

# PYTHON AGRO-VETO 2023

## Listes

<code>[]</code> -----	Créer une liste vide	<code>L.pop(k)</code> --	Renvoie le <del><math>k</math>ème élément</del> <sup>l'élément d'indice <math>k</math></sup> de la liste $L$ et l'enlève de $L$
<code>[a]*n</code> -----	Créer une liste avec $n$ fois l'élément $a$	<code>L.remove(a)</code>	Enlève une fois la valeur $a$ de la liste $L$ ( <b>la première</b> )
<code>L.append(a)</code>	Ajoute l'élément $a$ à la fin de la liste $L$	<code>max(L)</code> ----	Renvoie le plus grand élément de la liste $L$
<code>L1 + L2</code> ----	Concatène les deux listes $L1$ et $L2$	<code>min(L)</code> ----	Renvoie le plus petit élément de la liste $L$
<code>len(L)</code> ----	Renvoie le nombre d'éléments de la liste $L$	<code>sum(L)</code> ----	Renvoie la somme de tous les éléments de la liste $L$

## Numpy

<code>import numpy as np</code>	
<code>np.array()</code> -----	Transforme une liste en matrice <code>numpy</code>
<code>np.linspace(a,b,n)</code> ----	Crée une matrice ligne de $n$ valeurs uniformément réparties entre $a$ et $b$ (inclus)
<code>np.zeros([n,m])</code> -----	Crée la matrice nulle de taille $n \times m$
<code>np.eye(n)</code> -----	Crée la matrice identité de taille $n$
<code>np.diag(L)</code> -----	Crée la matrice diagonale dont les termes diagonaux sont les éléments de la liste $L$
<code>np.transpose(M)</code> -----	Renvoie la transposée de $M$
<code>np.dot(M,P)</code> -----	Renvoie le produit matriciel $MP$
<code>np.sum(M)</code> -----	Renvoie la somme de tous les éléments de $M$
<code>np.prod(M)</code> -----	Renvoie le produit de tous les éléments de $M$
<code>np.max(M)</code> -----	Renvoie le plus grand élément de $M$
<code>np.min(M)</code> -----	Renvoie le plus petit élément de $M$
<code>np.shape(M)</code> -----	Renvoie dans un couple le format de la matrice $M$
<code>np.size(M)</code> -----	Renvoie le nombre d'éléments de $M$

## Numpy.linalg

<code>import numpy.linalg as la</code>	
<code>la.inv(M)</code> -----	Renvoie l'inverse de la matrice $M$ si elle est inversible
<code>la.eigvals(M)</code> -----	Renvoie la liste des valeurs propres de $M$
<code>la.eig(M)</code> -----	Renvoie un couple $L,P$ où $L$ est la liste des valeurs propres de $M$ et $P$ la matrice de passage associée
<code>la.matrix_rank(M)</code> -----	Renvoie le rang de $M$

## Random

<code>import random as rd</code>	
<code>rd.random()</code> -----	Simule une réalisation d'une variable $X \hookrightarrow \mathcal{U}([0,1])$
<code>rd.randint(a,b)</code> ----	Simule une réalisation d'une variable $X \hookrightarrow \mathcal{U}([a,b])$
<code>rd.gauss(0,1)</code> -----	Simule une réalisation d'une variable $X \hookrightarrow \mathcal{N}(0,1)$
<code>rd.choice(L)</code> -----	Choisit aléatoirement un élément de la liste $L$

## Math

### Fonctions également présentes dans Numpy

<code>import math as m</code>			
<code>m.atan(x)</code> -----	Renvoie $\arctan(x)$	<code>m.sqrt(x)</code> --	Renvoie $\sqrt{x}$ si $x \geq 0$
<code>m.floor(x)</code> -----	Renvoie $\lfloor x \rfloor$	<code>m.log(x)</code> ----	Renvoie $\ln(x)$ si $x > 0$
<code>m.factorial(n)</code> --	Renvoie $n!$ si $n \in \mathbb{N}$	<code>m.exp(x)</code> ----	Renvoie $e^x$

## Logique

<code>a == b</code> ----	Teste l'égalité « $a = b$ »
<code>a != b</code> ----	Teste « $a \neq b$ »
<code>a &lt; b</code> ----	Teste « $a < b$ »
<code>a &lt;= b</code> ----	Teste « $a \leq b$ »
<code>a &gt; b</code> ----	Teste « $a > b$ »
<del><code>a &gt;= b</code></del> ----	Teste « $a \geq b$ »
<code>not A</code> -----	Renvoie la négation de $A$
<code>A and B</code> ----	Renvoie « $A$ et $B$ »
<code>A or B</code> ----	Renvoie « $A$ ou $B$ »
<code>True</code> -----	Constante booléenne « Vrai »
<code>False</code> -----	Constante booléenne « Faux »

## Matplotlib.pyplot

<code>import matplotlib.pyplot as plt</code>	
<code>plt.plot(X,Y,'+-r')</code> ----	Génère la courbe des points définis par les listes $X$ et $Y$ (abscisses et ordonnées) avec les options : <ul style="list-style-type: none"><li>symbole : <code>'.'</code> point, <code>'o'</code> rond, <code>'h'</code> hexagone, <code>'+'</code> plus, <code>'x'</code> croix, <code>'*'</code> étoile, ...</li><li>ligne : <code>'-'</code> trait plein, <code>'--'</code> pointillé, <code>'-.'</code> alterné, ...</li><li>couleur : <code>'b'</code> <b>bleu</b>, <code>'r'</code> <b>rouge</b>, <code>'g'</code> <b>vert</b>, <code>'c'</code> <b>cyan</b>, <code>'m'</code> <b>magenta</b>, <code>'k'</code> <b>noir</b>, ...</li></ul>
<code>plt.bar(X,Y)</code> -----	Génère l'histogramme des points définis par les listes $X$ et $Y$ (abscisses et ordonnées)
<code>plt.axis('equal')</code> -----	Rend le repère orthonormé
<code>plt.xlim(xmin,xmax)</code> ----	Fixe les bornes de l'axe des abscisses
<code>plt.ylim(ymin,ymax)</code> ----	Fixe les bornes de l'axe des ordonnées
<code>plt.show()</code> -----	Affiche le graphique