

# Consignes pour le projet Python

## 1. Effectifs

Comme pour tous les projets, le projet Python doit être réalisé en binôme. En aucun cas un groupe de trois signataires ou plus ne sera accepté. Il est possible de le faire seul, mais la notation sera la même que dans le cas d'un binôme.

## 2. Langage

Le projet devra être réalisé en Python, à l'exclusion de tout autre langage. Il est possible de recourir à diverses bibliothèques / packages. Ceux-ci devront être explicitement signalés dans un document joint au code lui-même, afin que la correction ne ressemble pas à un jeu de piste.

## 3. Emprunt de code et IA

Il est possible d'emprunter du code, soit à un être humain, soit à une IA. Toutefois, en pareil cas, deux règles doivent impérativement être respectées :

- Ces emprunts doivent être explicitement signalés, en commentaire ou dans une documentation jointe.
- Vous devez être capable d'expliquer et de modifier *dans le détail* l'intégralité du code qui figure dans le projet. Autrement dit : pas de « boîte noire » (ou magique).

## 4. Sujet

Le sujet est libre. On veillera cependant à faire un choix qui ne soit pas trop éloigné ni des possibilités du langage, ni de ce qui a été traité en cours. Seront ainsi à éviter les applications requérant une interface graphique, les animations en temps réel, et même celles qui appelleraient une gestion un peu élaborée de bases de données (au-delà de simples fichiers textes ou binaires).

Deux pistes peuvent être suggérées :

- des utilitaires, liés en particulier au traitement des chaînes de caractère (analyse syntaxique et lexicale, datamining...), à partir du web ou de documents fournis
- des jeux de société (ne nécessitant pas un affichage trop visuel).

## 5. Documents

Le seul document obligatoire est le code source. Cependant, tout document annexe peut être fourni à la discrétion des auteurs : règles éventuelles du jeu (mais qui peuvent aussi être intégrées dans l'application), documentation technique, explications sur les choix effectués, les fonctionnalités qui auraient dû être développées mais ne l'ont pas été, etc.

## 6. Evaluation

L'évaluation se déroule en trois temps. Tout d'abord, le correcteur se comporte en utilisateur loyal, prenant en main le logiciel avec les éléments fournis. Ensuite, il teste la résistance du logiciel à des actions illégitimes, afin de vérifier sa robustesse. Enfin, il examine le code afin de juger de sa clarté, de sa pertinence et de sa construction optimale (factorisation des modules, absence de redondances).

## 7. Notation

La notation n'obéit pas à une grille, impossible à mettre en œuvre sur une diversité de sujets, mais à un mix informel entre quatre critères.

- intérêt du sujet : un utilitaire doit être... utile, un jeu doit être distrayant et attractif, etc.
- difficulté du sujet
- ampleur du sujet
- qualité de la réalisation (dont celle de l'interface et de son caractère *user-friendly*).

En plus de la note, chaque projet fera l'objet d'une appréciation rédigée.