Les dictionnaires

Objectifs :

- Comprendre ce qu'est un **dictionnaire**.
- Créer, modifier et parcourir un dictionnaire.
- Faire la différence entre une **liste** et un **dictionnaire**.
- Savoir utiliser les méthodes de base : `.keys()`, `.values()`, `.items()`, `.get()`.

Table des matières

I. Découverte des dictionnaires en Python....1
1. Introduction simple....1
a) Qu'est-ce qu'un dictionnaire ?....1
b) Syntaxe de base....1
2. Création et accès aux données....2
a) Alternative sécurisée :....2
3. Ajout, modification, suppression....2
a) Ajouter une nouvelle clé :....2
b) Modifier une valeur :....2
c) Supprimer une clé :....2
4. Parcourir un dictionnaire....2
a) Par les clés avec une boucle for :....2
b) Par clés et valeurs avec une boucle for et la méthode .items():....2
5. Méthodes utiles associées aux dictionnaires....2
6. Comparaison rapide : liste vs dictionnaire....2
7. Exercices pour s'entraîner....3
a) Exercice 1 — Création de dictionnaire....3
b) Exercice 2 — Accès sécurisé....3
c) Exercice 3 — Mise à jour....3
d) Exercice 4 — Parcours....3

I. Découverte des dictionnaires en Python

1. Introduction simple

a) Qu'est-ce qu'un dictionnaire ?

Un **dictionnaire** est une **structure de données** qui associe des **clés** à des **valeurs**, comme un **annuaire** téléphonique :

`"Jean" → "06 12 34 56 78"`

b) Syntaxe de base

```
personne = {
    "nom": "Alice",
    "âge": 20,
    "ville": "Lyon"
}
```

2. Création et accès aux données

```
print(personne["nom"])    # Alice
print(personne["âge"])    # 20
```

⚠️ Si la clé n'existe pas : erreur `KeyError`.

a) Alternative sécurisée :

```
print(personne.get("email", "Non renseigné"))  # Non renseigné
```

Si la clé n'existe pas la méthode get() retournera "Non renseigné".

3. Ajout, modification, suppression

a) Ajouter une nouvelle clé :

```
personne["email"] = "alice@example.com"
```

b) Modifier une valeur :

```
personne["âge"] = 21
```

c) Supprimer une clé :

```
del personne["ville"]
```

4. Parcourir un dictionnaire

a) Par les clés avec une boucle for :

```
for cle in personne:
    print(cle, ":", personne[cle])
```

b) Par clés et valeurs avec une boucle for et la méthode .items():

```
for cle, valeur in personne.items():
    print(f"{cle} → {valeur}")
```

5. Méthodes utiles associées aux dictionnaires

Méthode	Description
`.keys()`	Renvoie toutes les clés
`.values()`	Renvoie toutes les valeurs
`.items()`	Renvoie les paires clé-valeur
`.get(cle)`	Renvoie la valeur associée ou `None`
`cle in dico`	Vérifie l'existence d'une clé

6. Comparaison rapide : liste vs dictionnaire

Liste	Dictionnaire
Accès par **indice**	Accès par **clé**
Ordonnée par défaut	Non ordonné (jusqu'à Python 3.6+)
Représente un **groupe d'objets**	Représente une **relation**

7. Exercices pour s'entraîner

a) Exercice 1 — Création de dictionnaire

Crée un dictionnaire nommé `animal` contenant :

- `espèce` : "chat"
- `âge` : 3
- `nom` : "Miaou"

b) Exercice 2 — Accès sécurisé

Affiche le surnom (`"surnom"`) du chat en utilisant `.get()` avec une valeur par défaut `"Inconnu"`.

c) Exercice 3 — Mise à jour

Ajoute au dictionnaire un nouvel élément `"couleur"` de ton choix.

d) Exercice 4 — Parcours

Affiche toutes les informations de ton dictionnaire sous la forme :

```
espèce → chat
âge → 3
nom → Miaou
couleur → gris
```