy,450
505,1984,George Orwell,Dystopian,300

Chaque ligne du fichier représente un livre avec les informations suivantes : ID du livre, Titre, Auteur, Genre et Nombre de pages.

1. Écrire un code qui permet d'ouvrir le fichier, extraire toutes les informations et les stocker dans un dictionnaire comme suit :
   Bibliotheque = {
   101: {'titre': 'Harry Potter', 'auteur': 'JK Rowling', 'genre': 'Fantasy', 'pages': 500},
   202: {'titre': 'To Kill a Mockingbird', 'auteur': 'Harper Lee', 'genre': 'Fiction', 'pages': 350},
   303: {'titre': 'The Great Gatsby', 'auteur': 'F. Scott Fitzgerald', 'genre': 'Classic', 'pages': 200},
   404: {'titre': 'The Hobbit', 'auteur': 'J.R.R. Tolkien', 'genre': 'Fantasy', 'pages': 450},
   505: {'titre': '1984', 'auteur': 'George Orwell', 'genre': 'Dystopian', 'pages': 300}
   }
2. Écrire une procédure **afficher_livre(Bibliotheque, livre_id)** qui prend un dictionnaire et l'ID d'un livre en paramètre, et qui affiche les informations du livre.
3. Écrire une fonction **ajouter_livre(Bibliotheque, livre_id, titre, auteur, genre, pages)** qui prend un dictionnaire, les informations d'un nouveau livre, et qui ajoute ce livre à la bibliothèque.
4. Écrire une fonction **rechercher_livre(Bibliotheque, mot_cle)** qui prend un dictionnaire et un mot clé en paramètre, et qui retourne une liste des livres dont le titre ou l'auteur contient le mot clé.
5. Écrire une fonction **supprimer_livre(Bibliotheque, livre_id)** qui prend un dictionnaire et l'ID d'un livre en paramètre, et qui supprime le livre correspondant.

6. Écrire une fonction **nombre_total_pages(Bibliotheque)** qui prend un dictionnaire et qui retourne le nombre total de pages dans la bibliothèque.
7. Dans le programme principal, proposer à l'utilisateur un menu pour appeler les fonctions déclarées précédemment.

---------- MENU ----------
1 – Afficher un livre
2 – Ajouter un livre
3 - Rechercher un livre
4 – Supprimer un livre
5 - Afficher le nombre total de pages
6 - Quitter

Entrez votre choix :

- ## Exercice 21:

Un magasin souhaite informatiser la gestion de son inventaire. Les informations sur les produits sont stockées dans un fichier "inventaire.txt" avec le format suivant :

101,Ordinateur portable,HP,800,15
202,Smartphone,Samsung,600,20
303,Tablette,iPad,400,10
404,Imprimante,Epson,200,5

Chaque ligne du fichier représente un produit avec les informations suivantes : ID du produit, Nom du produit, Marque, Prix unitaire, et Stock disponible. Écrire un code qui permet d'ouvrir le fichier, extraire toutes les informations, et les stocker dans un dictionnaire comme suit :

Inventaire = {
101: {'nom_produit': 'Ordinateur portable', 'marque': 'HP', 'prix_unitaire': 800, 'stock_disponible': 15},
202: {'nom_produit': 'Smartphone', 'marque': 'Samsung', 'prix_unitaire': 600, 'stock_disponible': 20},
303: {'nom_produit': 'Tablette', 'marque': 'iPad', 'prix_unitaire': 400, 'stock_disponible': 10},
404: {'nom_produit': 'Imprimante', 'marque': 'Epson', 'prix_unitaire': 200, 'stock_disponible': 5}
}

1. **Ajout, Recherche et Suppression :**
    - **Écrire une fonction ajouter_produit(Inventaire, produit_id, nom, marque, prix, stock) qui prend un dictionnaire et les informations d'un nouveau produit, et qui ajoute ce produit à l'inventaire.**

- **Écrire une fonction rechercher_produit(Inventaire, mot_cle) qui prend un dictionnaire et un mot clé en paramètre, et qui retourne une liste des produits dont le nom ou la marque contient le mot clé.**
- **Écrire une fonction supprimer_produit(Inventaire, produit_id) qui prend un dictionnaire et l'ID d'un produit en paramètre, et qui supprime le produit correspondant.**

2. **Gestion des Ventes :**
   - **Écrire une fonction effectuer_vente(Inventaire, produit_id, quantite) qui prend un dictionnaire, l'ID d'un produit, et une quantité en paramètre, et qui met à jour le stock disponible après une vente.**
   - **Écrire une fonction afficher_stock_faible(Inventaire, seuil) qui prend un dictionnaire et un seuil en paramètre, et qui affiche les produits dont le stock disponible est inférieur au seuil.**
3. **Menu Principal :**

Dans le programme principal, proposer à l'utilisateur un menu pour appeler les fonctions déclarées précédemment :

---------- MENU ----------
1 – Afficher l'inventaire
2 – Ajouter un produit
3 - Rechercher un produit
4 – Supprimer un produit
5 - Effectuer une vente
6 - Afficher les produits avec un stock faible
7 - Quitter

Entrez votre choix :

- ## Exercice 22:

Créez un fichier "universite.txt" qui contient des informations sur les étudiants et les cours de l'université. Chaque ligne représente un enregistrement avec les informations suivantes :

1. **Gestion des étudiants et des cours :**

- Créez un fichier "universite.txt" qui contient des informations sur les étudiants et les cours de l'université. Chaque ligne représente un enregistrement avec les informations suivantes :
  101,John Doe,Informatique
  102,Jane Smith,Mathématiques
  103,Bob Johnson,Physique
  104,Alice Brown,Informatique
  201,Introduction à la Programmation,3
  202,Calcul Différentiel,4
  203,Électromagnétisme,3

    - Les trois premières colonnes représentent l'ID de l'étudiant, le nom de l'étudiant et le département dans lequel l'étudiant est inscrit.
    - Les trois dernières colonnes représentent l'ID du cours, le nom du cours et le nombre de crédits du cours.
  - Écrire un code qui lit ce fichier, crée deux dictionnaires distincts (**etudiants** et **cours**) pour stocker les informations.
2. **Opérations sur les données :**
  - Écrire une fonction **afficher_etudiants_par_departement(etudiants, departement)** qui prend un dictionnaire d'étudiants et un département en paramètre, et qui affiche tous les étudiants inscrits dans ce département.
  - Écrire une fonction **afficher_cours_par_etudiant(etudiants, cours, etudiant_id)** qui prend les dictionnaires d'étudiants et de cours, ainsi que l'ID d'un étudiant en paramètre, et qui affiche tous les cours auxquels cet étudiant est inscrit.
  - Écrire une fonction **calculer_total_credits(etudiants, cours, etudiant_id)** qui prend les dictionnaires d'étudiants et de cours, ainsi que l'ID d'un étudiant en paramètre, et qui retourne le nombre total de crédits que cet étudiant suit.
  - Écrire une fonction **afficher_cours_populaires(cours, seuil)** qui prend le dictionnaire de cours et un seuil en paramètre, et qui affiche tous les cours ayant un nombre d'étudiants inscrits supérieur au seuil.
3. **Menu Principal :**
   Dans le programme principal, proposer à l'utilisateur un menu pour appeler les fonctions déclarées précédemment :
   ---------- MENU ----------
   1 – Afficher les étudiants par département
   2 – Afficher les cours par étudiant
   3 - Calculer le total de crédits pour un étudiant
   4 - Afficher les cours populaires

5 - Quitter

Entrez votre choix :

Par :da-hab