

Exercices-chaines

November 22, 2019

1 Exercices sur les chaînes de caractères

```
[11]: # Liste des décimales de pi
liste_dec = [1,4,1,5,9,2,6,5,3,5,8,9,7,9,3,2,3,8,4,6,2,6,4,\
            3,3,8,3,2,7,9,5,0,2,8,8,4,1,9,7,1,6,9,3,9,9,3,\
            7,5,1,0,5,8,2,0,9,7,4,9,4,4,5,9,2,3,0,7,8,1,6,\
            4,0,6,2,8,6,2,0,8,9,9,8,6,2,8,0,3,4,8,2,5,3,4,2]

no_dept_aquitaine = [16, 17, 19, 23, 24, 33, 40, 46, 64, 79, 86, 87]

noms_dept_aquitaine = ["charente", "charente-maritime", "correze", "creuse", \
                       "dordogne", "gironde", "landes", "lot-et-garonne", \
                       "pyrenees-atlantiques", "deux-sevres", "vienne", "haute-vienne"]

texte1 = "Le narrateur rencontre dans un bus un jeune homme au long cou, coiffé d'un_
→chapeau mou orné d'une tresse tenant lieu de ruban. Ce quidam échange quelques mots_
→assez vifs avec un autre voyageur, puis va s'asseoir à une autre place. Un peu plus_
→tard, le narrateur revoit le même jeune homme cour de Rome devant la gare_
→Saint-Lazare en train de discuter avec un ami qui lui conseille d'ajuster (ou_
→d'ajouter) un bouton de son pardessus."

stop_words = ["le", "la", "un", "une", "un"]
```

1.1 Afficher les départements d'acquitaine, un par ligne, et commençant par une majuscule.

```
[33]: for d in noms_dept_aquitaine:
       print(d.capitalize())
```

```
Charente
Charente-maritime
Correze
Creuse
Dordogne
Gironde
Landes
Lot-et-garonne
Pyrenees-atlantiques
Deux-sevres
Vienne
Haute-vienne
```

```
[34]: # Une autre méthode passant par la création d'une liste
newlist = []
for d in noms_dept_aquitaine:
    newlist.append(d.capitalize())

for d in newlist:
    print(d)
```

Charente
 Charente-maritime
 Correze
 Creuse
 Dordogne
 Gironde
 Landes
 Lot-et-garonne
 Pyrenees-atlantiques
 Deux-sevres
 Vienne
 Haute-vienne

2 Afficher les départements d'aquitaine, un par ligne, de sorte que tous les mots commencent par une majuscule.

```
[4]: for d in noms_dept_aquitaine:
    l = d.split('-') # on découpe sur les tirets
    if len(l) == 1: # s'il n'y a pas de tirets
        print(d.capitalize()) # on affiche le mot avec une majuscule
    else: # sinon, il y a pls mots séparés par "-"
        for m in l[:-1]: # pour chacun d'entre eux sauf le dernier
            print(m.capitalize(), end="-") # on l'imprime avec une maj. puis '-'
        print(l[-1].capitalize()) # dernier mot de la liste: pas de '-'
```

Charente
 Charente-Maritime
 Correze
 Creuse
 Dordogne
 Gironde
 Landes
 Lot-Et-Garonne
 Pyrenees-Atlantiques
 Deux-Sevres
 Vienne
 Haute-Vienne

2.1 Afficher le texte1 entièrement en majuscules.

```
[5]: print(texte1.upper())
```

LE NARRATEUR RENCONTRE DANS UN BUS UN JEUNE HOMME AU LONG COU, COIFFÉ D'UN CHAPEAU MOU ORNÉ D'UNE TRESSE TENANT LIEU DE RUBAN. CE QUIDAM ÉCHANGE QUELQUES

MOTS ASSEZ VIFS AVEC UN AUTRE VOYAGEUR, PUIS VA S'ASSEOIR À UNE AUTRE PLACE. UN PEU PLUS TARD, LE NARRATEUR REVOIT LE MÊME JEUNE HOMME COUR DE ROME DEVANT LA GARE SAINT-LAZARE EN TRAIN DE DISCUTER AVEC UN AMI QUI LUI CONSEILLE D'AJUSTER (OU D'AJOUTER) UN BOUTON DE SON PARDESSUS.

2.2 Afficher les départements d'acquitaine, un par ligne, en allant du plus court au plus long.

```
[14]: # on cherche le plus petit, on l'affiche, et on l'enlève de la liste
# D'abord on définit une fonction qui cherche le plus petit et le renvoie
def plus_petit(l):
    min = l[0]
    for m in l:
        if len(m) < len(min):
            min = m
    return min

# on travaille sur une copie de la liste pour ne pas modifier la liste initiale
liste = noms_dept_aquitaine.copy()      # attention pour avoir une nouvelle liste on
→utilise .copy()

while len(liste) != 0:
    x = plus_petit(liste)
    print(x)
    liste.remove(x)
```

```
creuse
landes
vienne
correze
gironde
charente
dordogne
deux-sevres
haute-vienne
lot-et-garonne
charente-maritime
pyrenees-atlantiques
```

2.3 Afficher les départements d'acquitaine, un par ligne, de façon que tous les mots soient alignés à droite.

```
[15]: for m in noms_dept_aquitaine:
    nb_blancs = 80-len(m)
    for i in range(nb_blancs):
        print(" ",end="")
    print(m)
```

```
charente
charente-maritime
correze
creuse
dordogne
```

```
gironde
landes
lot-et-garonne
pyrenees-atlantiques
deux-sevres
vienne
haute-vienne
```

2.4 En utilisant la fonction `s.isupper()` afficher les indices de tous les débuts de phrase dans `texte1`.

```
[17]: # première tentative: on affiche les indices de chaque lettre en majuscule
c = 0
for l in texte1:
    if l.isupper():
        print(l,c)
    c += 1
```

```
L 0
C 127
U 231
R 297
S 317
L 323
```

```
[16]: # même tentative, mais on affiche quelques lettres pour vérifier que c'est satisfaisant
for i in range(len(texte1)):
    if texte1[i].isupper():
        print(texte1[i:i+10],i)
```

```
Le narrate 0
Ce quidam 127
Un peu plu 231
Rome devan 297
Saint-Laza 317
Lazare en 323
```

```
[18]: # les options précédentes affichaient aussi les noms propres.
# On ajoute la contrainte que soit la majuscule est au début du texte,
# soit elle est précédée d'un point suivi d'un autre caractère (un blanc)
for i in range(len(texte1)):
    if texte1[i].isupper():
        if i == 0 or texte1[i-2] == ".":
            print(texte1[i:i+10],i)
```

```
Le narrate 0
Ce quidam 127
Un peu plu 231
```

2.5 Afficher tous les mots du texte1 qui contiennent le motif "an".

```
[20]: texte_decoupe = texte1.split()
      for m in texte_decoupe:
          if m.find("an") != -1:
              print(m)
```

dans
tenant
ruban.
échange
devant

2.6 Afficher tous les mots du texte1 qui se terminent par "e".

```
[21]: for m in texte_decoupe:
      if m.endswith("e"):
          print(m)
```

Le
rencontre
jeune
homme
d'une
tresse
de
Ce
échange
autre
une
autre
le
le
même
jeune
homme
de
Rome
gare
Saint-Lazare
de
conseille
de

2.7 Afficher le texte1 dans lequel les "stop-words" sont remplacés par "xxx".

```
[23]: for m in texte_decoupe:
      if m in stop_words:
          print("XXX", end=" ")
      else:
          print(m, end=" ")
```

Le narrateur rencontre dans XXX bus XXX jeune homme au long cou, coiffé d'un

chapeau mou orné d'une tresse tenant lieu de ruban. Ce quidam échange quelques mots assez vifs avec XXX autre voyageur, puis va s'asseoir à XXX autre place. Un peu plus tard, XXX narrateur revoit XXX même jeune homme cour de Rome devant XXX gare Saint-Lazare en train de discuter avec XXX ami qui lui conseille d'ajuster (ou d'ajouter) XXX bouton de son pardessus.

2.8 Afficher tous les signes de ponctuation qui apparaissent dans texte1.

```
[26]: for lettre in texte1:
        if not lettre.isalpha() and lettre != ' ':
            print(lettre, end=" ")
    print()
```

, ' ' . , ' . , - ' (') .

2.9 Afficher le texte1 en remplaçant tous les blancs par "/".

```
[28]: for lettre in texte1:
        if lettre == ' ':
            print('/', end='')
        else:
            print(lettre, end='')
    print()
```

Le/narrateur/rencontre/dans/un/bus/un/jeune/homme/au/long/cou,/coiffé/d'un/chapeau/mou/orné/d'une/tresse/tenant/lieu/de/ruban./Ce/quidam/échange/quelques/mots/assez/vifs/avec/un/autre/voyageur,/puis/va/s'asseoir/à/une/autre/place./Un/peu/plus/tard,/le/narrateur/revoit/le/même/jeune/homme/cour/de/Rome/devant/la/gare/Saint-Lazare/en/train/de/discuter/avec/un/ami/qui/lui/conseille/d'ajuster/(ou/d'ajouter)/un/bouton/de/son/pardessus.

2.10 Définir une fonction ayant le même effet que startswith() (la fonction aura deux paramètres, la chaîne et le préfixe candidat)

```
[31]: def commence_avec(chaine,prefixe):
        for i in range(len(prefixe)):
            if chaine[i] != prefixe[i]:
                return False
        return True

# le code suivant permet de faire une série de tests
essais = ["maison", "vide-grenier", "Toulouse", "mais non", "mamie"]
prefixes = ["m", "ma", "mais", "To"]
for c in essais:
    for p in prefixes:
        if commence_avec(c,p):
            print ("La chaîne '%s' commence avec '%s' % (c,p))
        else:
            print ("La chaîne '%s' ne commence pas par '%s' % (c,p))
```

La chaîne 'maison' commence avec m

La chaîne 'maison' commence avec ma

La chaîne 'maison' commence avec mais
La chaîne 'maison' ne commence pas par To
La chaîne 'vide-grenier' ne commence pas par m
La chaîne 'vide-grenier' ne commence pas par ma
La chaîne 'vide-grenier' ne commence pas par mais
La chaîne 'vide-grenier' ne commence pas par To
La chaîne 'Toulouse' ne commence pas par m
La chaîne 'Toulouse' ne commence pas par ma
La chaîne 'Toulouse' ne commence pas par mais
La chaîne 'Toulouse' commence avec To
La chaîne 'mais non' commence avec m
La chaîne 'mais non' commence avec ma
La chaîne 'mais non' commence avec mais
La chaîne 'mais non' ne commence pas par To
La chaîne 'mamie' commence avec m
La chaîne 'mamie' commence avec ma
La chaîne 'mamie' ne commence pas par mais
La chaîne 'mamie' ne commence pas par To

2.11 Afficher les dix mots les plus fréquents de texte1. [Avancé: il est recommandé d'utiliser un dictionnaire]

Voir cours sur les dictionnaires