

## 1 Les dictionnaires

Ce sont des structures de données (**dict**) modifiables et non séquentielles; un élément n'est pas repéré à l'aide d'un indice mais à l'aide d'un nom (sa **clé**). Un élément est constitué d'un **champ** : la donnée d'une clé et d'une valeur.

- La clé est un nombre, une chaîne ou un *t-uplet*.
- La valeur peut être de type quelconque.

Le type **dict** permet la définition de conteneurs dont les valeurs sont repérées non plus par des indices mais par des clés.

### 1.1 Déclaration d'un dictionnaire

Un dictionnaire se définit par extension, par la suite des champs (la clé suivie de la valeur, séparées de :) entre accolades {}.

#### • Déclaration

```
In [1]: Dic = { 'A' : 0, 'B' : 1, 'C' : 2, 'D' : 3 }
```

Ce qui est identique à :

```
In [1]: Dic = {}
In [2]: Dic['A'] = 0
In [3]: Dic['B'] = 1
In [4]: Dic['C'] = 2
In [5]: Dic['D'] = 3
```

### 1.2 Opérations sur les dictionnaires

L'accès à un élément se fait à l'aide de sa clé :

```
In [6]: Dic['B']
Out[6]: 1
```

Pour ajouter un champ, affecter une valeur à une nouvelle clé. Pour supprimer un champ utiliser la fonction **del()**.

```
In [7]: Dic['E'] = 4      # Ajout d'un champ
In [8]: del(Dic['C'])    # Suppression d'un champ
In [9]: Dic
{'A' : 0, 'B' : 1, 'D' : 3, 'E' : 4}
```

### 1.3 Méthodes des dictionnaires

Les dictionnaires admettent les méthodes suivantes :

Méthodes des dictionnaires	
D.keys()	retourne la liste des clés du dictionnaire D
D.values()	retourne la liste des valeurs du dictionnaire D
D.items()	retourne la liste des champs (key,value) de D
D.copy()	retourne une copie du dictionnaire D

### 1.4 Exercices

#### Exercice 1. Comptage de caractères.

- 1) Écrire une fonction **comptage** prenant en argument le nom d'un fichier texte et qui renvoie un dictionnaire dont les clés sont les caractères apparaissant dans le texte, avec pour valeur le nombre d'apparitions.
- 2) Écrire une fonction **listes** prenant en argument un dictionnaire et qui renvoie les listes des clés et valeurs (dans le même ordre) du dictionnaire.
- 3) En déduire une fonction **caractères** prenant en argument le nom d'un fichier texte et qui affiche dans la console tous ses caractères et occurrences dans l'ordre décroissant du nombre d'occurrences.

#### Exercice 2. Cryptage de César à l'aide de dictionnaires

- 1) Définir une variable **alphabet** = "abcdefghijklmnopqrstuvwxyz"
- 2) Ecrire un script qui définit un dictionnaire de clés les lettres de l'alphabet et de valeurs leur position dans l'alphabet.
- 3) Ecrire une fonction **cesar(n)** prenant en argument un entier **n** et qui renvoie un dictionnaire comme en 2, mais dont les valeurs sont décalées de l'entier **n** modulo 26.
- 4) Ecrire une fonction **cryptage(texte,n)** prenant en argument un texte sous forme d'une chaîne de caractère et un entier **n** et qui le crypte selon le chiffrement de César avec décalage de **n**.
- 5) Ecrire une fonction **decryptage(texte,n)** qui effectue l'opération de déchiffrement correspondant.