



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**

Membre de **HONORIS UNITED UNIVERSITIES**

Travaux Pratiques

Structure de données

Module : Python

Dr. ABOUABID Hamza

2 janvier 2025

1 Les Listes

1.1 Exercice 1

1. **Créer une liste** nommée `fruits` contenant les éléments suivants : "pomme", "banane", "cerise".
2. **Ajouter** le fruit "orange" à la liste `fruits` en utilisant la méthode appropriée.
3. **Supprimer** "banane" de la liste `fruits`.
4. **Afficher** le troisième élément de la liste `fruits`.

Exercice 2

1. **Trier** la liste `fruits` par ordre alphabétique.
2. **Inverser** l'ordre des éléments dans la liste `fruits`.
3. **Vérifier** si "cerise" est présent dans la liste `fruits` et afficher le résultat.

Exercice 3

1. **Tri des Fréquences** : Vous disposez de la liste suivante représentant les fréquences (en Hz) utilisées dans un système de communication : `frequencies = [2400, 1800, 2100, 1900, 2300, 2000]`. Triez cette liste par ordre croissant.
2. **Recherche et Insertion** : Vérifiez si la fréquence 2200 Hz est présente dans la liste `frequencies`. Si ce n'est pas le cas, insérez-la à la position appropriée pour maintenir l'ordre croissant.
3. **Calcul de la Moyenne** : Calculez la fréquence moyenne utilisée dans la liste `frequencies`.
4. **Filtrage des Fréquences** : Créez une nouvelle liste `frequencies_sup` contenant uniquement les fréquences supérieures ou égales à 2000 Hz.

2 Les Tuples

2.1 Exercice 4

1. **Création d'un Tuple de Paramètres de Circuit** : Créez un tuple nommé `parametres_circuit` contenant les valeurs suivantes : (5, 12, 220, "AC"), représentant respectivement le courant (A), la tension (V), la fréquence (Hz), et le type de courant.
2. **Accès aux Éléments** : Affichez la tension (12 V) en accédant au deuxième élément du tuple `parametres_circuit`.
3. **Décomposition du Tuple** : Décomposez le tuple `parametres_circuit` en variables individuelles : `courant`, `tension`, `frequence`, `type_courant`.
4. **Utilisation dans une Fonction** : Écrivez une fonction `description_circuit` qui prend `parametres_circuit` comme argument et retourne une chaîne de caractères décrivant le circuit, par exemple : "Circuit de 5A, 12V, 220Hz en courant AC".

2.2 Exercice 5

1. **Tentative de Modification** : Essayez d'ajouter un élément "DC" au tuple `parametres_circuit`. Observez l'erreur générée et expliquez pourquoi cette opération échoue.
2. **Création d'un Nouveau Tuple** : Créez un nouveau tuple `parametres_circuit_modifie` en remplaçant "AC" par "DC" dans `parametres_circuit`.

3 Les Dictionnaires

3.1 Exercice 6

1. **Création d'un Dictionnaire de Composants** : Créez un dictionnaire nommé `composants_stock` avec les paires clé-valeur suivantes :
 - "Résistance" : 150
 - "Condensateur" : 80
 - "Inductance" : 60
 - "Transistor" : 45
2. **Accès aux Valeurs** : Affichez le nombre en stock de "Condensateur".
3. **Mise à Jour des Stocks** : Augmentez le stock de "Résistance" de 50 unités.
4. **Ajout de Nouveaux Composants** : Ajoutez les composants "Diode" avec un stock de 30 et "Microcontrôleur" avec un stock de 20 au dictionnaire `composants_stock`.
5. **Suppression d'un Composant** : Supprimez "Inductance" du dictionnaire `composants_stock`.

3.2 Exercice 7

1. **Affichage des Clés et Valeurs** : Affichez toutes les clés (noms des composants) et toutes les valeurs (stocks) du dictionnaire `composants_stock`.
2. **Recherche Sécurisée** : Utilisez la méthode `get` pour récupérer le stock de "Transistor" et celui de "Capteur", en fournissant une valeur par défaut de 0 si la clé n'existe pas.
3. **Itération sur le Dictionnaire** : Écrivez une boucle qui parcourt le dictionnaire `composants_stock` et affiche chaque composant avec son stock sous la forme : "Composant : Résistance, Stock : 200".
4. **Fusion de Dictionnaires** : Créez un deuxième dictionnaire `composants_suppl` contenant "Capteur" : 25 et "Relay" : 15. Mettez à jour `composants_stock` en y ajoutant les éléments de `composants_suppl`.

4 Comparaison des Structures

4.1 Exercice 8

Remplissez le tableau suivant en fonction des structures de données étudiées : listes, tuples et dictionnaires.

Caractéristique	Liste	Tuple	Dictionnaire
Ordonnée Modifiable Permet les doublons Accès par clé/indice Utilisation principale			

4.2 Exercice 9

1. **Choix de la Structure** : Dans le cadre de la gestion des stocks d'un laboratoire de génie électrique, justifiez l'utilisation d'un dictionnaire plutôt qu'une liste pour stocker les composants et leurs quantités.
2. **Optimisation des Performances** : Expliquez pourquoi un tuple serait préférable qu'une liste pour stocker les paramètres constants d'un circuit électrique.
3. **Gestion des Données** : Donnez un exemple concret où l'utilisation simultanée d'une liste et d'un dictionnaire serait bénéfique pour optimiser la gestion des données dans un projet industriel.

5 Projet Pratique

5.1 Gestion d'un Système de Surveillance de Production

Vous êtes chargé de développer un programme pour surveiller et analyser les données de production dans une usine industrielle. Utilisez les structures de données appropriées pour stocker et manipuler les informations.

1. **Collecte des Données** : Créez une liste `produits` contenant les noms des produits fabriqués : "Moteur Électrique", "Variateur de Fréquence", "Capteur de Température", "Automate Programmable".
2. **Enregistrement des Statistiques de Production** : Créez un dictionnaire `statistiques_production` où chaque clé est le nom d'un produit et chaque valeur est un tuple contenant le nombre d'unités produites ce mois-ci et le taux de défaut (en pourcentage). Initialisez-le avec les valeurs suivantes :
 - "Moteur Électrique" : (500, 2.5)
 - "Variateur de Fréquence" : (300, 1.8)
 - "Capteur de Température" : (800, 0.5)
 - "Automate Programmable" : (200, 3.0)
3. **Mise à Jour des Données** : Ajoutez un nouveau produit "Convertisseur DC-DC" avec (150, 2.2) dans le dictionnaire `statistiques_production`.
4. **Analyse des Données** : Écrivez un script Python qui parcourt le dictionnaire `statistiques_production` et affiche les produits dont le taux de défaut est supérieur à 2%.
5. **Calcul des Totaux** : Calculez le nombre total d'unités produites ce mois-ci et le taux de défaut moyen de tous les produits.
6. **Visualisation des Données** : Créez une liste `taux_defaut` contenant tous les taux de défaut et utilisez-la pour déterminer le produit le plus fiable (avec le taux de défaut le plus bas).

7. **Export des Résultats** : Convertissez le dictionnaire `statistiques_production` en une liste de tuples et sauvegardez-le dans un fichier `production_janvier.csv` au format CSV.