

## Semaine n°29 du 10 au 14 juin

## Informatique(Python) : cf exemples en annexe

- Tableau 2D, bibliothèque Numpy

## Dérivation

- Dérivabilité en un point, nombre dérivée en un point, fonction dérivée, équation de la tangente, dérivabilité à gauche et à droite.
- $f$  dérivable en  $x_0 \Rightarrow$  continue en  $x_0$  (démonstration exigible).
- Opération, dérivabilité et dérivée d'une composée, d'une réciproque.
- Dérivabilité et dérivée de Arctan (démonstration exigible)
- Dérivée et extrema, théorème de Rolle, théorème des accroissements finis.
- Dérivation et variation.
- Fonction de classe  $\mathcal{C}^n$  : définition, fonction de classe  $\mathcal{C}^n$  ( $n \in \mathbb{N}$ ), de classe  $\mathcal{C}^\infty$ , somme, produit, quotient et composition de fonctions de classe  $\mathcal{C}^\infty$ .

## Intégration

- intégrale d'une fonction continue sur un segment.
- relation de Chasles, linéarité, positivité et croissance de l'intégrale, inégalité triangulaire pour les intégrales, valeur moyenne et propriété (la valeur moyenne est comprise entre le minimum et le maximum de la fonction)
- Théorème fondamentale de l'analyse.

## Remarques aux colleurs

- Merci aussi de poser une petite question d'informatique (cf Annexe).

## Exemples de programmes informatiques

## Exercice 1

Ecrire en Python une fonction `existence` qui prend en entrée un tableau  $T$  et un nombre  $element$  et renvoie `True` si  $element$  se trouve dans le tableau  $T$ , `False` sinon.

```
def existence(T,element):
    a=len(T)      # nombre de lignes
    b=len(T[0])  # nombre de colonnes
    for i in range(a):      # parcours des lignes
        for j in range(b): # parcours des colonnes
            if T[i,j]==element: # on teste si T[i,j] est égal à élément
                return True
    return False # si on n' a pas trouvé element après avoir parcouru tout le tableau
```

## Exercice 2

Ecrire en Python une fonction `MaximumTableau` qui prend en entrée un tableau  $T$  et renvoie la plus grande valeur de ce tableau

```
def MaximumTableau(T):
    a=len(T)      # nombre de lignes
    b=len(T[0])   # nombre de colonnes
    maxi = T[0,0] # initialisation avec la première valeur du tableau
    for i in range(a):      # parcours des lignes
        for j in range(b):  # parcours des colonnes
            if T[i,j]>maxi:  # on teste si T[i,j] est plus grand
                maxi=T[i,j] # on a trouvé une plus grande valeur
    return maxi
```

### Exercice 3

Ecrire en Python une fonction Moyenne qui prend en entrée un tableau  $T$  et renvoie la moyenne de ses éléments :

```
def Moyenne(T):
    a=len(T)      # nombre de lignes
    b=len(T[0])   # nombre de colonnes
    S=0           # initialisation de la somme
    for i in range(a):      # parcours des lignes
        for j in range(b):  # parcours des colonnes
            S=S+T[i,j]      # on rajoute l'élément T[i,j]
    return S/(a*b)         # formule pour la moyenne
```

### Exercice 4

Ecrire une fonction `experience` qui prend en paramètre un entier  $n$  et simule  $n$  lancers successifs d'une pièce de monnaie équilibrée en renvoyant une liste aléatoire composée de  $n$  valeurs égales à 0 ou 1. On considérera que 0 correspond à Face et 1 à Pile.

```
from random import * # bibliothèque nécessaire pour créer des nombres aléatoires
def experience(n):
    L=[]              #liste vide initialement
    for i in range(n):      #on répète n fois
        L.append(randint(0,1)) # on rajoute 0 ou 1, choisi de manière aléatoire
    return L
```

### Exercice 5

Créer une fonction `SommeLignes` prenant en entrée un tableau  $T$  et qui renvoie une liste contenant la somme de chaque lignes.

```
def SommeLignes(T):
    n=len(T)      # nombre de lignes
    p=len(T[0])   # nombre de colonnes
    L=[]          # la liste est vide au debut
    for i in range(n): # parcours des lignes
        S=0       # initialisation pour la somme de la ligne i
        for j in range(p): # parcours des colonnes
            S = S + T[i,j]
        L.append(S)      # on met la somme de la ligne i dans le liste L
    return L
```