



## 1) Les dictionnaires

Nous allons maintenant étudier un autre type abstrait de données : les dictionnaires aussi appelés tableau associatif.

On retrouve une structure qui ressemble, à première vue, beaucoup à un tableau (à chaque élément on associe un indice de position). Mais au lieu d'associer chaque élément à un indice de position, dans un dictionnaire, on associe chaque élément (on parle de valeur dans un dictionnaire) à une clé, on dit qu'un dictionnaire contient des couples clé:valeur (chaque clé est associée à une valeur). Exemples de couples clé:valeur => prenom:Kevin, nom:Durand, date-naissance:17-05-2005. prenom, nom et date sont des clés ; Kevin, Durand et 17-05-2005 sont des valeurs.

Voici les opérations que l'on peut effectuer sur le type abstrait dictionnaire :

- ajout : on associe une nouvelle valeur à une nouvelle clé
- modif : on modifie un couple clé:valeur en remplaçant la valeur courante par une autre valeur (la clé restant identique)
- suppr : on supprime une clé (et donc la valeur qui lui est associée)
- rech : on recherche une valeur à l'aide de la clé associée à cette valeur.

Exemples : Soit le dictionnaire D composé des couples clé:valeur suivants => prenom:Kevin, nom:Durand, date-naissance:17-05-2005. Pour chaque exemple ci-dessous on repart du dictionnaire d'origine :

- ajout(D,tel:06060606) ; le dictionnaire D est maintenant composé des couples suivants : prenom:Kevin, nom:Durand, date-naissance:17-05-2005, tel:06060606
- modif(D,nom:Dupont) ; le dictionnaire D est maintenant composé des couples suivants : prenom:Kevin, nom:Dupont, date-naissance:17-05-2005
- suppr(D,date-naissance) ; le dictionnaire D est maintenant composé des couples suivants : prenom:Kevin, nom:Durand
- rech(D,prenom) ; la fonction renvoie Kevin

**ATTENTION** : les clés d'un dictionnaire sont forcément des types non-mutables. Un type est non-mutable s'il ne peut pas être modifié après sa création. Par exemple, les chaînes de caractères et les tuples sont non-mutables. A contrario, les listes Python sont mutables (une liste Python peut être modifiée après sa création avec la méthode append). Il n'est donc pas possible d'utiliser une liste Python comme clé d'un dictionnaire. En revanche, il est possible d'utiliser une chaîne de caractères ou un tuple.

## 2) Implémentation des dictionnaires

L'implémentation des dictionnaires dans les langages de programmation peut se faire à l'aide des tables de hachage. Les tables de hachages ainsi que les fonctions de hachages qui sont utilisées pour construire les tables de hachages, ne sont pas au programme de NSI. Cependant, l'utilisation des fonctions de hachages est omniprésente en informatique, il serait donc bon, pour votre "culture générale informatique", de connaître le principe des fonctions de hachages. Voici un texte qui vous permettra de comprendre le principe des fonctions de hachages : [c'est quoi le hachage](#) . Pour avoir quelques idées sur le principe des tables de hachages, je vous recommande le visionnage de cette vidéo : [wandida : les tables de hachage](#)

Si vous avez visionné la vidéo de wandida, vous avez déjà compris que l'algorithme de recherche dans une table de hachage a une complexité  $O(1)$  (le temps de recherche ne dépend pas du nombre d'éléments présents dans la table de hachage), alors que la complexité de l'algorithme de recherche dans un tableau non trié est  $O(n)$ . Comme l'implémentation des dictionnaires s'appuie sur les tables de hachage, on peut dire que l'algorithme de recherche d'un élément dans un dictionnaire a une complexité  $O(1)$  alors que l'algorithme de recherche d'un élément dans un tableau non trié a une complexité  $O(n)$ .

Python propose une implémentation des dictionnaires, nous avons déjà étudié cette implémentation l'année dernière, n'hésitez pas à vous référer à la ressource proposée l'an passé : [les dictionnaires en Python](#)



## activité 6.1

Écrivez un programme Python qui permettra de créer un dictionnaire avec les couples clé/valeur suivants :

- nom => Doe
- prenom => John
- age => 42

## activité 6.2

Soit les données suivantes :

prenom	nom	age
John	Doe	42
Pierre	Martin	23
Christine	Dune	27
Marie	Tellaz	12

Proposez une structure de données permettant de stocker ces informations (on attend un programme Python).

## activité 6.3

En vous aidant de ce qui a été vu en première, écrivez un programme Python permettant de lister l'ensemble des noms "stockés" à l'activité 6.2 (on attend une structure de type boucle)



## exercice 6.1

Soit le programme Python suivant :

```
inventaire = {'pommes': 430, 'bananes': 312, 'oranges' : 274, 'poires' : 137}

stock = 0
for fruit in inventaire.keys():
    if fruit != 'bananes':
        stock = stock + inventaire[fruit]
```

Quelle est la valeur de la variable stock après l'exécution de ce programme ?

## exercice 6.2

Soit le programme Python suivant :

```
P = [{"nom": "Turing", "prenom": "Alan", "age": 28}, {"nom": "Lovelace", "prenom": "Ada", "
```

Qu'obtient-on si on tape `P[1]['age']` dans une console Python ?

## exercice 6.3

Soit le programme Python suivant :

```
def ajoute(stock,element,quantite):
    if element in stock:
        stock[element] = stock[element] + quantite
    else:
        stock[element] = quantite

stock = { 'clous': 14, 'vis': 27, 'boulons': 8, 'écrous': 24 }
ajoute(stock, 'vis', 5)
ajoute(stock, 'chevilles', 3)
```

Quelle est la valeur de la variable stock à la fin de cette exécution ?

## exercices du bac

- [Sujet 6 2021 Exercice 2](#)
- [Sujet 10 2021 Exercice 2](#)
- [Sujet 6 2022 Exercice 1](#)

- [Sujet 6 2022 Exercice 3](#)
- [Sujet 7 2022 Exercice 2](#)



## Ce qu'il faut savoir

### Types abstraits dictionnaire

Dans un dictionnaire (aussi appelé tableau associatif), chaque élément (appelé valeur) est associé à une clé. Un dictionnaire contient donc des couples clé:valeur.

Voici les opérations que l'on peut effectuer sur le type abstrait dictionnaire :

- ajout : on associe une nouvelle valeur à une nouvelle clé
- modification : on modifie un couple clef:valeur en remplaçant la valeur courante par une autre valeur (la clé restant identique)
- suppression : on supprime une clé (et donc la valeur qui lui est associée)
- recherche : on recherche une valeur à l'aide de la clé associée à cette valeur.

### Implémentation des dictionnaires

Dans beaucoup de langage de programmation les dictionnaires sont implémentés à l'aide de tables de hachage.

### Algorithme de recherche dans un dictionnaire

L'algorithme de recherche d'un élément dans un dictionnaire a une complexité  $O(1)$ . La durée de recherche ne dépend pas du nombre d'éléments présents dans le dictionnaire.

## Ce qu'il faut savoir faire

Savoir utiliser les dictionnaires en Python