

# Activité : Dictionnaires

---

Un **dictionnaire** est une structure de données qui permet de mémoriser et d'enregistrer des données en utilisant des associations entre des **clefs** et des **valeurs**.

- Une **clef** est une chaîne de caractères, type **string**. Chaque **clef** du dictionnaire est **unique**. Cette unicité garantit l'accès à la valeur associée.
- La **valeur** associée à une **clef** peut être de tout type : nombre **int** ou **float**, une chaîne de caractères **string**, un booléen **bool**, un tableau **list** ou **tuple** et même un dictionnaire **dict**.

L'ordre est important dans la notation. En premier, on écrit la **clef**, en second, la **valeur** associée à la **clef**.

En python, un **dictionnaire** est une structure de données native notée entre **accolades**.

Un dictionnaire est créé :

- soit avec la fonction **dict()** ;
- soit en énumérant les clefs et les valeurs sous forme de paire '**clef**' : **valeur**, le tout noté entre des **accolades**, les différentes paires étant séparées par des virgules.

Quelques exemples de création de dictionnaires :

```
Entrée [1]: # création d'un dictionnaire vide avec la fonction dict()
dico_1 = dict()

print(type(dico_1))

<class 'dict'>
```

```
Entrée [2]: # création d'un dictionnaire vide avec des accolades
dico_2 = {}

print(type(dico_2))

<class 'dict'>
```

```
Entrée [3]: # création d'un dictionnaire non vide avec dict
dico_3 = dict(clef_1='val_1',clef_2='val_2')

print(dico_3)
print(type(dico_3))

{'clef_1': 'val_1', 'clef_2': 'val_2'}
<class 'dict'>
```

```
Entrée [4]: # création d'un dictionnaire non vide entre accolades
dico_4 = {'clef_1':'val_1','clef_2':'val_2'}

print(dico_4)
print(type(dico_4))

{'clef_1': 'val_1', 'clef_2': 'val_2'}
<class 'dict'>
```

# Créer des dictionnaires

- 1) Notre premier dictionnaire contient les chiffres du système décimal. Chaque clef est le chiffre écrit en lettre et chaque valeur est le chiffre lui-même. Par exemple, pour le chiffre zéro, la clef est **zéro** et la valeur **0**.
  - a) Créer un dictionnaire vide **chiffres**.
  - b) L'ajout d'une donnée dans un dictionnaire suppose d'insérer une nouvelle clef et la valeur associée. La syntaxe est : **chiffres['clef'] = valeur**.  
Ajouter au dictionnaire **chiffres** les dix chiffres du système décimal.
  - c) Compléter le dictionnaire **chiffres** pour avoir tous les chiffres du système hexadécimal.
- 2) Notre second dictionnaire contient les capitales de pays européens. Chaque clef est un nom de pays et la valeur associée est une capitale.
  - a) Créer un dictionnaire **pays\_capitales** contenant 5 pays européens avec leurs capitales.
  - b) Ajouter deux pays européens et leurs capitales.

## Fonctions et méthodes d'un dictionnaire

Certaines fonctions vues avec les tableaux en Python (liste et tuple) s'appliquent aussi aux dictionnaires.

- 1) La fonction **len** renvoie le nombre d'éléments d'un dictionnaire.  
Écrire une instruction python donnant le nombre de valeurs des deux dictionnaires créés précédemment.
- 2) Les fonctions **min** et **max** s'appliquent-elles aux dictionnaires ?
- 3) Il existe des méthodes propres aux dictionnaires. En voici quatre :
  - La méthode **keys()** renvoie les clefs d'un dictionnaire ; Le type renvoyé est objet itérable, ce qui permet de parcourir son contenu avec une boucle.
  - La méthode **values()** renvoie les valeurs d'un dictionnaire ; Le type renvoyé est objet itérable, ce qui permet de parcourir son contenu avec une boucle.
  - La méthode **items()** renvoie les paires clefs et valeurs d'un dictionnaire ; Le type renvoyé est objet itérable, ce qui permet de parcourir son contenu avec une boucle.
  - La méthode **get(clef)** prend en paramètre une clef du dictionnaire et renvoie la valeur associée à cette clef.

Écrire des instructions en Python pour :

- a) afficher les clefs du dictionnaire **chiffres**.
- b) afficher les valeurs du dictionnaire **pays\_capitales**.
- c) afficher les items du dictionnaire **pays\_capitales**.
- d) afficher les chiffres impairs du dictionnaire **chiffres**.

# Manipuler un dictionnaire

Le fichier `capitales.py` contient les différents pays du monde et leurs capitales réunis dans un même dictionnaire. Pour utiliser ce dictionnaire, vous devez enregistrer ce fichier dans le même dossier que votre fichier de travail, puis réaliser un `import`.

- 1) Importer le dictionnaire `capitales` en saisissant l'instruction python suivante :  
`from capitales import capitales`
- 2) Combien de données contient le dictionnaire `capitales` ?
- 3) Écrire une instruction python qui affiche la capitale du **Belize**.
- 4) Écrire un script Python qui permet d'obtenir le pays dont la capitale est **Port Moresby**.
- 5) On veut créer un tableau `Pays_P` contenant tous les pays commençant par la lettre P. Écrire un script python qui crée ce tableau (attention ce n'est pas un affichage mais une variable).
- 6) On veut créer un tableau `Capitales_C` contenant toutes les capitales commençant par la lettre C. Écrire un script python qui crée ce tableau.
- 7) On veut créer un tableau `Pays_C` contenant tous les pays dont la capitale commence par la lettre C. Écrire un script python qui crée ce tableau (attention ce n'est pas un affichage mais une variable).
- 8) Écrire une fonction `recherche` avec 2 paramètres `nature` et `lettre`. Le paramètre `nature` accepte deux valeurs qui sont `pays` ou `capitale`. Le paramètre `lettre` prend comme valeur une `lettre` de l'alphabet. La fonction renvoie une liste contenant toutes les valeurs (pays ou capitales) commençant par la lettre passée en argument.
- 9) Certains pays ont un nom de capitale identique au nom du pays. Écrire un script python qui crée une liste contenant ces pays.