

Quizz_2020_6

February 29, 2020

Ecrire une fonction `ngrammes()` qui prend en entrée deux paramètres: une liste de mots (par exemple une phrase qui a déjà été découpée par `split()`), et un entier strictement positif. La fonction doit *afficher*, un par ligne, tous les n -grammes de la liste, (les n -grammes sont les sous-listes contiguës de longueur n).

En cas de difficulté pour répondre directement, vous pouvez commencer par proposer une fonction n'ayant qu'un seul paramètre, une liste de mots, qui affiche à l'écran un par ligne tous les trigrammes de la liste. (trigramme = 3-gramme.)

```
[2]: def trigrammes(l):
      for i in range(len(l)-2):
          print(l[i],l[i+1],l[i+2])

      def ngrammes(l,n):
          for i in range(len(l)-n+1):
              for j in range(n):
                  print(l[i+j], end=' ')
              print()
```

```
[4]: test = "Mon beau navire ô ma mémoire avons nous assez navigué".split()
      trigrammes(test)
      print()
      ngrammes(test,4)
```

```
Mon beau navire
beau navire ô
navire ô ma
ô ma mémoire
ma mémoire avons
mémoire avons nous
avons nous assez
nous assez navigué
```

```
Mon beau navire ô
beau navire ô ma
navire ô ma mémoire
ô ma mémoire avons
ma mémoire avons nous
```

mémoire avons nous assez
avons nous assez navigué

```
[6]: # Variantes inspirées par certaines copies:

# Construction de la ligne par concaténation puis affichage
def ngrammes_conc(l,n):
    for i in range(len(l)-n+1):
        ligne = ""
        for j in range(n):
            ligne += l[i+j] + ' '
        print(ligne)

# Utilisation des slices pour créer une sous-liste
def ngrammes_slice(l,n):
    for i in range(len(l)-(n-1)):
        for w in l[i:i+n]:
            print(w, end = ' ')
        print()

# Variante qui ne produit pas exactement l'affichage demandé:
def ngrammes_slice_lst(l,n):
    for i in range(len(l)-(n-1)):
        print(l[i:i+n])
```