

# Mélangeur de mots

## Travaux Pratiques 06 du module 05 - Les variables complexes

Avant de démarrer ce TP, il convient d'avoir suivi les vidéos des modules 1 à 6 de ce cours.

### Durée estimée

Environ 2 heures

## Énoncé

Écrire un programme qui mélange les lettres (sauf la première et la dernière) de chaque mot d'une phrase, et qui affiche la phrase avec les mots mélangés.

### Exemple

- Selon une étude de l'Université de Cambridge, l'ordre des lettres dans un mot n'a pas d'importance, la seule chose importante est que la première et la dernière lettre restent à la bonne place. Le reste peut être dans un désordre total et vous pouvez toujours lire sans problème. C'est parce que le cerveau humain ne lit pas chaque lettre elle-même, mais le mot comme un tout.

### Explication

Une étude de l'université de Cambridge a montré que l'on peut sans problème lire un texte dont les lettres sont dans le désordre pour peu que la première et la dernière lettre de chaque mot restent à la bonne place. Ceci montre que le cerveau ne lit pas toutes les lettres, mais prend le mot comme un tout. La preuve : avouez que vous n'avez pas eu de mal à lire ce texte.

### Conseils

Procéder étapes par étapes, pour mélanger d'abord un mot, puis une phrase.

Utiliser des constantes pour tester toujours la même phrase (déjà normalisée) avant de tester une phrase saisie, afin d'éprouver votre mécanisme. Ne cherchez pas à optimiser dès le début.

### Objectif / Niveau

1. Essentiel : Afficher un mot (de plus de 6 lettres) dont les lettres ont été mélangées.
2. Attendu : Afficher une phrase, dont les lettres de chaque mot ont été mélangées.
3. Avancé : Optimiser l'écriture de votre code pour gérer les apostrophes et la ponctuation.

## Solution

Des propositions de solution pour ce TP sont placées dans les éléments en téléchargement liés à ce module.

## Astuces

Vous pouvez vous intéresser aux fonctions de manipulations de chaînes de caractères « split » et « join ».

Utiliser la fonction « randrange » du package « random » pour obtenir une valeur aléatoire comprise entre zéro et un maximum (exclu) indiqué comme ci-dessous par exemple :

```
from random import randrange
print("randrange(9) = ", randrange(9)) # Soit un nombre entier compris entre 0 et 8

# Utilisation d'une lambda :
nb_aleatoire = lambda max_inclu: randrange(max_inclu + 1) # rappel : le maximum est exclu
print("nb_aleatoire(9) = ", nb_aleatoire(9)) # Soit un nombre entier compris entre 0 et 9

# Retourne un nombre entier aléatoire compris entre un mini et un maxi :
nb_aleatoire2 = lambda mini, maxi: randrange(mini, maxi + 1)
```