

Construisez votre premier projet IA de A à Z

Méthode + Pratique - Jour 2



Rappel du Jour 1 et mise en contexte

Les 4 approches vues hier

1 Prompting

Instructions directes aux modèles

2 RAG

Retrieval Augmented Generation

3 Agents

Systèmes autonomes avec outils

4 Fine-tuning

Adaptation des modèles à vos données

⊗ Alerte MIT/Fortune

95% des projets pilotes en GenAI échouent

Causes principales :

- Cadrage flou
- Absence de métriques
- Manque d'intégration réelle

Objectif du jour

Structurer et démarrer concrètement

1. Apprendre la méthodologie

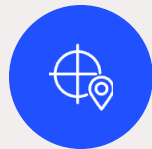
Maîtriser les 7 étapes clés pour cadrer et exécuter un projet IA réussi

2. Passer à la pratique

Démarrer votre application en "vibe coding" avec Cursor

- Mise en place de l'environnement
- Structure du projet
- Premières lignes de code

Les 7 étapes clés pour créer un projet IA



1. Cadrage stratégique

Définir un problème précis et un objectif mesurable avec KPIs clairs



2. Constitution des données

Identifier et nettoyer FAQ, rapports, emails, bases de données existantes



3. Choix de l'approche

Sélectionner entre Prompting, RAG, Agent ou Fine-tuning selon votre cas d'usage



4. Prototype ciblé

Une seule fonctionnalité critique - éviter de viser trop large

Les 7 étapes clés (suite)



5. Tests et itération

Feedback utilisateur rapide, principe 80/20. Validez la robustesse avant d'élargir le périmètre.



6. Déploiement MVP

Mise en production limitée (5-10 utilisateurs pilotes) pour valider l'usage réel.



7. Surveillance continue

Suivi des performances, des coûts, et adaptation progressive selon les retours.

Interaction en direct

Partagez votre cas d'usage, votons pour l'approche IA adaptée, et définissons ensemble un KPI simple de succès.

Créer un projet IA en vibe coding avec Cursor

Arborescence recommandée

```
mlxpress-app/  
├─ app/|  
  │─ main.py    # logique métier|  
  └─ ui.py      # interface  
├─ configs/|  
  └─ settings.example.env|  
├─ data/        # fichiers d'exemple  
├─ outputs/     # résultats générés  
├─ tests/|  
├─ requirements.txt  
├─ .gitignore  
└─ README.md
```



Cette structure permet une organisation claire des différentes parties de votre application et facilite la collaboration.

Créer l'environnement virtuel

Option Conda

```
conda create -n mlxpress python=3.11 -yconda activate  
mlxpress
```

Option venv

```
python -m venv .venv# macOS / Linuxsource  
.venv/bin/activate# Windows.venv\Scripts\activate
```

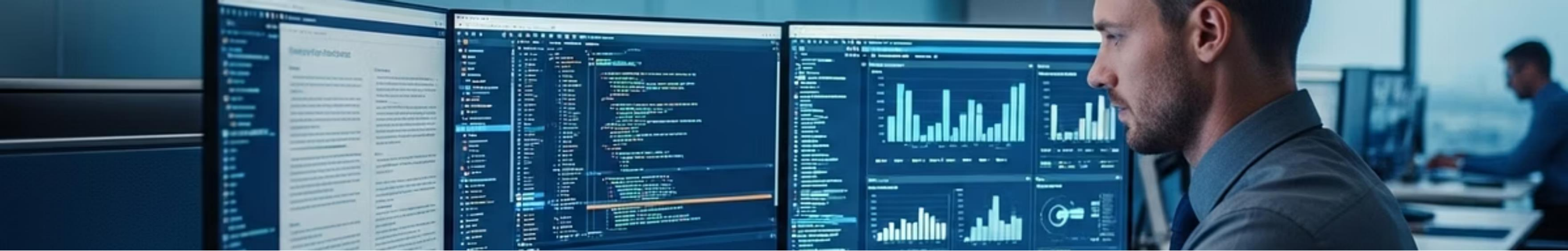
Ajouter ipykernel

```
pip install ipykernelpython -m ipykernel install --user \  
--name=mlxpress \  --display-name "Python (mlxpress)"
```

Dépendances de base (requirements.txt)

```
gradiopython-dotenvopenaiollama  # si usage  
localipykernel
```

Un environnement virtuel garantit l'isolation des dépendances et rend votre projet facilement partageable et reproductible.



Bonnes pratiques de développement

Fichier .env

Stockez vos clés API et paramètres sensibles dans un fichier .env qui ne sera jamais partagé sur GitHub.

```
OPENAI_API_KEY=sk-  
xxx...MODEL_NAME=gpt-  
4DEBUG_MODE=True
```

Fichier .gitignore

Excluez les fichiers sensibles ou inutiles du versionnement.

```
.env.venv/__pycache__/outputs/*.py  
c
```

README.md

Documentez clairement comment installer, configurer et utiliser votre projet.

```
# MLXpressApplication IA pour  
[votre cas d'usage]##  
Installation...
```


Votre mission pour aujourd'hui

Créer votre environnement

Choisissez conda ou venv et ajoutez
ipykernel

Préparer Cursor

Formulez vos premiers prompts de
vibe coding



Installer les dépendances

Utilisez le fichier requirements.txt
fourni

Monter l'arborescence

Suivez la structure présentée pour
votre projet

Partagez vos avancées

Dans le groupe WhatsApp :

- 1 Une [capture d'écran](#) de votre arborescence projet complète
- 2 Le [cas d'usage](#) que vous souhaitez prototyper en détail
- 3 Vos [questions](#) éventuelles sur la structure ou l'approche

Nous analyserons ensemble les propositions les plus prometteuses et identifierons les ajustements nécessaires pour la suite du bootcamp.

Conseil pratique

Commencez petit mais pensez grand ! Un MVP fonctionnel vaut mieux qu'un projet ambitieux inachevé.

