

La procédure de rendu est la même que pour les DM précédents. Déposer un fichier python sur le site (vous pouvez le nommer comme bon vous semble, à condition d'éviter les caractères spéciaux ou accentués et il faut que l'extension de fichier soit ".py").

Exercice 1

On définit les suites $(a_n)_{n \in \mathbb{N}}$ et $(b_n)_{n \in \mathbb{N}}$ par :

$$\begin{cases} a_0 = u \\ v_0 = v \end{cases} \quad \text{et } \forall n \in \mathbb{N} \begin{cases} a_{n+1} = \sqrt{a_n b_n} \\ b_{n+1} = \frac{2}{1/a_{n+1} + 1/b_n} \end{cases}$$

où u et v sont des réels strictement positifs.

1. Créer une fonction **iterer** d'un seul argument p , où p représente le couple (a_n, b_n) , et qui renvoie le couple (a_{n+1}, b_{n+1}) . Tester cette fonction en calculant par exemple **iterer([3, 2])** dans la console.
2. Créer une fonction **suite** de trois arguments u, v et n qui renvoie le couple (a_n, b_n) . Tester cette fonction en calculant **iterer(3, 2, 5)** et **iterer(3, 2, 10)**. Que conjecturer ?
3. On admet que pour deux réels u, v strictement positifs donnés, les suites $(a_n)_{n \in \mathbb{N}}$ et $(b_n)_{n \in \mathbb{N}}$ convergent vers une même limite ℓ qu'elles encadrent (pour $n \geq 1$, $a_n \geq \ell \geq b_n$). Créer une fonction (moyenne) de deux arguments u et v qui renvoie une valeur approchée à 10^{-10} près par défaut de cette limite ℓ .
4. Faire tracer **moyenne(1, x)** pour x variant de 1 à 10 avec un pas de 0.1

Exercice 2

1. Créer une fonction **initialise(n)** qui renvoie un tableau numpy à 2 dimensions représentant la matrice carrée de taille n dont tous les coefficients sont nuls sauf sur la première colonne et la diagonale où ils valent 1.

Indication : on pourra utiliser les fonction **np.zeros(n, p, dtype=int)** ou **np.eye(n, dtype=int)** pour obtenir une liste de liste de la bonne taille.

2. Créer une fonction **pascal(n)** qui renvoie une matrice carrée de taille $n + 1$ dont le coefficient d'indice (i, j) est $\binom{i}{j}$ (indices donnés en convention numpy, c'est à dire qu'ils commencent à 0). Par exemple l'appel de **pascal(3)** renvoie

```
array([[ 1.,  0.,  0.,  0.],
       [ 1.,  1.,  0.,  0.],
       [ 1.,  2.,  1.,  0.],
       [ 1.,  3.,  3.,  1.]])
```

Votre fonction devra pouvoir exécuter **pascal(500)**, ce qui paraît assez peu probable si vous utilisez la formule factorielle...

Exercice 3

1. Ecrire une fonction **binaire** d'argument un entier naturel n et qui renvoie la liste des chiffres, bit de poids fort en tête, de l'écriture en base 2 de n . Par exemple, **binaire(23)** renvoie $[1, 0, 1, 1, 1]$.

2. Ecrire une fonction **nombre_de_uns** d'argument un entier naturel n , qui renvoie le nombre de chiffre 1 dans l'écriture binaire de n . **nombre_de_uns(23)** renvoie ainsi 4.

3. Soit n un entier naturel. On dit que n est un 2-palindrome si sa représentation en base 2 est la même qu'elle soit écrite de gauche à droite ou de droite à gauche. Par exemple, 9 est un 2-palindrome car 9 s'écrit 1001 en base 2.

Ecrire une fonction **palindrome** d'argument un entier naturel n , qui renvoie un booléen indiquant si n est un 2-palindrome.