ECG2 – Saint Just Mémento de Python

Les commandes ci-dessous sont les indispensables au programme.
D'autres commandes peuvent être définies de façon ponctuelle dans les sujets de concours.

I. Commandes de base dans Python

1°) commandes élémentaires

- **#** commentaire
- **print( )** affichage print(« valeur : »,i)
- **input** lecture de valeur n=int(input(«donner n »)) pour un entier, x=float(input(« donner x »)) pour un réel
- **x=x+1** s'écrit aussi **x+=1**

2°) connecteurs logiques et structures conditionnelles

| ***and*** : *et* | ***or*** : *ou* | ***not*** : *négation* |

True False

== (être égal à), <, >, < =, >=, != (être différent de)

a==b teste si a est égal à b. **a !=b** teste si a est différent de b

- **if** *condition* **:**
  suite d'instructions
  elif *condition2* **:**
  suite d'instructions2
  else:
  suite d'instructions3

3°) structures itératives.

- **range(n)** suite des entiers de 0 à n-1. **range(a,b)** suite des entiers de a à b-1
  list(range(n)) transformer en liste pour afficher
- **for i in range(n):**
  bloc d'instructions

Variantes : **for i in range(a,b)** (i varie de a à b-1)

for i in T où **T** peut être une matrice, un vecteur, une chaîne de caractères

- **while** *condition***:**
  bloc d'instructions

4°) opérations et fonctions élémentaires.

- $+,\ *,\ -,\ /,**$ $(puissance),\ //\ (quotient\ dans\ la\ division\ euclidienne)$

5°) fonctions .

def nomdelafonction(x_1 , , x_n)**:** # plusieurs paramètres possibles, ou aucun…
suite d'instructions
return valeur souhaitée

Il est possible d'écrire une fonction qui ne renvoie pas de valeur.

Exemple

```
def f(x,y) :
    return x**2+3*y
```

6°) Importer une librairie.

import … as
from … import *

II. LIBRAIRIE NUMPY

import numpy as np

1°) Constantes classiques

np.e : constante e=exp(1)
np.pi : constante pi=3.14…

2°) Fonctions usuelles

np.exp, np.log, np.sin, np.cos,
np.sqrt : racine carrée
np.abs : valeur absolue
np.floor : partie entière

Ces fonctions peuvent s'appliquer à des valeurs numériques, des vecteurs ou des matrices.

3°) Matrices : définitions

A=np.array([[1, 2 , 3], [4 , 5, 6]]) définit la matrice $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$

L=np.array([1,2,3]) définit la matrice ligne $L = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$.

C=np.array([[1],[2],[3]]) définit la matrice colonne $C = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$.

B=np.zeros([2,3]) renvoie la matrice nulle de taille 2x3

J=np.ones([2,3]) renvoie la matrice remplie de 1 de taille 2x3

I=np.eye(3) renvoie la matrice identité de taille 3x3

a,b =np.shape(A) renvoie la taille de la matrice A

np.linspace : matrice ligne de valeurs régulièrement espacées. Np.linspace(1,10,20) place 20 valeurs entre 1 et 10. Attention il s'agit d'une matrice unidimensionnelle (un seul crochet).

np.arange(début, fin, pas) : matrice ligne de valeurs régulièrement espacées avec un certain pas.
Attention il s'agit aussi d'une matrice unidimensionnelle (un seul crochet).

4°) opérations usuelles sur les matrices

A+B, A-B, k*A

np.dot(A,B) pour le produit matriciel
np.transpose(A) pour la transposée de A

Attention A*B est le produit case à case

5°) Autres opérations sur les matrices

np.sum somme les cases de la matrice
np.min renvoie le minimum des cases de la matrice
np.max renvoie le maximum des cases de la matrice
np.mean renvoie la moyenne des cases de la matrice
np.cumsum renvoie la somme cumulée des cases de la matrice
np.median renvoie la médiane des valeurs de la matrice
np.var renvoie la variance des valeurs de la matrice
np.std renvoie l'écart type des valeurs de la matrice

III. LIBRAIRIE NUMPY.LINALG

import numpy.linalg as al

al.inv : inverse d'une matrice
al.matrix_rank :. rang d'une matrice
al.matrix_power : puissance d'une matrice
al.solve : al.solve(A,B) résout l'équation AX=B où A matrice carrée et B matrice colonne
al.eig : spectre et vecteurs propres d'une matrice (valeurs propres = eigen values)

IV. LIBRAIRIE NUMPY.RANDOM

import numpy.random as rd

rd.random : loi uniforme sur [0,1] rd.random(3) renvoie trois réels au hasard dans [0,1]
rd.binomial : loi binômiale
rd.randint : loi uniforme discrète rd.randint(5,10) renvoie un entier de [[5,9]]
rd.geometric : loi géométrique. rd.geometric(0.3) simule la loi géométrique de paramètre 0.3
rd.poisson : loi de Poisson. rd.poisson(4) simule la loi de Poisson de paramètre 4
rd.exponential : loi exponentielle
ATTENTION rd.exponential(1/lambda) pour simuler la loi E(lambda) (on prend l'espérance comme paramètre).
rd.normal : loi normale
ATTENTION : rd.normal(m,sigma) : le deuxième paramètre est l'écart-type.
rd.gamma : loi gamma

On peut générer au choix un nombre aléatoire, un vecteur aléatoire ou une matrice à coefficients aléatoires.

Exemple :
rd.binomial(10,0.2) : un nombre aléatoire simulation de la loi B(10,0.2)
rd.binomial(10,0.2,100) : vecteur aléatoire contenant 100 simulations de la loi B(10,0.2)
rd.binomial(10,0.2,[100,10]) : matrice aléatoire de taille 100 x 10 dont les coefficients sont des simulations de la loi B(10,0.2)

V. LIBRAIRIE SCIPY.SPECIAL

import scipy.special as sp

sp.ndtr : fonction Φ sp.ndtr(x) calcule Φ(x)

VI. LIBRAIRIE MATPLOTLIB.PYPLOT

import matplotlib.pyplot as plt

plt.plot : tracé de courbes plt.plot(x,y) relie les points dont les coordonnées sont données dans x et y
plt.show : affichage de la courbe à la fin d'un problème
on pourra utiliser **xlim, ylim, axis, grid, legend.** axis(« equal ») pour un repère orthonormé
plt.hist
plt.contour : tracés de lignes de niveau en lien avec np.meshgrid
plt.quiver : tracé de gradients

VI. FONCTION AXES3D

from mpl_toolkits.mplot3d import Axes3D
ax = Axes3D(plt.figure())

ax.plot_surface. La maîtrise de cette fonction n'est pas exigible.
En lien avec np.meshgrid. Il s'agira plutôt d'interpréter des graphiques.