

OBJECTIFS DE CETTE FORMATION :

À l'issue de la formation, les participants seront capables de :

- comprendre le fonctionnement et les limites des modèles de langage ;
- identifier les principes fondamentaux du prompt engineering ;
- rédiger des prompts clairs, structurés et adaptés aux tâches de développement ;
- utiliser les principales techniques de prompting pour améliorer la qualité des réponses ;
- intégrer les modèles d'intelligence artificielle dans un processus de développement logiciel ;

PARTICIPANT : Développeurs, ingénieurs logiciels, lead développeur, architectes techniques, DevOps et professionnels techniques utilisant des outils d'intelligence artificielle générative.

PRE-REQUIS : Connaître les principes fondamentaux du développement logiciel, être à l'aise avec la lecture de code et la documentation technique.

MODALITES ET PEDAGOGIE :

- Questionnaire d'évaluation en amont et en fin de formation
- Cas pratiques et études de cas, Quizz
- Horaires : 9h00 à 12h30 – 13h30 à 17h00
- Nombre maximum de stagiaires : 8
- Méthode interactive et intuitive
- Support Stagiaire
- Assistance téléphonique
- Attestation de fin de stage

PROGRAMME**1. Comprendre les modèles de langage**

- Principes de fonctionnement d'un modèle de langage
- Notions de tokens, contexte et génération de réponses
- Forces et limites des modèles
- Hallucinations, biais et fiabilité des résultats

2. Les fondamentaux du prompt engineering

- Définition et objectifs du prompt engineering
- Structure d'un prompt efficace
- Formulation d'instructions claires
- Ajout du contexte, des contraintes et du format attendu
- Erreurs fréquentes dans la rédaction des prompts

3. Les principales techniques de prompting

- Zero-shot, one-shot et few-shot prompting
- Prompting par rôle
- Décomposition d'une tâche complexe
- Utilisation d'exemples
- Enchaînement de plusieurs instructions
- Demande de vérification et d'amélioration d'une réponse

4. Utiliser l'IA dans le développement logiciel

- Génération et explication de code
- Refactorisation et amélioration de code
- Détection d'erreurs
- Génération de tests unitaires
- Création de documentation technique
- Analyse de logs et de messages d'erreur
- Génération de requêtes SQL

5. Structurer et fiabiliser les résultats

- Définition d'un format de sortie
- Génération de réponses en JSON, Markdown ou tableau
- Mise en place de critères de validation
- Vérification de la cohérence et de la qualité des réponses
- Création de prompts réutilisables

6. Sécurité et bonnes pratiques

- Protection des données sensibles
- Risques de prompt injection
- Validation des contenus générés
- Limites de l'exécution automatique de code
- Supervision humaine
- documentation des prompts