

# Les dictionnaires

## 1 Définition

### 1.1 Définition

#### Définition 8.1

Un dictionnaire python est une structure de donnée qui permet d'associer une valeur à une clé. Cette clé peut être un entier ou un str.  
L'ensemble clé-valeur est appelé entrée.

#### Remarques

- Les dictionnaires ne comportent pas d'ordre. On ne peut donc pas trouver un élément via sa position, mais seulement via sa clé.
- Python possède nativement cette structure de données, ce qui n'est pas vrai pour tous les langages de programmation. Il faudrait alors l'implémenter si on souhaite l'utiliser.

### 1.2 Interface d'un dictionnaire

Voici l'interface d'un dictionnaire :

**creer\_dico()** Permet de créer un dictionnaire vide

**Inserer(dico,c,v)** Permet d'ajouter l'entrée c-v

**supprimer(dico,c)** Permet de supprimer l'entrée dont la clé est c

**lire(dico,c)** Permet de retourner la valeur de l'entrée dont la clé est c.

## 2 Utilisation des dictionnaires Python

L'implémentation des dictionnaires Python a été vu en première. Se référer au cours de l'an passé :-)

## 3 Complexité de recherche d'un élément dans un dictionnaire

Un dictionnaire n'est pas une séquence ! Il n'y a pas d'ordre, et les éléments ne sont pas repérés par des indices.

Pour un tableau (list python) par exemple, la complexité de recherche d'un élément est linéaire, et dépend de la taille du tableau.

Ce n'est pas du tout le cas d'un dictionnaire : la complexité de recherche d'un élément est constante (ne dépend pas de la taille du dictionnaire!!). Cela est dû à son implémentation avec des tables de hachage (ce n'est pas au programme de terminale, mais c'est très intéressant..Ne pas hésiter à faire des recherches sur le Web!)

## 4 Exercices

### Exercice 8.1

Ecrire le script Python qui permet d'obtenir une structure de données de type dict qui doit être :

```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8, 'J': 9, 'K': 10,
 'L': 11, 'M': 12, 'N': 13, 'O': 14, 'P': 15, 'Q': 16, 'R': 17, 'S': 18, 'T': 19, 'U': 20,
 'V': 21, 'W': 22, 'X': 23, 'Y': 24, 'Z': 25}
```

### Exercice 8.2

Le chiffrement de César

A chaque lettre de l'alphabet on associe un nombre de 0 à 25. On ajoute à ce nombre un nombre choisi par exemple 7. Le reste de la division euclidienne du nombre obtenue par 26 correspond au chiffre qui codera la lettre initiale.

la lettre Y est associée à 24, on lui ajoute 7 on obtient 31. Le reste de la division euclidienne de 31 par 26 est 5. Ce qui donne E. Y est donc codé par E.

1. Quel script écrire pour obtenir le dictionnaire de la table de codage/décodage de l'alphabet avec la méthode du chiffrement de César avec un décalage de 7?

```
{'A': 'H', 'B': 'I', 'C': 'J', 'D': 'K', 'E': 'L', 'F': 'M', 'G': 'N', 'H': 'O', 'I': 'P',
 'J': 'Q', 'K': 'R', 'L': 'S', 'M': 'T', 'N': 'U', 'O': 'V', 'P': 'W', 'Q': 'X', 'R': 'Y',
 'S': 'Z', 'T': 'A', 'U': 'B', 'V': 'C', 'W': 'D', 'X': 'E', 'Y': 'F', 'Z': 'G'}
```

Table codage

```
{'H': 'A', 'I': 'B', 'J': 'C', 'K': 'D', 'L': 'E', 'M': 'F', 'N': 'G', 'O': 'H', 'P': 'I',
 'Q': 'J', 'R': 'K', 'S': 'L', 'T': 'M', 'U': 'N', 'V': 'O', 'W': 'P', 'X': 'Q', 'Y': 'R',
 'Z': 'S', 'A': 'T', 'B': 'U', 'C': 'V', 'D': 'W', 'E': 'X', 'F': 'Y', 'G': 'Z'}
```

Table décodage

2. Ecrire une fonction en Python `codage(mot)` qui prend en argument un chaîne de caractère écrit en lettre capitale et qui renvoie le mot codé par le chiffrement de César.

☞ Utiliser la table de codage!

```
>>> codage('VIVELANSI')
'CPCLSHUZZP'
```

3. Ecrire une fonction en Python `decodage(mot)` qui prend en argument une chaîne de caractère en majuscule et qui renvoie le mot décodé par le chiffrement de César. Décoder le message 'CVBZWVB-CLGWHZZLYHSLELYJPJLZBPCHUA'.

### Exercice 8.3

1. Un mot a été codé en 'SIRMF' avec un chiffrement de César. Décoder ce mot.

### Exercice 8.4

On considère le dico suivant :

```
dico={'nsi' :18,'maths':17,'svt':14,'français':14,'lv1':8,'physique':12,'HG':11}
```

1. Ecrire cet affichage :

```
En nsi , Jean a 18 de moyenne
En maths , Jean a 17 de moyenne
En svt , Jean a 14 de moyenne
En français , Jean a 14 de moyenne
En lv1 , Jean a 8 de moyenne
En physique , Jean a 12 de moyenne
En HG , Jean a 11 de moyenne
```

2. Effacer les informations relatives au Français dans le dictionnaire

### Exercice 8.5

Voici une citation célèbre de Gandhi :

La vie est un mystère qu'il faut vivre, et non un problème à résoudre.

Créer un dictionnaire qui associe à chaque lettre (clé) son occurrence (valeur). Par exemple la lettre 'a' apparaît deux fois.

```
{ 'l': 1, 'a': 2, 'v': 3, 'i': 3, 'e': 7, 's': 3, 't': 4, 'u': 5, 'n': 4, 'm': 2, 'y': 1, ' ' : 1, 'r': 5, 'q': 1, 'l': 2, 'f': 1, 'o': 3, 'p': 1, 'b': 1, 'à': 1, 'é': 1, 'd': 1 }
```

### Exercice 8.6

Le Titanic appareille de Southampton (Angleterre) le mercredi 10 avril à 12 h 15 .

Six heures plus tard, à 18 h 15, il fait escale dans la rade de Cherbourg. Il y débarque 24 passagers et en embarque 274, amenés par les transbordeurs Nomadic et Traffic. Il appareille à 20 h 10.

Le Titanic fait route vers l'Irlande. Il arrive à Queenstown (aujourd'hui Cobh) le 11 avril à 11 h 30. Il débarque 7 passagers et en embarque 120. À 13 h 30, le paquebot appareille et entame sa traversée de l'Atlantique vers New York .

Le 14 avril, à 23 h 40 (heure locale, GMT-3), il percute un iceberg au large de Terre-Neuve. Il sombre le 15 avril à 2 h 20, causant la mort de 1 524 personnes.

Le fichier `titanic.csv` se trouve ici .

1. Importer le contenu du fichier dans une liste nommée `titanic` (une liste de dictionnaire) :

```
>>> titanic
[{'PassengerId': '1', 'Survived': '0', 'Pclass': '3', 'Name': 'Braund, Mr. Owen Harri
s', 'Sex': 'male', 'Age': '22', 'SibSp': '1', 'Parch': '0', 'Ticket': 'A/5 21171', 'F
are': '7.25', 'Cabin': '', 'Embarked': 'S'}, {'PassengerId': '2', 'Survived': '1', 'P
class': '1', 'Name': 'Cumings, Mrs. John Bradley (Florence Briggs Thayer)', 'Sex': 'f
```

2. Écrire un script qui détermine le nombre de survivants
3. Écrire un programme qui affiche le pourcentage de survivants par classe
4. Écrire un programme qui donne le nombre de survivants embarqués à Cherbourg

### Exercice 8.7

Lorsqu'on cherche un mot dans un dictionnaire (un vrai), on tourne peu de pages pour le trouver puisqu'ils sont triés par ordre alphabétique.

Maintenant, si on souhaite trouver tous les mots dans la seconde lettre est e, c'est impossible à moins de trier les mots par leur seconde lettre : il faudrait un dictionnaire différent pour chaque position de lettre. L'objectif de cet exercice est de concevoir une sorte de dictionnaire qui permette de retrouver tous les mots ayant telle lettre à telle position.

Voici une liste de mots :

```
mots = ['edward', 'catelyn', 'robb', 'sansa', 'arya', 'brandon',
        'rickon', 'theon', 'rorbert', 'cersei', 'tywin', 'jaime',
        'tyrion', 'shae', 'bronn', 'lancel', 'joffrey', 'sandor',
        'vargs', 'renly', 'a' ]
```

1. construire un dictionnaire dico\_bien\_choisi avec :
  - La clé sous la forme d'un tuple (i,l), où i est la position de la lettre l (on commence à 0) et l est une lettre
  - La valeur associée à chaque clé est une liste contenant tous les mots contenant la lettre l à la i-ème position.
2. Créer une fonction liste\_lettre\_position de paramètres :
  - d : le dico bien choisi (dict)
  - l : la lettre choisi (str)
  - p : la position de la lettre (int)

```
>>> liste_lettre_position(d, 'r', 2)
['rorbert', 'cersei', 'tyrion', 'vargs']
```

### Exercice 8.8

Voici un message codé avec un chiffrement de César :

```
message =
"CFIJHLFETYVITYVLEDFKUREJLEUZTKZFEERZIVLEMIRZFEKFL
IEVGVLUVGRXVJGFLICVKIFLMVIGLZJHLZCJJFEKKIZVJGRIFU
IVRCGYRSVKZHLVDRZEKVEREKJZFEJFLYRZKVKIFLMVIKFLJCVJ
DFKJUFEKCRJVTFEUVCVKIVVJKVTVJKZDGFJJZSCVRDFZEJUVK
IZVICVJDFKJGRICVLIJVTFEUVCVKIVZCWRLUIRZKLEUZTKZFE
ERZIVUZWWWVIVEKGFLITYRHLVGFJJKZFEUVCVKIV"
```

Décoder ce message en utilisant une approche fréquentielle : créer un dictionnaire avec comme clé les lettres de A à Z et comme valeur associé le pourcentage de cette lettre dans le message codé.

Ensuite, créer une fonction qui utilise ce dictionnaire afin de décoder le mot

☞ Le 'E' est la lettre la plus fréquente !