

TP 4 : les Collections en Python

◆ Exercice 1 : Structurer des Données avec namedtuple

Un club sportif souhaite enregistrer les informations de ses adhérents (Nom, Âge, Sport pratiqué).

1. Créez une structure **Adherent** avec namedtuple.
 2. Instanciez **3 adhérents** avec des données réelles.
 3. Affichez les informations de chaque adhérent.
-

◆ Exercice 2 : Manipuler une deque

Un bus ne peut embarquer que **3 passagers** à la fois.

1. Créez une deque pour représenter les passagers.
 2. Ajoutez **3 noms** au départ.
 3. Un passager descend (popleft()) et un autre monte (append()).
 4. Affichez la liste des passagers après le changement.
-

◆ Exercice 3 : Compter les occurrences avec Counter

Un magasin analyse les produits les plus achetés par ses clients.

1. Créez une liste avec **10 produits** (certains en double).
 2. Utilisez Counter pour afficher la fréquence de chaque produit.
 3. Affichez les **2 articles les plus achetés** (most_common()).
-

◆ Exercice 4 : Gérer des configurations avec ChainMap

Une application utilise une **configuration par défaut** mais l'utilisateur peut la personnaliser.

1. Définissez un **dictionnaire** config_default avec theme: clair, langue: fr.
 2. Définissez un **dictionnaire** config_utilisateur où l'utilisateur choisit theme: sombre.
 3. Fusionnez ces configurations avec ChainMap.
 4. Affichez la configuration finale (theme et langue).
-

◆ Exercice 5 : Trier avec OrderedDict

Un restaurant veut enregistrer les commandes des clients dans l'ordre.

1. Créez un OrderedDict pour stocker **3 commandes** (nom du client → plat commandé).
2. Affichez les commandes dans l'ordre d'ajout.

3. Ajoutez une nouvelle commande et affichez le dictionnaire mis à jour.
-

♦ **Exercice 6 : Gérer des valeurs par défaut avec defaultdict**

Un lycée stocke les matières suivies par chaque élève.

1. Utilisez un defaultdict où **chaque élève a une liste de matières**.
2. Ajoutez des matières pour **2 élèves**.
3. Affichez les matières de chaque élève.