

Cours : Les dictionnaires en Python

Qu'est-ce qu'un dictionnaire ?

Un **dictionnaire** est une structure de données qui associe **des clés à des valeurs**.

Il fonctionne comme un vrai dictionnaire :

Vous cherchez un mot (clé), vous obtenez sa définition (valeur).

```
mon_dict = {"nom": "Alice", "age": 25, "ville": "Paris"}
```

- Les **clés** doivent être **uniques**.
 - Les **valeurs** peuvent être de **n'importe quel type**.
 - Les dictionnaires sont **mutables** (on peut les modifier).
-

Créer un dictionnaire

```
d = {"a": 1, "b": 2}
```

Ou progressivement :

```
d = {}
d["a"] = 1
d["b"] = 2
```

Même avec des types variés :

```
d = {"nom": "Alice", "age": 25, "notes": [15, 14, 16]}
```

Accéder à une valeur

```
d = {"nom": "Bob", "age": 30}
print(d["nom"])      # Bob
print(d.get("age")) # 30
```

⚠ Si la clé n'existe pas :

- `d["ville"]` → ✗ erreur
- `d.get("ville")` → 🤖 renvoie None (ou une valeur par défaut)

```
d.get("ville", "Inconnue") # "Inconnue"
```

Modifier / Ajouter / Supprimer

Action	Syntaxe
Ajouter / Modifier	<code>d["clé"] = valeur</code>
Supprimer	<code>d.pop("clé")</code>
Supprimer (sécurisé)	<code>d.pop("clé", valeur_si_absente")</code>
Vider le dictionnaire	<code>d.clear()</code>

```
d["age"] = 31    # modification  
d["ville"] = "Paris" # ajout  
d.pop("age")      # suppression
```

Parcourir un dictionnaire

- ◆ Parcourir les clés

```
for key in d.keys():  
    print(key)
```

- ◆ Parcourir les valeurs

```
for value in d.values():  
    print(value)
```

- ◆ Parcourir les deux

```
for key, value in d.items():
    print(f"{key} -> {value}")
```

Méthodes utiles

Méthode	Description
d.keys()	Liste des clés
d.values()	Liste des valeurs
d.items()	Paires clé/valeur
d.update(dict2)	Fusion de dictionnaires
len(d)	Nombre de clés

```
d1 = {"a": 1}
d2 = {"b": 2}
d1.update(d2) # {"a": 1, "b": 2}
```

Dictionnaire imbriqué

```
classe = {
    "Alice": {"age": 15, "moyenne": 14},
    "Bob": {"age": 16, "moyenne": 12}
}

print(classe["Alice"]["moyenne"]) # 14
classe["Bob"]["moyenne"] += 1
```

Créer un dictionnaire depuis deux listes

```
fruits = ["pomme", "banane", "cerise"]
prix = [2.5, 1.8, 3.2]

d = dict(zip(fruits, prix))
```

```
print(d)
# {'pomme': 2.5, 'banane': 1.8, 'cerise': 3.2}
```

Compréhension de dictionnaire

Comme pour les listes, on peut créer un dictionnaire en une ligne :

```
d = {x: x**2 for x in range(5)}
print(d)
# {0: 0, 1: 1, 2: 4, 3: 9, 4: 16}
```

Résumé

✓ Structure clé → valeur ✓ Création, modification, suppression ✓ Parcours avec .keys(), .values(), .items() ✓ Dictionnaires imbriqués ✓ Génération automatique (zip, compréhension)

Exemple complet

```
etudiant = {
    "nom": "Alice",
    "age": 17,
    "notes": {"Maths": 15, "Français": 14, "Physique": 16}
}

# 1. Accéder à la note de Physique
print(etudiant["notes"]["Physique"])

# 2. Ajouter une note
etudiant["notes"]["Histoire"] = 13

# 3. Calculer la moyenne
moyenne = sum(etudiant["notes"].values()) / len(etudiant["notes"])
print(f"Moyenne = {moyenne}")
```