

# TD n°7

## Les dictionnaires

Les **dictionnaires** ressemblent aux `listes` mais chaque item d'un dictionnaire est composé de 2 parties, on parle de paires "**clé / valeur**" ou "**key / value**"

Exemple de dictionnaire :

```
>>> monDico = {"nom": "Durand", "prenom": "Christophe", "date_naissance": "29/02/1981"}
```

On utilise des accolades `{ }` pour définir le début et la fin du `**dictionnaire**`.

Dans le dictionnaire 'monDico' :

"nom", "prenom" et "date\_naissance" sont des **clés**

"Durand", "Christophe" et "29/02/1981" sont des **valeurs**

La clé "nom" est associée à la valeur "Durand"

La clé "prenom" est associée à la valeur "Christophe"

La clé "date\_naissance" est associée à la valeur "29/02/1981"

Les **clés** sont des **objets non mutables** (int, float, str, tuple ...) et doivent être **uniques**.

Les **valeurs** peuvent être de tout type (int, float, str, bool, tuple, list,...)

Les **dictionnaires** sont des **objets mutables**.

Pour créer un `**dictionnaire**`, il est aussi possible de procéder comme suit :

```
>>> monDico = { }  
>>> monDico["nom"] = "Durand"  
>>> monDico["prenom"] = "Christophe"  
>>> monDico["date_naissance"] = "29/02/1981"
```

Pour afficher le contenu d'un dictionnaire, ou une valeur à partir de sa clé :

```
>>> monDico  
{'nom': 'Durand', 'prenom': 'Christophe', 'date_naissance': '29/02/1981'}  
>>> monDico['prenom']  
'Christophe'  
>>> monDico['nom']  
'Durand'  
>>> monDico['date_naissance']  
'29/02/1981'
```

Remarque : si on recherche dans un dictionnaire avec une clé inexistante, on obtient un message d'erreur (KeyError) :

```
>>> monDico['name']
Traceback (most recent call last):
  File "<pyshell>", line 1, in <module>
KeyError: 'name'
```

Il est donc préférable d'utiliser la méthode ``get`` qui ne renverra pas de message d'erreur (sauf si le spécifiez) :

```
>>> monDico.get('nom')
Durand
>>> monDico.get('name')
>>> monDico.get('name',"Attention : clé inexistante")
'Attention : clé inexistante'
```

Il est facile d'ajouter un élément à un dictionnaire :

```
>>> monDico['lieu_naissance'] = 'Bonneville'
>>> monDico
{'nom': 'Durand', 'prenom': 'Christophe', 'date_naissance': '29/02/1981',
'lieu_naissance': 'Bonneville'}
```

On peut facilement vérifier l'existence d'une clé dans un dictionnaire avec l'instruction ``in`` :

```
>>> 'nom' in monDico
True
>>> 'name' in monDico
False
```

L'instruction ``del`` permet de supprimer un item (paire "clé / valeur") du dictionnaire :

```
>>> del monDico['date_naissance']
>>> monDico
{'nom': 'Durand', 'prenom': 'Christophe', 'lieu_naissance': 'Bonneville'}
```

Il est possible de modifier une valeur :

```
>>> monDico['lieu_naissance'] = 'Paris'
>>> monDico
{'nom': 'Durand', 'prenom': 'Christophe', 'lieu_naissance': 'Paris'}
```

Il est possible de parcourir un dictionnaire à l'aide d'une boucle **for**.  
Le parcours peut se faire selon les clés ou les valeurs ou les deux.

#### Parcours selon les clés : méthode ``keys()``

```
>>> mesFruits = {"poire":3, "pomme":4, "orange":2, "pamplemousse":1}
>>> for fruit in mesFruits.keys():
    print (fruit)
poire
pomme
orange
pamplemousse
```

#### Parcours selon les valeurs : méthode ``values()``

```
>>> mesFruits = {"poire":3, "pomme":4, "orange":2, "pamplemousse":1}
>>> for quantite in mesFruits.values():
    print (quantite)
3
4
2
1
```

#### Parcours selon les couples clés, valeurs : méthode ``items()`` :

```
>>> mesFruits={"poire":3, "pomme":4, "orange":2, "pamplemousse":1}
>>> print ("Stock de fruits :")
    for fruit,quantite in mesFruits.items():
        print (fruit + " : " + str(quantite))
Stock de fruits :
poire : 3
pomme : 4
orange : 2
pamplemousse : 1
```

On peut obtenir facilement la liste de clés et des valeurs d'un dictionnaire :

```
>>> monDico.values()
dict_values(['Durand', 'Christophe', '29/02/1981'])
>>> monDico.keys()
dict_keys(['nom', 'prenom', 'naissance'])
>>> monDico.items()
dict_items([('nom', 'Durand'), ('prenom', 'Christophe'), ('naissance', '29/02/1981')])
```

### **Exercice 1**

Soit le dictionnaire défini par

```
>>> d = {'nom': 'Dupuis', 'prenom': 'Jacque', 'age': 30}
```

1. Ecrire une instruction pour corriger l'erreur dans le prénom, la bonne valeur est 'Jacques'.
2. Afficher la liste des clés du dictionnaire.
3. Afficher la liste des valeurs du dictionnaire.
4. Afficher la liste des paires clé/valeur du dictionnaire.
5. Ecrire la phrase "Jacques Dupuis a 30 ans".

### **Exercice 2**

Créer une fonction '**occurrence**' qui prend en paramètre un texte et qui retourne un dictionnaire précisant le nombre d'occurrences de chaque caractère dans le texte.

### **Exercice 3**

Dans le cadre d'un projet qui consiste à étiqueter d'un code-barre les ouvrages scolaires d'un établissement de 3 000 élèves, prévoir un ou plusieurs dictionnaire(s) permettant de connaître les livres que possède chaque élève.

### **Exercice 4 (facultatif)**

Écrire une fonction '**inverse\_dico**' qui prend en paramètre un premier dictionnaire et qui retourne un deuxième dictionnaire en échangeant les clés et les valeurs du premier (ce qui permettrait par exemple de transformer un dictionnaire anglais/français en un dictionnaire français/anglais).

*On suppose que le dictionnaire ne contient pas plusieurs valeurs identiques.*

### **Exercice 5 (facultatif)**

Écrire une fonction '**occurrence\_mot**' qui prend en paramètre un texte et qui retourne un dictionnaire précisant le nombre d'occurrences de chaque mot dans le texte.

Aide : utiliser les méthodes `split()` et `strip()` des chaînes de caractères.