

# Formation Cursor

Cette formation Cursor s'adresse aux développeurs, prompt engineers et DevOps souhaitant exploiter les atouts de l'IA pour coder plus vite et mieux. Durant ce cours, vous apprendrez à installer et configurer l'IDE Cursor sur votre environnement et à choisir le bon modèle LLM pour vos projets. Vous maîtriserez le chat et la complétion multi-ligne pour générer, refactorer et documenter le code, tout en définissant des règles CursorRules pour garantir style, linting et sécurité. Vous découvrirez comment comparer les performances et coûts de GPT-5, Claude ou Mistral et comment intégrer Cursor à un pipeline CI/CD pour livrer un mini-projet de bout en bout.

## Durée

1 jour

## Objectifs pédagogiques

- ♦ Installer et configurer l'IDE Cursor AI sur son environnement et sélectionner un modèle LLM approprié
- ♦ Utiliser les fonctionnalités de chat et de complétion multiligne de Cursor pour générer, refactoriser et documenter du code dans plusieurs fichiers
- ♦ Développer et appliquer des règles CursorRules personnalisées pour faire respecter les conventions de codage, les pratiques de linting et de sécurité
- ♦ Intégrer Cursor dans un workflow CI/CD basé sur Git (par exemple, GitHub Actions) pour automatiser les pull requests, les tests et les notifications, aboutissant à un mini-projet livré de bout en bout avec l'aide de l'IA.

## Public

Développeurs (front, back, full-stack),  
prompt engineers, ingénieurs DevOps,  
lead techs...

## Prérequis

Expérience pratique d'au moins un langage de programmation et des outils de développement courants, familiarité avec Git/GitHub ou des outils similaires de contrôle de version et pipelines CI/CD, capacité à naviguer dans une interface en ligne de commande et à utiliser un IDE.

## Programme de formation

### Phase d'inclusion

#### Module 1 : Prise en main et installation de Cursor AI

- ◆ Procédure d'installation et d'activation de la licence sur différents OS
- ◆ Tour complet de l'interface : explorateur de code, chat contextuel, panneau diff
- ◆ Choisir un LLM par défaut (GPT-5, Mistral, Claude) et configurer les préférences
- ◆ Comparaison des forces et limites de Cursor par rapport à Copilot ou Claude

**Exemples d'activités pratiques** : installer Cursor sur l'ordinateur du participant et lancer un 'Hello World', personnaliser les raccourcis clavier et le thème de l'éditeur, explorer l'interface et repérer les panneaux diff et chat.

#### Module 2 : Génération et refactoring de code

- ◆ Comprendre la complétion multi-ligne prédictive et l'anticipation contextuelle
- ◆ Prompts courts vs détaillés ; tags @file/@symbol pour le contexte multi-fichiers
- ◆ Fonctions de diff inline et génération de messages de commit IA
- ◆ Génération automatique de tests et de documentation
- ◆ Gestion de projets multi-fichiers et limites de tokens

**Exemples d'activités pratiques** : construire une API CRUD à l'aide du chat Cursor et des prompts, refactoriser un code existant et analyser le diff généré, écrire et exécuter des tests unitaires générés par Cursor.

#### Module 3 : Personnalisation avec CursorRules et prompt engineering

- ◆ Structure et scopes d'un fichier CursorRules (projet, utilisateur, mémoire)
- ◆ Règles de style, linting, architecture et gestion des secrets
- ◆ Partage et versionnement des CursorRules au sein de l'équipe
- ◆ Fondamentaux du prompt engineering : few-shot, chain-of-thought et prompt chaining
- ◆ Gestion du contexte multi-fichiers et bibliothèque de prompts versionnée

**Exemples d'activités pratiques** : écrire un fichier CursorRules pour un projet open source et l'appliquer, concevoir un jeu de prompts avancés pour générer une API REST documentée et ses tests, tester la sécurité avec des exemples de secrets.

#### Module 4 : Comparaison des LLM et vue d'ensemble du mode Agent

- ◆ Présentation des modèles supportés et de leurs différences (vitesse, coût, qualité)
- ◆ Fonctionnement du mode Auto-select vs modèle fixe et monitoring des tokens
- ◆ Vue d'ensemble du mode Agent : YOLO mode sécurisé, allow-list, checkpoints et roll-back
- ◆ Génération de scripts Shell/SQL/Terraform via prompts
- ◆ Exploration du concept de 'vibe coding' et de la transformation maquette → code

**Exemples d'activités pratiques :** benchmarker deux modèles LLM sur une tâche identique, configurer un agent pour automatiser l'initialisation d'un projet React ou API et analyser ses logs, personnaliser allow-lists et checkpoints pour exécuter des scripts Shell de manière sécurisée.

## Module 5 : Intégration CI/CD et collaboration d'équipe

- ◆ Création automatique de pull-requests, tests et merges assistés par Cursor
- ◆ Notifications et mentions ciblées via Slack ou autres outils de collaboration
- ◆ Partage des CursorRules via un repository Git ou gist pour garantir la cohérence
- ◆ Mise en place d'un pipeline complet GitHub Actions avec Cursor Agent
- ◆ Mesurer l'impact et la qualité : feedback de code review, scores de tests, tokens consommés

**Exemples d'activités pratiques :** Configurer un repository GitHub et un workflow CI/CD automatisé avec Cursor Agent, créer une pull-request avec code généré via Cursor et déclencher des tests automatiques, mettre en place des notifications Slack pour les revues de code et mesurer les gains de productivité.

## Module 6 : Projet synthèse et bonnes pratiques

- ◆ Choisir un mini-projet (API Node ou front React) et planifier les étapes
- ◆ Appliquer l'ensemble des modules : installation, prompts, CursorRules, multi-LLM, agent, CI/CD
- ◆ Checklist qualité et sécurité, bonnes pratiques long terme et maintenance
- ◆ Débrief collectif : erreurs fréquentes, retours sur l'expérience et amélioration continue

**Exemples d'activités pratiques :** Développer en groupes un projet from scratch en utilisant Cursor de bout en bout, présenter le projet, les métriques (vitesse, tokens, coûts) et le retour d'expérience, discuter des améliorations possibles et des prochaines étapes d'adoption en entreprise.

## Moyens et méthodes pédagogiques

- ♦ La formation alterne entre présentations des concepts théoriques et mises en application à travers d'ateliers et exercices pratiques (hors formation de type séminaire).
- ♦ Les participants bénéficient des retours d'expérience terrains du formateur ou de la formatrice
- ♦ Un support de cours numérique est fourni aux stagiaires

## Modalités d'évaluation

- ♦ **En amont de la session de formation**, un questionnaire d'auto-positionnement est remis aux participants, afin qu'ils situent leurs connaissances et compétences déjà acquises par rapport au thème de la formation.
- ♦ **En cours de formation**, l'évaluation se fait sous forme d'ateliers, exercices et travaux pratiques de validation, de retour d'observation et/ou de partage d'expérience, en cohérence avec les objectifs pédagogiques visés.
- ♦ **En fin de session**, le formateur évalue les compétences et connaissances acquises par les apprenants grâce à un questionnaire reprenant les mêmes éléments que l'auto-positionnement, permettant ainsi une analyse détaillée de leur progression.