

TD du 09/03/2022

Résoudre une grille de SUDOKU

4	9		2				7	
5		3						2
		2	3		4		8	6
			1	5	8			
						3	6	
		1	7	6				
		8	4					
	2	5					7	
6					3	7	9	

Données d'entrée :

Elles se présente sous la forme d'un fichier CSV (*fichier avec champs séparés par des ' ; '*)

```
4;9;.;2;.;.;.;7;.
5;.;3;.;.;.;.;2
.;.;2;3;.;4;.;8;6
.;.;.;1;5;8;.;.;.
.;.;.;.;.;3;6;.
.;.;1;7;6;.;.;.;.
.;.;8;4;.;.;.;.;.
.;2;5;.;.;.;7;.;.
6;.;.;.;3;7;9;.;.
```

A faire :

- 1 – Définir les règles à imposer sur la grille pour la résolution du problème
- 2 – Définir la structure des données afin de faciliter l'exécution des règles
- 3 – Définir une méthode de résolution
- 4 – Coder !!!!!!!

Rappel de la fonction **range()** de Python :

```
range (start, stop, inc)
```

Permet de créer un générateur de nombre allant de **start** à **stop-1** avec un pas de **inc**.
(l'incrément est optionnel, par défaut c'est 1)

Exemple :

```
for i in range(0,3) :  
    print(i)
```

donnera le résultat suivant :

```
> 0  
> 1  
> 2
```

Exercice :

Créer une fonction qui permet de retourner la liste des valeurs de chaque bloc en fonction de son numéro sachant que les blocs sont attribués ainsi:

Numéro Bloc	X début	X fin	Y début	Y fin
0	0	3	0	3
1	0	3	3	6
2	0	3	6	9
3	3	6	0	3
4	3	6	3	6
5	3	6	6	9
6	6	9	0	3
7	6	9	3	6
8	6	9	6	9

Objectif si on reçoit « 1 » obtenir : range(0,3) pour x et range(3,6) pour y par exemple.

```
for x in range(0,3) :  
    for y in range(3,6) :  
        liste_valeurs_bloc_choisi.append(sudoku_grille[x][y])
```

déterminer **expression** pour le cas général :

```
for x in range(expression) :  
    for y in range(3,6) :  
        liste_valeurs_bloc_choisi.append(sudoku_grille[x][y])
```