

TP5 : Les collection (Les dictionnaires)

Les dictionnaires

- Un dictionnaire est **une collection de paires clé-valeur**, où chaque **clé est unique**.
- Les dictionnaires **sont mutables**, permettant d'ajouter, de supprimer ou de modifier des éléments.
- Les **clés** doivent être de **types immuables** (comme des chaînes, des nombres ou des tuples).
- Les **valeurs peuvent** être de **n'importe quel type**, offrant une grande flexibilité dans la structuration des données.

Exemple :

```
# Création d'un dictionnaire vide
dictionnaire_vide = {}

# Création d'un dictionnaire avec différents types d'objets
dictionnaire_varie = {
    "nom": "Hamza",      # Chaîne de caractères
    "age": 20,           # nombre
    "est_etudiant": True, # Booléen
    "notes": [85, 90, 78], # Une Liste
    "adresse": {         # Dictionnaire imbriqué
        "ville": « CasaBlanca",
        "code_postal": 12400 }
}
```

Opération sur les dictionnaires

```
# Création d'un dictionnaire
mon_dictionnaire = {
    "nom": "Youssef",
    "age": 30,
    "ville": "Rabat" }

# 1. Accéder à une valeur par sa clé
nom = mon_dictionnaire["nom"]

# 2. Ajouter une nouvelle paire clé-valeur
mon_dictionnaire["profession"] = "Ingénieur"

# 3. Modifier une valeur existante
mon_dictionnaire["age"] = 31

# 4. Supprimer un élément par clé avec del
del mon_dictionnaire["profession"]

# 5. Vérification de l'existence d'une clé
est_dans_dictionnaire = "nom" in mon_dictionnaire

# 6. Longueur du dictionnaire
longueur = len(mon_dictionnaire)
print("Longueur du dictionnaire :", longueur)

# 7. Parcourir les clés et les valeurs
for cle, valeur in mon_dictionnaire.items():
    print(f"{cle} : {valeur}")
```

Exercice 1 :

Ecrire une fonction Occurrences qui prend en paramètre un mot et renvoie un dictionnaire qui pour chaque caractère du mot associe le nombre d'occurrence de chaque lettre. On précisera que les autres caractères ., ;! ? sont exclus :

Exemple : pour la chaîne = "Google", le programme renvoie le dictionnaire: `d = {'g': 2, 'o': 2, 'l': 1, 'e': 1}`

Exercice 2 :

Ecrire une fonction Python qui reçoit une chaîne de caractère et retourne un dictionnaire dont les clés sont les caractères de la chaîne saisie et les valeurs sont la liste des positions des caractères dans la chaîne.

Exemple : pour la chaîne = "Google", le programme renvoie le dictionnaire: `d={'g':[1,4], 'o':[2,3], 'l':[5], 'e':[6]}`



Exercice 3 : Regroupement d'anagrammes

Écrire une fonction qui prend en paramètre une liste de mots et retourne un dictionnaire où :

- Les **clés** sont des **tuples** contenant les caractères d'un mot triés par ordre alphabétique.
- Les **valeurs** sont des **listes de mots originaux** qui sont des **anagrammes** entre eux (i.e., ils contiennent exactement les mêmes lettres, dans un ordre différent).

Règles à respecter :

- Ignorer la casse (traiter tous les mots en minuscules).
- Ignorer la ponctuation .,:!/? dans les mots.
- Un mot ne doit pas apparaître dans plusieurs groupes.

Exemple : pour la liste : `mots = ["écoute", "couteu", "google", "gogole", "apple"]`, le programme renvoie le dictionnaire :

```
d = {
    ('c', 'e', 'o', 't', 'u', 'é') : ['écoute', 'couteu'],
    ('e', 'g', 'g', 'l', 'o', 'o') : ['google', 'gogole'],
    ('a', 'e', 'l', 'p', 'p') : ['apple']
}
```

Rappel : Un **tuple** est une liste qui ne peut plus être modifiée.

```
# Création d'un tuple vide
mon_tuple = ()

# Création d'un tuple avec des valeurs
mon_tuple = (1, "ok", "olivier")

# Création d'un tuple avec une seule valeur
mon_tuple = ("ok",)

# Afficher une valeur d'un tuple
mon_tuple[0] # Résultat : 1

# Création d'un tuple par la fonction tuple()
mon_tuple = tuple(("API -2", "ENSAM", 2025)) # note the double round-brackets
```

Exercice 4 : Gestion des notes d'un étudiant

Écrire un programme Python qui permet de gérer les notes d'un étudiant en suivant les étapes ci-dessous.

1 Saisie des données

- Demander à l'utilisateur de saisir les notes de 4 modules :
 - Mathématiques
 - Physique
 - Français
 - Informatique
- Stocker les notes dans un dictionnaire où :
 - **Clé** : Nom du module (chaîne de caractères).
 - **Valeur** : Note (nombre réel).

2 Calculs et affichages

Le programme doit réaliser les actions suivantes :

- Calculer et afficher la moyenne générale de l'étudiant.
- Identifier et afficher :
 - La meilleure note obtenue.
 - Le module correspondant à cette note.
- Afficher la liste des modules réussis (note ≥ 10).
- Afficher la liste des modules à rattraper (note < 10) avec : Le nombre de points manquants pour atteindre la moyenne (10).

Exemple : pour le dictionnaire

Notes = {

Mathématiques : 14

Physique : 8

Français : 12

Informatique : 9

},

le programme renvoie le résultat suivant :

- *Moyenne générale : 10.75*
- *Meilleure note : 14 (Mathématiques)*
- *Modules réussis : ['Mathématiques', 'Français']*
- *Modules à rattraper :*
 - *Physique (2.0 points manquants)*
 - *Informatique (1.0 points manquants)*

Exercice 5 : Chiffrement des données personnelles de la carte nationale (Chiffrement de César)

Vous êtes chargé de développer un programme pour sécuriser des données personnelles de la carte nationale en utilisant un chiffrement simple appelé **chiffrement de César**. Ce chiffrement consiste à décaler chaque caractère de l'alphabet et chaque chiffre d'un nombre fixe de positions dans l'ordre. Dans cet exercice, nous utiliserons un décalage de 3 positions.

Objectifs :

1. **Récupération des informations** : Implémentez une fonction permettant de récupérer les informations personnelles d'une carte nationale dans un dictionnaire (Nom, Prénom, Numéro de la CINE).
2. **Chiffrement des données personnelles** : Implémentez une fonction qui applique un décalage de 3 sur chaque caractère d'une donnée personnelle (lettres et chiffres uniquement).
3. **Déchiffrement des données personnelles** : Implémentez une fonction qui inverse le décalage de 3 pour retrouver les données originales.

Explications :

- Chiffrement des lettres :
 - Les lettres de l'alphabet sont décalées de 3 positions. Par exemple, "A" devient "D", "B" devient "E", "C" devient "F", etc.
 - Le décalage est cyclique : après "Z", on revient à "A". Donc, "Y" devient "B" et "Z" devient "C".
- Chiffrement des chiffres :
 - Les chiffres sont également décalés de 3 positions. Par exemple, "1" devient "4", "2" devient "5", "3" devient "6", etc.
 - Le décalage des chiffres est aussi cyclique : après "9", on revient à "0". Donc, "8" devient "1", "9" devient "2", etc.

Exemple de chiffrement :

- " Mohamed Amine " devient " Prkdphg Dplqh "
- " Ali " devient " Dol "
- "AB123456 " devient " DE456789 "