

TD 08

Structure de données : dictionnaire

Lycée Louis Thuillier - Informatique - PCSIB -

Dictionnaire

En Python un dictionnaire est "une liste de paires". C'est un ensemble de paire, *i.e.* d'associations entre

▷ une **clé**

▷ et une **valeurs**, associées à la clé

Un dictionnaire se définit entre accolade {} :

```
dico = { clé_1 : valeur_1 , clé_2 : valeur_2 }
```

On note {} le dictionnaire vide.

🚨🚨🚨 **Attention !** aux deux points et aux virgules!!!

```
>>> edt={"Math" : 10 , "Physique" : 8 , "Chimie" : 4}
```

Dans un "vrai" dictionnaire, on retrouve la même structure :

▷ la clé est le mot représentant un concept

▷ la valeur est la définition du concept

Utilisation d'un dictionnaire

Avec un dictionnaire nommé *dico* je peux :

♡ **Commande** ♡. **Accéder et modifier une valeur**

▷ accéder à la **valeur** associé à une **clé** via :

```
dico[clé_1]
```

Dans une liste, la clé est la position de l'élément.

▷ je peux modifier la **valeur** associé à une **clé** via :

```
dico[clé_1] = nouvelle valeur
```

Exemple 1 : J'accède à la valeur associée à la clé "Math" du dictionnaire *edt* et j'incrmente la valeur associée à la clé "Physique".

```
>>> edt["Math"]
10

>>> edt["Physique"]=edt["Physique"]+50

>>> edt
{'Math': 10, 'Physique': 58, 'Chimie': 4}
```

| **Application 1** : Mettez dans l'emploi du temps le nombre d'heure de physique dont vous rêvez.

♡ **Commande** ♡. **Ajouter un élément à un dictionnaire** Un élément d'un dictionnaire est une **clé** et une **valeur**. Pour ajouter un nouvel élément :

```
dico[clé_new] = nouvelle_valeur
```

*** **Attention !** c'est la même commande que pour modifier une valeur associée à une clé **mais** la nouvelle clé définie n'existe pas.

Exemple 2 : Ajoutons quelques heures de SI à l'emploi du temps.

```
>>> edt["SI"]=4

>>> edt
{'Math': 10, 'Physique': 58, 'Chimie': 4, 'SI': 4}
```

| *Application 2 : Ajoutez le français et l'anglais à votre emploi du temps.*

Application 3 : Créer un dictionnaire où :

- ▷ les **clés** sont les entiers n entre 0 et 10^4
- ▷ les **valeurs** sont des chaînes de caractères "pair" ou "impair" suivant la parité de n

Application 4 :

1. Créer une fonction *nombre_diviseur* qui prend en entrée un entier n et renvoie le nombre de diviseur de n .
2. Créer un dictionnaire où :
 - ▷ les **clés** sont les entiers n entre 0 et 10^4
 - ▷ les **valeurs** sont le nombre de diviseurs de n
3. Avec le dictionnaire ainsi créé, donner le nombre de nombre premier entre 1 et 10^4 .
4. Créer un dictionnaire où :
 - ▷ les clé sont les entier m entre 0 et 10^4
 - ▷ les **valeurs** sont le nombre d'entiers n entre 0 et 10^4 qui ont m diviseurs

Itération dans un dictionnaire

Un dictionnaire est une structure **itérable** : tout comme une liste, on peut "se promener" dans un dictionnaire. **Sauf que**, contrairement à une liste, les clés ne sont pas des nombres entier compris entre 0 et $\text{len}(\text{Liste}) - 1$.

♡ **Commande** ♡. On génère successivement toutes les clés d'un dictionnaire *dico* avec la commande :

for x in dico.keys() :

Exemple 3 : On veut récupérer toutes les matières (clés) et les heures de cours (valeurs) du dictionnaire edt :

```
>>> for x in edt.keys():
...     print("On a ", edt[x], "heures de ", x)
...
...
On a  10 heures de  Math
On a  58 heures de  Physique
On a  4 heures de  Chimie
On a  4 heures de  SI
```

| *Application 5 : A l'aide d'une boucle, diviser par deux toutes les heures de cours **sauf la physique**.*

Exercice classique

On souhaite écrire une fonction *occurrences* qui, recevant une chaîne de caractères, renvoie un dictionnaire qui associe à chaque lettre minuscule de l'alphabet le nombre de fois qu'elle apparaît dans la chaîne.

1. Qui jouera la rôle de clé ? De valeur ?
2. Ecrire la fonction *occurences*.
3. (*) Ecrire une fonction *occurence_ordonnées* qui, recevant une chaîne de caractères, renvoie un dictionnaire qui associe à chaque lettre minuscule de l'alphabet le nombre de fois qu'elle apparaît dans la chaîne mais les lettres devront être rangées par ordre décroissant d'apparition.