

Dictionnaires : Exercices en classe

Exercice 1. Compréhension

Que donnent les codes suivants ?

a)

```
ani1 = {}
ani1["nom"] = "girafe"
ani1["taille"] = 5.0
ani1["poids"] = 1100
print(ani1)
```

b)

```
ani2 = {"nom": "singe", "poids": 70, "taille": 1.75}
print(ani2["poids"])
```

c)

```
ani3 = {"nom": "lion", "poids": 140, "taille": 2.25}
ani3["poids"] = ani3["poids"] + 20
print(ani3)
```

d)

```
for key in ani3:
    ... print(key, ani3[key])
```

e)

```
"poids" in ani3
"age" in ani2
```

f)

```
animaux = [ani1, ani2, ani3]
for ani in animaux.keys():
    ... print(ani['nom'])
```

Exercice 2. parcours de dictionnaires

```
courses = {
    "Beurre": 250,
    "Farine": 420,
    "Sucre": 200,
    "Œufs": 90,
    "levure": 5,
    "raton-laveur": 450
}
for ingredient in courses.keys():
    courses[ingredient] = courses[ingredient] * 2
print(courses)
```

S=0

```
for masse in courses.values():
    S = S + masse
print(S)
```

Exercice 3. Fonctions mystères

Que renvoient les fonctions suivantes, appliquées aux dictionnaires donnés:

def f(dico): t=[] for clef in dico.keys(): t=t+[clef] return t	def g(dico): t=[] for clef in dico.keys(): t=t+[dico[clef]] return t
--	--

Dictionnaires :

panier={"pommes":7, "salade":1, "couleur":"Jaune"}	foret={} foret[(1,1)]= "peuplier" foret[(1,3)]= "chene" foret[(3,1)]= "hetre"	alien={} alien['x']=37 alien['y']=-12 alien['type']="3ème" alien['niveau']=13
--	--	---

Exercice 4. Ecrire le dico

Ecrire le code python donnant un dictionnaire représentant cette liste, avec en clef le nom du produit et en valeur le prixTTC

Nom du produit	Prix TTC
Sabre laser	229.0
Mitendo DX	127.30
Coussin Linux	74.50
Slip Goldorak	29.90
Station Nextpresso	184.60

Exercice 5. base d'étudiants

On regroupe souvent plusieurs dictionnaires du même styles dans un tableau. Cela permet de faire une petite base de données :

Soit la variable suivante :

```
BaseUPMC = [
{"nom": 'GARGA' , "prenom": 'Amel' , "id": 20231343 , "notes":
[12 , 8 , 11 , 17 , 9 ]},
{"nom": 'POLO' , "prenom": 'Marcello' , "id": 20342241 ,
"notes": [9 , 11 , 19 , 3 ] },
{"nom": 'AMANGEAI' , "prenom": 'Hildegard' , "id": 20244229 ,
"notes": [15 , 11 , 7 , 14 , 12 ] },
{"nom": 'DENT' , "prenom": 'Arthur' , "id": 42424242 , "notes":
[8 , 4 , 9 , 4 , 12 , 5 ] },
{"nom": 'ALEZE' , "prenom": 'Blaise' , "id": 30012024 , "notes":
[17 , 15 , 20 , 14 , 18 , 16 , 20 ] }
]
```

De quoi s'agit-t'il ?

Que vaut `len(BaseUPMC)` ?

Que vaut `BaseUPMC[0].keys()` ?

Que vaut `BaseUPMC[2].values()` ?

Qu'affiche :

```
for eleve in BaseUPMC:
    print(eleve["prenom"])
```

Qu'affiche :

```
for eleve in BaseUPMC:
    notes=eleve["notes"]
    S=0
    for note in notes:
        S=S+note
    print(S/len(notes))
```

Exercices de programmation

Exercice 1. Gestion de panier

Un magasin recense ses produits avec un dictionnaire nommé **produits**, **nom (clef) prixTTC(valeur)** comme dans l'exercice 1.4

Nom du produit	Prix TTC
Sabre laser	229.0
Mitendo DX	127.30
Coussin Linux	74.50
Slip Goldorak	29.90
Station Nextpresso	184.60

- 1) Ecrire une fonction nommée **calculePrixHT** qui prend en entrée un dictionnaire de produits: **prods** qui renvoie un autre dictionnaire (**result**) ou on aura modifié la valeur **prixTTC** par **prixHT** pour chaque clef du tableau.

Rappel : Pour calculer le prixHT, on divise le prixTTC par 1,20

- 2) Ecrire une fonction nommée **achat()** qui prend en entrée un dictionnaire de produit: **prods** ainsi qu'une liste de quantité de chaque produit: **quantites** qui renvoie un **Tuple (totalTTC, totalHT)** contenant le prix total TTC et HT à payer pour la quantité de produits achetés.

Exemple :

```
produits={"fraises":2.25,"pommes":1.30,"bananes":0.70}
```

```
et q=[2,3,4]
```

```
achat(produits, q) doit calculer 2*2.25+3*1.30+4*0.70 et renvoyer le tuple (11.20,9.33)
```

Exercice 2. moyenne Eleve

En vous inspirant de l'exercice 5 'exercices en classe: base d'étudiants', écrire la fonction **moyenneEleve** qui :

- ✓ prend en entrée deux variables :
base : la base d'élèves
nom : un nom de famille
- ✓ calcule et renvoie la moyenne des notes si le nom est dans la base, et **"inconnu"** sinon.

Exercice 3. todo liste

Le dictionnaire **todo** contient les clefs suivantes :

titre, description, priorité (un entier), couleur

Le tableau **todoListe** contient plusieurs dictionnaires de cette forme

Ecrire une fonction nommée **todoPrio()** qui:

prend en entrées une **todoListe** et un nombre entier P

renvoie un tableau contenant les titres de tous les dictionnaires dont la priorité est supérieure à P.

Ecrire une fonction nommée **todoCouleur()** qui :

prend en entrée une **todoListe** et une couleur

renvoie un tableau contenant les titres de tous les dictionnaires qui ont cette couleur

Exemples :

```
todo1={"titre":"voiture","description":"aller au garage","priorité":3,"couleur":"rouge"}
```

```
todo2={"titre":"enfants","description":"trouver babysitter","priorité":3,"couleur":"vert"}
```

```
todo3={"titre":"courses","description":"payer amap","priorité":1,"couleur":" rouge "}
```

```
tdL1=[todo1,todo2,todo3]
```

```
todoPrio(tdL1,2) renvoie ["voiture","enfants"]
```

```
todoCouleur(tdL1,"rouge") renvoie ["enfants","courses"]
```

Exercice 4. Décompte de lettres

La fonction **fichierToString** prend en entrée

un nom de fichier : **nomFichier**

lis le texte contenu dans ce fichier et le renvoie comme une chaîne de caractère: **t**

```
def fichierToString(nomFichier):
    texte = open(nomFichier, "r")
    t=""
    for ligne in texte:
        t=t+str(ligne)
    return t
```

Compléter la fonction **compteLettre** suivante qui

- ✓ prend en entrée une chaîne de caractères : **texte**
- ✓ renvoie un dictionnaire (**dico**) contenant
 - en clef : une lettre
 - en valeur : le nombre d'occurrence de chacune des lettres du texte.

```
def compteLettre(texte):
    dico={}
    for lettre in texte:
        if lettre in dico:
            dico[lettre]=...
        else:
            dico[lettre]=...
    return dico
```

Complétez la fonction **majusculesSansPonctuation** qui :

- ✓ prend en entrée une chaîne de caractères : **P**
- ✓ renvoie la chaîne **Pcodee** réalisée ainsi :
 - Pour chaque lettre de **P**,
 - si c'est une majuscule, on la conserve
 - si c'est une minuscule, on la remplace par sa majuscule,
 - sinon on la remplace par ""

```
import unicodecode
def majusculesSansPonctuation(P):
    P = unicodecode.unicodecode(P)
```

```
#supprime les accents
Pcodee=""
for lettre in P:
    code = ord(lettre)
    if
        lettreC=
    elif
        lettreC=
    else:
        lettreC=
    Pcodee=Pcodee+lettreC
return Pcodee
```

En utilisant cette dernière fonction, modifier la fonction **compteLettre** pour qu'elle ne compte que les lettres de l'alphabet.

Défi : comment renvoyer un tableau ou les lettres sont données dans l'ordre alphabétique ?

Exercice 5. desserts

```
Dessert = {
'gateau chocolat' : {'chocolat', 'oeuf', 'farine', 'sucre', 'beurre'},
'gateau yaourt' : {'yaourt', 'oeuf', 'farine', 'sucre'},
'crepes' : {'oeuf', 'farine', 'lait'},
'quatre-quarts' : {'oeuf', 'farine', 'beurre', 'sucre'},
'kouign amann' : {'farine', 'beurre', 'sucre'}
}
```