
MPSI - ITC - TD 5

TRANCHES ET LISTES DE LISTES

Exercice 1 (Création d'identifiant)

Dans certains services en ligne, tels que Prépas Plus, l'identifiant d'un·e élève est créé en prenant la première lettre de son prénom, suivie des premières lettres de son nom de famille.

1. Codez une fonction `identifiant(prenom, nom)` qui prend en entrée deux chaînes de caractères, et qui renvoie la chaîne constituée de la première lettre du prénom, suivi des 5 premières lettres du nom de famille.

Votre fonction utilisera au moins une tranche.

Vous pouvez supposer que le nom est de taille 5 ou plus. Si ce n'est pas le cas, votre fonction a le droit de lever une erreur.

Si vous le souhaitez, vous pouvez utiliser la méthode `upper()` des chaînes de caractères pour passer toutes les lettres d'une chaîne en majuscules.

Exercice 2 (Alias, copies superficielles et copies en profondeur)

Exécutez chacun de ces codes, observez ce qu'ils affichent, et prenez le temps d'être sûr·e que vous comprenez ce qui se passe.

Code 1 :

```
liste = [3, 5, 7]
alias = liste
alias[1] = 14

print("À la fin, liste contient ", liste)
print("À la fin, alias contient ", alias)
```

Code 2 :

```
liste = [3, 5, 7]
copie = liste.copy()
copie[1] = 14

print("À la fin, liste contient ", liste)
print("À la fin, copie contient ", copie)
```

Code 3 :

```
grille = [[0] * 3 for i in range(3)]
alias = grille
alias[0] = [1, 2, 3]

print("À la fin, grille contient ", grille)
print("À la fin, alias contient ", alias)
```

Code 4 :

```
grille = [[0] * 3 for i in range(3)]
copie_superficielle = grille.copy()
copie_superficielle[0] = [1, 2, 3]
copie_superficielle[1][0] = 4

print("À la fin, grille contient ", grille)
print("À la fin, copie_superficielle contient ", copie_superficielle)
```

Code 5 :

```
from copy import deepcopy

grille = [[0] * 3 for i in range(3)]
copie_en_profondeur = deepcopy(grille)
copie_en_profondeur[0] = [1, 2, 3]
copie_en_profondeur[1][0] = 4

print("À la fin, grille contient ", grille)
print("À la fin, copie_en_profondeur contient ", copie_en_profondeur)
```

Exercice 3 (Les mots mêlés)

Dans cet exercice on s'intéresse à la création de grilles de « mots mêlés » tel qu'on peut en trouver dans des journaux et des magazines.

	v	y	y	m	i
o		q	a	u	n
r		h	y	h	f
d		a	l	g	o
i		p	c	v	l

Une grille de mots mêlés.

1. Codez une fonction `grille_de_a` qui prend en entrée deux entiers n et m et qui renvoie une grille avec n lignes et m colonnes, avec dans chaque case la lettre 'a'.
On utilisera pour représenter la grille, une liste de listes. Chacune des listes internes représentera une ligne de la grille. Par exemple, `grille_de_a(2, 3)` devra renvoyer `[ ['a', 'a', 'a'], ['a', 'a', 'a'] ]` tandis que `grille_de_a(3, 2)` devra renvoyer `[ ['a', 'a'], ['a', 'a'], ['a', 'a'] ]`.
Attention, on veillera à ce que les différentes lignes ne soient pas des alias les unes des autres. En particulier, la liste externe ne sera pas obtenue à l'aide d'une multiplication de listes (mais, les lettres n'étant pas mutables, chaque ligne peut, quand à elle, être obtenue avec une multiplication de liste).
2. Coder une fonction `alphabet` qui ne prend pas d'arguments et qui renvoie une grille de taille 5×5 contenant les lettres 'a' jusqu'à 'y'.
La première ligne de la grille doit correspondre aux lettres 'a' à 'e'.
On utilisera la fonction `chr` qui prend en argument un entier et renvoie le caractère associé dans le code ascii. Par exemple, `chr(97)` renvoie 'a', `chr(98)` renvoie 'b', ..., et `chr(122)` renvoie 'z'. L'inverse de la fonction `chr` est la fonction `ord`.
3. Codez une fonction `insérer_horizontalement(grille, i, j, mot)` qui modifie la grille donnée en entrée pour que à la ligne d'indice i , sur les lettres d'indices j et suivants, apparaisse le mot donné.
4. Comment généraliser la fonction précédente pour pouvoir choisir la direction dans laquelle est inséré le mot ?