



L'ÉCOLE DE L'INTELLIGENCE ARTIFICIELLE

# Mastère 1

**EXAMEN** Versionning de code et

Programmation Python:

*Implémentation de fonction  
mathématiques et intégration à Github*

*ANNEE*

*2021 – 2022*



## Présentation du projet

Cet examen consiste à mettre en application les notions de programmation Python et de versionning de code étudié au S1.1. Les supports des cours et la documentation Python sont autorisés. Aucun rendu ne sera corrigé s'il est rendu **3h00 après le début de l'examen**.

**Tout rendu identique** où jugé fortement similaire ne **sera pas corrigé**.

### *1. Création de fonction mathématique simple en Python (12 points) :*

Dans cette partie, vous devrez implémenter en Python les fonctions ci-dessous :

a) Implémenter la fonction polynomiale ci-dessous :

$$f(x) = x^3 - 1,5x^2 - 6x + 5$$

Test du code :

Polynome(5) = 62.5

b) Implémenter la fonction **factorielle (Approche récursive ou classique)** :

Soit  $n$  un entier naturel, on définit la fonction factorielle comme :

$$n! = \prod_{1 \leq i \leq n} i = 1 \times 2 \times 3 \times \dots \times (n-1) \times n.$$

Si la valeur  $n$  est égale à 0 alors **la fonction factorielle donnera 1**. Sinon, on procèdera aux produits de tous les entiers **allant de 1 jusqu'au nombre saisi**.

Voici un exemple de cette suite :

$$1! = 1$$

$$2! = 1 \times 2 = 2$$

$$3! = 1 \times 2 \times 3 = 6$$

.

.

.

$$n! = 1 * 2 * 3 * \dots * (n-1) * n$$

Test du code :

```
factorielle(5)
```

120

c) Implémenter la suite de **Fibonnaci**

La [suite de Fibonacci](#) est une suite de nombres entiers de 0, 1, 1, 2, 3, 5, 8 ....

Les deux **premiers termes sont 0 et 1**. Tous les autres termes sont obtenus en ajoutant les deux termes précédents. Cela signifie que le nième terme est la somme des **(n-1)ème** et **(n-2)ème** terme.

Voici un exemple de cette suite :

$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$
0	1	1	2	3	5	8	13	21

Test du code :

```
Entrez un nombre: 6
La suite fibonacci est :
0 , 1, 1, 2, 3, 5,
```

*2. Création de fonction comportant des modules de gestions des exceptions (5 points) :*

En utilisant l'une des trois fonctions mathématiques précédentes, implémenter un script permettant de gérer les exceptions pour les cas suivants :

- Saisie d'une **chaîne de caractère** dans les arguments de la fonction
- Saisie un **nombre complexe**
- Saisie d'un **nombre négatif**
- Saisie d'un très **grand nombre**
- Saisie d'un très **petit nombre**

On pourra s'aider de la correction du **TP Pythagore** figurant sur le Classroom.

*3. Question Bonus : Implémentation de la formule Pricer d'option avec Python*

Ce modèle mathématique provient du travail de trois chercheurs ; Fischer Black, Myron Scholes et Bob Merton. Leurs recherches ont mérité le prix Nobel de l'économie en 1997.

Les options étaient déjà fréquemment transigées sur les marchés boursiers, mais il n'y avait aucune méthode mathématique précise pour trouver le prix théorique que devrait valoir une option. Les investisseurs avaient une idée générale de ce qui pouvait influencer le prix d'une option, mais cette formule vient confirmer le tout et vient aussi établir la relation causale entre toutes les variables.

Voici la formule du modèle Black-Scholes-Merton:

$$c = S N(d_1) - X e^{-rT} N(d_2)$$
$$p = X e^{-rT} N(-d_2) - S N(-d_1)$$

L'équation C évalue le prix d'une option d'achat (*call option*) alors que l'équation P calcule le prix d'une option de vente (*put option*). De plus, les variables d1 ainsi que d2 possèdent leur propre formule dans ce modèle :

$$d_1 = \frac{\ln(S / X) + (r + \sigma^2 / 2)T}{\sigma \sqrt{T}}$$
$$d_2 = \frac{\ln(S / X) + (r - \sigma^2 / 2)T}{\sigma \sqrt{T}} = d_1 - \sigma \sqrt{T}$$

À première vue, cette formule semble imposante mais il suffit d'expliquer chacune des 5 variables utilisées :

S = Prix actuel de l'action

X = Prix d'exercice de l'option

T = Temps restant avant l'expiration de l'option, en pourcentage d'une année

r = Taux d'intérêt sans risque

$\sigma$  = volatilité implicite du prix de l'action, mesurée par un décimal

#### **4. Déploiement des 4 scripts sur la plateforme Github (3 points) :**

Vous constituerez enfin, un dossier comportant l'ensemble des scripts implémentés sur votre compte GitHub. Vous y uploaderez ces fichiers sur ledit dossier. Deux possibilités pour ce faire : l'upload des fichiers manuellement via le bouton « upload » sur **github** et les commandes **git**.

**A noter que la première option ne rapporte qu'un point tandis que la seconde permet d'obtenir les 3 points.**

Vous ajouterez un fichier Markdown (.md) sur Github comportant l'ensemble des commandes git utilisées.

## Livrables

Les livrables attendus pour chaque étudiant sont :

- 4 scripts python au format .py
- Un lien vers le dossier github sur lequel les scripts ont été uploader (push)
- Le fichier .md contenant l'ensemble des instructions git permettant de push les fichiers depuis un repository local

Sur la partie « **gestion des exceptions** », on valorisera particulièrement la capacité de l'élève à aller au-delà des consignes et à proposer des solutions permettant de rendre l'outil développé **le plus robuste possible**.

## Modalités de rendu

### *A. Rendu écrit*

Le rendu écrit doit se faire sur **Classroom**, dans le devoir intitulé « **Projet Versionning de code et Python** », de préférence dans un *dossier zip* avec tous vos documents ainsi que le lien vers votre dossier github.

## Références/liens utiles :

- **For Loop** : [https://www.w3schools.com/python/python\\_for\\_loops.asp](https://www.w3schools.com/python/python_for_loops.asp)
- **Condition** : [https://www.w3schools.com/python/python\\_conditions.asp](https://www.w3schools.com/python/python_conditions.asp)
- **Black & Scholes** : <https://educationfinance.ca/investissements/le-modele-black-scholes-merton-pour-evaluer-le-prix-des-options/>