

Programmation scientifique en Python

v3.1



Programmation scientifique en Python de [Dr Michaël GUEDJ](#) est mis à disposition selon les termes de la [licence Creative Commons Attribution 4.0 International](#).

Fondé(e) sur une œuvre à https://github.com/michaelguedj/ens__prog_sci_python.

Table des matières

TP – Utilisation de fonctions mathématiques.....	3
Sujet d'étude 1 : opérations binaires.....	6
Sujet d'étude 2 : tournoi de jeu vidéo.....	8
Sujet d'étude 3 : fibre optique.....	9
TP – Opérations sur points.....	12
TP – Nombres complexes.....	14

TP – Utilisation de fonctions mathématiques

Exercice 1

Calculer en Python :

- $\sin \pi - \cos \pi$
- $\sin(\pi/2) - \cos(\pi/2)$
- $\cos(\pi)$
- $\cos(\pi + 2\pi)$
- $\cos 5$
- $\cos \sqrt{5}$

Exercice 2

Implémenter la fonction : $f(x, y) = \sin x - \cos y$.

Mathématiquement, $f(\pi, \pi) = 1$ et $f(\pi/2, \pi) = 2$.

Exercice 3

Calculer en Python :

- $2\sqrt{2} + 2$
- $2\sqrt{2+2}$
- $\frac{3^3 + 3^5}{2} + \sqrt{3}$

Exercice 4

Implémenter la fonction :

$$f(x, y, z) = \frac{x^2 + \sqrt{y+2} + \cos z}{3}$$

Mathématiquement, $f(2, -2, \pi) = 1$

Exercice 5

Calculer en Python :

- $2 \ln 2 + 3$
- $2 \log 2 + 3$
- $2 \log_2 2 + 3$

Exercice 6

Implémenter la fonction :

$$f(x) = \frac{2 \log x + 3}{5}$$

Vérifiez que $f(10) = 1$.

Exercice 7

Implémenter la fonction :

$$g(x, y) = \frac{5 \log x}{5} + y$$

Vérifiez que :

- $g(10, 7) = 8$
- $g(10, 8) = 9$
- $g(10, 501) = 502$

Exercice 8

Calculer en Python :

- e^3
- $e^{\frac{3}{2}}$
- $e^{\ln 301}$
- $10^{\log 501}$

Rappel : mathématiquement on a :

- $e^{\ln 301} = 301$
- $10^{\log 501} = 501$

Exercice 9

Soit l'équation $y = 3 \ln \frac{x+2}{5}$ (pour $x > -2$).

Théorème.

$$5 \cdot e^{\frac{y}{3}} - 2 = x$$

Preuve.

$$y = 3 \ln \frac{x+2}{5} \Leftrightarrow \frac{y}{3} = \ln \frac{x+2}{5} \Leftrightarrow e^{\frac{y}{3}} = e^{\ln \frac{x+2}{5}} \Leftrightarrow e^{\frac{y}{3}} = \frac{x+2}{5} \Leftrightarrow 5 \cdot e^{\frac{y}{3}} = x+2 \Leftrightarrow 5 \cdot e^{\frac{y}{3}} - 2 = x$$

□

1. Implémenter la fonction : $toto(x) = 3 \ln \frac{x+2}{5}$
2. Implémenter la fonction : $tata(y) = 5 \cdot e^{\frac{y}{3}} - 2$

Mathématiquement, pour tout k réel positif, $toto(tata(k)) = tata(toto(k)) = k$.

3. Via l'interpréteur de code, l'égalité est elle vérifiée pour $k=5$? Idem pour $k=101$?

Sujet d'étude 1 : opérations binaires

Nous nous intéressons dans un premier temps aux opérations binaires (simples) ; puis étendons ces opérations pour gérer les opérations binaires bits à bits sur les octets.

Opérations binaires

- La négation binaire NON est définie par : $\text{NON}(0)=1$ et $\text{NON}(1)=0$.
- La disjonction binaire OU est définie par : $1 \text{ OU } 1 = 1$; $1 \text{ OU } 0 = 1$; $0 \text{ OU } 1 = 1$ et $0 \text{ OU } 0 = 0$.
- La conjonction binaire ET est définie par : $1 \text{ ET } 1 = 1$; $1 \text{ ET } 0 = 0$; $0 \text{ ET } 1 = 0$ et $0 \text{ ET } 0 = 0$.

Exemple d'opérations bits à bits sur octets

- $\text{NON}(00110000) = 11001111$
- $00110000 \text{ OU } 11000011 = 11110011$
- $00110011 \text{ ET } 11000011 = 00000011$

Partie A : Algorithmique

1 – Opérations binaires

Fonctions

- 1- Réaliser la fonction **non** qui prend en argument un bit et retourne sa négation.
- 2- Réaliser la fonction **ou** qui prend en argument deux bits et retourne leur disjonction.
- 3- Réaliser la fonction **et** qui prend en argument deux bits et retourne leur conjonction.

2 – Opérations bits à bits sur les octets

Structure de données

Déterminer un type de donnée nécessaire pour stocker un octet (8 bits).

Fonctions

- 4- Réaliser la fonction **non_oct** qui prend en argument un octet et retourne sa négation bit à bit.
- 5- Réaliser la fonction **ou_oct** qui prend en argument deux octets et retourne leur disjonction bit à bit.
- 6- Réaliser la fonction **et_oct** qui prend en argument deux octets et retourne leur conjonction bit à bit.

Intéressez-vous à l'extension de vos résultats, pour gérer des opérations bit à bit, sur des données binaires composées de 4 octets.

Partie B : Implémentation

- Planter en Python les fonctions de la Partie A.
- Définir un jeu de test qui vous servira à tester la bonne implantation des fonctions demandées.

Sujet d'étude 2 : tournoi de jeu vidéo

Contexte d'un tournoi de jeu vidéo.

1. Comment stocker les scores des joueurs d'une équipe ?
2. Fonction prenant en argument les scores d'une équipe, et retournant le nombre de joueurs d'une équipe ?
3. Fonction prenant en argument les scores d'une équipe, et retournant Vrai si l'équipe n'a pas de joueurs ; et Faux sinon ?
4. Procédure prenant en argument les scores d'une équipe, et affichant les scores qui sont strictement positifs.
5. Procédure prenant en argument les scores d'une équipe, ainsi qu'un certain score x , et affichant les scores supérieurs ou égaux au score x .
6. Fonction prenant en argument les scores d'une équipe, et retournant la moyenne des scores.

$$\mu = \frac{1}{n} \sum_{i=0}^{n-1} \text{scores}[i]$$

(Le nombre de joueurs n est supposé non nul).

7. Fonction prenant en argument les scores d'une équipe, et retournant la variance des scores.

$$V = \frac{1}{n} \sum_{i=0}^{n-1} (\text{scores}[i] - \mu)^2$$

(Le nombre de joueurs n est supposé non nul).

Sujet d'étude 3 : fibre optique

Une fibre optique est un fil, en verre ou en plastique, très fin, qui a la propriété d'être un conducteur de la lumière ; elle sert dans la transmission de données, offrant un débit d'information nettement supérieur à celui des câbles coaxiaux ; et peut servir de support à un réseau « large bande » par lequel transitent aussi bien la télévision, le téléphone, la visioconférence ou les données informatiques.

Ci-dessous, l'extrait d'un article du Figaro datant du 17/07/2015.

2016, année de l'accélération pour la fibre en France

En France, la fibre optique va se généraliser à partir de 2016 par la mise en place de « réseaux d'initiative publique ».

La fibre optique s'apprête à sortir des plus grandes villes de France. D'ici à la fin de l'année, plusieurs dizaines de milliers de lignes auront été activées dans de nouveaux « réseaux d'initiative publique » détenus par les collectivités territoriales.

« Nous atteindrons notre rythme de croisière fin 2016, avec un million de nouvelles prises raccordables chaque année. Ce sera une lame de fond », explique Antoine Darodes, qui pilote le plan "France Très Haut Débit".

L'atténuation caractérise l'affaiblissement du signal au cours de la propagation. Le principal atout des fibres optiques est une atténuation extrêmement faible.

Le coefficient d'atténuation A , exprimée en dB/km, est donné par la relation :

$$A = \frac{1}{L} \cdot 10 \cdot \log \frac{P_e}{P_s}$$

Où :

- L est la longueur de la fibre optique (en km) ;
- P_e (en milliwatt ou mW) est la puissance lumineuse d'entrée ;
- P_s (en milliwatt ou mW) est la puissance lumineuse de sortie.

Nous souhaitons connaître les coefficients d'atténuation pour des fibres optiques dont les références sont FO1, FO2 et FO3.

Un technicien mesure une puissance lumineuse d'entrée sur chaque fibre (FO1, FO2 et FO3) de 5 mW.

En outre, chaque fibre a une longueur de 5 km.

Des techniciens mesurent les puissances lumineuses de sortie sur chaque fibre, et notent leurs résultats dans le tableau ci-dessous.

Référence de la fibre optique	FO1	FO2	FO3
L (Longueur de la fibre) (en km)	5	5	5
P_e (puissance lumineuse d'entrée) (en mW)	5	5	5
P_s (puissance lumineuse de sortie) (en mW)	0.8891	1.7741	3.7495

Ainsi, dans ce qui suit, nous considérerons que :

- Toute fibre est de longueur $L=5$ km ;
- Toute puissance lumineuse d'entrée P_e est fixée à 5 mW.

Partie A : Algorithmique

Type de donnée

1- Déterminer un type de donnée permettant de stocker les puissances de sortie de l'ensemble des fibres FO1, FO2 et FO3.

2- Déterminer un type de données permettant de stocker les puissances de sortie d'un ensemble de fibres optiques.

Fonctions

3- Réaliser la fonction **coefficient_attenuation** prenant en paramètre une puissance de sortie mesurée sur une fibre optique, et retournant le coefficient d'atténuation correspondant.

Procédures traitantes des fibres FO1, FO2 et FO3

4- Réaliser la procédure **afficher** prenant en paramètre les puissances de sortie des fibres FO1, FO2 et FO3, et affichant ces puissances de sortie (on affichera une puissance par ligne).

5- Réaliser une procédure prenant en paramètre les puissances de sortie des fibres FO1, FO2 et FO3, et affichant les coefficients d'atténuations correspondants à ces puissances de sortie.

Procédures traitantes d'un ensemble de fibres

6- Réaliser la procédure **afficher2** prenant en paramètre les puissances de sortie d'un ensemble de fibres, et affichant ces puissances de sortie (on affichera une puissance par ligne).

7- Réaliser une procédure prenant en paramètre les puissances de sortie d'un ensemble de fibres, et affichant les coefficients d'atténuations correspondants à ces puissances de sortie.

Partie B : Implémentation

Planter en Python les fonctions/procédures de la Partie A.

Définir un jeu de test qui vous servira à tester la bonne implantation des fonctions/procédures demandées.

TP – Opérations sur points

Nous utilisons la Programmation Orientée Objet (POO) pour manipuler les points (2D).

Ci-après, une première implémentation.

```
import math
sqrt = math.sqrt

class Point():

    def __init__(self, x, y):
        self.x = x
        self.y = y

    def norme(self):
        return sqrt(self.x**2 + self.y**2)

    def produit_scalaire(self, p):
        return True #A FAIRE

    def addition(self, p):
        return Point(self.x + p.x, \
                      self.y + p.y)

    def distance(self, p):
        return True #A FAIRE

    # pour l'impression
    def __str__(self):
        return "(" + \
            str(self.x) + ", " + \
            str(self.y) + \
            ")"

if __name__ == "__main__":

    p1 = Point(5, 2)
    print(p1)
```

Exercice 1

Tester le code ci-avant.

Exercice 2

Créer (*instancier*) le point $p1=(3,4)$ et afficher-le.

Exercice 3

Afficher la norme du point $p1$ par la commande :

```
print(p1.norme())
```

Exercice 4

Afficher la norme du point $p2$.

Exercice 5

Afficher la somme des points $p1$ et $p2$ par la commande :

```
print(p1.addition(p2))
```

Exercice 6

Tester la commande :

```
print(p2.addition(p1))
```

Exercice 7

Implémenter la méthode **produit_scalaire**, prenant en argument un point, et retournant le produit scalaire du « point objet » (*self*) et du « point argument ».

Rappel : pour deux points de \mathbb{R}^2 : $p=(x,y)$ et $p'=(x',y')$, le produit scalaire de p et p' est défini par :

$$p \cdot p' = x \cdot x' + y \cdot y'$$

Tester les commandes :

```
print(p1.produit_scalaire(p2))
```

et

```
print(p2.produit_scalaire(p1))
```

Exercice 8

Implémenter la méthode **distance**, prenant en argument un point, et retournant la distance euclidienne du « point objet » (*self*) et du « point argument ».

Rappel : pour deux points de \mathbb{R}^2 : $p=(x,y)$ et $p'=(x',y')$, la distance euclidienne de p et p' est définie par :

$$dist(p, p') = \sqrt{(x-x')^2 + (y-y')^2}$$

Tester les commandes :

```
print(p1.distance(p2))
```

et

```
print(p2.distance(p1))
```

TP – Nombres complexes

Un nombre complexe z se présente, en général, sous **forme algébrique**, comme une somme :

$$a + i \cdot b$$

Où :

- a et b sont des nombres réels quelconques ;
- i (l'unité imaginaire) est un nombre particulier tel que $i^2 = -1$.
- Le réel a , est appelé **partie réelle** de z , et se note $\Re(z)$;
- Le réel b , est appelé **partie imaginaire** de z , et se note $\Im(z)$.

Deux nombres complexes sont égaux si et seulement si : ils ont la même partie réelle et la même partie imaginaire :

$$\forall z, z' \in \mathbb{C},$$

$$z = z' \quad \text{iff} \quad [\Re(z) = \Re(z') \text{ et } \Im(z) = \Im(z')]$$

Un nombre complexe z est dit **imaginaire pur** si sa partie réelle est nulle ; dans ce cas il s'écrit sous la forme :

$$z = i \cdot b$$

Un nombre complexe, dont la partie imaginaire est nulle, est dit **réel**.

Exercices

1. Définir une classe **Complexe**, qui nous permettra de manipuler les nombres complexes.
Implémenter :
2. Une méthode d'affichage d'un nombre complexe ;
3. Une méthode effectuant la somme de deux nombres complexes ;
4. Une méthode calculant l'opposé d'un nombre complexe (l'opposé de $a + b \cdot i$ est $-a - b \cdot i$) ;
5. Une méthode retournant Vrai si le nombre complexe considéré est un imaginaire pur (i.e. : $z = 0 + b \cdot i$) ; et Faux sinon.

6. Une méthode retournant le conjugué du nombre complexe considéré (le conjugué de $a+b.i$ est $a-b.i$) ;
7. Une méthode retournant le module du nombre complexe considéré (le module de $z=a+b.i$ (noté $|z|$) est défini par : $|z|=\sqrt{a^2+b^2}$).
8. Une méthode retournant la multiplication de deux nombres complexes
 $((a+i.b)\times(a'+i.b'))=(aa'-bb')+(ab'+ba').i$).