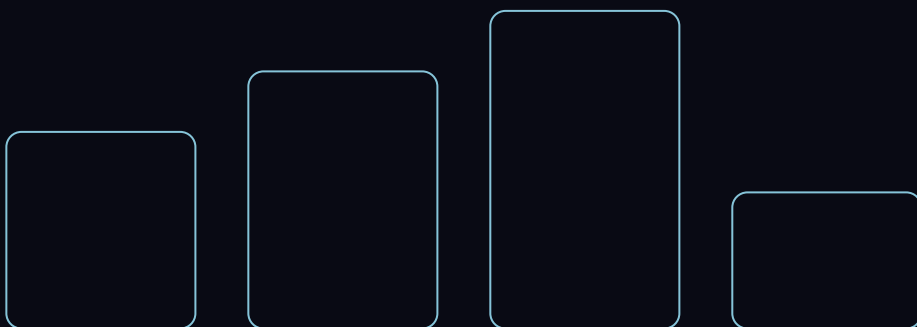


Édition Future.

---

# Créer une petite application avec l'IA

Un prototype local complet pour apprendre à demander et vérifier du code.



---

GUIDE PRATIQUE

EXEMPLES ET EXERCICES CORRIGÉS

# Votre parcours

Lisez les repères, puis réalisez les exercices avec vos propres données non confidentielles.

Avant de commencer

1. Écrire un besoin observable
2. Comprendre les trois couches
3. Exemple complet
4. Lire avant d'accepter
5. Tester le parcours réel
6. Atelier de correction
7. Ce qui change avant une vraie mise en service

Références

# Avant de commencer

Le « vibe coding » désigne couramment une manière de construire avec des instructions en langage naturel et du code proposé par une IA. Le vocabulaire ne change pas les responsabilités : il faut vérifier ce que fait le programme, ce qu'il enregistre et ce qu'il permet à un utilisateur d'accomplir.

Ce cours propose un petit tableau de tâches personnel, sans compte ni serveur de données. Il fonctionne dans un navigateur et conserve les tâches sur ce navigateur. Ce périmètre convient à l'apprentissage ; il ne constitue pas une application collaborative prête à accueillir des données clients.

Vous aurez besoin d'un éditeur de texte, d'un navigateur et, pour servir le fichier localement, de Python si celui-ci est déjà disponible. N'installez pas un ensemble d'outils dont vous ne comprenez pas le rôle pour réaliser cet exercice.

## 1. Écrire un besoin observable

« Crée une application moderne » ne décrit pas une fonction vérifiable. « Ajouter une tâche, la marquer terminée, la supprimer et retrouver la liste après rechargement » donne un point de départ testable.

**Cas fictif.** Nora prépare les étapes d'un atelier. Elle veut noter « Acheter le papier », « Préparer les outils » et « Imprimer les consignes ». Une autre personne n'a pas besoin d'accéder à cette liste. Elle accepte que les données restent sur un seul navigateur et qu'elles puissent disparaître si le stockage du navigateur est effacé.

Écrivez aussi les limites : aucune notification, aucun paiement, aucune synchronisation et aucune authentification pour ce prototype. Ce ne sont pas des fonctions interdites ; elles demanderaient d'autres choix et d'autres contrôles.

### Demande initiale à une IA de code

Construis un tableau de tâches personnel dans un seul fichier HTML, sans dépendance externe. L'utilisateur peut ajouter une tâche non vide, la terminer, la supprimer et retrouver la liste après rechargement. Utilise le stockage local du navigateur, traite les erreurs de lecture et d'écriture, affiche le texte saisi comme du texte. Prévois un label de champ et des boutons accessibles. Explique comment lancer le fichier et donne des tests avec leurs résultats attendus.

Ce prompt fixe un contrat. Il ne prouve pas que le code rendu le respecte.

## 2. Comprendre les trois couches

Le HTML décrit les éléments de la page : titre, formulaire, champ et liste. Le CSS les présente. Le JavaScript réagit aux actions, met à jour les données et reconstruit la liste visible.

Dans notre exemple, les tâches forment un tableau d'objets. Chaque objet possède un texte et un état terminé. Le stockage local conserve une représentation JSON de ce tableau. Lors du chargement, le programme relit cette représentation et vérifie les données avant de les utiliser.