

Exercices de Révision (NSI)

Types Construits & Données en Tables

Partie A : Les Types Construits (p-uplets, listes, dictionnaires)

Exercice 1 - Compréhension de p-uplets (Tuples)

On définit la variable suivante : `mon_tuple = (10, 20, 30, 40)`.

1. Que renvoie l'instruction `mon_tuple[2]` ?
2. L'instruction `mon_tuple[1] = 25` provoque une erreur. Expliquez pourquoi.
3. Comment créer un nouveau tuple contenant les mêmes valeurs suivies de la valeur 50 ?

Exercice 2 - Listes en compréhension

Complétez le code ci-dessous pour générer la liste des carrés des nombres pairs de 0 à 20 inclus.

```
1 nombres = [x**2 for x in range(.....) if .....]  
2 # Résultat attendu : [0, 4, 16, 36, ..., 400]  
3
```

Exercice 3 - Dictionnaires et accès

Soit le dictionnaire suivant représentant un élève :

```
1 eleve = {"nom": "Dupont", "prenom": "Jean", "notes": [12, 15, 14]}  
2
```

Écrivez les instructions Python pour :

1. Accéder au prénom de l'élève.
2. Ajouter une nouvelle note (18) à la liste des notes.
3. Modifier le nom de famille en "Durand".

Exercice 4 - Algorithme de recherche

Écrivez une fonction `compte_occurrences(texte)` qui prend une chaîne de caractères en entrée et renvoie un dictionnaire où :

- Les clés sont les caractères présents dans le texte.
- Les valeurs sont le nombre d'apparitions de chaque caractère.

Exemple : `"aba" → {'a': 2, 'b': 1}`

Exercice 5 - *Analyse de code*

Que va afficher le code suivant ? Détaillez le déroulement.

```
1 dico = {"pomme": 3, "poire": 5, "banane": 2}
2 total = 0
3 for cle, valeur in dico.items():
4     if valeur > 2:
5         total = total + valeur
6 print(total)
7
```

Exercice 6 - *Listes de listes*

On représente une grille de morpion par une liste de listes :

```
1 grille = [
2     ["X", "O", "X"],
3     [" ", "X", "O"],
4     ["O", " ", "X"]
5 ]
6
```

Quelle instruction permet d'accéder à l'élément situé au centre de la grille (2ème ligne, 2ème colonne) ?

Exercice 7 - *Construction de structure*

Vous disposez de deux listes :

```
1 matieres = ["Maths", "NSI", "Anglais"]
2 moyennes = [14, 16, 12]
3
```

Écrivez une boucle (ou une compréhension) pour créer le dictionnaire suivant :
{"Maths": 14, "NSI": 16, "Anglais": 12}

Partie B : Données en Tables (CSV, Traitement)

Exercice 8 - *Structure de données*

Pour traiter un fichier CSV en Python, on utilise souvent une *liste de dictionnaires*. Donnez un exemple (en écrivant le code Python littéral) d'une variable `table` contenant deux enregistrements d'un carnet d'adresses (champs : "nom", "email").

Exercice 9 - *Compléter le code (Chargement CSV)*

Le fichier `eleves.csv` contient des données avec des séparateurs point-virgule. Complétez la fonction.

```
1 import csv
2
3 def charger_csv(nom_fichier):
4     table = []
5     with open(nom_fichier, mode='...', encoding='utf-8') as f:
6         lecteur = csv.DictReader(f, delimiter='...')
```

```
7         for ligne in lecteur:
8             table.append(dict(...))
9     return table
10
```

Exercice 10 - *Filtrage de données*

On dispose d'une table `films` (liste de dictionnaires) où chaque dictionnaire a les clés : "titre", "annee", "realisateur".

Écrivez une fonction `filtrer_recents(table, annee_min)` qui renvoie une nouvelle table contenant uniquement les films sortis après ou pendant `annee_min`.

Exercice 11 - *Projection*

À partir de la table `films` de l'exercice précédent, écrivez une compréhension de liste pour obtenir uniquement la liste des titres de tous les films.

Exercice 12 - *Tri de table*

On souhaite trier la table `eleves` (qui contient les clés "nom" et "age") par ordre décroissant d'âge. Complétez l'instruction suivante :

```
1 table_triee = sorted(eleves, key=lambda x: ..., reverse=...)
2
```

Exercice 13 - *Analyse de fusion*

Soient deux tables :

- Eleves : [{"id": 1, "nom": "Alice"}, {"id": 2, "nom": "Bob"}]
- Notes : [{"id_eleve": 1, "note": 15}, {"id_eleve": 3, "note": 12}]

On effectue une jointure (fusion) sur les champs `id` et `id_eleve`.

Quelle sera la taille de la table résultante (nombre de lignes) si on ne garde que les correspondances exactes (Inner Join) ? Justifiez.

Exercice 14 - *Recherche de cohérence*

Vous avez importé une table CSV, mais certaines lignes sont incomplètes (valeurs vides). Écrivez un code qui parcourt une table et affiche le numéro de la ligne (l'index) si la valeur associée à la clé "email" est une chaîne vide "".

Exercice 15 - *Synthèse*

On dispose d'une liste de tuples représentant un panier de courses :

`panier = [("Pomme", 3), ("Banane", 2)]` (Article, Quantité).

On dispose d'un dictionnaire des prix :

`prix = {"Pomme": 0.5, "Banane": 1.2, "Poire": 0.8}`.

Écrivez une fonction qui calcule le prix total du panier.

Exercice 16 - *Analyse de données Streaming (Problème bilan)*

Vous travaillez sur un fichier `hits.csv` qui recense les écoutes de morceaux de musique sur une plateforme de streaming. Le fichier est formaté avec des points-virgules.

Aperçu du fichier `hits.csv` :

titre	artiste	ecoutes
Shape of You	Ed Sheeran	2000
Blinding Lights	The Weeknd	1500
Perfect	Ed Sheeran	1200
...

Objectif : Déterminer quel est l'artiste le plus écouté au total (cumul de tous ses titres).

Question 1 : Importation

Écrivez une fonction `charger_hits(fichier)` qui lit le fichier CSV et renvoie une **liste de dictionnaires**.

Attention : les "ecoutes" sont des chaînes de caractères dans le CSV, il faut les convertir en entiers (int) lors du chargement.

Question 2 : Agrégation (Le cœur du problème)

Écrivez une fonction `cumul_par_artiste(table)` qui prend la table en paramètre et renvoie un **dictionnaire** où :

- Les clés sont les noms des artistes.
- Les valeurs sont le nombre total d'écoutes cumulées.

Indice : Parcourez la table. Si l'artiste n'est pas dans le dictionnaire, créez l'entrée. S'il y est, ajoutez les écoutes à la valeur existante.

Question 3 : Recherche du maximum

À partir du dictionnaire obtenu à la question 2, écrivez une fonction (ou un script) qui affiche le nom de l'artiste ayant le plus grand nombre d'écoutes.

Espace réponse (brouillon de code) :