

# Chapitre 6 : Les Dictionnaires en Python

CHAFAOUI Mohamed-Reda

Centre de Formation 2 Mille (CF2M)

October 2, 2025



# Plan

- 1 Qu'est-ce qu'un dictionnaire ?
- 2 Dictionnaires vs Listes
- 3 Opérations de base
- 4 Méthodes utiles
- 5 Parcourir un dictionnaire
- 6 Structures plus complexes
- 7 Lien avec JSON



# Définition

- Un dictionnaire est une **collection de paires clé → valeur**.
- Chaque **clé** est unique et permet d'accéder rapidement à sa **valeur**.
- Contrairement à une liste, l'ordre des éléments n'est pas garanti.

## Exemple

```
etudiant = {"nom": "Alice", "age": 22, "note": 15.5}
```

# Comparaison avec les listes

- **Liste** : accès par un index numérique.
- **Dictionnaire** : accès par une clé choisie.

## Exemple Liste

```
notes = [15, 12, 18]  
print(notes[0]) # 15
```

## Exemple Dictionnaire

```
etudiant = {"nom": "Alice", "note": 15}  
print(etudiant["note"]) # 15
```

# Accéder, ajouter, modifier, supprimer

- Accéder : `print(etudiant["nom"])` → Alice
- Ajouter : `etudiant["ville"] = "Paris"`
- Modifier : `etudiant["age"] = 23`
- Supprimer : `del etudiant["note"]`

## Quelques méthodes principales

- `d.keys()` → liste des clés
- `d.values()` → liste des valeurs
- `d.items()` → couples (clé, valeur)
- `d.get("cle", default)` → évite une erreur
- `d.pop("cle")` → supprime et renvoie la valeur

### Exemple

```
print(etudiant.get("note", "pas de note"))
```



# Boucles sur un dictionnaire

- Parcourir uniquement les clés :

```
for cle in etudiant:  print(cle)
```

- Parcourir clés et valeurs :

```
for cle, valeur in etudiant.items():  
    print(cle, ":", valeur)
```

# Combiner dictionnaires et listes

- **Liste de dictionnaires** : utile pour gérer plusieurs objets.

## Exemple

```
etudiants = [  
    {"nom": "Alice", "note": 15},  
    {"nom": "Bob", "note": 12} ]  
print(etudiants[0]["nom"]) # Alice
```



# Dictionnaire de listes

- Un dictionnaire peut contenir des listes comme valeurs.

## Exemple

```
notes = {"math": [15, 12], "info": [18, 17]}  
print(notes["math"]) # [15, 12]
```

# Dictionnaires et JSON

- Les dictionnaires ressemblent beaucoup au format **JSON**.
- JSON est très utilisé pour stocker et échanger des données (API, fichiers).

## Exemple

```
import json  
print(json.dumps(etudiant, indent=2))
```



# Conclusion

- Les dictionnaires associent des clés à des valeurs.
- Plus flexibles que les listes pour organiser des données complexes.
- Utilisés partout en Python : JSON, APIs, bases de données...



# Questions ?

Merci pour votre attention !  
Avez-vous des questions ?

