

Triangle de Pascal

L'objectif de cette activité est de créer une fonction Python permettant de générer un triangle de Pascal de taille donnée. On donne ci-dessous le triangle de Pascal de taille 5.

	0	1	2	3	4	5
0	1					
1	1	1				
2	1	2	1			
3	1	3	3			
4	1	4	6	4	1	
5	1	5	10	10	5	1

Les lignes et les colonnes sont numérotées à partir de 0 (tout comme les listes en Python). Le principe de construction est le suivant :

- les lignes 0 et 1 sont données et constituées de 1 ;
- pour tout $j \geq 2$, la ligne i est constituée de $i + 1$ nombres ; elle commence et se termine par 1 et, pour tout j compris entre 1 et $i - 1$, le nombre de la ligne i et de la colonne j s'obtient en ajoutant les deux nombres de la ligne $i - 1$ situés aux colonnes $j - 1$ et j .

Par exemple, le nombre situé à la ligne 5 et à la colonne 3, est obtenu comme la somme des deux nombres situés à la ligne 4 et aux colonnes 2 et 3.

1. Le triangle de Pascal en Python sera une liste de liste.
Ainsi, le triangle de Pascal de taille 1 est le liste $T = [[1], [1, 1]]$.
(a) Pour la liste T précédente, donner $T[0]$ et $T[1]$.
(b) Quelle est la liste correspondant au triangle de Pascal de taille 2 ?
2. On considère la fonction Python suivante, dont l'objectif est d'imprimer la ligne i du triangle de Pascal T .

```
def ligne(T,i):
    T.append([1])
    for j in range(1,i):
        T[i].append(T[i-1][j-1]+T[i-1][j])
    T[i].append(1)
    print(T[i])
```

- (a) Qu'effectue l'instruction en ligne 2 ? Quelle différence avec l'instruction de la ligne 5 ?
 - (b) Quelles valeurs la variable j prend(elle dans la boucle des lignes 3 et 4 ?
 - (c) A quelle règle correspond la ligne 4 ?
3. Compléter la fonction Python suivante afin qu'elle imprime un triangle de Pascal de taille n .

```
def Pascal(n):
    T=[[1],[1,1]]
    print(T[0])
    print(T[1])
    for i in range(...):
        ligne(T,i)
```

4. Implanter les fonctions précédentes et les tester en exécutant `Pascal(10)`.