





Mon parcours



✓ PROJET VALIDÉ

Développez un programme logiciel en Python

Mission Cours Ressources Évaluation



Mis à jour le mardi 10 janvier 2023

Si vous avez commencé ce projet avant le 21/12/2022, vous avez débuté votre travail sur <u>ce</u> <u>projet archivé</u>. Vous pouvez continuer l'existant ou recommencer avec ce nouveau projet, dont les modifications s'alignent avec les retours étudiants.

Avant de démarrer votre travail sur ce projet, nous vous conseillons de :

- 1. Lire le scénario en entier, chaque section du projet ainsi que les documents fournis ;
- 2. Consulter <u>le guide des étapes clés</u> avec des recommandations et des ressources pour organiser votre travail ;
- 3. Préparer une liste de questions pour votre première session de mentorat.

Scénario

Vous êtes un développeur junior depuis deux mois et vous travaillez en freelance pour écrire des scripts simples afin d'aider les petites entreprises locales à gérer leur inventaire.

Elie, votre amie et elle aussi développeuse Python, est membre du club d'échecs local. Elle vous a expliqué que les tournois du club sont actuellement organisés à la main, mais que les membres du club aimeraient pouvoir les gérer avec un logiciel. Le club avait trouvé une application convenable, mais elle avait besoin d'une connexion Internet – qui n'est pas disponible pour tous les tournois – et l'abonnement mensuel était trop cher.



Vous suggérez que vous pourriez écrire un outil qui permette de gérer les tournois pour aider le club, mais qui fonctionne hors ligne. Elie aime bien votre idée et elle dit qu'elle en discutera la prochaine

fois qu'elle se rendra au club. Comme elle s'occupe déjà de plusieurs contrats en freelance, elle accepte de vous recommander comme candidat idéal pour développer cette application.



Vous recevez un mail d'Elie:

De: Elie **À:** Vous

Objet : logiciel de tournoi d'échecs

Salut!

J'ai parlé aux autres membres du club d'échecs de la création de notre propre logiciel de gestion de tournoi et ils sont très intéressés. Comme je n'ai pas le temps de travailler sur ce projet, ils seraient heureux si tu pouvais les aider.

Le point le plus important est que le logiciel fonctionne hors ligne, au cas où le site ne dispose pas d'une connexion à Internet. Nous avons également besoin d'un moyen facile de sauvegarder et de revoir les résultats des tournois.

Je sais bien que tu as écrit des scripts et logiciels Python simples et c'est une bonne opportunité pour toi de travailler sur un projet plus important. Tu dois utiliser la **programmation orientée objet** pour créer et améliorer les classes qui représentent les entités dans le programme : joueurs, tournois et tours.

Tu devras suivre la conception Modèle-Vue-Contrôleur (MVC) :

- Créer des **classes** qui te serviront de **modèles** pour le tournoi, les joueurs, les matchs et les rondes.
- Utiliser des **vues** pour afficher les classements, les appariements et d'autres statistiques.
- Écrire des **contrôleurs** pour accepter les données de l'utilisateur, produire les résultats des matchs, lancer de nouveaux tournois, etc.

Ton code doit être propre et maintenable, alors assure-toi de suivre les directives de style de code – la **PEP 8** en particulier.

Voici le fichier ci-joint contenant les spécifications techniques que j'ai compilé en discutant avec les organisateurs du club d'échecs. Après mûre réflexion, j'ai décidé qu'il serait mieux d'utiliser le format JSON pour les fichiers de données.

Assure-toi que le repository contient un **README** pour aider le club à utiliser le programme, ainsi qu'un fichier **flake8-html** pour me montrer que le code n'a pas d'erreurs de linting.

Bon courage!

Pièce jointe :

Spécifications techniques

Cette occasion vous permettra de développer vos compétences en Python et de vous faire connaître en tant que développeur d'applications. Avec tout cela en main, vous êtes prêt à commencer!

Pour vous aider à compléter ce projet, vous pouvez suivre <u>ces exemples des étapes clés</u> pour le projet. Vous trouverez des recommandations, des outils ou des ressources utiles pour chaque étape.

Livrables

- 1. Un document TXT contenant le lien vers le **repository GitHub** qui comprend :
 - Le code de l'application, tel que prescrit dans la spécification technique ;
 - Un répertoire contenant un fichier HTML, généré par **flake8-html**, ne montrant aucune erreur de peluchage dans le code ;
 - Un fichier **README.md** contenant des instructions claires sur la manière d'exécuter le programme, de l'utiliser et de générer un nouveau fichier flake8-html.

Pour faciliter votre passage devant le jury, déposez sur la plateforme, dans un dossier zip nommé "*Titre_du_projet_nom_prénom*", tous les livrables du projet comme suit : Nom_Prénom_n° du livrable_nom du livrable_date de démarrage du projet. Cela donnera :

• Nom_Prénom_1_repo_mmaaaa.

Par exemple, ce livrable peut être nommé comme suit : Dupont_Jean_1_repo_012022

Soutenance

Durant la présentation orale, l'évaluateur interprétera le rôle d'Élie, votre amie qui représente le club d'échecs. La soutenance est structurée de la manière suivante :

Présentation des livrables (15 minutes)

- Pendant la mise en œuvre du programme, faites-en la démonstration :
 - un nouveau tournoi peut être créé et joué du début à la fin, avec une mise à jour correcte des points des joueurs;
 - un exemple de rapport peut être généré;
 - l'état du programme peut être sauvegardé et chargé à tout moment entre deux actions de l'utilisateur.
- Vous donnerez un aperçu de la structure du programme et, tout en montrant des exemples tirés de votre code, vous indiquerez où et comment vous avez adhéré au modèle de conception MVC.

• Discussion (10 minutes)

 L'évaluateur jouera le rôle d'Élie, la représentante du club d'échecs. Il vous challengera sur votre méthode et vos livrables. • Il peut également vous demander de montrer une partie du programme que vous n'avez pas démontrée dans votre présentation.

• Debrief (5 minutes)

• À la fin de la soutenance, l'évaluateur arrêtera de jouer le rôle d'Élie, pour vous permettre de débriefer ensemble.

Votre présentation devrait durer 15 minutes (+/- 5 minutes). Puisque le respect des durées des présentations est important en milieu professionnel, les présentations en dessous de 10 minutes ou au-dessus de 20 minutes peuvent être refusées.

Compétences évaluées



Structurer le code d'un programme Python en utilisant un design pattern



Écrire un code Python robuste en utilisant la PEP 8



Utiliser la programmation orientée objet pour développer un programme Python



