

Les dictionnaires et recherche en table - Feuille d'exercices

► Exercice 1 - Exercice débranché - ★

1. On considère les scripts ci-dessous. Pour chacune des expressions, l'associer au(x) tableau(x) associatif(s) correspondant(s) :

```
dico={
    'keyboard': 'clavier',
    'souris': 'mouse'
    'computer': 'ordinateur' }
```

```
dico = {}
dico['keyboard'] = 'clavier'
dico['mouse'] = 'souris'
dico['computer'] = 'ordinateur'
```

```
dico = {}
dico['mouse'] = 'souris'
dico['keyboard'] = 'clavier'
dico['computer'] = 'ordinateur'
```

'keyboard'	'mouse'	'computer'
'clavier'	'souris'	'ordinateur'

'keyboard'	'souris'	'computer'
'clavier'	'mouse'	'ordinateur'

'mouse'	'keyboard'	'computer'
'souris'	'clavier'	'ordinateur'

2. On considère dans cette question, le dictionnaire capitales déclaré ainsi :

```
capitales={'France': 'Paris', 'Allemagne': 'Berlin', 'Belgique': 'Bruxelles', 'France': 'Marseille'}
```

Après exécution de l'instruction `capitales` dans la boucle d'interaction, on obtient (choisir la bonne réponse) :

Réponse a : {'France': 'Marseille', 'Allemagne': 'Berlin', 'Belgique': 'Bruxelles'}

Réponse b : {'France': 'Paris', 'Allemagne': 'Berlin', 'Belgique': 'Bruxelles'}

Réponse c : {'France': 'Paris', 'Allemagne': 'Berlin', 'Belgique': 'Bruxelles', 'France': 'Marseille'}

3. On considère le script donné ci-dessous :

```
1 capitaleseuro={'France': 'Paris', 'Suisse': 'Berne', 'Belgique': 'Bruxelles', 'Espagne': 'Madrid'}
2 capitaleseuro['Islande']='Reykjavik'
3 print(capitaleseuro)
4 print(len(capitaleseuro))
5 print(capitaleseuro['France'])
6 del(capitaleseuro['Belgique'])
7 print(capitaleseuro)
8 print(list(capitaleseuro.keys()))
9 print(capitaleseuro['Italie'])
10 print(capitaleseuro[0])
```

Pour chacune des lignes d'instructions du script ci-dessus, dire quelle est la réponse obtenue dans la boucle d'interaction après son interprétation :

(a) ligne 3

(c) ligne 5

(e) ligne 8

(g) ligne 10

(b) ligne 4

(d) ligne 7

(f) ligne 9

► Exercice 2 - débranché en grande partie - ★

On considère le dictionnaire suivant saisi dans la boucle d'interaction :

```
>>> valise={"Pantalon":3, "Cravate":0, "Chemise":2, "T-Shirt":5, "Caleçon":7, "pull":3,
...         "paire de chaussettes":7}
```

Ce dictionnaire symbolise le contenu de la valise de Max pour partir en vacances pendant une semaine.

1. A quel tableau associatif correspond ce dictionnaire valise?

Pour répondre à cette question, on complétera un tableau de ce type :

Clef
Valeur

2. Quelle instruction doit-on saisir dans la boucle d'interaction (console) pour obtenir la valeur associée à la cle 'Cravate'?
3. Max a acheté un chapeau et veut l'ajouter à sa valise. Quelle instruction doit on saisir dans la boucle d'interaction pour mettre à jour sa valise?
4. La valeur associée à la cle 'Cravate' étant nulle, on désire retirer du dictionnaire le couple (clé,valeur) correspondant. Quelle instruction doit-on saisir dans la boucle d'interaction?
5. Question facultative : Max veut partir finalement en vacances pour deux semaines. Il doit donc amener le double de chaque sorte de vêtements. Quelles instructions doit-on saisir pour obtenir le dictionnaire valise mis à jour avec les nouvelles quantités?
6. Vérifier les réponses à l'aide d'un IDLE et les corriger si besoin.

▷ Exercice 3 - Affichages ★

1. Ecrire une fonction **affiche** qui prend en paramètre un dictionnaire **d** et qui affiche toutes les associations clés/valeurs du dictionnaire **d**.
Par exemple, `affiche({'pommes': 430, 'bananes': 312, 'oranges' : -274})` doit afficher :
pommes 430
bananes 312
oranges -274
2. Ecrire une fonction **test** qui prend en paramètre un dictionnaire **d** et une valeur **v** et qui affiche toutes les clés du dictionnaire **d** associées à une valeur supérieure ou égale à **v**.
Par exemple, `test({'pommes': 430, 'bananes': 312, 'oranges' : -274}, 350)` doit afficher :
pommes 430

▷ Exercice 4

Dans cet exercice, nous nous familiarisons avec les manipulations sur les dictionnaires sur une thématique d'un magasin en ligne.

La base des prix des produits du magasin en ligne considérée est donnée par la table suivante :

Nom du produit	Clavier	Souris	Ecran	Tour PC	Casque	Casque VR	Cle USB
Prix en €(TTC)	38,99	11,59	125,00	379,89	17,84	350,69	16,99

1. Créer un dictionnaire nommé **base_prix** correspondant à la table précédente.
2. Ecrire une fonction **disponibilite** qui prend en paramètre un dictionnaire de base de prix **b** et un nom de produit **p** et qui renvoie True si le produit **p** est dans la base et False sinon.
3. Ecrire une fonction **prix_moyen** qui prend en paramètre un dictionnaire de base de prix **b** et qui renvoie le prix moyen des produits présents dans la base.
4. Ecrire une fonction **fourchette_prix** qui prend en paramètres un prix minimum **mini** , un prix maximum **maxi** et un dictionnaire de base de prix **b** et qui affiche les produits disponibles dans cette fourchette de prix.
5. Le panier est un concept important dans les magasins en ligne. On représentera le panier par un dictionnaire :
 - les noms de produit comme clés
 - une quantité d'achat comme valeurs associées

Créer un dictionnaire nommé **panier** correspondant à l'achat de trois clés usb, d'une souris gaming et d'un clavier gaming.

6. Ecrire une fonction *stock* qui prend en paramètres un dictionnaire de panier d'achat **p** et un dictionnaire de base de prix **b** et qui retourne True si tous les produits sont disponibles ou False sinon.

Remarques :

- On suppose que si un article du panier est présent dans la base de prix, il est également disponible dans la quantité demandée.
- La fonction **stock** devra appeler la fonction **disponibilite**.

7. Ecrire une fonction **facture** qui prend en paramètres un dictionnaire de panier d'achat **p** et un dictionnaire de base de prix **b** et qui retourne le montant total de la facture à payer.

Remarque : on suppose que tous les articles du panier sont tous disponibles dans les quantités demandées dans la base de prix

▷ **Exercice 5** - Compréhension en vrac - ★★

Cet exercice propose d'approfondir les expressions de compréhension de listes et de dictionnaire.

Soient les variables suivantes :

```
Liste=[2,5,12,31,2,17,31,42,2]
Dico={'xx':'bli','zy':'blo','cuicui':'toutou','miaou':'toutou'}
```

1. Question débranchée : Donner le résultat d'évaluation des expressions suivantes :

```
expression1=[(k,Dico[k]) for k in Dico ]
expression2=[(k,v) for (k,v) in Dico.items() ]
expression3=[Dico[k] for k in Dico ]
expression4=[v for (k,v) in Dico.items() ]
expression5={ k:Dico[k] for k in Dico }
expression6={ Dico[k]:k for k in Dico }
expression7={ (v+v):k for (k,v) in Dico.items() }
```

2. Vérifier les réponses à l'aide d'un IDLE et les corriger si besoin.

▷ **Exercice 6** - Site WEB

1. Pour créer un compte sur un site WEB, un utilisateur doit donner un nom, un login et un mot de passe. Ensuite, pour accéder au site, l'utilisateur ne doit fournir que le couple (login,mdp) afin de s'identifier; Par ailleurs, le site stocke tous les identifiants et la date de dernière connexion dans un dictionnaire D1 au format :

{ **login** : {'nom' : **nom**, 'mdp' : **mdp**, 'date' : date } } où les champs en gras sont entrés par l'utilisateur.

Voici le contenu du dictionnaire D1 aujourd'hui :

```
{'LouLouCMoi': {'nom':'Lou','mdp':'3juillet2005@!','date':'16/12/2018' },
'oznE': {'nom':'Enzo','mdp':'Mbappe78','date':'23/2/2019'}}
```

- (a) Ecrire une ligne de code permettant d'afficher le nombre d'utilisateurs inscrits aujourd'hui.
- (b) Ecrire une fonction **auth** qui accepte trois paramètres : **d** (type dict), **login** (type str) et **mdp** (type str) telle que :
- si le couple (login,mdp) existe, elle retourne le message : "Bonjour **nom**! Quoi de neuf depuis le **date**?" où **nom** et **date** sont les valeurs associées aux clés login et mdp

- si le login existe mais le mdp ne correspond pas, elle retourne : "Mot de passe INCORRECT"
- si le login n'existe pas, elle retourne : "Identifiant Inconnu, veuillez créer un compte"

2. Que renvoie l'instruction suivante?

```
print(D1 ['LouLouCMoi'] ['mdp'])
```

3. Ecrire une fonction **creation** qui accepte quatre paramètres : **d** (type dict), **login** (type str), **mdp** (type str) et **nom** (type str). On suppose que l'on dispose d'une fonction **jour** qui est appelée sans arguments et renvoie la date du jour au format 'JJ/MM/AAAA'.
- Si la clé **login** est déjà présente alors **creation** retourne le message : "Login existant, veuillez choisir un autre identifiant"
 - Sinon **creation** retourne le dictionnaire **d** actualisé