

FIGURES ALPHANUMÉRIQUES

Rappel : multiplication de collection

Vous pouvez multiplier n'importe quelle collection `c` (une liste, un tuple, une chaîne de caractères) par n'importe quel entier `n` pour obtenir une collection du même type de `c`, de taille `len(c)*n` et qui contient les éléments de `c` répétés `n` fois.

Par exemple, l'expression python `'ab' * 3` s'évalue vers la chaîne de caractères `'ababab'`.

Attention, ce type de multiplication ne commute pas. Il faut bien mettre la collection à gauche du signe multiplier, et le nombre à droite, sinon il y aura une erreur de type.

Exercice 1 (Questions de triangle)

Cet exercice est à la fois simple mais très important pour mieux comprendre ce qui arrive plus tard dans l'année (notamment le parcours de graphe). Vous allez étudier deux fonctions récursives et il vous faut être capable de déterminer la figure alphanumérique obtenue en fonction de la **position** de l'appel récursif.

```
def triangle1(n) :  
    print('* '*n)  
    if n > 0 :  
        triangle1(n-1)  
  
def triangle2(n) :  
    if n>0 :  
        triangle2(n-1)  
    print('* '*n)
```

1. Comparer les deux fonctions `triangle1` et `triangle2`. Sans les exécuter, pensez-vous qu'elles afficheront les mêmes lignes ? Dans le même ordre ?
2. Prédire la figure obtenue pour chaque fonction.
3. Tester vos prédictions !
4. Rédiger une conclusion.

Exercice 2 (Figures)

```

*      *      *
*      *
*
*
*      *
*      *      *

```

Figure 1

```

*
*   *
*
*   *   *
*
*   *
*
*   *

```

Figure 2

[illegible]

Figure 3

Figure 4

Reproduire les figures ci-dessus. Pour la figure 4, vous définirez une fonction qui permet de choisir les 2 caractères utilisés pour la figure ainsi que la taille `n` de la figure. Si vous avez le temps et de l'imagination, proposez vos propres figures !