Les Tuples, Dictionnaires et Ensembles en Python

A.MAZZA
MPSI 2-3
CPGE Oujda

31 janvier 2025

Introduction

Pourquoi ces structures ?

- Python propose plusieurs structures de données puissantes.
- Nous allons explorer les tuples, les dictionnaires et les ensembles.
- Ces structures sont fondamentales pour optimiser le stockage et la manipulation des données.

Les Tuples

Définition : Un tuple est une structure de données immuable qui stocke plusieurs éléments.

Exemple

```
t = (1, 2, 3)
```

Principales caractéristiques :

- Accès par index : `t[0]`
- Déballage : `a, b, c = t`
- Concaténation : `t1 + t2`
- Recherche d'un élément : `valeur in t`

Les Tuples

Méthodes et fonctions utiles :

- `len(t)` : Longueur du tuple.
- `min(t)` : Élément minimum du tuple.
- `max(t)` : Élément maximum du tuple.
- `t.index(x)` : Trouve l'index de l'élément `x`.
- `t.count(x)` : Compte le nombre d'occurrences de `x` dans le tuple.

Les Dictionnaires

Définition : Un dictionnaire stocke des paires `clé → valeur`.

Exemple

```
d = {"nom": "Omar", "âge": 25}
```

Opérations principales :

- Accès aux valeurs : `d["nom"]`
- Modification : `d["âge"] = 26`
- Suppression d'une clé : `del d["nom"]`
- Parcours : `for key, value in d.items()`

Les Dictionnaires

Méthodes utiles :

- `len(d)` : Nombre de paires clé-valeur.
- `d.keys()` : Liste des clés du dictionnaire.
- `d.values()` : Liste des valeurs.
- `d.items()` : Liste des paires clé-valeur.
- `d.get(k)` : Renvoie la valeur associée à la clé `k`, ou `None` si la clé n'existe pas.
- `d.pop(k)` : Supprime et renvoie la valeur associée à la clé `k`.
- `d.update(d2)` : Fusionne un autre dictionnaire `d2` dans `d`.

Les Ensembles

Définition : Un ensemble est une collection non ordonnée et sans doublons.

Exemple

`s = {1, 2, 3, 3}` $\rightarrow$ `{1, 2, 3}`

Opérations principales :

- Ajout d'un élément : `s.add(4)`
- Suppression : `s.remove(2)`
- Vérification d'appartenance : `4 in s`
- Opérations ensemblistes : `A | B, A  B, A - B`

Les Ensembles

Méthodes utiles :

- `len(s)` : Nombre d'éléments dans l'ensemble.
- `s.add(x)` : Ajoute l'élément `x`.
- `s.remove(x)` : Supprime l'élément `x` (lève une exception si l'élément n'existe pas).
- `s.discard(x)` : Supprime l'élément `x` sans lever d'exception si l'élément n'existe pas.
- `s.pop()` : Retire et renvoie un élément arbitraire de l'ensemble.
- `s.union(t)` ou `A | B` : Union de deux ensembles.
- `s.intersection(t)` ou `A & B` : Intersection de deux ensembles.
- `s.difference(t)` ou `A - B` : Différence entre deux ensembles.
- `s.issubset(t)` : Vérifie si `s` est un sous-ensemble de `t`.
- `s.issuperset(t)` : Vérifie si `s` est un sur-ensemble de `t`.
- `s.symmetric_difference(t)` : Renvoie la différence symétrique entre deux ensembles.

Exercices

- **Tuples** :
  - Échanger deux éléments d'un tuple.
  - Trouver l'élément le plus grand dans un tuple.
  - Fusionner deux tuples.
  - Utiliser `count()` et `index()` pour analyser un tuple.
- **Dictionnaires** :
  - Inverser un dictionnaire.
  - Fusionner deux dictionnaires en un seul.
  - Trouver la clé associée à la plus grande valeur dans un dictionnaire.
  - Utiliser `get()`, `keys()` et `items()`.
- **Ensembles** :
  - Trouver la différence entre deux ensembles.
  - Vérifier si un ensemble est un sous-ensemble d'un autre.
  - Trouver l'intersection entre plusieurs ensembles.
  - Utiliser `add()`, `remove()` et

Exercice Supplémentaire

Problème : Gestion d'une liste de contacts

- Créez un dictionnaire où chaque clé est un nom de contact et la valeur est un tuple contenant le numéro de téléphone et l'adresse email du contact.
- Ajoutez au dictionnaire un nouveau contact avec un numéro de téléphone et une adresse email.
- Modifiez l'email d'un contact existant.
- Supprimez un contact du dictionnaire.
- Créez un ensemble contenant les noms des contacts.
- Vérifiez si un contact existe dans l'ensemble des contacts.
- Affichez tous les contacts sous forme de tableau (nom, numéro de téléphone, email).

Solution : Gestion d'une liste de contacts (1/2)

Code Python : Création et mise à jour du dictionnaire

```
# Création du dictionnaire des contacts
contacts = {
    "Ali": ("0612345678", "ali@example.com"),
    "Sara": ("0698765432", "sara@example.com"),
    "Othman": ("0678901234", "othman@example.com")
}
# Ajout d'un nouveau contact
contacts["Alae"] = ("0654321987", "alae@example.com")
# Modification de l'email d'un contact existant
contacts["Ali"] = (contacts["Ali"][0], "ali.new@example.com")
# Suppression d'un contact
del contacts["Othman"]
```

Solution : Gestion d'une liste de contacts (2/2)

Code Python : Opérations sur les ensembles et affichage

```python
# Création d'un ensemble contenant les noms des contacts
noms_contacts = set(contacts.keys())
# Vérification si un contact existe
nom_recherche = "Alae"
if nom_recherche in noms_contacts:
    print(f"{nom_recherche} est dans la liste des contacts.")
else:
    print(f"{nom_recherche} n'est pas dans la liste des contacts.")
# Affichage des contacts sous forme de tableau
print("\nListe des contacts :")
print(f"{'Nom':<10} {'Téléphone':<12} {'Email'}")
print("-" * 40)
for nom, (telephone, email) in contacts.items():
    print(f"{nom:<10} {telephone:<12} {email}")
```

Conclusion

Résumé :

- **Tuples** : immuables et indexés.
- **Dictionnaires** : clés associées à des valeurs.
- **Ensembles** : collections uniques et non ordonnées.

Questions ?

N'hésitez pas à poser vos questions !