



### Ce qu'il faut savoir

- un dictionnaire permet de “stocker” des données
- chaque élément d'un dictionnaire est composé de 2 parties, on parle de paires "clé/valeur" (exemple : `mon_dico = {"nom": "Durand", "prenom": "Christophe", "date de naissance": "29/02/1981"}`)
- pour afficher une “valeur” particulière, on utilise la notation `mon_dico[nom_de_la_clé]`
- il est possible de parcourir l'ensemble des clés d'un dictionnaire à l'aide d'une boucle for en utilisant `keys`
- il est possible de parcourir l'ensemble des valeurs d'un dictionnaire à l'aide d'une boucle for en utilisant `values`
- il est possible de parcourir l'ensemble des clés et des valeurs (en même temps) d'un dictionnaire à l'aide d'une boucle for en utilisant `items`

### Ce qu'il faut savoir faire

- A. construire un dictionnaire
- B. utiliser la notation `mon_dico[nom_de_la_clé]` afin d'utiliser une valeur particulière
- C. parcourir l'ensemble des valeurs
- D. parcourir l'ensemble des clés
- E. parcourir l'ensemble des clés et des valeurs (en même temps)

#### A.1 Compter les occurrences d'éléments dans une liste \*

Écrire une fonction qui prend une liste d'éléments (nombres ou chaînes de caractères) et renvoie un dictionnaire où chaque clé est un élément de la liste et la valeur est le nombre d'occurrences de cet élément.

# Entrée : ['pomme', 'banane', 'pomme', 'orange', 'banane', 'banane']

# Sortie : {'pomme': 2, 'banane': 3, 'orange': 1}

## A.2 Inverser un dictionnaire \*\*

Écrire une fonction qui prend un dictionnaire où les clés sont des chaînes de caractères et les valeurs sont des entiers. La fonction doit renvoyer un dictionnaire où les clés et les valeurs sont inversées.

**Exemple :**

# Entrée : {'a': 1, 'b': 2, 'c': 3}

# Sortie : {1: 'a', 2: 'b', 3: 'c'}

## A.3 Supprimer les doublons d'une liste et les compter \*\*

Écrire une fonction qui prend une liste de valeurs (par exemple, des chaînes de caractères ou des entiers) et renvoie un dictionnaire avec chaque élément unique comme clé et le nombre d'occurrences comme valeur.

**Exemple :**

# Entrée : ['apple', 'banana', 'apple', 'orange', 'banana']

# Sortie : {'apple': 2, 'banana': 2, 'orange': 1}

## B.1 Vérifier si une clé existe dans un dictionnaire \*

Écrire une fonction qui prend un dictionnaire et une clé, et qui renvoie **True** si la clé existe dans le dictionnaire et **False** sinon.

## **B.2 Trouver la clé avec la valeur maximale \*\***

Écrire une fonction qui prend un dictionnaire et renvoie la clé associée à la valeur maximale.

**Entrée** = {'a': 1, 'b': 5, 'c': 3}

## **C.1 Extraire les clés ou les valeurs d'un dictionnaire \***

Écrire une fonction qui prend un dictionnaire et retourne une liste contenant toutes les clés, ou une autre fonction qui retourne toutes les valeurs.

**Exemple :**

**# Entrée** : {'a': 1, 'b': 2, 'c': 3}

**# Sortie** : {1: 'a', 2: 'b', 3: 'c'}

## **C.2 Trier un dictionnaire par valeur \*\***

Écrire une fonction qui prend un dictionnaire et renvoie une liste de tuples, triée par ordre croissant des valeurs du dictionnaire.

**# Entrée** : {'a': 3, 'b': 1, 'c': 2}

**# Sortie** : [('b', 1), ('c', 2), ('a', 3)]

### D.1 Supprimer un élément d'un dictionnaire \*\*

*Écrire une fonction qui prend un dictionnaire et une clé en entrée, et qui supprime l'élément correspondant à cette clé dans le dictionnaire. Si la clé n'existe pas, la fonction doit afficher un message d'erreur.*

**# Entrée :** {'a': 1, 'b': 2, 'c': 3}, 'b' clé à supprimer

**# Sortie :** {'a': 1, 'c': 3}

**# Entrée :** {'a': 1, 'b': 2}, 'd'

**# Sortie :** "Clé 'd' non trouvée."

### E.1 Fusionner deux dictionnaires \*\*

*Écrire une fonction qui fusionne deux dictionnaires. Si une même clé existe dans les deux dictionnaires, la valeur du second dictionnaire doit remplacer celle du premier.*

**Exemple :**

**# Dictionnaires :** {'a': 1, 'b': 2}, {'b': 3, 'c': 4}

**# Sortie :** {'a': 1, 'b': 3, 'c': 4}

### E.2 Vérification des étudiants ayant réussi \*\*

*Imaginons que tu as un dictionnaire représentant des étudiants et leurs notes. Écris une fonction qui prend un seuil de réussite et renvoie une liste des étudiants qui ont réussi (dont la note est supérieure ou égale au seuil).*

**Exemple :**

**# Entrée :** {'Alice': 15, 'Bob': 8, 'Charlie': 18}, seuil = 10

**# Sortie :** ['Alice', 'Charlie']