

# Dictionnaires

- Le **dictionnaire** a été vu en MPSI comme une boîte noire.

On va revoir et compléter son usage.

On présentera le principe du **hachage** qui est utilisé pour gérer cette structure dans la mémoire de l'ordinateur.

- La **programmation dynamique** (inventée en 1950 par Bellman) permet de résoudre efficacement certains problèmes d'optimisation, en résolvant des problèmes identiques plus petits appelés **sous problèmes** dont **on doit mémoriser les résultats** pour éviter de répéter des calculs.

On obtient ainsi une **réduction de la complexité temporelle** au prix d'une **augmentation de la complexité spatiale**.

- Un **dictionnaire** (ou **tableau associatif**) est une collection de couples (clé, valeur).

Chaque **clé** peut être :

- un entier ;
- un flottant ;
- une chaîne de caractère ;
- un tuple.

*On ne peut pas prendre pour clé une liste ou un autre dictionnaire.*

Chaque **valeur** est un objet Python quelconque.

Les clés d'un dictionnaire ne sont pas forcément toutes de même nature, de même que ses valeurs.

## ■ Pour créer un dictionnaire :

- on peut créer un dictionnaire vide qu'on remplit ensuite :

```
dico = {}  
dico['a'] = 1  
dico['b'] = 2  
dico['c'] = 3
```

- ou énumérer ses éléments sous la forme « clé : valeur » séparés par des virgules :

```
dico = { 'a' : 1, 'b' : 2, 'c' : 3 }
```

- ou le définir par compréhension (comme pour une liste) :

```
dico = { 'abc'[i] : i+1 for i in range(3) }
```

## ■ Pour connaître la taille d'un dictionnaire :

```
len(dico)
```

- On peut ajouter un nouveau couple (clé, valeur) simplement en déclarant sa valeur :

```
dico['d'] = 4
```

Attention : si la clé existe déjà dans le dictionnaire, l'ancienne valeur est écrasée par la nouvelle.

- L'instruction

```
B = x in dico
```

affecte à la variable B la valeur `True` si x est une clé de dico (et `False` sinon).

- On peut supprimer un couple (clé, valeur) en le désignant par sa clé :

```
del dico['c']
```

ou supprimer un couple (clé, valeur) et récupérer la valeur dans une variable Python par la méthode `pop` (comme pour les listes) :

```
x = dico.pop('c')
```

## ■ Pour parcourir un dictionnaire :

- l'instruction

```
for x in dico :
```

entame une boucle dans laquelle `x` prend successivement la valeur de toutes les **clés** dans un ordre quelconque ;

- l'instruction

```
for x in dico.keys() :
```

fait la même chose ;

- l'instruction

```
for x in dico.values() :
```

entame une boucle dans laquelle `x` vaut successivement toutes les **valeurs** présentes dans le dictionnaire dans un ordre quelconque ;

- ```
for x in dico.items() :
```

entame une boucle dans laquelle `x` vaut successivement tous les **couples (clé, valeur)** dans un ordre quelconque.