

Informatique PCSI

Corrigé TP 3

Exercice 1

De manière générale, on écrit : `from module import fonction`. C'est la réponse 3.

Exercice 2

1. Les fonctions du module.

```
def maximum(a, b):
    """ a et b sont des nombres de type int ou float
    la fonction renvoie le maximum des deux nombres """
    if a > b:
        return a
    else:
        return b

def minimum(a, b):
    """ a et b sont des nombres de type int ou float
    la fonction renvoie le minimum des deux nombres """
    if a < b:
        return a
    else:
        return b
```

2. Il faut importer le module ou la fonction utilisée (`from mesfonctions import maximum`). Le fichier suivant et le fichier `mesfonctions.py` sont placés dans le même répertoire.

```
# autre méthode: on importe le module et on le renomme mf
import mesfonctions as mf

def maxi(a, b, c, d):
    """ a, b, c et d sont des nombres de type int ou float
    la fonction renvoie le maximum des quatre nombres """
    m1 = mf.maximum(a, b)
    m2 = mf.maximum(c, d)
    return mf.maximum(m1, m2)
```

Exercice 3

```
import matplotlib.pyplot as plt
from random import randint

def marche(n): # n nombre d'étapes
    y = 0
    yliste = [0] # la liste des ordonnées
    for i in range(n):
        y = y + 2 * randint(0, 1) - 1 # pour choisir -1 ou 1 "au hasard"
```

```

        yliste.append(y)
    xliste = list(range(n+1))
    plt.plot(xliste, yliste, "r")
    plt.grid()
    plt.show()

marche(500)

```

Exercice 4

On écrit dans un fichier nommé `stats.py` les codes des trois fonctions.

```

def somme(liste):
    s = 0
    for x in liste:
        s = s + x
    return s

def moyenne(liste):
    s = somme(liste)
    n = len(liste)
    return s / n

def variance(liste):
    n = len(liste)
    m1 = moyenne(liste)
    carres = [x ** 2 for x in liste]
    m2 = moyenne(carres)
    return m2 - m1 ** 2

```

Le fichier est enregistré dans un répertoire. Dans ce même répertoire on écrit un fichier `test.py`.

```

import stats as st

liste = [1, 2, 3, 4, 5]
print(st.somme(liste))
print(st.moyenne(liste))
print(st.variance(liste))

```

Exercice 5

Il faut créer la liste qui est donnée en paramètre à la fonction `moyenne` du module `stats`.

```

import stats as st

def moyenne_pond(liste):
    liste2 = []
    for [x, n] in liste:
        for i in range(n):
            liste2.append(x)
    return st.moyenne(liste2)

```