

Interrogation – Python (avec ordinateur) : listes, tuples, dictionnaires (Pratique)

Instructions : Vous pouvez exécuter et tester votre code. Appelez moi après chaque question/exercice pour que je vérifies.

Exercice 1 – Catalogue simplifié

On veut représenter un petit catalogue de produits. Chaque produit est représenté par un **tuple** :

```
(nom, prix)
```

Et l'ensemble du catalogue est stocké dans une **liste**.

On vous donne :

```
catalogue = [  
    ("stylo", 2.5),  
    ("cahier", 3.2),  
    ("gomme", 1.0),  
    ("crayon", 1.5)  
]
```

1. Afficher le prix du produit "cahier" en utilisant la liste et le tuple.
2. Ajouter un nouveau produit "feutre" coûtant 2.0 au catalogue.
3. Recopier et compléter la fonction recherche(nom) qui prend le nom d'un produit et retourne son prix, ou "Non trouvé" si le produit n'est pas dans le catalogue.

```
def recherche(nom):  
    for n, prix in ...:  
        if ...:  
            return prix  
    return ...
```

4. Écrire une fonction plus_cher(catalogue) qui retourne le **tuple du produit le plus cher**.

Exercice 2 – Informations sur des élèves

On veut stocker les informations de plusieurs élèves dans une liste de dictionnaires. Chaque dictionnaire contient :

```
{
    "nom": ... ,
    "age": ... ,
    "notes": [...]  # une liste de notes
}
```

On vous donne :

```
classe = [
    {"nom": "Alice", "age": 16, "notes": [14, 17, 15]},
    {"nom": "Benoit", "age": 15, "notes": [10, 11, 13]},
    {"nom": "Chloe", "age": 16, "notes": [18, 19, 17]}
]
```

1. Grace à la notation avec les crochets, afficher la **première note** de Benoit.
2. Ajouter la note 16 à la liste de notes de Alice.
3. Créer une fonction `moyenne(notes)` qui retourne la moyenne d'une liste de notes (on ne va pas chercher la liste dans le dictionnaire, simplement calculer la moyenne d'une liste).
4. Afficher le nom de l'élève ayant la **meilleure moyenne**.

Exercice 3 – Base de données miniaturisée

On veut représenter une mini base de données de films par catégorie. Chaque catégorie est une **clé du dictionnaire**, et la valeur associée est une **liste de tuples** :

```
films = {
    "Action": [("Inception", 2010), ("Mad Max", 2015)],
    "Animation": [("Wall-E", 2008), ("Soul", 2020)],
    "Science-fiction": [("Interstellar", 2014), ("Dune", 2021)]
}
```

1. Afficher l'année de sortie du film "Wall-E".
2. Ajouter dans "Action" le film "Gladiator" (2000).
3. Recopier et compléter la fonction `tous_les_films(films)` qui retourne **une liste contenant tous les titres**, toutes catégories confondues.

```
def tous_les_films(films):  
    res = []  
    for v in films....:  
        for titre,date in ...:  
            res...  
    return res
```