

PROGRAMME DE FORMATION

Développeur augmenté par l'IA

Développeur

Générer, comprendre, refactorer, tester, réviser et sécuriser du code avec l'IA — en gardant la maîtrise de l'architecture et de la qualité.

13

modules

54leçons dont 31
exercices**17 h 04**

de contenu

93

thèmes couverts

Ce que contient la formation

Un parcours complet et opérationnel : chaque notion est immédiatement mise en pratique sur des cas réels du métier, avec un corrigé. Les collaborateurs avancent en autonomie, à leur rythme, et repartent avec des méthodes réutilisables dès le lendemain.

Présentations		22	41 %
Exercices pratiques		31	57 %
Quiz de validation		1	2 %

Thèmes couverts

7 principes	8 paliers	Abstraction	ADR	Agent	Agents	Anti-patterns	Antiséche	API	Atelier
Automatisation	Autonomie	Bilan	Cas limites	Code smells	Commit	Communication	Complexité		
Comprendre du code	Compréhension de code	Confidentialité	Contrôle d'accès	Conventions	Copilot / Claude Code / MCP				
Couverture	Critères de réussite	Dette	Dette technique	Documentation	Données de test	Données sensibles	DRY		
Durcissement	Débogage	Esprit critique	Exercice	Exercices guidés	Extraction	Fixtures	Fonctionnalité		
Garder la maîtrise	Gherkin	Git & CI	Génération de code	Générer du code	IDOR	Injection	Legacy	Matériau fourni	
MCP	Migration	Mocks	Modèle	Modèle mental	Multi-fichiers	Méthode C-R-T-F-C	Onboarding	OWASP	
Parcours	Passerelle	Passerelles	Portage	Pour aller plus loin	Priorisation	Profil Développeur	Prompt engineering		
Property-based	Prérequis	Prérequis : bases en dev	Pull request	Qualité	Quiz	Refactoring	Regex		
Revue de code	Récapitulatif	Sans inventer	Scaffolding	Secrets	Secrets & PI	Skills	Spec-Driven Development		
spec-kit	SQL	Supply chain	Synthèse	Sécurité	Templates	Tests	Tests unitaires	Tips avancés	Typage
Validation									

Modalités

- 100 % en ligne, en autonomie — ou en présentiel intra-entreprise
- Version audio de chaque présentation, pour apprendre en mobilité
- Exercices guidés et corrigés sur des cas réels du métier
- Suivi de progression individuel, leçon par leçon
- Quiz de validation des acquis avec score et recommandations

Prérequis

Aucun prérequis technique. Parcours accessible du débutant à l'avancé : chaque exercice est proposé à plusieurs niveaux de difficulté.

Sommaire des modules

13 modules, 54 leçons, 17 h 04 de contenu.

01	Introduction	1 leçon	10 min
02	Cadrage & confidentialité	1 leçon	30 min
03	Prompt engineering	6 leçons	2 h 25
04	Générer & comprendre du code	9 leçons	2 h 15
05	Refactoring, dette & migration	8 leçons	2 h 10
06	Tests & qualité avec l'IA	8 leçons	2 h 05
07	Revue de code & sécurité	8 leçons	2 h
08	Atelier de bout en bout — IDE & agents	6 leçons	3 h 25
09	Autonomie & ancrage	1 leçon	20 min
10	Documenter & rendre compte	1 leçon	20 min
11	Garder la maîtrise	1 leçon	25 min
12	Pour aller plus loin	3 leçons	39 min
13	Validation des acquis	1 leçon	20 min

01 INTRODUCTION

1 leçon · 10 min

Présentation

Feuille de route

🕒 10 min

Votre parcours en 8 paliers : de la littérature IA et du prompting jusqu'aux agents de code, en gardant la maîtrise de l'architecture, de la qualité et de la sécurité.

🔊 audio

8 paliers · Prérequis : bases en dev · Profil Développeur

02 CADRAGE & CONFIDENTIALITÉ

1 leçon · 30 min

Présentation

Charte & règle de tri du code

🕒 30 min

La règle d'or avant tout usage de l'IA : classer chaque artefact par sensibilité (●●●) et ne jamais exposer de code propriétaire, de secrets ni de données réelles.

🔊 audio

Prérequis · Confidentialité · Secrets & PI

03 PROMPT ENGINEERING

6 leçons dont 1 exercice · 2 h 25

Présentation

Atelier littérature IA

🕒 25 min

Comprendre ce qu'est — et n'est pas — un grand modèle de langage appliqué au code, savoir le piloter par un bon prompt, et toujours relire ce qu'il produit.

🔊 audio

Atelier · Prompt engineering · Modèle mental

Présentation

Les bases : C-R-T-F-C

🕒 15 min

Pourquoi bien demander change tout, comment l'IA « lit » votre demande, et la méthode C-R-T-F-C pour cadrer un prompt de génération de code.

🔊 audio

Prompt engineering · Méthode C-R-T-F-C

Présentation

Principes & tips avancés

🕒 20 min

Les 7 principes qui font un bon prompt, puis une douzaine de techniques avancées (few-shot, chain of thought, grounding, auto-critique) appliquées au code.

🔊 audio

Prompt engineering · 7 principes · Tips avancés

Présentation

Anti-patterns, galerie & templates

🕒 15 min

Les erreurs qui ruinent un prompt de dev, une galerie avant/après orientée code, et des templates prêts à copier-coller.

🔊 audio

Prompt engineering · Anti-patterns · Templates

Exercice

Exercices de prompting

🕒 1 h

12 exercices guidés, matériel et prompts fournis (rien à inventer), avec correction en accordéon — du générique au code. Vos premiers vrais échanges avec un assistant IA sur du code.

Exercices guidés · Prompt engineering · Matériel fourni

Présentation

Mesurer & antisèche

🕒 10 min

Reconnaître un bon prompt de code à des critères objectifs, et l'antisèche à garder sous la main.

🔊 audio

Prompt engineering · Antisèche

04 GÉNÉRER & COMPRENDRE DU CODE

9 leçons dont 7 exercices · 2 h 15

Présentation

Introduction & rituel

⌚ 10 min

Le palier de l'adhésion : prouver la valeur de l'IA pour produire et comprendre du code.
Comment lire chaque exercice, et le fil rouge fictif NovaShop.

🔊 audio

Générer du code · Comprendre du code

Exercice

Ex. 1 — Générer une fonction depuis une spec

⌚ 20 min

Cadrer un prompt pour obtenir une fonction correcte, avec sa signature, ses cas limites et la gestion d'erreur — et savoir juger ce qui est produit.

🔊 audio

Exercice · Génération de code

Exercice

Ex. 2 — Scaffolding d'un endpoint REST

⌚ 20 min

Faire générer la structure d'un endpoint CRUD (route, validation, réponse) et repérer ce qui reste à câbler et à sécuriser soi-même.

🔊 audio

Exercice · Scaffolding · API

Exercice

Ex. 3 — Expliquer du code inconnu

⌚ 15 min

Faire expliquer un bout de code dense ligne à ligne puis en résumé, et vérifier l'explication plutôt que de la croire sur parole.

🔊 audio

Exercice · Compréhension de code

Exercice

Ex. 4 — Explorer une base de code inconnue

⌚ 20 min

S'onboarder plus vite sur un dépôt legacy : retrouver le point d'entrée, le fil d'une fonctionnalité et la carte mentale du projet.

🔊 audio

Exercice · Legacy · Onboarding

Exercice

Ex. 5 — Porter un code d'un langage à l'autre + confidentialité

⌚ 15 min

Traduire un snippet d'un langage vers un autre, tout en appliquant le réflexe de tri : ne jamais coller du code propriétaire ou des secrets dans un outil non maîtrisé.

🔊 audio

Exercice · Portage · Confidentialité

Exercice

Ex. 6 — Générer une regex / requête SQL (chain of thought)

⌚ 15 min

Obtenir une expression régulière ou une requête correcte en forçant l'IA à raisonner étape par étape, puis la tester sur des cas concrets.

🔊 audio

Exercice · Regex · SQL

Exercice

Ex. 7 — Données de test & fixtures réalistes

⌚ 15 min

Générer rapidement un jeu de données de test valides, aux limites et invalides, sans jamais exposer de données de production.

🔊 audio

Exercice · Données de test

Présentation

Synthèse du palier

⌚ 5 min

Ce que le palier a fait travailler, et le fil conducteur : l'IA produit le brouillon, vous restez l'auteur qui comprend, relit et intègre.

🔊 audio

Synthèse · Critères de réussite

05 REFACTORING, DETTE & MIGRATION

8 leçons dont 6 exercices · 2 h 10

Présentation

Introduction & rituel

⌚ 10 min

Améliorer du code existant sans changer son comportement, avec l'IA. Le réflexe transverse : un filet de tests d'abord, des pas petits et vérifiables ensuite.

🔊 audio

Refactoring · Dette technique

Exercice

Ex. 1 — Découper une fonction trop longue

⌚ 20 min

Extraire, nommer et clarifier une fonction fourre-tout, à comportement strictement identique, et vérifier l'équivalence.

🔊 audio

Exercice · Extraction

Exercice

Ex. 2 — Réduire la complexité & les code smells

⌚ 20 min

Faire diagnostiquer les mauvaises odeurs d'un code (duplication, conditions imbriquées) et appliquer les corrections une à une.

🔊 audio

Exercice · Code smells · Complexité

Exercice

Ex. 3 — Typier & documenter un module

⌚ 15 min

Ajouter types et docstrings cohérents à un module non typé, et faire de l'IA un relecteur de cohérence nom/comportement.

🔊 audio

Exercice · Typage · Documentation

Exercice

Ex. 4 — Migrer une API / bibliothèque par étapes

⌚ 25 min

Conduire une migration (callbacks → async/await, v1 → v2) en pas sûrs et testés, sans tout casser d'un coup.

🔊 audio

Exercice · Migration

Exercice

Ex. 5 — Dédupliquer (DRY) sans sur-abstraire

⌚ 20 min

Factoriser du code répété en gardant la lisibilité, et reconnaître quand une abstraction proposée par l'IA fait plus de mal que de bien.

🔊 audio

Exercice · DRY · Abstraction

Exercice

Ex. 6 — Plan de réduction de dette

⌚ 15 min

Transformer une revue de dette technique en plan d'actions priorisé (valeur/risque), exploitable par l'équipe — l'IA propose, vous arbitrez.

🔊 audio

Exercice · Dette · Priorisation

Présentation

Synthèse du palier

⌚ 5 min

Ce que le palier a fait travailler, et le rappel : on ne refactorise jamais sans filet de tests, et chaque pas reste réversible.

🔊 audio

Synthèse · Critères de réussite

06 TESTS & QUALITÉ AVEC L'IA

8 leçons dont 6 exercices · 2 h 05

Présentation

Introduction & rituel

⌚ 10 min

Faire écrire des tests par l'IA sans lui déléguer le jugement. Le réflexe transverse : un test qui passe ne prouve rien si vous n'avez pas lu ce qu'il vérifie.

🔊 audio

Tests · Qualité

06 TESTS & QUALITÉ AVEC L'IA

8 leçons dont 6 exercices · 2 h 05

Exercice

Ex. 1 — Tests unitaires d'une fonction

⌚ 20 min

Générer une batterie de tests unitaires (nominal, limites, erreurs) pour une fonction, et juger leur pertinence au-delà du « vert ».

🔊 audio

Exercice · Tests unitaires

Exercice

Ex. 2 — Des critères Gherkin aux tests

⌚ 20 min

Transformer des critères d'acceptation Given-When-Then (produits par le PO/BA) en tests automatisés — la passerelle directe avec la conception et le QA.

🔊 audio

Exercice · Gherkin · Passerelle

Exercice

Ex. 3 — Cas limites & tests de propriété

⌚ 20 min

Faire chercher les bords par l'IA (valeurs nulles, débordements, Unicode) et découvrir les tests basés sur des propriétés invariantes.

🔊 audio

Exercice · Cas limites · Property-based

Exercice

Ex. 4 — Couverture & code non testé

⌚ 15 min

À partir d'un rapport de couverture, cibler le code à risque non testé plutôt que de courir après 100 % — la couverture n'est pas la qualité.

🔊 audio

Exercice · Couverture

Exercice

Ex. 5 — Mocks, fixtures & doubles de test

⌚ 20 min

Isoler une dépendance (base, API externe, horloge) avec un mock ou un fixture généré par l'IA, sans tester l'outil de mock à la place du code.

🔊 audio

Exercice · Mocks · Fixtures

Exercice

Ex. 6 — Comprendre & réparer un test rouge

⌚ 15 min

Faire diagnostiquer un échec de test par l'IA, distinguer un vrai bug d'un test mal écrit, et corriger le bon côté.

🔊 audio

Exercice · Débogage

Présentation

Synthèse du palier

⌚ 5 min

Ce que le palier a fait travailler, et le rappel : vous signez les tests, donc vous lisez ce qu'ils affirment avant de leur faire confiance.

🔊 audio

Synthèse · Critères de réussite

07 REVUE DE CODE & SÉCURITÉ

8 leçons dont 6 exercices · 2 h

Présentation

Introduction & rituel

⌚ 10 min

Utiliser l'IA comme un relecteur infatigable — sur la lisibilité, les bugs et surtout la sécurité (OWASP) — sans lui déléguer la décision de merger.

🔊 audio

Revue de code · Sécurité

Exercice

Ex. 1 — Revue de code assistée d'une PR

⌚ 20 min

Faire passer une pull request au crible (lisibilité, bugs, perf, tests manquants) et trier les remarques de l'IA du bruit au critique.

🔊 audio

Exercice · Pull request

07 REVUE DE CODE & SÉCURITÉ

8 leçons dont 6 exercices · 2 h

Exercice

Ex. 2 — Détecter une injection SQL

⌚ 20 min

Repérer une injection SQL dans du code réaliste, comprendre la faille et la corriger par requête paramétrée — la vulnérabilité n°1.

🔊 audio

Exercice · OWASP · Injection

Exercice

Ex. 3 — Secrets en dur & données sensibles

⌚ 15 min

Faire débusquer secrets codés en dur, logs trop bavards et données personnelles exposées, et appliquer les bons réflexes (variables d'env, secrets manager).

🔊 audio

Exercice · Secrets · Données sensibles

Exercice

Ex. 4 — Contrôle d'accès & IDOR

⌚ 20 min

Repérer un défaut de contrôle d'accès (un utilisateur accède aux données d'un autre) — la faille que ni un test vert ni un linter ne voient.

🔊 audio

Exercice · Contrôle d'accès · IDOR

Exercice

Ex. 5 — Durcir la validation des entrées

⌚ 15 min

Faire générer une validation d'entrées robuste (typage, bornes, liste blanche) et comprendre pourquoi on valide côté serveur, jamais seulement côté client.

🔊 audio

Exercice · Validation · Durcissement

Exercice

Ex. 6 — Commentaire de revue & message de commit

⌚ 15 min

Transformer un constat brut en commentaire de revue constructif et en message de commit clair, directement actionnables par l'auteur du code.

🔊 audio

Exercice · Communication · Commit

Présentation

Synthèse du palier

⌚ 5 min

Ce que le palier a fait travailler, et le rappel : l'IA accélère la détection, mais la décision de sécurité et de merge reste humaine et tracée.

🔊 audio

Synthèse · Critères de réussite

08 ATELIER DE BOUT EN BOUT — IDE & AGENTS

6 leçons dont 4 exercices · 3 h 25

Présentation

Cadrage, outils & confidentialité

⌚ 25 min

L'atelier de bout en bout sur une vraie application RH (espace salarié) : panorama des assistants d'IDE et agents (Copilot, Claude Code, MCP), la chaîne user story → code → tests → revue, et les règles de confidentialité.

🔊 audio

Atelier · Copilot / Claude Code / MCP · Confidentialité

Exercice

Parcours 1 — Implémenter une fonctionnalité (modèle)

⌚ 50 min

Le parcours modèle : d'une user story et de ses critères Gherkin jusqu'au code livré et testé, en pilotant l'assistant pas à pas — l'export d'un CRA mensuel.

🔊 audio

Parcours · Fonctionnalité · Modèle

Exercice

Parcours 2 — Comprendre & corriger un bug

⌚ 40 min

Partir d'un ticket de bug et d'une stack trace, localiser la cause dans le code existant avec l'IA, écrire le test qui échoue, puis corriger.

🔊 audio

Parcours · Débogage · Legacy

08 ATELIER DE BOUT EN BOUT — IDE & AGENTS

6 leçons dont 4 exercices · 3 h 25

Exercice

Parcours 3 — Refactorer & sécuriser un module

⌚ 40 min

Sur le changement de mot de passe : poser un filet de tests, refactorer, puis durcir la sécurité (validation, anti-énumération) en gardant la maîtrise.

🔊 audio

Parcours · Refactoring · Sécurité

Exercice

Parcours 4 — Piloter un agent de code

⌚ 40 min

Confier à un agent (type Claude Code) une tâche multi-fichiers — plan, exécution, relecture du diff — et savoir où poser les garde-fous.

🔊 audio

Parcours · Agent · Multi-fichiers

Présentation

Récapitulatif & passerelles

⌚ 10 min

Ce que l'atelier a fait travailler, les passerelles vers les formations PO/BA (qui spécifie) et QA (qui éprouve), et le rappel : compte de test + nettoyage avant d'agir sur du réel.

🔊 audio

Récapitulatif · Passerelles

09 AUTONOMIE & ANCRAGE

1 leçon · 20 min

Présentation

Git, CI & conventions d'équipe

⌚ 20 min

Ancrer les nouveaux réflexes dans la durée : intégrer l'IA au flux Git et à la CI, partager des conventions de prompts, garder l'humain dans la boucle, et installer un rituel d'équipe.

🔊 audio

Autonomie · Git & CI · Conventions

10 DOCUMENTER & RENDRE COMPTE

1 leçon · 20 min

Présentation

Docstrings, README, ADR & notes de version

⌚ 20 min

Transformer du code en documentation claire — docstrings, README, décisions d'architecture (ADR), messages de commit et notes de version — avec l'IA, sans qu'elle invente les faits techniques.

🔊 audio

Documentation · ADR · Sans inventer

11 GARDER LA MAÎTRISE

1 leçon dont 1 exercice · 25 min

Exercice

Risques propres à l'IA pour le développeur

⌚ 25 min

Les pièges qu'aucune relecture rapide ne révèle : code plausible mais faux, dépendance fantôme et risque supply chain, faille introduite, dette cachée, sur-confiance et fuite de code propriétaire.

🔊 audio

Garder la maîtrise · Supply chain · Esprit critique

12 POUR ALLER PLUS LOIN

3 leçons · 39 min

Présentation

MCP & agents — relier l'IA au dépôt et aux outils

🕒 12 min

🔊 audio

Ce qu'est un MCP, ce connecteur qui relie l'IA à vos outils (dépôt, tickets, base de données, doc), et ce que les agents de code changent pour un développeur — avec les garde-fous.

Pour aller plus loin · MCP · Agents

Présentation

Skills — créez votre premier skill dev

🕒 12 min

🔊 audio

Un skill apprend une de vos procédures à Claude pour la rejouer d'un mot. Deux exemples à fabriquer — générateur de tests et revue de sécurité — puis de quoi créer tous les autres seul.

Pour aller plus loin · Skills · Automatisation

Présentation

Spec Kit — de la spécification au code

🕒 15 min

🔊 audio

Le projet open source spec-kit (GitHub) outille le Spec-Driven Development : installation dans Claude Code, et surtout reprise d'un projet réel déjà spécifié par le PO (constitution + spec + clarifications) pour dérouler le flux développeur /speckit-plan → /speckit-tasks → /speckit-implement sur le panier de NovaShop.

Pour aller plus loin · Spec-Driven Development · spec-kit

13 VALIDATION DES ACQUIS

1 leçon · 20 min

Quiz

Quiz de validation des acquis

🕒 20 min

Questions à choix multiple sur l'ensemble de la formation, de difficultés variées. Score, corrigé commenté et recommandations de chapitres à revoir.

Quiz · Bilan · Validation

Et après ?

À l'issue du parcours, chaque collaborateur dispose d'une méthode de travail avec l'IA — pas d'une simple liste d'outils. Les supports restent accessibles après la formation, ce qui permet de revenir sur un exercice au moment où le besoin se présente réellement dans le travail.

PUBLIC VISÉ

Développeur

PRÉREQUIS

Aucun prérequis technique. Parcours accessible du débutant à l'avancé : chaque exercice est proposé à plusieurs niveaux de difficulté.

MODALITÉS D'ÉVALUATION

Positionnement à l'entrée, exercices évalués tout au long du parcours, puis quiz de validation final (score et chapitres à revoir). Sanction : attestation de fin de formation.

DÉLAI D'ACCÈS

Formation intra-entreprise sur mesure : mise en œuvre généralement sous 2 à 4 semaines après validation de la convention.

ACCESSIBILITÉ

Formation adaptable aux situations de handicap. Référent handicap : Emanuel Henriot — emanuel.henriot@haapii.fr — 06 04 52 12 46.

HAAPII Services

Emanuel Henriot · 06 04 52 12 46 · emanuel.henriot@haapii.fr
ora-ia.fr/formation

Document commercial — non contractuel — contenu susceptible d'évoluer.