

Mis à jour le 15/05/2026

S'inscrire

Formation Superpowers

3 jours (21 heures)

Présentation

Superpowers est un framework de skills pour agents IA de développement, conçu pour structurer la manière dont les assistants de code spécifient, planifient, testent, relisent et valident leurs implémentations.

Notre formation Superpowers vous permettra de maîtriser une méthodologie complète pour encadrer l'usage des agents IA de programmation dans vos projets logiciels.

Vous apprendrez à installer et utiliser Superpowers avec des environnements comme Claude Code, Codex, Gemini CLI, Cursor, OpenCode ou GitHub Copilot CLI.

Vous serez en mesure de transformer une demande imprécise en spécification claire, de produire un plan d'implémentation détaillé, d'appliquer le TDD, de conduire une revue de code structurée et de vérifier les résultats avant livraison.

Grâce à une approche orientée ingénierie logicielle, vous apprendrez à éviter les dérives fréquentes des agents IA : surcodage, absence de tests, corrections superficielles, mauvaise compréhension du besoin ou validation sans preuve.

À l'issue de cette formation, vous serez capable d'intégrer Superpowers dans vos workflows de développement afin de produire du code plus fiable, mieux testé et plus facilement maintenable.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la technologie et ses nouveautés.

Objectifs

- Comprendre le rôle de Superpowers dans les workflows de développement assistés par IA.
- Installer et configurer Superpowers avec un agent IA de programmation.
- Structurer les phases de brainstorming, spécification et planification.
- Appliquer le TDD, la revue de code et la vérification systématique.
- Industrialiser l'usage des agents IA de développement en équipe.

Public visé

- Développeurs logiciels
- Ingénieurs IA
- Ingénieurs DevOps
- Tech Leads
- Architectes logiciels
- Équipes utilisant des agents IA de développement

Pré-requis

- Bonnes bases en développement logiciel
- Connaissances de base en Git
- Compréhension des workflows de développement et de revue de code

Pré-requis techniques

- Ordinateur portable avec 8 Go de RAM minimum et droits d'administration.
- Connexion Internet stable pour accéder aux agents IA, dépôts Git et outils de développement.
- Éditeur de code installé, comme Visual Studio Code ou Cursor.
- Client Git installé et configuré.
- Accès à au moins un agent IA compatible : Claude Code, Codex, Gemini CLI, Cursor, OpenCode ou GitHub Copilot CLI.

Programme de notre formation Superpowers

[Jour 1 - Matin]

Comprendre Superpowers et les agents IA de développement

- Comprendre le rôle de Superpowers comme méthodologie pour agents IA de programmation
- Différencier Superpowers d'un modèle IA, d'un IDE ou d'une simple collection de prompts
- Identifier les plateformes compatibles : Claude Code, Codex, Gemini CLI, Cursor, OpenCode et GitHub Copilot CLI
- Comprendre les limites classiques des agents IA : surcodage, absence de tests, corrections superficielles
- Positionner Superpowers dans un workflow de développement logiciel assisté par IA
- Atelier pratique : Analyse d'un workflow agent IA non structuré et identification des risques.

[Jour 1 - Après-midi]

Installation, configuration et prise en main

- Installer Superpowers dans un environnement compatible
- Comprendre l'organisation des skills et des instructions initiales
- Configurer Superpowers pour un agent IA de développement
- Vérifier que l'agent déclenche correctement les procédures attendues
- Adapter l'environnement de travail selon l'outil utilisé
- Atelier pratique : Installation de Superpowers et validation du déclenchement des skills.

Brainstorming, clarification et cadrage fonctionnel

- Utiliser le workflow de brainstorming pour clarifier une demande avant de coder
- Transformer une idée vague en besoin exploitable par un agent IA
- Identifier les questions nécessaires avant la conception
- Limiter les interprétations erronées et les hypothèses implicites
- Structurer les échanges pour obtenir une demande stable
- Atelier pratique : Cadrage d'une fonctionnalité à partir d'une demande utilisateur imprécise.

[Jour 2 - Matin]

Design, spécification et plan d'implémentation

- Formaliser une demande sous forme de spec claire et vérifiable
- Découper une fonctionnalité en livrables cohérents
- Créer un plan d'implémentation détaillé avant le développement
- Appliquer les principes YAGNI et DRY dans la planification
- Préparer des tâches petites, testables et contrôlables par l'agent IA
- Atelier pratique : Rédaction d'une spec et d'un plan d'implémentation pour un mini-projet.

[Jour 2 - Après-midi]

Développement guidé par TDD avec agents IA

- Comprendre l'approche Test-Driven Development appliquée aux agents IA
- Forcer l'agent à écrire un test qui échoue avant le code
- Limiter le code produit au strict nécessaire pour faire passer le test
- Refactorer après validation tout en conservant la couverture de test
- Éviter les implémentations prématurées et les comportements non vérifiés
- Atelier pratique : Implémentation d'une fonctionnalité avec cycle red, green, refactor.

Debugging systématique et correction par la cause racine

- Utiliser une méthodologie de debugging systématique
- Éviter les corrections superficielles basées sur des symptômes

- Collecter des preuves avant de modifier le code
- Identifier la cause racine d'un bug avec l'aide de l'agent IA
- Valider la correction par des tests et observations reproductibles
- Atelier pratique : Diagnostic d'un bug, identification de la cause racine et correction vérifiée.

[Jour 3 - Matin]

Sous-agents, exécution par lots et workflows avancés

- Comprendre l'usage des sous-agents dans le développement assisté par IA
- Découper le travail en tâches indépendantes et vérifiables
- Organiser une exécution par lots sans perte de contrôle
- Utiliser les skills adaptés selon la situation : plan, implémentation, review, feedback
- Limiter les risques de divergence entre le plan et le code produit
- Atelier pratique : Réalisation d'un développement par tâches guidées avec contrôle intermédiaire.

[Jour 3 - Après-midi]

Code review, vérification et fin de branche

- Mettre en place une code review structurée avec Superpowers
- Comparer l'implémentation au plan initial et aux critères d'acceptation
- Détecter les problèmes bloquants avant de déclarer une tâche terminée
- Exécuter les tests et collecter les preuves de bon fonctionnement
- Gérer la fin de branche : merge, pull request, conservation ou suppression
- Atelier pratique : Revue complète d'une branche, correction des écarts et validation finale.

Industrialisation de l'usage de Superpowers en équipe

- Définir des standards d'utilisation de Superpowers pour une équipe de développement
- Structurer les workflows agent IA dans un contexte projet
- Mettre en place des règles de qualité, vérification et documentation
- Former les équipes à l'usage discipliné des agents IA de programmation
- Mesurer les gains, limites et risques liés à l'AI-assisted development
- Atelier pratique : Création d'un workflow d'équipe pour spécifier, planifier, coder, tester et relire avec Superpowers.

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son

inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.