
PYTHON STRIKE FOR PERFECTION

MOIS 1

Semaine 1 : Projets de renforcement des bases

Jour 1 : Calculatrice simple

- Crée une calculatrice capable d'effectuer les 4 opérations de base (addition, soustraction, multiplication, division).
- Ajoute la gestion des erreurs, par exemple division par zéro.

Jour 2 : Analyseur de texte

- Construis un programme qui analyse un texte donné :
 - Compte le nombre de mots, de phrases et de caractères.
 - Identifie les mots les plus fréquents.

Jour 3 : Générateur de mots de passe

- Développe un générateur de mots de passe aléatoires avec les options suivantes :
 - Longueur personnalisée.
 - Inclure des majuscules, des chiffres, et des caractères spéciaux.

Jour 4 : Convertisseur d'unités

- Crée un outil permettant de convertir des unités (par exemple, température : Celsius ↔ Fahrenheit, distance : kilomètres ↔ miles).
- Ajoute un menu interactif pour choisir les conversions.

Jour 5 : Jeu "Devine le nombre"

- Programme un jeu où l'utilisateur doit deviner un nombre aléatoire généré par l'ordinateur.
- Fournis des indices "plus haut" ou "plus bas".

Week-end : Repos ou amélioration des projets.

Semaine 2 : Projets intermédiaires

Jour 1 : Gestionnaire de tâches (To-Do List)

- Crée une application en console permettant de :
 - Ajouter, afficher et supprimer des tâches.
 - Marquer des tâches comme terminées.

Jour 2 : Simulation de lancer de dés

- Programme une simulation de lancer de dés :
 - Permet de choisir le nombre de dés et leur type (6 faces, 20 faces, etc.).
 - Affiche les résultats et le total.

Jour 3 : Générateur de questions de mathématiques

- Crée un programme qui génère des questions aléatoires de calcul mental (addition, multiplication, etc.) et évalue les réponses de l'utilisateur.

Jour 4 : Convertisseur de devises

- Connecte-toi à une API (par exemple, ExchangeRatesAPI) pour convertir des montants entre différentes devises.
- Assure-toi de gérer les erreurs de connexion.

Jour 5 : Jeu du pendu

- Programme le jeu du pendu en console.
 - Choisis un mot aléatoire dans une liste prédéfinie.
 - Permet plusieurs essais avant de perdre.

Week-end : Peaufine les projets ou explore une librairie Python (par exemple, pandas ou matplotlib).

Semaine 3 : Projets avancés

Jour 1 : Analyse de fichiers CSV

- Utilise la librairie pandas pour lire un fichier CSV et :
 - Analyser des données (somme, moyenne, maximum, etc.).
 - Visualiser des résultats simples avec matplotlib.

Jour 2 : Gestionnaire de contacts

- Programme une application permettant de :
 - Ajouter, supprimer et rechercher des contacts.
 - Sauvegarder et charger les contacts dans un fichier JSON.

Jour 3 : Horloge numérique

- Crée une horloge numérique en temps réel en utilisant la librairie tkinter.
 - Ajoute un design simple et une mise à jour dynamique de l'heure.

Jour 4 : Web scraper

- Utilise BeautifulSoup pour scraper les titres et liens d'un site web (par exemple, des articles d'actualité).
- Stocke les résultats dans un fichier texte ou JSON.

Jour 5 : Analyse de texte avancée

- Développe un programme capable de :
 - Analyser la tonalité d'un texte (positif, négatif, neutre).
 - Utilise une librairie comme TextBlob ou nltk.

Week-end : Documente tes projets ou partage-les sur GitHub.

Semaine 4 : Projets mini-applications

Jour 1 : Bot Telegram

- Crée un bot Telegram en utilisant python-telegram-bot.
 - Configure des réponses automatiques simples.

Jour 2 : Quiz interactif

- Conçois un quiz interactif avec des questions à choix multiple.
 - Enregistre les scores et affiche les résultats à la fin.

Jour 3 : Gestionnaire de notes personnelles

- Crée une application permettant de :
 - Ajouter, modifier, supprimer et rechercher des notes.
 - Sauvegarder les notes dans un fichier local.

Jour 4 : Application météo

- Utilise une API météo (par exemple, OpenWeatherMap) pour afficher :
 - La température actuelle et les prévisions pour une ville donnée.

Jour 5 : Déploie un projet

- Choisis un des projets et déploie-le en ligne avec Flask ou Streamlit.
 - Héberge-le sur un service gratuit comme Heroku ou Streamlit Cloud.

Week-end : Revue et bilan.

Voici un programme pour un mois de projets plus avancés, conçu pour consolider tes acquis et aborder des concepts plus complexes en Python. L'objectif est d'élargir tes compétences en intégrant des outils comme les bases de données, les API avancées, et des bibliothèques spécifiques pour le développement ou la data.

MOIS 2

Semaine 1 : Applications pratiques et gestion de données

Jour 1 : Gestionnaire de bibliothèque

- Crée une application en console permettant de :
 - Ajouter des livres (titre, auteur, année).
 - Rechercher un livre par différents critères.
 - Sauvegarder les données dans une base SQLite.

Jour 2 : Analyse de logs

- Développe un programme pour analyser des fichiers logs (format texte) d'un serveur.
 - Identifie les IP les plus fréquentes.
 - Analyse les codes de réponse (200, 404, etc.).

Jour 3 : Dashboard interactif (avec Streamlit)

- Crée un tableau de bord simple pour visualiser des données CSV.

- Affiche des graphiques (histogrammes, diagrammes circulaires).
- Permet des filtres interactifs.

Jour 4 : Gestionnaire d'inventaire

- Programme une application de gestion d'inventaire avec :
 - Ajout, suppression et modification de produits.
 - Gestion des stocks faibles.
 - Base SQLite pour stocker les données.

Jour 5 : Explorateur de fichiers

- Crée un explorateur de fichiers en console ou avec tkinter.
 - Permet de naviguer dans les dossiers, ouvrir des fichiers, et afficher des informations (taille, type, date de modification).
-

Semaine 2 : Projets axés sur le web et les APIs

Jour 1 : Scraper avancé avec Selenium

- Programme un scraper pour extraire des données dynamiques d'un site web (par exemple, les prix d'un e-commerce).
 - Ajoute une gestion des cookies et des délais aléatoires.

Jour 2 : Application de gestion de tâches avec Flask

- Développe une application web avec Flask :
 - Ajoute des fonctionnalités pour créer et gérer des tâches.
 - Sauvegarde les données dans une base SQLite.

Jour 3 : Bot Twitter avec tweepy

- Programme un bot qui :
 - Publie automatiquement un tweet chaque jour.
 - Récupère et répond aux mentions.

Jour 4 : Analyse de données en temps réel

- Utilise une API publique (par exemple, Binance pour les cryptomonnaies) pour afficher des données mises à jour en temps réel.
 - Affiche les tendances sous forme de graphique.

Jour 5 : API REST personnelle

- Crée une API REST avec Flask ou FastAPI pour gérer un service (par exemple, un carnet d'adresses).
 - Ajoute des routes pour créer, lire, mettre à jour, et supprimer des données.
-

Semaine 3 : Data Science et Machine Learning

Jour 1 : Visualisation avancée avec matplotlib et seaborn

- Utilise un dataset réel (par exemple, Iris ou Titanic).
 - Affiche des graphiques avancés (boxplot, heatmap, etc.).
 - Explique les insights issus des visualisations.

Jour 2 : Analyse exploratoire de données

- Charge un dataset complexe (par exemple, house-prices sur Kaggle).
 - Analyse les corrélations, les valeurs manquantes, et les distributions.

Jour 3 : Modèle de régression linéaire

- Programme un modèle de régression linéaire avec scikit-learn pour prédire une variable (par exemple, les prix des maisons).
 - Évalue la performance du modèle (RMSE, R^2).

Jour 4 : Clustering avec K-Means

- Applique le clustering sur un dataset (par exemple, des clients d'un e-commerce).
 - Visualise les clusters obtenus.

Jour 5 : Détection de spam

- Programme un détecteur de spam en utilisant un modèle NLP (avec sklearn ou nltk).
 - Entraîne le modèle sur un dataset d'emails.
-

Semaine 4 : Déploiement et automatisation

Jour 1 : Déploiement d'une application Flask

- Déploie une application Flask sur Heroku ou Render.
 - Ajoute une base de données PostgreSQL en backend.

Jour 2 : Automatisation avec celery

- Programme une application qui planifie des tâches en arrière-plan (par exemple, envoyer un email à une heure précise).

Jour 3 : Analyse de sentiment avec une API

- Utilise l'API de sentiment d'AWS Comprehend ou TextBlob.
 - Analyse les sentiments d'un ensemble de tweets ou de commentaires.

Jour 4 : Gestion d'images avec Pillow

- Crée une application qui :
 - Redimensionne, rogne et applique des filtres à des images.
 - Sauvegarde les images modifiées.

Jour 5 : Programme une mini-IA avec ChatGPT API

- Utilise l'API de OpenAI pour :
 - Créer un assistant virtuel qui répond à des questions basiques.
 - Enregistre les conversations pour une analyse future.

Week-end : Revue et consolidation

- Rédige une documentation pour tes projets.
- Publie tes meilleurs projets sur GitHub.
- Identifie les points que tu souhaites approfondir le mois suivant.

Avec ce programme, tu pourras travailler sur des concepts avancés comme les bases de données, le scraping dynamique, les modèles de machine learning, et le déploiement d'applications. Cela te donnera une solide expérience pratique pour tes futurs projets.

Voici un programme encore plus avancé pour le mois suivant, axé sur des projets à impact concret, la maîtrise des technologies connexes (bases de données, machine learning avancé, déploiement), et l'utilisation de frameworks et outils pour créer des applications robustes.

MOIS 3

Semaine 1 : Développement d'applications complexes

Jour 1 : Gestionnaire de projets complet

- Développe une application web avec Django ou Flask qui permet :
 - La gestion de projets (tâches, deadlines, assignation).
 - Une authentification utilisateur (inscription, connexion).
 - Une base PostgreSQL pour stocker les données.

Jour 2 : Intégration d'API tierces

- Intègre une API externe (par exemple, Google Calendar ou Slack) à l'application précédente pour envoyer des rappels ou synchroniser des tâches.
- Implémente une gestion des erreurs robuste.

Jour 3 : Application de messagerie instantanée

- Programme une application de chat en temps réel avec Flask-SocketIO ou Django Channels.
 - Implémente des fonctionnalités de base comme les salons de discussion.
 - Sauvegarde les messages dans une base de données.

Jour 4 : Visualisation interactive avec Dash

- Crée un tableau de bord interactif pour analyser un dataset complexe (par exemple, les ventes d'une entreprise).
 - Implémente des graphiques dynamiques, des filtres, et des indicateurs clés.

Jour 5 : Gestionnaire de fichiers avancé

- Programme une application web permettant d'uploader, télécharger, et organiser des fichiers.
 - Implémente une prévisualisation des fichiers et des restrictions sur leur taille/type.
-

Semaine 2 : Machine Learning avancé

Jour 1 : Réseaux de neurones avec TensorFlow ou PyTorch

- Programme un modèle pour reconnaître des chiffres manuscrits (dataset MNIST).
 - Implémente un réseau simple avec 2 à 3 couches.
 - Ajoute une évaluation de performance.

Jour 2 : Système de recommandation

- Crée un système de recommandation pour un dataset d'e-commerce (par exemple, MovieLens).
 - Implémente un modèle de filtrage collaboratif ou basé sur le contenu.

Jour 3 : Modèle NLP avancé

- Entraîne un modèle de classification de texte (par exemple, détection de toxicité dans les commentaires).
 - Utilise Hugging Face Transformers pour appliquer un modèle pré-entraîné comme BERT.

Jour 4 : Analyse des séries temporelles

- Analyse un dataset de séries temporelles (par exemple, données météorologiques ou boursières).
 - Implémente un modèle ARIMA ou LSTM pour prédire des tendances.

Jour 5 : Génération d'images avec GANs

- Entraîne un GAN (Generative Adversarial Network) simple pour générer des images basiques (par exemple, des chiffres manuscrits).
 - Utilise TensorFlow ou PyTorch pour l'implémentation.

Semaine 3 : Cloud et déploiement avancé

Jour 1 : Déploiement d'un modèle ML

- Déploie un modèle ML sur une API Flask ou FastAPI.
 - Héberge l'application sur AWS ou Google Cloud.
 - Implémente une interface simple pour interagir avec le modèle.

Jour 2 : Infrastructure as Code (IaC)

- Apprends les bases de Terraform ou CloudFormation pour créer une infrastructure cloud.
 - Programme une configuration pour déployer automatiquement un serveur d'application.

Jour 3 : Pipelines CI/CD avec GitHub Actions

- Configure un pipeline CI/CD pour automatiser le test et le déploiement d'une application.
 - Implémente des tests unitaires et de déploiement automatique sur Heroku ou Docker Hub.

Jour 4 : Conteneurisation avec Docker

- Conteneurise une application web ou ML complète.
 - Crée un docker-compose.yml pour gérer plusieurs services (application, base de données, etc.).

Jour 5 : Kubernetes de base

- Crée un cluster Kubernetes local avec Minikube.
 - Déploie une application conteneurisée avec plusieurs pods et un load balancer.

Semaine 4 : Applications complètes et automatisation

Jour 1 : Développement d'un site e-commerce minimaliste

- Programme un site e-commerce avec Django ou Flask.
 - Fonctionnalités : gestion des produits, panier, et commandes.
 - Intègre un système de paiement avec Stripe ou PayPal.

Jour 2 : Application de reconnaissance faciale

- Implémente une application qui :
 - Utilise OpenCV et un modèle pré-entraîné pour détecter des visages dans des images ou des vidéos.
 - Enregistre les résultats dans une base de données.

Jour 3 : Automatisation de tests avec Selenium

- Programme des tests automatisés pour un site web (par exemple, naviguer, remplir un formulaire, et vérifier les résultats).

Jour 4 : Projet personnel avancé

- Planifie et commence un projet personnel ambitieux :
 - Une application SaaS.
 - Un jeu interactif en Python.
 - Un outil d'automatisation pour un besoin spécifique.

Jour 5 : Finalisation et déploiement du projet personnel

- Termine et déploie ton projet personnel.
 - Documente le projet et prépare une présentation pour le partager avec d'autres (GitHub, portfolio).

Week-end : Revue et objectifs futurs

- Analyse les compétences que tu as acquises.
- Publie des articles ou vidéos pour présenter tes projets (partage d'expérience).
- Élabore un plan pour approfondir un domaine particulier (Data Science, DevOps, NLP, etc.).

Ce programme est conçu pour t'amener à un niveau avancé tout en te préparant à des problématiques concrètes dans le monde professionnel. Avec ces projets, tu maîtriseras non seulement Python, mais aussi des concepts complexes liés au cloud, à l'apprentissage automatique, et au déploiement d'applications.

Voici un programme pour un quatrième mois encore plus avancé, axé sur la création de projets sophistiqués, la résolution de problèmes complexes, et l'exploration des concepts de pointe en intelligence artificielle, cloud computing, et développement logiciel à grande échelle.

Mois 4

Semaine 1 : Intelligence artificielle et systèmes complexes

Jour 1 : Chatbot conversationnel avancé

- Crée un chatbot avec les modèles pré-entraînés de Hugging Face (GPT, BERT).
 - Ajoute des réponses contextuelles et une capacité d'apprentissage continu (fine-tuning).
 - Implémente une interface web interactive avec Gradio.

Jour 2 : Vision par ordinateur avec YOLO

- Implémente un détecteur d'objets avec YOLOv5.
 - Applique-le sur des vidéos en temps réel pour identifier des objets comme des voitures, des piétons, etc.

Jour 3 : Réseaux de neurones convolutionnels avancés (CNN)

- Entraîne un CNN pour classifier des images plus complexes (par exemple, CIFAR-100).
 - Implémente des techniques de régularisation (Dropout, Batch Normalization).

Jour 4 : Algorithmes génétiques

- Programme une solution basée sur un algorithme génétique pour résoudre un problème d'optimisation (par exemple, le problème du voyageur de commerce).

Jour 5 : Système expert avec règles

- Programme un système expert pour un domaine précis (par exemple, diagnostic médical ou assistance technique).
 - Utilise des bibliothèques comme experta pour construire des règles complexes.

Semaine 2 : Big Data et systèmes distribués

Jour 1 : Traitement de Big Data avec PySpark

- Analyse un dataset massif avec PySpark.
 - Implémente une transformation des données et des calculs distribués.
 - Sauvegarde les résultats dans un format Parquet.

Jour 2 : Streaming de données en temps réel

- Crée un pipeline de streaming avec Kafka et PySpark.
 - Traite des données en temps réel (par exemple, flux de clics sur un site).

Jour 3 : Base de données NoSQL avec MongoDB

- Développe une application qui stocke des données non structurées dans MongoDB.
 - Implémente des recherches avancées (indexation, filtres complexes).

Jour 4 : Intégration des bases relationnelles et NoSQL

- Crée un projet combinant PostgreSQL pour les données structurées et MongoDB pour les données non structurées.
 - Implémente une API REST pour interagir avec ces bases.

Jour 5 : Introduction à Hadoop

- Configure un cluster Hadoop local.
 - Implémente un job MapReduce simple pour traiter un dataset.

Semaine 3 : DevOps et automatisation avancée

Jour 1 : Microservices avec Kubernetes

- Crée une architecture basée sur des microservices déployés sur Kubernetes.
 - Implémente un service pour gérer des utilisateurs et un autre pour gérer des produits.
 - Utilise un ingress pour gérer le trafic.

Jour 2 : Observabilité avec Prometheus et Grafana

- Configure Prometheus pour surveiller une application Python.
 - Affiche des métriques (temps de réponse, utilisation des ressources) sur un tableau de bord Grafana.

Jour 3 : Automatisation des déploiements avec Jenkins

- Configure un pipeline Jenkins pour :
 - Tester, conteneuriser, et déployer automatiquement une application.

Jour 4 : Déploiement serverless

- Déploie une fonction serverless avec AWS Lambda ou Google Cloud Functions.
 - Implémente une fonction de traitement d'images ou d'analyse de texte.

Jour 5 : Infrastructure avancée avec Terraform

- Programme une infrastructure cloud complète :
 - Crée un cluster Kubernetes, une base de données, et des configurations réseau.
-

Semaine 4 : Projets interconnectés et recherche avancée

Jour 1 : Système de recommandations en production

- Déploie un système de recommandations (livres, films, produits) avec Flask ou FastAPI.
 - Entraîne un modèle de filtrage collaboratif et implémente des recommandations personnalisées.

Jour 2 : Projet NLP avec un modèle pré-entraîné

- Fine-tune un modèle GPT ou T5 pour répondre à un problème spécifique, comme :
 - Génération de rapports.
 - Résumé automatique de textes longs.

Jour 3 : Automatisation complexe avec Python et cloud

- Crée un système automatisé qui collecte des données depuis le web, les nettoie, et les charge dans une base cloud (par exemple, BigQuery).
 - Intègre des notifications Slack pour le suivi.

Jour 4 : Application de Deep Reinforcement Learning

- Implémente un algorithme DQN (Deep Q-Network) pour résoudre un problème de contrôle, comme jouer à un jeu simple (Pac-Man, par exemple).

Jour 5 : Projet open-source collaboratif

- Contribue à un projet open-source sur GitHub.

- Propose une nouvelle fonctionnalité, corrige un bug ou améliore la documentation.

Week-end : Bilan et vision à long terme

- Documente les projets du mois avec des rapports et des démonstrations.
- Publie tes réalisations sur ton portfolio ou ton GitHub.
- Planifie des objectifs pour les prochains mois :
 - Approfondir un domaine (vision par ordinateur, NLP, systèmes distribués).
 - Apprendre un langage complémentaire (Scala, Rust).
 - Explorer davantage les architectures cloud et la gestion à grande échelle.

Avec ce programme, tu auras abordé des concepts avancés comme les systèmes distribués, les modèles de machine learning à la pointe, et les architectures cloud modernes. Cela te prépare non seulement à des projets complexes, mais aussi à des problématiques réelles rencontrées en entreprise.

Pour un cinquième mois encore plus avancé, nous allons nous concentrer sur des projets intégrant des technologies de pointe, des algorithmes complexes, et des pratiques de développement logiciel à grande échelle. Ce programme est conçu pour repousser les limites de tes compétences et viser des réalisations proches des standards professionnels.

MOIS 5

Semaine 1 : Intelligence artificielle de pointe

Jour 1 : Modèles de diffusion (Diffusion Models)

- Implémente un modèle de diffusion pour générer des images ou des données synthétiques.
 - Utilise PyTorch ou TensorFlow.

- Apprends les bases des architectures comme DALL-E ou Stable Diffusion.

Jour 2 : Applications avancées en NLP

- Programme un modèle de traduction automatique (seq2seq avec attention).
 - Compare les résultats d'un modèle entraîné à partir de zéro et d'un modèle pré-entraîné comme MarianMT.

Jour 3 : Transformer de zéro

- Implémente un Transformer (architecture de base de GPT/BERT).
 - Concentre-toi sur la compréhension de l'auto-attention et de la normalisation.

Jour 4 : Entraînement distribué

- Configure l'entraînement distribué d'un modèle ML sur plusieurs GPU ou machines avec PyTorch Lightning ou Horovod.

Jour 5 : Application multimodale

- Construis un modèle multimodal qui associe texte et images (par exemple, texte vers image).
 - Utilise CLIP ou une architecture similaire.

Semaine 2 : Systèmes distribués et cloud avancé

Jour 1 : Traitement de flux avec Apache Flink

- Implémente un pipeline de traitement de données en flux avec Apache Flink.
 - Analyse des données en temps réel, comme un flux Twitter ou des logs système.

Jour 2 : Cluster Kubernetes avancé

- Configure un cluster Kubernetes sur AWS EKS ou Google GKE.
 - Implémente une architecture hautement résiliente avec des auto-scalers.

Jour 3 : Déploiement d'une architecture microservices complexe

- Déploie une application composée de plusieurs microservices :

- Service de gestion utilisateur.
- Service de recommandations.
- Service de facturation.
- Utilise un gestionnaire d'API comme Istio.

Jour 4 : Base de données distribuée avec Cassandra

- Configure et utilise Apache Cassandra pour gérer des données distribuées à grande échelle.
 - Implémente un système pour stocker et analyser des logs massifs.

Jour 5 : Infrastructure complète avec Terraform et Ansible

- Crée une infrastructure cloud entièrement automatisée avec Terraform.
 - Utilise Ansible pour configurer les machines virtuelles et les services.

Semaine 3 : Projets d'impact et algorithmes complexes

Jour 1 : Détection de fraude avec ML

- Développe un modèle pour détecter les fraudes financières à partir de données transactionnelles.
 - Combine des modèles supervisés (Random Forest, XGBoost) et non supervisés (Isolation Forest).

Jour 2 : Simulation basée sur des agents

- Implémente une simulation d'agents pour résoudre un problème complexe (exemple : gestion du trafic urbain).
 - Utilise mesa ou construis une solution personnalisée.

Jour 3 : Algorithmes de graphes avancés

- Implémente des algorithmes comme PageRank ou recherche de plus courts chemins dans un graphe.
 - Applique-les à des données réelles (exemple : graphe de réseau social).

Jour 4 : Systèmes recommandateurs hybrides

- Crée un système recommandateur combinant filtrage collaboratif, contenu, et approches hybrides.

- Déploie ce système sur une application web.

Jour 5 : Modèle RL pour un environnement complexe

- Implémente un modèle d'apprentissage par renforcement pour un environnement complexe (par exemple, gestion de stocks ou robotique).
 - Utilise une bibliothèque comme Stable-Baselines3.
-

Semaine 4 : Développement logiciel à grande échelle

Jour 1 : Optimisation et scalabilité

- Optimise une application Python pour gérer un grand volume de données (profilage, parallélisme avec multiprocessing ou asyncio).

Jour 2 : Application SaaS

- Conçois une application SaaS complète :
 - Implémente une gestion multi-tenant (plusieurs clients avec des données séparées).
 - Ajoute un système de facturation par abonnement.

Jour 3 : Frameworks ML spécialisés

- Explore des frameworks ML spécialisés comme LightGBM, CatBoost ou AutoML.
 - Crée un modèle AutoML pour sélectionner automatiquement les meilleurs hyperparamètres.

Jour 4 : Sécurité logicielle avancée

- Implémente des pratiques de sécurité avancées :
 - Ajoute des tests de sécurité automatisés.
 - Configure un système de surveillance et d'alertes pour les menaces.

Jour 5 : Collaboration open-source avancée

- Contribue à un projet open-source complexe.
 - Implémente une fonctionnalité importante ou corrige une faille critique.
-

Week-end : Intégration et bilan

Jour 1 : Intégration des compétences

- Combine toutes les compétences apprises en créant un projet final ambitieux (par exemple, une plateforme de gestion intégrée avec IA, visualisation, et déploiement cloud).

Jour 2 : Rétrospective et planification future

- Identifie les domaines à approfondir ou les compétences complémentaires à apprendre (Rust, Scala, Spark, systèmes embarqués, etc.).
 - Prépare ton portfolio pour démontrer tes projets avancés à des employeurs ou clients potentiels.
-

Résultats attendus :

- Maîtrise des systèmes distribués, des algorithmes complexes, et des architectures modernes.
- Capacité à développer et déployer des solutions professionnelles et innovantes.
- Portfolio impressionnant démontrant des projets concrets et des compétences avancées.

Les ressources à utiliser

Mois 1 : Renforcement des bases et projets simples

1. Automatisation de tâches (Semaine 1)

- **Ressource** : [Automate the Boring Stuff with Python](#) (Livre et cours en ligne).
- **Ressource YouTube** : Chaîne [Tech With Tim](#) pour des projets simples en Python.

2. Bases des APIs et web scraping (Semaine 2)

- **Ressource** : Documentation de Requests (lien officiel) et BeautifulSoup (lien officiel).
- **Tutoriel vidéo** : Web Scraping with Python.

3. Pandas et visualisation (Semaine 3)

- **Ressource** : Pandas Official Documentation et Matplotlib Guide.
- **Cours** : [Data Analysis with Python - freeCodeCamp](#).

4. Projets simples (Semaine 4)

- **Ressource** : Python Projects on Real Python.
-

Mois 2 : Projets intermédiaires

1. SQL et bases de données relationnelles (Semaine 1)

- **Ressource** : [SQL Tutorial](#).
- **Cours** : [Database Management with PostgreSQL](#).

2. API avancées et FastAPI (Semaine 2)

- **Ressource** : FastAPI Documentation.
- **Tutoriel vidéo** : [FastAPI Full Course](#).

3. Programmation orientée objet et design patterns (Semaine 3)

- **Ressource** : Dive Into Design Patterns.
- **Cours** : Python OOP Tutorial on Real Python.

4. Analyse de données avancée (Semaine 4)

- **Ressource** : Kaggle Courses - Pandas, NumPy.
 - **Projet pratique** : Datasets Kaggle (lien).
-

Mois 3 : Projets avancés et concepts de machine learning

1. Machine Learning avec Scikit-learn (Semaine 1)

- **Ressource** : Scikit-learn Official Documentation.
- **Cours complet** : [ML avec Python - freeCodeCamp](#).

2. Deep Learning avec TensorFlow ou PyTorch (Semaine 2)

- **Ressource** : PyTorch Official Tutorials ou TensorFlow Guide.
- **Cours vidéo** : [Deep Learning Specialization - Coursera](#).

3. Traitement du langage naturel (NLP) (Semaine 3)

- **Ressource** : [Natural Language Processing with Python \(NLTK\)](#).
- **Projet pratique** : NLP avec Hugging Face.

4. Projets ML déployés (Semaine 4)

- **Ressource** : Flask for ML deployment.
 - **Cours** : Deploy Machine Learning Models - Udemy.
-

Mois 4 : Concepts avancés et projets professionnels

1. Vision par ordinateur (Semaine 1)

- **Ressource** : OpenCV Tutorials.
- **Tutoriel vidéo** : [Computer Vision with Python](#).

2. Big Data et PySpark (Semaine 2)

- **Ressource** : [PySpark Documentation](#).
- **Cours** : Data Engineering with PySpark - Udacity.

3. Microservices et Kubernetes (Semaine 3)

- **Ressource** : Kubernetes Basics.
- **Cours vidéo** : Microservices with Kubernetes - KodeKloud.

4. Projets NLP avancés (Semaine 4)

- **Ressource** : Hugging Face Tutorials.
 - **Cours spécialisé** : [NLP Specialization - Coursera](#).
-

Mois 5 : Niveau expert

1. Modèles de diffusion et multimodalité (Semaine 1)

- **Ressource** : [DALL-E Tutorials](#).
- **Tutoriel vidéo** : [Deep Learning for Generative Models](#).

2. Systèmes distribués et Big Data avancé (Semaine 2)

- **Ressource** : [Apache Flink Documentation](#).
- **Cours** : Distributed Systems with Kubernetes - edX.

3. Deep Reinforcement Learning (Semaine 3)

- **Ressource** : [Spinning Up in Deep RL](#).
- **Tutoriel** : [Deep Reinforcement Learning Explained](#).

4. Systèmes avancés et sécurité logicielle (Semaine 4)

- **Ressource** : [OWASP Security Guides](#).
- **Cours** : [Advanced Cloud Security - Coursera](#).