

Encore et encore des dictionnaires

Projet 1 : Capitales du monde

Déclarer un dictionnaire nommé `capitale` qui aura pour clé le nom (`str`) d'un pays, et pour valeur le nom (`str`) de sa capitale. Ce dictionnaire aura pour longueur minimale 40.

Ecrire la fonction `afficher_capitale(dico)` qui prendra un dictionnaire en paramètre et qui demandera à l'utilisateur (`input`) le nom d'un pays et affichera la phrase : " Le pays dont le nom est ... a pour capitale ..." (il y aura évidemment quelque chose à la place des "...").

Si le pays ne fait pas partie des clés du dictionnaire, la fonction affichera "Je ne connais pas ce pays."

Ecrire la fonction `afficher_pays(dico)` qui prendra un dictionnaire en paramètre et qui demandera à l'utilisateur (`input`) le nom d'une capitale et affichera la phrase : " ... est la capitale du pays dont le nom est ..." (il y aura évidemment quelque chose à la place des "...").

Si la capitale ne fait pas partie des valeurs du dictionnaire, la fonction affichera "Je ne connais pas cette capitale."

Projet 2 : des chiffres et des lettres

Déclarer le dictionnaire `chiffre` qui contiendra les 10 chiffres (de 0 à 9) en chiffres (`str`) pour la clé, et en toute lettre (`str`) pour la valeur (ex: "3" : "trois").

Ecrire la fonction `convertir_en_lettres(message)` :

- qui prendra en paramètre un message (`str`)
- et qui retournera ce message (`str`) réécrit en remplaçant les éventuels chiffres présents par leur écriture en toute lettre

```
>>> convertir_en_lettres("il était haut comme 3 pommes !")
"il était haut comme trois pommes !"
```

Attention, si le chiffre fait partie d'un nombre, il ne sera pas converti

```
>>> convertir_en_lettres("J'ai eu 0 sur 20 en maths...")
"J'ai eu zéro sur 20 en maths..."
```

Et non pas

```
"J'ai eu zéro sur deuxzéros en maths..."
```

Stratégie : comment vérifier qu'un chiffre fait parti d'un nombre ?

Projet 3 : Vente en ligne

Voici le dictionnaire dico_prix dont les clés sont des articles (str) et les valeurs le prix (int ou float) en euros.

```
dico_prix={ "appareil photo" : 359 , "ordinateur" : 499 , "tele" : 599, "radio" : 49 , "enceinte Bluetooth" : 79 , "smartphone" : 189, "barre de son" : 219}
```

Dans un fichier vente_en_ligne.py (à créer) :

- déclarer ce dictionnaire
- écrire la fonction prix_de(article, dico) qui prendra en paramètre le nom d'un article et un dictionnaire des prix qui affichera "le prix de l'article ... est ... €" (il y aura évidemment quelque chose à la place des "..."). Si l'article n'est pas dans les clés du dictionnaire, la fonction affichera "l'article ... n'est pas vendu dans notre magasin".

```
>>> prix_de('radio', dico_prix)
le prix de l'article radio est 49 €
>>> prix_de('micro-onde', dico_prix)
l'article micro-onde n'est pas vendu par dans notre magasin
```
- écrire la fonction liste_articles_dont_le_prix_est_superieur_a(valeur, dico) qui retournera la liste (initialement vide) des noms des articles parmi les clés du dictionnaire dont le prix est supérieure à la valeur passée en paramètre.

```
>>> liste_articles_dont_le_prix_est_superieur_a(200, dico_prix)
['appareil photo', 'ordinateur', 'tele', 'barre de son']
>>> liste_articles_dont_le_prix_est_superieur_a(900, dico_prix)
[]
```
- écrire la fonction le_plus_cher(dico) qui retournera le nom de l'article le plus cher parmi les clés du dictionnaire.

```
>>> le_plus_cher(dico_prix)
'tele'
```
- écrire la fonction catalogue(dico) qui affichera tous les articles du dictionnaire passé en paramètres (un article par ligne), suivi de son prix en euros

```
>>> catalogue(dico_prix)
appareil photo : 359 €
ordinateur : 499 €
tele : 599 €
radio : 49 €
enceinte Bluetooth : 79 €
smartphone : 189 €
barre de son : 219 €
```