

EXERCICE 1

:

```
# coding: utf8
'''Voici une proposition de correction pour les exercices ;
Quand il y en a, les réponses de la console sont inscrites
en commentaire après la commande'''

#QUESTION: 1
dico={
'keyboard': 'clavier',
'souris': 'mouse',
'computer': 'ordinateur' }
```

tableau 2

:

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

tableau 1 & 3

:

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

tableau 1 & 3

:

```
#QUESTION: 2
capitales={'France': 'Paris', 'Allemagne': 'Berlin', 'Belgique': 'Bruxelles', 'France'
: 'Marseille'}
capitales
```

:

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

:

```
#QUESTION: 3
capitaleseuro={'France':'Paris','Suisse':'Berne','Belgique':'Bruxelles','Espagne':'Madrid'}
capitaleseuro['Islande']='Reykjavik'
print(capitaleseuro)
```

```
{'France': 'Paris', 'Suisse': 'Berne', 'Belgique': 'Bruxelles', 'Espagne': 'Madrid', 'Islande': 'Reykjavik'}
```

:

```
print(len(capitaleseuro))
```

5

:

```
print(capitaleseuro['France'])
```

Paris

:

```
del(capitaleseuro['Belgique']) #supprime la clé 'Belgique'
print(capitaleseuro)
```

```
{'France': 'Paris', 'Suisse': 'Berne', 'Espagne': 'Madrid', 'Islande': 'Reykjavik'}
```

:

```
print(list(capitaleseuro.keys()))
```

```
['France', 'Suisse', 'Espagne', 'Islande']
```

]:

```
print(capitaleseuro['Italie'])#KeyError car la clé n'existe pas
```

```
-----
-----
KeyError                                Traceback (most recent call
1 last)
<ipython-input-10-a727d9d6fa5c> in <module>
----> 1 print(capitaleseuro['Italie'])#KeyError car la clé n'existe
pas
```

```
KeyError: 'Italie'
```

]:

```
print(capitaleseuro[0])#KeyError car la clé n'existe pas
```

```
-----  
-----  
KeyError                                Traceback (most recent call  
1 last)  
<ipython-input-11-bd62206bed4e> in <module>  
----> 1 print(capitaleseuro[0])#KeyError car la clé n'existe pas  
  
KeyError: 0
```

EXERCICE 2

]:

```
#QUESTION: 1  
valise={"Pantalon":3,"Cravate":0,"Chemise":2,"T-Shirt":5,"Caleçon":7,"pull":3,"p  
aire de chaussettes":7}
```

valise possède la structure d'un dictionnaire en python.

]:

```
#QUESTION: 2  
valise['Cravate']
```

]:

0

]:

```
#QUESTION: 3  
valise['Chapeau']=1
```

]:

```
#QUESTION: 4  
del valise['Cravate']
```

]:

```
#QUESTION: 5  
valise={k: 2*v for k,v in valise.items()}  
valise
```

]:

```
{'Pantalon': 6,  
 'Chemise': 4,  
 'T-Shirt': 10,  
 'Caleçon': 14,  
 'pull': 6,  
 'paire de chaussettes': 14,  
 'Chapeau': 2}
```

EXERCICE 3

]:

```
#QUESTION: 1
def affiche(d):
    for i in d:
        print(i, '/', d[i], end='; ')
affiche({'pommes': 430, 'bananes': 312, 'oranges': -274})
```

pommes / 430; bananes / 312; oranges / -274;

]:

```
#QUESTION: 2
def test(d,v):
    for i in d:
        if d[i]>=v:
            print(i, '/', d[i], end='; ')
test({'pommes': 430, 'bananes': 312, 'oranges': -274}, 350)
```

pommes / 430;

EXERCICE 4

]:

```
#QUESTION: 1
base_prix={'clavier':38.99,'souris':11.59,'ecran':125.0,
           'tour':379.89,'casque':17.84,'casque':350.59,'cle_usb':16.99}
```

]:

```
#QUESTION: 2
def disponibilite(b,produit):
    return produit in b
print(disponibilite(base_prix,'souris'))
```

True

]:

```
#QUESTION: 3
def prix_moyen(b):
    somme=0
    for prix in b.values():
        somme=somme+prix
    return somme/len(b)
print(prix_moyen(base_prix))
```

153.84166666666667

]:

```
#QUESTION: 4
#version 1: Parcours sur les clés
def fourchette_prix(mini,maxi,b):
    for p in b:
        if b[p]>=mini and b[p]<=maxi:
            print(p,b[p])
fourchette_prix(15,150,base_prix)
```

clavier 38.99
ecran 125.0
cle_usb 16.99

]:

```
#version 2 Parcours sur les clé et les valeurs avec la méthode .items()
def fourchette_prix2(mini,maxi,b):
    for produit,prix in b.items():
        if prix>=mini and prix<=maxi:
            print(produit,':',prix,end='; ')
fourchette_prix2(15,150,base_prix)
```

clavier : 38.99; ecran : 125.0; cle_usb : 16.99;

]:

```
#QUESTION: 5
panier={'clavier':1,'souris':1,'cle_usb':3}
```

]:

```
#QUESTION: 6
#version 1
def stock1(p,b):
    d=True
    for produit in p:
        d=disponibilite(b,produit) and d
    return d
#version 2
def stock2(p,b):
    for produit in p:
        if not disponibilite(b,produit):
            return False
    return True
print(stock1(panier,base_prix))
print(stock2(panier,base_prix))
```

True
True

]:

```
print(stock1({'cafetiere':1},base_prix))
print(stock2({'cafetiere':1},base_prix))
```

False
False

]:

```
#QUESTION: 7
def facture(p,b):
    f=0
    for produit in p:
        f=b[produit]+f
    return f
print(facture(panier,base_prix))
```

67.57

EXERCICE 5

]:

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

]:

```
expression1=[(k,Dico[k]) for k in Dico ]
print(expression1)
```

```
[('xx', 'bli'), ('yzy', 'blo'), ('cuicui', 'toutou'), ('miaou', 'toutou')]
```

]:

```
expression2=[(k,v) for (k,v) in Dico.items() ]
print(expression2)
```

```
[('xx', 'bli'), ('yzy', 'blo'), ('cuicui', 'toutou'), ('miaou', 'toutou')]
```

]:

```
expression3=[Dico[k] for k in Dico ]
print(expression3)
```

```
['bli', 'blo', 'toutou', 'toutou']
```

]:

```
expression4=[v for (k,v) in Dico.items() ]
print(expression4)
```

```
['bli', 'blo', 'toutou', 'toutou']
```

]:

```
expression5={ k:Dico[k] for k in Dico }
print(expression5)
```

```
{'xx': 'bli', 'yzy': 'blo', 'cuicui': 'toutou', 'miaou': 'toutou'}
```

]:

```
expression6={ Dico[k]:k for k in Dico }  
print(expression6)
```

```
{'bli': 'xx', 'blo': 'zyz', 'toutou': 'miaou'}
```

]:

```
expression7={ (v+v):k for (k,v) in Dico.items() }  
print(expression7)
```

```
{'bilibli': 'xx', 'bloblo': 'zyz', 'toutoutoutou': 'miaou'}
```

EXERCICE 6

]:

```
#QUESTION: 1 a  
D1={'LouLouCMoi': {'nom':'Lou','mdp':'3juillet2005@!','date':'16/12/2018' },  
   'oznE': {'nom':'Enzo','mdp':'Mbappe78','date':'23/2/2019'}}  
print(len(D1))
```

2

]:

```
#QUESTION: 1 b  
def auth(d,login,mdp):  
    if login not in d.keys():  
        return "Identifiant Inconnu, veuillez creer un compte"  
    elif d[login]['mdp']!=mdp:  
        return "Mot de Passe INCORRECT"  
    else :  
        return "Bonjour "+d[login]['nom']+". Quoi de neuf depuis le "+ d[login]['date'] + " ?"  
print(auth(D1,'LouLouCMoi','3juillet2005@!'))
```

Bonjour Lou. Quoi de neuf depuis le 16/12/2018 ?

]:

```
print(auth(D1,'oznE','Neymar!!!'))
```

Mot de Passe INCORRECT

]:

```
print(auth(D1,'Try','to_hack_this!!!'))
```

Identifiant Inconnu, veuillez creer un compte

]:

```
#QUESTION: 2  
print(D1 ['LouLouCMoi'] ['mdp'])
```

3juillet2005@!

]:

```
#QUESTION: 3
def creation(d,login,mdp,nom):
    #aujourd'hui=jour()*****Appel d'une fonction fictive jour()
    if login in d.keys():
        return "Login existant, veuillez choisir un autre identifiant"
        #d[login]={'nom':nom,'mdp':mdp,'date':aujourd'hui}
    else :
        d[login]={'nom':nom,'mdp':mdp,'date':'22/06/2019'}
    return d
print(creation(D1,'the_brat','n3wforc3','Patrice'))
print(creation(D1,'the_brat','n3wforc3','Patrice'))
```

```
{'LouLouCMoi': {'nom': 'Lou', 'mdp': '3juillet2005@!', 'date': '16/12/2018'}, 'oznE': {'nom': 'Enzo', 'mdp': 'Mbappe78', 'date': '23/2/2019'}, 'the_brat': {'nom': 'Patrice', 'mdp': 'n3wforc3', 'date': '22/06/2019'}}
```

Login existant, veuillez choisir un autre identifiant