LES DICTIONNAIRES

Les éléments d'une liste ou d'un tuple sont ordonnés et on accède à un élément grâce à sa position en utilisant un numéro qu'on appelle l'indice de l'élément. Un dictionnaire en python est une sorte de liste mais au lieu d'utiliser des index, on utilise des clés alphanumériques. On peut faire l'analogie avec un dictionnaire de français où on accède à une définition avec un mot. Contrairement aux listes qui sont délimitées par des crochets, on utilise des accolades pour les dictionnaires.

I. Création d'un dictionnaire vide.

1.1 Création d'un dictionnaire vide.

Pour initialiser un dictionnaire , on utile la syntaxe suivante :

```
>>> dicto = {} # ou a = dict()
```

1.2 Création d'un dictionnaire non vide.

On peut assimiler un dictionnaire à un **ensemble de couples (clé, valeur)**, les clés étant le plus souvent des nombres ou des chaînes de caractères.

```
>>> dicto1 = {'nom' : 'KELLER', 'prénom' : 'Stéphane'}
```

II. Ajout de valeurs dans un dictionnaire.

Pour ajouter des valeurs à un dictionnaire vide ou non, il faut indiquer une **nouvelle clé** ainsi qu'une **valeur** :

```
>>> dicto2 = {} #création d'un dictionnaire vide
>>> dicto2['nom'] = 'KELLER'
>>> dicto2['prénom'] = 'Stéphane'
>>> dicto2
{'nom': 'KELLER', 'prénom' : 'Stéphane'}
>>> dicto2['âge'] = 40
>>> dicto2
{'âge' :  40, 'nom' : 'KELLER', 'prénom' :  'Stéphane'}
```

III. Récupération d'une valeur dans un dictionnaire.

La méthode **get** vous permet de récupérer une valeur dans un dictionnaire et si la clé est introuvable, vous pouvez donner une valeur à retourner par défaut :

```
>>> dicto2.get('âge')
40
>>> dicto1.get('âge')

>>> dicto2.get('âge', 'âge inconnu') # 'âge inconnu' est la valeur à retourner par défaut
40
>>> dicto1.get('âge', 'âge inconnu')
' âge inconnu'
```

IV. Vérification de l'existence d'une clé dans un dictionnaire.

On peut utiliser l'opérateur **in** pour vérifier la présence d'une **clé** que l'on cherche :

```
>>> 'âge' in dicto2
True
>>> 'âge' in dicto1
False
>>>
```

Attention, cela ne fonctionne qu'avec une clé et non une valeur.

```
>>> 40 in dicto2
False
```

V. Suppression d'une clé dans un dictionnaire.

Il est possible de supprimer une entrée en indiquant sa clé, comme pour les listes :

```
>>> del dicto1['nom']
>>> dicto1
{'prénom': 'Stéphane'}
```

VI. Parcours et affichage d'un dictionnaire.

6.1 Affichage des clés.

On utilise **keys()**.

```
for cle in dicto2.keys():
    print('Clé =', cle)

>>>
Clé = âge
Clé = nom
Clé = prénom
>>>
```

6.2 Affichage des valeurs.

On utilise **values()**.

```
for valeur in dicto2.values():
    print('Valeur =', valeur)
```

```
>>>
Valeur = 40
Valeur = KELLER
Valeur = Stéphane
>>>
```

6.3 Affichage des clés et des valeurs.

On utilise **items()**.

```
for item in dicto2.items():
   print('Couple =', item)
```

```
>>>
Couple = ('âge', 40)
Couple = ('nom', 'KELLER')
Couple = ('prénom', 'Stéphane')
>>>
```

```
for cle, valeur in dicto2.items():
   print('La clé', cle, 'correspond à la valeur', valeur)
```

```
>>>
La clé âge correspond à la valeur 40
La clé nom correspond à la valeur KELLER
La clé prénom correspond à la valeur Stéphane
>>>
```

6.4 Affichage des valeurs à partir des clés d'un dictionnaire.

```
graphe = {'A' : ('B', 'D', 'E'), 'B' : ('A', 'C'), 'C' : ('B', 'D'), 'D' : ('A', 'C', 'E'), 'E' : ('A', 'D', 'F',
'G'), 'F' : ('E', 'G'), 'G' : ('E', 'F', 'H'), 'H' : ('G')}

#Affichage des valeurs à partir d'une clé saisie
cle = input('Entrer une clé : ')
print('Clé =', cle)
for valeur in graphe[cle]:
   print('Valeurs =', valeur)
```

```
>>>
Clé = E
Valeurs = A
Valeurs = D
Valeurs = F
Valeurs = G
>>>
```

```
#Affichage des valeurs à partir des clés à l'aide d'une boucle
for cle in graphe.keys():
   print('Clé =', cle)
   for valeur in graphe[cle]:
      print('Valeurs =', valeur)
```

```
>>>
Clé = E
Valeurs = A
```

```
Valeurs = D
Valeurs = F
Valeurs = G
...
Clé = H
Valeurs = G
>>>
```

VII. Modification d'une ou plusieurs valeurs dans un dictionnaire.

7.1 Avec une boucle for et item().

```
dicto3 = {'nom' : 'KELLER', 'prénom' : 'Stéphane', 'âge' :  40}
for cle, valeur in dicto3.items():
    if cle == 'âge':
        dicto3[cle] = 50
```

7.2 Utilisation de update().

```
>>> dicto3 = {'nom' : 'KELLER', 'prénom' : 'Stéphane', 'âge' :  40}
>>> dicto3.update({'âge' : 50})
>>> dicto3
{'nom' : 'KELLER', 'prénom' : 'Stéphane', 'âge' :  50}
```

VIII. Copie indépendante d'un dictionnaire.

Comme pour toute variable, vous ne pouvez pas copier un dictionnaire en faisant dicto1 = dicto2 car Python effectue une *copie par référence et non une copie par recopie.*

```
dicto4 = {'nom' : 'KELLER', 'prénom' : 'Stéphane', 'âge' : 40}
print('dicto4 =', dicto4)
dicto5 = dicto4
dicto4['âge'] = 20
print('dicto4 =', dicto4)
print('dicto5 =', dicto5)

>>>
dicto4 = {'nom': 'KELLER', 'prénom': 'Stéphane', 'âge': 40}
dicto4 = {'nom': 'KELLER', 'prénom': 'Stéphane', 'âge': 20}
dicto5 = {'nom': 'KELLER', 'prénom': 'Stéphane', 'âge': 20}
>>>
```

Pour effectuer une copie indépendante, il faut effectuer une *copie par recopie* avec le module **copy**.

```
from copy import copy
dicto6 = {'nom' : 'KELLER', 'prénom' : 'Stéphane', 'âge' : 40}
print('dicto6 =', dicto6)
dicto7 = dicto6.copy()
dicto6['âge'] = 20
```

```
print('dicto6 =', dicto6)
print('dicto7 =', dicto7)
```

```
>>>
dicto6 = {'nom': 'KELLER', 'prénom': 'Stéphane', 'âge': 40}
dicto6 = {'nom': 'KELLER', 'prénom': 'Stéphane', 'âge': 20}
dicto7 = {'nom': 'KELLER', 'prénom': 'Stéphane', 'âge': 40}
>>>
```

IX. Fusionner des dictionnaires.

Il faut utiliser **update()**.

```
dicto8 = {'nom': 'KELLER', 'prénom': 'Stéphane'}
dicto9 = {'âge': 20}
print('dicto8 =', dicto8)
dicto8.update(dicto9)
print('dicto8 =', dicto8)

>>>
dicto8 = {'nom': 'KELLER', 'prénom': 'Stéphane'}
dicto8 = {'nom': 'KELLER', 'prénom': 'Stéphane', 'âge': 20}
>>>
```

LES DICTIONNAIRES - TESTS

I. Écrire une fonction occurrences(txt) renvoyant un dictionnaire avec le nombre d'occurrences de chaque caractère dans un texte. Ouvrir et utiliser le fichier texte.txt.
II. Écrire une fonction li_to_dict(li) qui convertit une liste en dictionnaire. Les éléments de la liste deviennent les clés du dictionnaire, et leurs valeurs correspondent à leur nombre d'occurrences dans la liste.
III. Écrire une fonction fusionnant deux dictionnaires de la manière suivante : si dict1 = {'Alban' : 2, 'Clément' : 1} et dict2 = {'Alban' : 1, 'Clément' : 3, 'Stéphane' : 2}, alors fusion(dict1, dict2) renverra {'Alban' : 3, 'Clément' : 4, 'Stéphane' : 2}.
IV. L'ARN contient le codage des protéines, ou des chaines d'acides aminés. La séquence AUG, par exemple, correspond à la méthionine, noté M. Le dictionnaire ci-dessous donne les correspondances entre les séquences d'ARN, à prendre par groupes de trois nucléotides, et les acides aminés.
dicto_gen = {'UUU' : 'F', 'UUC' : 'F', 'UUG' : 'L', 'UUA' : 'L', 'UCU' : 'S', 'UCC' : 'S', 'UCG' : 'S', 'UCA' : 'S', 'UAU' : 'Y', 'UAC' : 'Y', 'UAG' : 'STOP', 'UAA' : 'STOP', 'UGU' : 'C', 'UGC' : 'C', 'UGG' : 'W', 'UGA' : 'STOP', 'CUU' : 'L', 'CUC' : 'L', 'CUG' : 'L', 'CUA' : 'L', 'CCU' : 'P', 'CCC' : 'P', 'CCG' : 'P', 'CCA' : 'P', 'CGU' : 'R', 'CGC' : 'R', 'CGG' : 'R', 'CGA' : 'R', 'CAU' : 'H', 'CAC' : 'H', 'CAG' : 'Q', 'CAA' : 'Q', 'ACU' : 'T', 'ACC' : 'T', 'ACG' : 'T', 'ACA' : 'T', 'AUG' : 'M', 'AUU' : 'I', 'AUC' : 'I', 'AUA' : 'I', 'AAU' : 'N', 'AAC' : 'N', 'AAG' : 'K', 'AAA' : 'K', 'AGU' : 'S', 'AGC' : 'S', 'AGG' : 'R', 'AGA' : 'R', 'GUU' : 'V', 'GUC' : 'V', 'GUG' : 'V', 'GUA' : 'V', 'GCU' : 'A', 'GCC' : 'A', 'GCG' : 'A', 'GCA' : 'A', 'GGU' : 'G', 'GGC' : 'G', 'GGG' : 'G', 'GGA' : 'G', 'GAU' : 'D', 'GAC' : 'D', 'GAG' : 'E', 'GAA' : 'E'}
Écrire une fonction traduction(arn) qui traduit une chaine d'ARN en protéine. On suppose que la longueur de la chaine d'ARN est un multiple de trois. Ainsi, traduction('UUCAGUGGG') renverra 'FSG'.