

Python: Les dictionnaires

Les dictionnaires sont une structure de données qui permet de stocker des paires clé-valeur, où chaque clé est unique et associée à une valeur. Si vous êtes débutant en programmation Python et fan de Donjons et Dragons (D&D), cet article est pour vous ! Nous allons explorer les dictionnaires à travers des exemples concrets inspirés de l'univers de D&D.

 Exploration

Qu'est-ce qu'un Dictionnaire en Python ?

Un dictionnaire en Python est une **collection non ordonnée de paires clé-valeur**. Chaque **clé** dans un dictionnaire doit être **unique** et immuable (on ne peut pas changer le nom d'une clé), tandis que **les valeurs peuvent être de n'importe quel type** (ce sont des variables).

Création d'un Dictionnaire

Pour créer un dictionnaire, vous pouvez utiliser des accolades `{}` ou la fonction `dict()`.

```
1  # Utilisation des accolades
2  personnage = {
3      "nom": "Gandalf",
4      "classe": "Magicien",
5      "niveau": 5
6  }
7
8  # Utilisation de la fonction dict()
9  personnage = dict(nom="Gandalf", classe="Magicien", niveau=5)
```

Accès aux Valeurs

La syntaxe pour accéder aux valeurs d'un objet `dict` est très proche de celle utilisée pour accéder aux éléments d'une liste, sauf qu'au lieu d'utiliser un index numérique (0,1...) on utilise une chaîne de caractères.

Vous pouvez **accéder** aux valeurs d'un dictionnaire en utilisant les clés correspondantes.

```
1  nom = personnage["nom"]
2  classe = personnage["classe"]
3  niveau = personnage["niveau"]
```

```
4  
5 | print(f"Nom: {nom}, Classe: {classe}, Niveau: {niveau}")
```

Comapraison entre une liste et un dict

On peut s'imaginer une liste de cette manière:

Index	Valeur
0	"Humain"
1	"Nain"
2	"Elf"

On peut s'imaginer un dict de cette manière:

Index	Valeur
"nom"	"Gandalf"
"classe"	"Magicien"
"niveau"	5

Ajout et Modification de Paires Clé-Valeur

Vous pouvez **ajouter** de nouvelles paires clé-valeur ou **modifier** les valeurs de paires existantes simplement en utilisant la syntaxe de l'index.

```
1 | # Ajout d'une nouvelle paire clé-valeur  
2 | personnage["race"] = "Maia"  
3 |  
4 | # Modification d'une paire existante  
5 | personnage["niveau"] = 6
```

Suppression de Paires Clé-Valeur

Vous pouvez supprimer des paires clé-valeur avec la méthode `del` ou la méthode `pop()`.

```
1 | # Utilisation de del  
2 | del personnage["race"]  
3 |  
4 | # Utilisation de pop()  
5 | niveau = personnage.pop("niveau")
```

Exemple Simple de Dictionnaire en Python avec Donjons et Dragons

Pour illustrer l'utilisation des dictionnaires en Python avec un exemple simple de Donjons et Dragons (D&D), nous allons créer un dictionnaire représentant un personnage de jeu avec des attributs de base.

Définition du Personnage

Imaginons un personnage nommé "Aragorn" avec les attributs suivants :

- Nom
- Classe
- Race
- Niveau
- Points de Vie

Nous allons créer un dictionnaire pour représenter ce personnage.

```
1  # Création du dictionnaire pour le personnage
2  personnage = {
3      "nom": "Aragorn",
4      "classe": "Rôdeur",
5      "race": "Humain",
6      "niveau": 5,
7      "points_de_vie": 68
8  }
9
10 # Affichage du dictionnaire
11 print(personnage)
```

Explication du Code

1. Création du Dictionnaire:

- Nous utilisons des accolades `{}` pour définir le dictionnaire.
- Chaque paire clé-valeur est **séparée par une virgule**.
- Les clés sont des chaînes de caractères (par exemple, `"nom"`) et les valeurs peuvent être de n'importe quel type (chaînes de caractères, entiers, etc.).

2. Affichage du Dictionnaire:

- La fonction `print(personnage)` affiche le contenu du dictionnaire.

Accéder aux Valeurs du Dictionnaire

Pour accéder aux valeurs associées aux clés, vous pouvez utiliser la syntaxe de l'indexation.

```
1  # Accéder et afficher les valeurs des attributs du personnage
2  nom = personnage["nom"]
3  classe = personnage["classe"]
```

```

4 | race = personnage["race"]
5 | niveau = personnage["niveau"]
6 | points_de_vie = personnage["points_de_vie"]
7 |
8 | print(f"Nom: {nom}")
9 | print(f"Classe: {classe}")
10 | print(f"Race: {race}")
11 | print(f"Niveau: {niveau}")
12 | print(f"Points de Vie: {points_de_vie}")

```

Vous pouvez aussi utiliser cette syntaxe pour ne pas devoir créer de variables intermédiaires:

```

1 | # Accéder et afficher les valeurs des attributs du personnage
2 | print(f"Nom: {personnage['nom']}")
3 | print(f"Classe: {personnage['classe']}")
4 | print(f"Race: {personnage['race']}")
5 | print(f"Niveau: {personnage['niveau']}")
6 | print(f"Points de Vie: {personnage['points_de_vie']}")

```

Ou, encore, sans utiliser les f-strings:

```

1 | # Accéder et afficher les valeurs des attributs du personnage
2 | print("Nom: " + personnage['nom'])
3 | print("Classe: " + personnage['classe'])
4 | print("Race: " + personnage['race'])
5 | print("Niveau: " + str(personnage['niveau']))
6 | print("Points de Vie: " + str(personnage['points_de_vie']))

```

Ajouter ou Modifier des Valeurs

Vous pouvez ajouter de nouveaux attributs ou modifier les attributs existants du personnage.

```

1 | # Ajouter un nouvel attribut
2 | personnage["arme"] = "Épée"
3 |
4 | # Modifier un attribut existant
5 | personnage["points_de_vie"] = 75
6 |
7 | # Afficher le dictionnaire mis à jour
8 | print(personnage)

```

Supprimer des Valeurs

Pour supprimer un attribut du personnage, utilisez le mot-clé `del`.

```

1 | # Supprimer l'attribut "arme"
2 | del personnage["arme"]
3 |
4 | # Afficher le dictionnaire mis à jour
5 | print(personnage)

```

Résumé

Voici le code complet avec des commentaires expliquant chaque étape :

```
1  # Création du dictionnaire pour le personnage
2  personnage = {
3      "nom": "Aragorn",
4      "classe": "Rôdeur",
5      "race": "Humain",
6      "niveau": 5,
7      "points_de_vie": 68
8  }
9
10 # Affichage du dictionnaire
11 print(personnage)
12
13 # Accéder et afficher les valeurs des attributs du personnage
14 print("Nom: " + personnage['nom'])
15 print("Classe: " + personnage['classe'])
16 print("Race: " + personnage['race'])
17 print("Niveau: " + str(personnage['niveau']))
18 print("Points de Vie: " + str(personnage['points_de_vie']))
19
20 # Ajouter un nouvel attribut
21 personnage["arme"] = "Épée"
22
23 # Modifier un attribut existant
24 personnage["points_de_vie"] = 75
25
26 # Afficher le dictionnaire mis à jour
27 print(personnage)
28
29 # Supprimer l'attribut "arme"
30 del personnage["arme"]
31
32 # Afficher le dictionnaire mis à jour
33 print(personnage)
```

Conclusion

Les dictionnaires sont une structure de données extrêmement utile et polyvalente en Python. Ils vous permettent de stocker et de manipuler des données de manière efficace et intuitive. En utilisant des exemples issus du monde de Donjons et Dragons, nous avons exploré comment créer, accéder, modifier et supprimer des éléments dans un dictionnaire.

Que vous soyez en train de gérer des personnages, des monstres, ou tout autre type de données, les dictionnaires vous offrent une solution flexible pour organiser et manipuler vos informations. Commencez à les utiliser dans vos projets Python et découvrez toute leur puissance par vous-même !

Exercices

Exercice 1 : Créer un Dictionnaire

Objectif : Créer un dictionnaire simple représentant un livre avec les clés suivantes : titre, auteur, année.

Instruction :

Créez un dictionnaire nommé `livre` avec les clés "titre", "auteur", et "année", et assignez-lui les valeurs de votre choix.

```
1 | # Votre code ici
2 | livre = {
3 |     "titre": "Le Seigneur des Anneaux",
4 |     "auteur": "J.R.R. Tolkien",
5 |     "annee": 1954
6 | }
7 |
8 | print(livre)
```

Exercice 2 : Accéder aux Valeurs

Objectif : Accéder aux valeurs d'un dictionnaire et les afficher.

Instruction :

Utilisez le dictionnaire `livre` de l'exercice précédent. Accédez et affichez les valeurs des clés "titre" et "auteur".

```
1 | # Votre code ici
2 | titre = livre["titre"]
3 | auteur = livre["auteur"]
4 |
5 | print(f"Titre: {titre}")
6 | print(f"Auteur: {auteur}")
```

Exercice 3 : Ajouter une Nouvelle Clé-Valeur

Objectif : Ajouter une nouvelle paire clé-valeur à un dictionnaire existant.

Instruction :

Ajoutez une nouvelle clé "genre" au dictionnaire `livre` avec la valeur "Fantasy".

```
1 | # Votre code ici
2 | livre["genre"] = "Fantasy"
3 |
4 | print(livre)
```

Exercice 4 : Modifier une Valeur

Objectif : Modifier la valeur associée à une clé dans un dictionnaire.

Instruction :

Modifiez la valeur de la clé "année" dans le dictionnaire `livre` pour qu'elle soit 1955.

```
1 | # Votre code ici
2 | livre["annee"] = 1955
3 |
4 | print(livre)
```

Exercice 5 : Supprimer une Clé-Valeur

Objectif : Supprimer une paire clé-valeur d'un dictionnaire.

Instruction :

Supprimez la clé "genre" du dictionnaire `livre`.

```
1 | # Votre code ici
2 | del livre["genre"]
3 |
4 | print(livre)
```