

# IA – analyse et production de code informatique

## INFORMATIONS GÉNÉRALES

**Type de formation :** Formation continue

**Éligible au CPF :** Non

**Domaine :** IA, Big Data et Bases de données

**Action collective :** Non

**Filière :** IA

**Rubrique :** ChatGPT - BARD

**Code de formation :** IA051

## € Tarifs

**Prix public :** 1 610 €

### Tarif & financement :

Nous vous accompagnons pour trouver la meilleure solution de financement parmi les suivantes :

- **Le plan de développement des compétences** de votre entreprise : rapprochez-vous de votre service RH.
- **Le dispositif FNE-Formation.**
- **L'OPCO** (opérateurs de compétences) de votre entreprise.
- **Pôle Emploi** sous réserve de l'acceptation de votre dossier par votre conseiller Pôle Emploi.
- **CPF -MonCompteFormation**

Contactez nous pour plus d'information

## PRÉSENTATION

### Objectifs & compétences

Comprendre quels sont les apports de l'IA dans le process de développement, les principaux outils et savoir faire le choix adéquat selon le projet.

### Public visé

Développeurs, chefs de projet

### Pré-requis

Connaissances de Python ou d'un autre langage de programmation structuré.  
Expériences en développement

## 📍 Lieux & Horaires

**Durée :** 14 heures

### Délai d'accès :

Jusqu'à 8 jours avant le début de la formation

## PROGRAMME

### 1) Apports de l'IA

Interventions à plusieurs stades du process de développement  
Vérification de la qualité du code, détection d'erreurs, de failles de sécurité, vérification de la syntaxe, des règles de développement,  
Analyse de code pour générer de la documentation, pour la gestion de sources,  
Génération de tests automatisés.  
Utilisation de modèles d'apprentissage automatique.  
Proposition de modules de codes, autocomplétion, génération de codes complets.  
Intérêts : aide aux développeurs, gains de temps, contrôle exhaustif, meilleure documentation, ..  
Quelques outils phares d'analyse de code : Pylint, Checkstyle, pycodestyle, Black, CodeQL,  
d'autocomplétion : Tabnine, Kite,  
et de production de code : Alphacode, GitHub Copilot, Codex.

### 2) Pycodestyle

Objectif : vérification de respect des conventions d'écriture PEP8 (Python Enhancement Proposal 8),  
Fonctionnement : architecture modulaire, outil léger,  
Intégration aux principaux IDE (VS Code, Pycharm, JupyterNotebook, ...) renvoi direct aux erreurs lors de l'édition du code.  
Configuration en mode utilisateur ou projet.

### Atelier : Installation de pycodestyle

Mise en oeuvre sur un programme simple,  
Affichage des erreurs dans un code source,  
Intégration à JupyterNotebook

### 3) Pylint

Objectif : vérification du respect de la PEP8,  
Détection d'erreurs de programmation, aide au refactoring,  
Configuration des priorités utilisateur,  
Intégration continue, intégration avec les principaux éditeurs et ide

**Atelier : Installation de Pylint**

## 📅 Prochaines sessions

Consultez-nous pour les prochaines sessions.

Configuration de la détection d'erreurs,  
Désactivation des vérifications de règles d'écriture et du refactoring,  
Test sur des programmes caractéristiques.

#### **4) CodeQL**

Objectif : recherche de failles de sécurité dans du code

Fonctionnement :

génération d'une base de données à partir du code, exécution de requêtes sur cette base pour détecter les failles.

Langages traités, architecture : s

#### **Atelier : Mise en oeuvre de CodeQL pour Python**

Ecriture de requêtes basiques pour du code Python

Utilisation de CodeQL library for Python

#### **5 ) Alphacode**

Présentation du projet Google deepmind

Principe de fonctionnement

Résultats obtenus sur CodeForces

#### **6) Copilot**

Présentation du projet Git alimenté par OpenAI Codex.

Fonction : assistant virtuel en programmation.

Fourniture de suggestions de lignes entières de codes ou de fonction entières

Langages supportés

#### **Atelier : Démonstration sur GITHUB de l'utilisation de Copilot**

#### **7) Codex**

Projet : OpenAI Codex, module du projet OpenAI,

production de code informatique à partir de requêtes exprimées en langage naturel

Ressources disponibles : bibliothèque de codes en ligne, hackathons, librairies Python, etc

...

#### **Atelier : Démonstration avec OpenAI Codex**

Génération de code Python à partir d'un cahier des charges simple

Tests et amélioration du code depuis l'interface en langage naturel.

#### **8) Les limites et risques**

Nécessité d'une nouvelle organisation des tâches de développeurs.

Définition du problème à résoudre, définition des contraintes, des jeux d'essai, etc ...

Potentiels risques de sécurité, risques juridiques : origine des solutions utilisées, propriété du code/

Erreurs de compréhension, d'analyse du problème à résoudre.

Importance des contrôles humains à mettre en place.

## **MODALITÉS**

### **Modalités**

**Modalités** : en présentiel, distanciel ou mixte – Horaires de 9H à 12H30 et de 14H à 17H30 soit 7H – Intra et Inter entreprise

**Pédagogie** : essentiellement participative et ludique, centrée sur l'expérience, l'immersion et la mise en pratique. Alternance d'apports théoriques et d'outils pratiques.

**Ressources techniques et pédagogiques** : Support de formation au format PDF ou PPT Ordinateur, vidéoprojecteur, Tableau blanc, Visioconférence : Cisco Webex / Teams / Zoom

**Pendant la formation** : mises en situation, autodiagnostic, travail individuel ou en sous-groupe sur des cas réels

### **Méthode**

**Fin de formation** : entretien individuel

**Satisfaction des participants** : questionnaire de satisfaction réalisé en fin de formation

**Assiduité** : certificat de réalisation (validation des acquis)