

Les dictionnaires

UNE ALTERNATIVE AUX LISTES

Les dictionnaires

Les dictionnaires sont une façon d'organiser les données qui associent une **clé** à une **valeur**.

La syntaxe du langage Python fait en sorte qu'on peut les voir comme une liste dans laquelle on désigne la case à manipuler par une **chaîne de caractères**, au lieu d'un **indice**.

Dictionnaire

- Associe des clés à des valeurs
- N'est pas conçu pour être parcouru séquentiellement
- L'accès aux données se fait à l'aide du nom de la clé (**chaîne de caractères**)

Liste

- Associe un indice à une valeur
- Est conçu pour être parcouru séquentiellement
- L'accès aux données se fait à l'aide d'un indice (**int**)

Créer un dictionnaire

Créer un dictionnaire est similaire à la création d'une liste. Au lieu d'utiliser les **crochets** `[ ]`, il suffit d'utiliser les **accolades** `{ }`.

```
# Créé un dictionnaire vide
dict_vide = {}
```

```
# dictionnaire qui associe le nom d'animaux à leur longévité
dict_animal_longevite = {
    "pékan": 10,
    "castor": 12,
    "rat": 2
}
```

```
# Créé un dictionnaire qui associe chaque voyelle au nombre de
# fois qu'elle a été rencontrée dans un mot
dict_compte_voyelles = {
    "a":0,
    "e":0,
    "i":0,
    "o":0,
    "u":0,
    "y":0
}
```

Manipulations d'un dictionnaire

L'accès à un élément d'un dictionnaire est similaire à une liste, mais avec une **clé** (chaîne de caractères)

```
# accès à la valeur de la clé "a" (donc 0)
compte_voyelle_a = dict_compte_voyelles["a"]

print(f"nombre de a: {compte_voyelle_a}")
```

Pour **modifier** une valeur du dictionnaire il suffit de la réassigner

```
dict_compte_voyelles["e"] = dict_compte_voyelles["e"] + 1
```

Attention aux types de données!

Manipulations d'un dictionnaire

Pour **ajouter** de nouveaux éléments dans le dictionnaire, il suffit de faire une assignation avec une nouvelle clé

```
# dictionnaire qui associe le nom d'animaux à leur longévité
dict_animal_longevite = {
    "pékan": 10,
    "castor": 12,
    "rat": 2
}

dict_animal_longevite["chat"] = 15
```

Pour **retirer** un élément

```
del dict_animal_longevite["castor"]
```

Ou

```
dict_animal_longevite.pop("castor")
```

Manipulations d'un dictionnaire

Pour **vérifier** si une clé est présente dans un dictionnaire, on utilise le mot réservé « in »:

```
if "castor" in dict_animal_longevite:  
    print("Castor est présent dans le dictionnaire")
```
