

1. On veut créer un dictionnaire associant au nom de chaque mois son nombre de jours. Pourquoi la déclaration suivante est-elle fautive informatiquement parlant (données exactes)?

```
dicomois = {janvier : 31 , fevrier : 28 , mars : 31 , avril : 30 , mai : 31 , juin : 30 , juillet : 31 , aout : 31 , septembre : 30 , octobre : 31 , novembre : 30 , decembre : 31}.
```

2. Le dictionnaire étant correctement défini, compléter l'instruction suivante :

```
print('En mai il y a ' , , 'jours.').
```

3. La tradition recommande de ne manger des huîtres que pendant les mois "en r", c'est-à-dire qui ont un "r" dans leur nom (la raison en est sans doute que les mois sans "r" sont les plus chauds, ce qui complique le transport et la conservation des huîtres sorties de l'eau). Définir en compréhension la liste des mois où il est déconseillé de manger des huîtres.

4. Compléter le dictionnaire avec le mois "fevrierbis" qui a 29 jours, pour les années bissextiles.

5. À partir de dicomois, créer la liste L1 des noms de mois et la liste L2 des nombres de jours.

6. La fonction **sum** s'utilise ainsi : *sum( expression dépendant de k for k in itérable if condition )*. À partir de L2, calculer la durée moyenne de l'année terrestre, en jours, sachant qu'il y a 97 années bissextiles dans quatre siècles.

7. En utilisant les commandes intégrées de Python pertinentes, écrire une instruction donnant à partir de ces listes le mois plus court.

8. En utilisant une liste auxiliaire à définir, écrire une instruction donnant le nombre de lettres du nom de mois le plus long (on ne demande pas quel est ce mois).

9. Compléter la fonction *minimum(L)*, 10. Écrire une fonction *compte(x,L)* qui renvoie équivalente à la fonction intégrée *min(L)* : le nombre d'occurrences du nombre *x* dans la liste L (donc équivalente à *L.count(x)*).

```
def minimum(L) :  
  
    for e in L :  
        if e < m :  
  
    return
```

## Correction

1. Les noms de mois doivent être **déclarés comme des chaînes de caractères** : 'janvier' ou "janvier", etc. Sinon, ils sont considérés comme des identifiants et l'interpréteur Python essaie de les évaluer, ce qui provoque un message d'erreur du type suivant :

**NameError : name 'janvier' is not defined.**

2. 

```
print('En mai il y a',mois['mai'],'jours.')
```

3. 

```
Lsans = [mois for mois in dicomois if 'r' not in mois]
```

4. 

```
dicomois['fevrierbis'] = 29
```

5. 

```
L1 = [nom for nom in dicomois] ; L2 = [dicomois[nom] for nom in dicomois]
```

6. 

```
anneemoy = ( 303*sum(n for n in L2 if n !=29)
              + 97*sum(n for n in L2 if n !=28) ) / 400
```

7. 

```
pluscourt = L1[ L2.index( min(L2) ) ]
```

8. 

```
nblettresmax = max( [ len(nom) for nom in L1 ] )
```

9. 

```
def minimum(L):
    m = L[0]
    for e in L:
        if e < m:
            m = e
    return m
```

10. 

```
def compte(x,L):
    nb = 0
    for e in L:
        if e == x:
            nb += 1
    return nb
```

- = FIN = -