

FORMATION

Comprendre les Outils IA de Code

Cette formation rassemble le parcours, les objectifs, les exercices, les scripts ou notes pédagogiques, les prompts et le plan d'action final pour appliquer Comprendre les Outils IA de Code sans devoir ouvrir plusieurs fichiers séparés.

Format

PDF A4 imprimable

Contenu

6 fichier(s) consolidé(s)

Usage

Lire, appliquer, personnaliser

Livrable

Guide complet + bonus

Document client neutre et prêt à vendre

Introduction

Cette formation rassemble le parcours, les objectifs, les exercices, les scripts ou notes pedagogiques, les prompts et le plan d'action final pour appliquer Comprendre les Outils IA de Code sans devoir ouvrir plusieurs fichiers separes.

Le contenu a ete reorganise pour eviter les doublons, separer les modes d'emploi des bonus et rendre les prompts, checklists, workflows et scripts directement utilisables.

Table des matieres

1. Comment utiliser cette ressource
2. Contenu principal
3. Objectifs et parcours de formation
4. Bibliotheque de prompts
5. Annexes structurees
6. Prochaine action
7. Conclusion

Comment utiliser cette ressource

1. Commencez par le guide principal pour comprendre le contexte et le resultat attendu.
2. Utilisez le workbook, les checklists ou les workflows comme espace de travail.
3. Copiez les prompts dans votre outil d'IA, remplacez les variables entre crochets et demandez une premiere version.
4. Ameliorer la reponse avec les conseils d'utilisation et sauvegardez les meilleures versions.
5. Terminez par la prochaine action proposee a la fin du document.

Objectifs et parcours de formation

A la fin de cette formation, vous devez disposer d'un système utilisable, de prompts adaptés à vos cas réels et d'un plan d'action pour continuer sans dépendre d'un fichier isolé.

- Comprendre les objectifs de chaque module.
- Appliquer les exercices dans le workbook.
- Utiliser les scripts ou notes comme support de révision.
- Finaliser un plan d'action mesurable.

Contenu principal

Programme de formation

Source integree : course.html

Bienvenue dans la formation

Cette formation explique concrètement les différences entre assistants IA, agents IA de code et outils d'automatisation, avec des règles simples pour choisir sans risquer son projet.

Cette formation contient les leçons, exemples, exercices et corrigés. Elle ne te demande pas de fabriquer la formation : elle te fait pratiquer directement.

Résultat attendu

À la fin, tu sais quel outil utiliser selon ton besoin et quelles règles appliquer avant modification de code.

Plan complet

Module	Ce que tu apprends	Livrable
1	Comprendre ce que ces outils font réellement.	Une matrice besoin outil.
2	Différencier réponse et action.	Une consigne agent sécurisée.
3	Utiliser les assistants pour penser et expliquer.	Un brief technique clair.
4	Comprendre l'usage codebase.	Une demande d'analyse codebase.
5	Comprendre l'usage création et modification.	Un prompt Codex prêt.
6	Comprendre l'usage assistant personnel.	Un workflow OpenClaw.
7	Prendre la bonne décision d'outil.	Une matrice de choix.
8	Éviter de casser un projet.	Une checklist agent IA code.

Module 1 - C'est quoi un outil IA de code ?

Leçon complète

Un outil IA de code peut expliquer, générer, modifier, tester ou documenter du code selon son niveau d'accès. La différence importante est le niveau d'action : certains répondent dans une conversation, d'autres agissent dans les fichiers.

Comprendre + proposer + modifier + valider.

Exemple traité

ChatGPT explique une erreur. Codex peut modifier un fichier. Claude Code peut analyser une codebase et proposer un plan de changement.

Exemple de résultat attendu

Tu dois savoir quand demander une explication et quand autoriser une modification.

Erreur à éviter

Laisser un agent modifier un projet sans plan ni sauvegarde.

Méthode à appliquer

1. Identifier le besoin
2. Choisir assistant ou agent
3. Limiter le périmètre
4. Demander un plan
5. Valider avant modification

Prompt d'application

Agis comme un coach opérationnel. Je suis [profil] et je suis la formation "Comprendre les Outils IA de Code".

Module : C'est quoi un outil IA de code ?.

Mon contexte réel : [décris ta situation].

Mon objectif : Une matrice besoin outil..

Produis une réponse directement utilisable avec :

1. un diagnostic rapide,
2. une méthode en étapes,
3. un exemple rempli adapté à mon contexte,
4. les erreurs à éviter,
5. une checklist de validation,
6. ma prochaine action concrète.

Contraintes : reste simple, spécifique, réaliste, sans promesse automatique. Si mon contexte est insuffisant, pose-moi jusqu'à 3 questions avant de répondre.

Exercice du module

Classe 6 tâches selon l'outil approprié.

Validation

- Besoin défini
- Risque évalué
- Outil choisi
- Périmètre limité
- Validation prévue

Module 2 - Assistant IA ou agent IA

Leçon complète

Un assistant te répond. Un agent peut agir dans ton environnement : lire fichiers, proposer patch, exécuter commandes. Plus l'outil agit, plus les garde-fous doivent être stricts.

Lecture seule avant modification.

Exemple traité

Avant de demander corrige ce bug, demande analyse le bug, liste les fichiers concernés et propose un plan sans modifier.

Exemple de résultat attendu

Le bon réflexe est d'exiger une analyse avant action.

Erreur à éviter

Donner une consigne trop large comme améliore tout le projet.

Méthode à appliquer

1. Demander analyse
2. Lire le plan
3. Limiter fichiers
4. Autoriser modification

5. Relire diff

Prompt d'application

Agis comme un coach opérationnel. Je suis [profil] et je suis la formation "Comprendre les Outils IA de Code".

Module : Assistant IA ou agent IA.

Mon contexte réel : [décris ta situation].

Mon objectif : Une consigne agent sécurisée..

Produis une réponse directement utilisable avec :

1. un diagnostic rapide,
2. une méthode en étapes,
3. un exemple rempli adapté à mon contexte,
4. les erreurs à éviter,
5. une checklist de validation,
6. ma prochaine action concrète.

Contraintes : reste simple, spécifique, réaliste, sans promesse automatique. Si mon contexte est insuffisant, pose-moi jusqu'à 3 questions avant de répondre.

Exercice du module

Transforme une demande risquée en demande sécurisée.

Validation

- Analyse d'abord
- Fichiers listés
- Plan demandé
- Modification limitée
- Diff relu

Module 3 - Claude et ChatGPT

Leçon complète

Claude et ChatGPT sont utiles pour clarifier une idée, expliquer du code, résumer une documentation, préparer une consigne technique ou comparer des options. Ils sont souvent le bon point de départ avant d'utiliser un agent.

Comprendre avant de modifier.

Exemple traité

Colle une erreur et demande : explique les causes probables, les informations manquantes et les tests à faire avant correction.

Exemple de résultat attendu

La réponse doit t'aider à décider, pas juste fournir du code.

Erreur à éviter

Demander directement du code sans comprendre le problème.

Méthode à appliquer

1. Coller contexte
2. Demander explication
3. Lister hypothèses
4. Choisir test
5. Préparer consigne agent

Prompt d'application

Agis comme un coach opérationnel. Je suis [profil] et je suis la formation "Comprendre les Outils IA de Code".

Module : Claude et ChatGPT.

Mon contexte réel : [décris ta situation].

Mon objectif : Un brief technique clair..

Produis une réponse directement utilisable avec :

1. un diagnostic rapide,
2. une méthode en étapes,
3. un exemple rempli adapté à mon contexte,
4. les erreurs à éviter,
5. une checklist de validation,
6. ma prochaine action concrète.

Contraintes : reste simple, spécifique, réaliste, sans promesse automatique. Si mon contexte est insuffisant, pose-moi jusqu'à 3 questions avant de répondre.

Exercice du module

Prépare un brief technique avec ChatGPT ou Claude.

Validation

- Contexte fourni
- Hypothèses listées
- Tests proposés
- Décision claire
- Consigne prête

Module 4 - Claude Code

Leçon complète

Claude Code est pertinent quand il faut comprendre une base de code, identifier des fichiers, planifier une modification et travailler dans un projet réel. Il doit être utilisé avec un périmètre précis.

Explorer + planifier + modifier + vérifier.

Exemple traité

Demande : analyse la structure du projet, identifie où ajouter cette fonctionnalité, propose un plan, n'écris pas de code pour l'instant.

Exemple de résultat attendu

La sortie doit être une carte du projet et un plan de modification.

Erreur à éviter

Lancer une modification sans savoir quels fichiers seront touchés.

Méthode à appliquer

1. Demander cartographie
2. Identifier fichiers
3. Valider plan
4. Modifier petit
5. Tester

Prompt d'application

Agis comme un coach opérationnel. Je suis [profil] et je suis la formation "Comprendre les Outils IA de Code".

Module : Claude Code.

Mon contexte réel : [décris ta situation].

Mon objectif : Une demande d'analyse codebase..

Produis une réponse directement utilisable avec :

1. un diagnostic rapide,
2. une méthode en étapes,
3. un exemple rempli adapté à mon contexte,
4. les erreurs à éviter,
5. une checklist de validation,
6. ma prochaine action concrète.

Contraintes : reste simple, spécifique, réaliste, sans promesse automatique. Si mon contexte est insuffisant, pose-moi jusqu'à 3 questions avant de répondre.

Exercice du module

Écris une consigne Claude Code lecture seule.

Validation

- Lecture seule
- Fichiers clés
- Plan attendu
- Risques listés
- Validation humaine

Module 5 - Codex

Leçon complète

Codex est utile pour créer ou modifier des projets, ajouter une fonctionnalité, corriger un bug, refactoriser ou préparer un déploiement. La qualité dépend de ta consigne et de la validation finale.

Objectif + fichiers + contraintes + tests.

Exemple traité

Ajoute une section pricing sur la page d'accueil en respectant le design existant, sans modifier les autres pages, puis indique comment valider.

Exemple de résultat attendu

Le résultat doit être limité, testable et cohérent avec le projet.

Erreur à éviter

Demander un site complet sans cahier des charges.

Méthode à appliquer

1. Décrire objectif
2. Limiter périmètre
3. Donner contraintes
4. Demander validation
5. Tester

Prompt d'application

Agis comme un coach opérationnel. Je suis [profil] et je suis la formation "Comprendre les Outils IA de Code".

Module : Codex.

Mon contexte réel : [décris ta situation].

Mon objectif : Un prompt Codex prêt..

Produis une réponse directement utilisable avec :

1. un diagnostic rapide,
2. une méthode en étapes,
3. un exemple rempli adapté à mon contexte,
4. les erreurs à éviter,
5. une checklist de validation,
6. ma prochaine action concrète.

Contraintes : reste simple, spécifique, réaliste, sans promesse automatique. Si mon contexte est insuffisant, pose-moi jusqu'à 3 questions avant de répondre.

Exercice du module

Écris un prompt Codex pour une petite fonctionnalité.

Validation

- Objectif clair
- Périmètre limité
- Contraintes présentes
- Tests demandés
- Résultat vérifiable

Module 6 - OpenClaw

Leçon complète

OpenClaw se positionne davantage comme assistant de productivité, organisation ou automatisation personnelle selon ton environnement. Il est utile pour gérer routines, idées et workflows.

Routine + déclencheur + résultat + garde-fous.

Exemple traité

Chaque matin, résume mes priorités, propose 3 actions, archive les idées non urgentes.

Exemple de résultat attendu

Le workflow doit aider sans créer de dépendance aveugle.

Erreur à éviter

Automatiser une routine qui n'est pas encore claire.

Méthode à appliquer

1. Décrire routine
2. Définir déclencheur
3. Préciser résultat
4. Ajouter garde-fous
5. Tester

Prompt d'application

Agis comme un coach opérationnel. Je suis [profil] et je suis la formation "Comprendre les Outils IA de Code".

Module : OpenClaw.

Mon contexte réel : [décris ta situation].

Mon objectif : Un workflow OpenClaw..

Produis une réponse directement utilisable avec :

1. un diagnostic rapide,
2. une méthode en étapes,
3. un exemple rempli adapté à mon contexte,
4. les erreurs à éviter,
5. une checklist de validation,
6. ma prochaine action concrète.

Contraintes : reste simple, spécifique, réaliste, sans promesse automatique. Si mon contexte est insuffisant, pose-moi jusqu'à 3 questions avant de répondre.

Exercice du module

Crée un workflow personnel simple.

Validation

- Routine définie
- Déclencheur clair
- Résultat attendu
- Garde-fous
- Test prévu

Module 7 - Choisir selon son besoin

Leçon complète

Choisis l'outil selon la tâche : comprendre, écrire, analyser, modifier ou automatiser. Plus la tâche touche au code réel, plus tu dois exiger un plan et des tests.

Besoin + risque + niveau + outil.

Exemple traité

Besoin : expliquer une erreur = ChatGPT/Claude. Modifier un composant = Codex/Claude Code avec plan. Automatiser routine = OpenClaw.

Exemple de résultat attendu

La matrice doit éviter d'utiliser un agent quand un assistant suffit.

Erreur à éviter

Utiliser l'outil le plus puissant pour une tâche simple.

Méthode à appliquer

1. Définir tâche
2. Évaluer risque
3. Choisir outil
4. Écrire consigne
5. Valider

Prompt d'application

Agis comme un coach opérationnel. Je suis [profil] et je suis la formation "Comprendre les Outils IA de Code".

Module : Choisir selon son besoin.

Mon contexte réel : [décris ta situation].

Mon objectif : Une matrice de choix..

Produis une réponse directement utilisable avec :

1. un diagnostic rapide,
2. une méthode en étapes,
3. un exemple rempli adapté à mon contexte,
4. les erreurs à éviter,
5. une checklist de validation,
6. ma prochaine action concrète.

Contraintes : reste simple, spécifique, réaliste, sans promesse automatique. Si mon contexte est insuffisant, pose-moi jusqu'à 3 questions avant de répondre.

Exercice du module

Remplis la matrice besoin outil.

Validation

- Besoin écrit
- Risque noté
- Outil choisi
- Justification
- Plan de validation

Module 8 - Sécurité avant modification

Leçon complète

Avant toute modification de code, vérifie l'état du projet, sauvegarde, demande un plan, limite le périmètre, lis le diff et teste. Ce rituel réduit les erreurs coûteuses.

Sauvegarde + plan + patch limité + test + revue.

Exemple traité

Avant de demander une correction : vérifie branche Git, copie erreur exacte, demande hypothèses, puis patch minimal.

Exemple de résultat attendu

La demande finale doit inclure ce qui ne doit pas être modifié.

Erreur à éviter

Autoriser des changements massifs sans revue.

Méthode à appliquer

1. Vérifier état
2. Sauvegarder
3. Demander plan
4. Limiter modification
5. Tester

Prompt d'application

Agis comme un coach opérationnel. Je suis [profil] et je suis la formation "Comprendre les Outils IA de Code".

Module : Sécurité avant modification.

Mon contexte réel : [décris ta situation].

Mon objectif : Une checklist agent IA code..

Produis une réponse directement utilisable avec :

1. un diagnostic rapide,
2. une méthode en étapes,
3. un exemple rempli adapté à mon contexte,
4. les erreurs à éviter,
5. une checklist de validation,
6. ma prochaine action concrète.

Contraintes : reste simple, spécifique, réaliste, sans promesse automatique. Si mon contexte est insuffisant, pose-moi jusqu'à 3 questions avant de répondre.

Exercice du module

Crée ta checklist sécurité.

Validation

- Sauvegarde prévue
- Plan demandé
- Périmètre limité
- Diff relu
- Tests faits

Étude de cas finale

Cas final : tu dois aider une personne débutante à utiliser ChatGPT pour organiser sa semaine. Tu vas créer un prompt, générer un planning, puis corriger la réponse.

Étape	Travail à produire
1	Écrire le contexte de la personne
2	Créer le prompt complet
3	Générer le planning
4	Identifier ce qui manque
5	Demander une version corrigée

Erreurs fréquentes

Erreur	Pourquoi c'est un problème	Correction
Prompt vague	Réponse trop générale	Ajouter contexte et format
Pas de relecture	Risque d'erreur	Vérifier avant utilisation
Trop de demandes	Réponse confuse	Découper en étapes

Plan d'action final

Jour	Action	Preuve concrète
Jour 1	Créer 3 prompts simples	Prompts sauvegardés
Jour 2	Tester résumé et email	Deux résultats utiles
Jour 3	Créer le prompt final	Version sauvegardée

Workbook et exercices

Source integree : workbook.html

Mode d'emploi

Ce workbook accompagne la formation. Il ne sert pas à concevoir une formation : il sert à appliquer les leçons sur un cas réel et à produire un résultat concret.

Diagnostic de départ

Question	Exemple	Espace de réponse
Quel est ton objectif principal ?	Je veux appliquer cette formation sur un projet réel et obtenir un livrable utilisable.	
Quel est ton niveau actuel ?	Débutant	
Quel livrable veux-tu obtenir ?	À la fin, tu sais quel outil utiliser selon ton besoin et quelles règles appliquer avant modification de code.	

Exercice 1 - C'est quoi un outil IA de code ?

Consigne

Classe 6 tâches selon l'outil approprié.

Question guidée	Exemple	Ta réponse
Quel contexte vas-tu utiliser ?	Je ne sais pas quel outil choisir.	
Quel résultat veux-tu obtenir ?	Une matrice besoin outil.	
Comment vas-tu vérifier la qualité ?	Chaque outil est associé à un usage clair.	

Zone de travail

Corrigé type

Tu dois savoir quand demander une explication et quand autoriser une modification.

Exercice 2 - Assistant IA ou agent IA

Consigne

Transforme une demande risquée en demande sécurisée.

Question guidée	Exemple	Ta réponse
Quel contexte vas-tu utiliser ?	Je veux utiliser un agent sans casser mon projet.	
Quel résultat veux-tu obtenir ?	Une consigne agent sécurisée.	
Comment vas-tu vérifier la qualité ?	La consigne impose analyse avant modification.	

Zone de travail

Corrigé type

Le bon réflexe est d'exiger une analyse avant action.

Exercice 3 - Claude et ChatGPT

Consigne

Prépare un brief technique avec ChatGPT ou Claude.

Question guidée	Exemple	Ta réponse
Quel contexte vas-tu utiliser ?	J'ai un problème technique à comprendre.	
Quel résultat veux-tu obtenir ?	Un brief technique clair.	
Comment vas-tu vérifier la qualité ?	Le brief contient contexte, hypothèses et tests.	

Zone de travail

Corrigé type

La réponse doit t'aider à décider, pas juste fournir du code.

Exercice 4 - Claude Code

Consigne

Écris une consigne Claude Code lecture seule.

Question guidée	Exemple	Ta réponse
Quel contexte vas-tu utiliser ?	Je veux comprendre un projet existant.	
Quel résultat veux-tu obtenir ?	Une demande d'analyse codebase.	
Comment vas-tu vérifier la qualité ?	La demande interdit les modifications au départ.	

Zone de travail

Corrigé type

La sortie doit être une carte du projet et un plan de modification.

Exercice 5 - Codex

Consigne

Écris un prompt Codex pour une petite fonctionnalité.

Question guidée	Exemple	Ta réponse
Quel contexte vas-tu utiliser ?	Je veux ajouter ou corriger quelque chose dans un site.	
Quel résultat veux-tu obtenir ?	Un prompt Codex prêt.	
Comment vas-tu vérifier la qualité ?	Le prompt mentionne fichiers, contraintes et validation.	

Zone de travail

Corrigé type

Le résultat doit être limité, testable et cohérent avec le projet.

Exercice 6 - OpenClaw

Consigne

Crée un workflow personnel simple.

Question guidée	Exemple	Ta réponse
Quel contexte vas-tu utiliser ?	Je veux automatiser une petite routine.	
Quel résultat veux-tu obtenir ?	Un workflow OpenClaw.	
Comment vas-tu vérifier la qualité ?	Le workflow est clair et contrôlable.	

Zone de travail

Corrigé type

Le workflow doit aider sans créer de dépendance aveugle.

Exercice 7 - Choisir selon son besoin

Consigne

Remplis la matrice besoin outil.

Question guidée	Exemple	Ta réponse
Quel contexte vas-tu utiliser ?	J'hésite entre plusieurs outils IA.	
Quel résultat veux-tu obtenir ?	Une matrice de choix.	
Comment vas-tu vérifier la qualité ?	Le choix est justifié par le risque et le besoin.	

Zone de travail

Corrigé type

La matrice doit éviter d'utiliser un agent quand un assistant suffit.

Exercice 8 - Sécurité avant modification

Consigne

Crée ta checklist sécurité.

Question guidée	Exemple	Ta réponse
Quel contexte vas-tu utiliser ?	Je veux laisser un agent toucher un projet.	
Quel résultat veux-tu obtenir ?	Une checklist agent IA code.	
Comment vas-tu vérifier la qualité ?	La checklist empêche les modifications non contrôlées.	

Zone de travail

Corrigé type

La demande finale doit inclure ce qui ne doit pas être modifié.

Synthèse finale

Élément	Espace de réponse
Meilleur résultat produit	
Ce qui doit être relu	
Prompt à garder	
Prochaine action	

Checklist

Source integree : checklist.html

Checklist de validation

Utilise cette checklist à la fin de la formation pour vérifier que tu as bien produit le livrable attendu.

Module	Critère de réussite
C'est quoi un outil IA de code ?	Besoin défini; Risque évalué; Outil choisi; Périmètre limité; Validation prévue
Assistant IA ou agent IA	Analyse d'abord; Fichiers listés; Plan demandé; Modification limitée; Diff relu
Claude et ChatGPT	Contexte fourni; Hypothèses listées; Tests proposés; Décision claire; Consigne prête
Claude Code	Lecture seule; Fichiers clés; Plan attendu; Risques listés; Validation humaine
Codex	Objectif clair; Périmètre limité; Contraintes présentes; Tests demandés; Résultat vérifiable
OpenClaw	Routine définie; Déclencheur clair; Résultat attendu; Garde-fous; Test prévu
Choisir selon son besoin	Besoin écrit; Risque noté; Outil choisi; Justification; Plan de validation
Sécurité avant modification	Sauvegarde prévue; Plan demandé; Périmètre limité; Diff relu; Tests faits

Contrôle final

- Je sais expliquer ce que fait ChatGPT
- Je sais écrire un prompt structuré
- J'ai testé au moins 3 usages
- J'ai sauvegardé mon prompt final

Avant de considérer la formation terminée

Point	Statut
J'ai appliqué la formation à un vrai cas	
J'ai produit un livrable utilisable	
J'ai relu les réponses IA	
J'ai corrigé les formulations trop vagues	
J'ai défini la prochaine action	

Bibliothèque de prompts

Prompts prêts a copier depuis prompts.txt

PROMPT

PROMPTS.TXT

Prompt 1

PROMPT

PROMPTS.TXT

Prompt 2

PROMPT

PROMPTS.TXT

Prompt 3

PROMPT

PROMPTS.TXT

Prompt 4

PROMPT

PROMPTS.TXT

Prompt 5

PROMPT

PROMPTS.TXT

Prompt 6

PROMPT

PROMPTS.TXT

Prompt 7

PROMPT

PROMPTS.TXT

Prompt 8

PROMPT

PROMPTS.TXT

Prompt 9

PROMPT

PROMPTS.TXT

Prompt 10

PROMPT

PROMPTS.TXT

Prompt 11

PROMPT

PROMPTS.TXT

Prompt 12

PROMPT

PROMPTS.TXT

Prompt 13

PROMPT

PROMPTS.TXT

Prompt 14

PROMPT

PROMPTS.TXT

Prompt 15

PROMPT

PROMPTS.TXT

Prompt 16

PROMPT

PROMPTS.TXT

Prompt 17

PROMPT

PROMPTS.TXT

Prompt 18

PROMPT

PROMPTS.TXT

Prompt 19

PROMPT

PROMPTS.TXT

Prompt 20

Annexes structurees

Scripts video et notes pedagogiques

Notes de cours

Source integree : course-notes.txt

Notes de cours complètes - Comprendre les Outils IA de Code

Aucune vidéo n'est fournie. Ce fichier contient les scripts écrits des leçons pour lecture, enregistrement éventuel ou support pédagogique.

MODULE 1 - C'est quoi un outil IA de code ?

Objectif . Comprendre ce que ces outils font réellement.

Leçon :

Un outil IA de code peut expliquer, générer, modifier, tester ou documenter du code selon son niveau d'accès. La différence importante est le niveau d'action : certains répondent dans une conversation, d'autres agissent dans les fichiers.

Méthode :

- Identifier le besoin
- Choisir assistant ou agent
- Limiter le périmètre
- Demander un plan
- Valider avant modification

Exemple traité :

ChatGPT explique une erreur. Codex peut modifier un fichier. Claude Code peut analyser une codebase et proposer un plan de changement.

Résultat attendu :

Tu dois savoir quand demander une explication et quand autoriser une modification.

Exercice .

Classe 6 tâches selon l'outil approprié.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 2 - Assistant IA ou agent IA

Objectif . Différencier réponse et action.

Leçon :

Un assistant te répond. Un agent peut agir dans ton environnement : lire fichiers, proposer patch, exécuter commandes. Plus l'outil agit, plus les garde-fous doivent être stricts.

Méthode :

- Demander analyse
- Lire le plan
- Limiter fichiers
- Autoriser modification
- Relire diff

Exemple traité :

Avant de demander corrige ce bug, demande analyse le bug, liste les fichiers concernés et propose un plan sans modifier.

Résultat attendu :

Le bon réflexe est d'exiger une analyse avant action.

Exercice .

Transforme une demande risquée en demande sécurisée.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 3 - Claude et ChatGPT

Objectif . Utiliser les assistants pour penser et expliquer.

Leçon :

Claude et ChatGPT sont utiles pour clarifier une idée, expliquer du code, résumer une documentation, préparer une consigne technique ou comparer des options. Ils sont souvent le bon point de départ avant d'utiliser un agent.

Méthode :

- Coller contexte
- Demander explication
- Lister hypothèses
- Choisir test
- Préparer consigne agent

Exemple traité :

Colle une erreur et demande : explique les causes probables, les informations manquantes et les tests à faire avant correction.

Résultat attendu :

La réponse doit t'aider à décider, pas juste fournir du code.

Exercice .

Prépare un brief technique avec ChatGPT ou Claude.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 4 - Claude Code

Objectif . Comprendre l'usage codebase.

Leçon :

Claude Code est pertinent quand il faut comprendre une base de code, identifier des fichiers, planifier une modification et travailler dans un projet réel. Il doit être utilisé avec un périmètre précis.

Méthode :

- Demander cartographie
- Identifier fichiers
- Valider plan
- Modifier petit
- Tester

Exemple traité :

Demande : analyse la structure du projet, identifie où ajouter cette fonctionnalité, propose un plan, n'écris pas de code pour l'instant.

Résultat attendu :

La sortie doit être une carte du projet et un plan de modification.

Exercice .

Écris une consigne Claude Code lecture seule.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 5 - Codex

Objectif . Comprendre l'usage création et modification.

Leçon :

Codex est utile pour créer ou modifier des projets, ajouter une fonctionnalité, corriger un bug, refactoriser ou préparer un déploiement. La qualité dépend de ta consigne et de la validation finale.

Méthode :

- Décrire objectif
- Limiter périmètre
- Donner contraintes
- Demander validation
- Tester

Exemple traité :

Ajoute une section pricing sur la page d'accueil en respectant le design existant, sans modifier les autres pages, puis indique comment valider.

Résultat attendu :

Le résultat doit être limité, testable et cohérent avec le projet.

Exercice .

Écris un prompt Codex pour une petite fonctionnalité.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 6 - OpenClaw

Objectif . Comprendre l'usage assistant personnel.

Leçon :

OpenClaw se positionne davantage comme assistant de productivité, organisation ou automatisation personnelle selon ton environnement. Il est utile pour gérer routines, idées et workflows.

Méthode :

- Décrire routine
- Définir déclencheur
- Préciser résultat
- Ajouter garde-fous
- Tester

Exemple traité :

Chaque matin, résume mes priorités, propose 3 actions, archive les idées non urgentes.

Résultat attendu :

Le workflow doit aider sans créer de dépendance aveugle.

Exercice .

Crée un workflow personnel simple.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 7 - Choisir selon son besoin

Objectif . Prendre la bonne décision d'outil.

Leçon :

Choisis l'outil selon la tâche : comprendre, écrire, analyser, modifier ou automatiser. Plus la tâche touche au code réel, plus tu dois exiger un plan et des tests.

Méthode :

- Définir tâche
- Évaluer risque
- Choisir outil
- Écrire consigne
- Valider

Exemple traité :

Besoin : expliquer une erreur = ChatGPT/Claude. Modifier un composant = Codex/Claude Code avec plan. Automatiser routine = OpenClaw.

Résultat attendu :

La matrice doit éviter d'utiliser un agent quand un assistant suffit.

Exercice .

Remplis la matrice besoin outil.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 8 - Sécurité avant modification

Objectif . Éviter de casser un projet.

Leçon :

Avant toute modification de code, vérifie l'état du projet, sauvegarde, demande un plan, limite le périmètre, lis le diff et teste. Ce rituel réduit les erreurs coûteuses.

Méthode :

- Vérifier état
- Sauvegarder
- Demander plan
- Limiter modification
- Tester

Exemple traité :

Avant de demander une correction : vérifie branche Git, copie erreur exacte, demande hypothèses, puis patch minimal.

Résultat attendu :

La demande finale doit inclure ce qui ne doit pas être modifié.

Exercice .

Crée ta checklist sécurité.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

Scripts video

Source integree : video-scripts.txt

Notes de cours complètes - Comprendre les Outils IA de Code

Aucune vidéo n'est fournie. Ce fichier contient les scripts écrits des leçons pour lecture, enregistrement éventuel ou support pédagogique.

MODULE 1 - C'est quoi un outil IA de code ?

Objectif . Comprendre ce que ces outils font réellement.

Leçon :

Un outil IA de code peut expliquer, générer, modifier, tester ou documenter du code selon son niveau d'accès. La différence importante est le niveau d'action : certains répondent dans une conversation, d'autres agissent dans les fichiers.

Méthode :

- Identifier le besoin
- Choisir assistant ou agent
- Limiter le périmètre
- Demander un plan
- Valider avant modification

Exemple traité :

ChatGPT explique une erreur. Codex peut modifier un fichier. Claude Code peut analyser une codebase et proposer un plan de changement.

Résultat attendu :

Tu dois savoir quand demander une explication et quand autoriser une modification.

Exercice .

Classe 6 tâches selon l'outil approprié.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 2 - Assistant IA ou agent IA

Objectif . Différencier réponse et action.

Leçon :

Un assistant te répond. Un agent peut agir dans ton environnement : lire fichiers, proposer patch, exécuter commandes. Plus l'outil agit, plus les garde-fous doivent être stricts.

Méthode :

- Demander analyse
- Lire le plan
- Limiter fichiers
- Autoriser modification
- Relire diff

Exemple traité :

Avant de demander corrige ce bug, demande analyse le bug, liste les fichiers concernés et propose un plan sans modifier.

Résultat attendu :

Le bon réflexe est d'exiger une analyse avant action.

Exercice .

Transforme une demande risquée en demande sécurisée.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 3 - Claude et ChatGPT

Objectif . Utiliser les assistants pour penser et expliquer.

Leçon :

Claude et ChatGPT sont utiles pour clarifier une idée, expliquer du code, résumer une documentation, préparer une consigne technique ou comparer des options. Ils sont souvent le bon point de départ avant d'utiliser un agent.

Méthode :

- Coller contexte
- Demander explication
- Lister hypothèses
- Choisir test
- Préparer consigne agent

Exemple traité :

Colle une erreur et demande : explique les causes probables, les informations manquantes et les tests à faire avant correction.

Résultat attendu :

La réponse doit t'aider à décider, pas juste fournir du code.

Exercice .

Prépare un brief technique avec ChatGPT ou Claude.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 4 - Claude Code

Objectif . Comprendre l'usage codebase.

Leçon :

Claude Code est pertinent quand il faut comprendre une base de code, identifier des fichiers, planifier une modification et travailler dans un projet réel. Il doit être utilisé avec un périmètre précis.

Méthode :

- Demander cartographie
- Identifier fichiers
- Valider plan
- Modifier petit
- Tester

Exemple traité :

Demande : analyse la structure du projet, identifie où ajouter cette fonctionnalité, propose un plan, n'écris pas de code pour l'instant.

Résultat attendu :

La sortie doit être une carte du projet et un plan de modification.

Exercice .

Écris une consigne Claude Code lecture seule.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 5 - Codex

Objectif . Comprendre l'usage création et modification.

Leçon :

Codex est utile pour créer ou modifier des projets, ajouter une fonctionnalité, corriger un bug, refactoriser ou préparer un déploiement. La qualité dépend de ta consigne et de la validation finale.

Méthode :

- Décrire objectif

- Limiter périmètre
- Donner contraintes
- Demander validation
- Tester

Exemple traité :

Ajoute une section pricing sur la page d'accueil en respectant le design existant, sans modifier les autres pages, puis indique comment valider.

Résultat attendu :

Le résultat doit être limité, testable et cohérent avec le projet.

Exercice .

Écris un prompt Codex pour une petite fonctionnalité.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 6 - OpenClaw

Objectif . Comprendre l'usage assistant personnel.

Leçon :

OpenClaw se positionne davantage comme assistant de productivité, organisation ou automatisation personnelle selon ton environnement. Il est utile pour gérer routines, idées et workflows.

Méthode :

- Décrire routine
- Définir déclencheur
- Préciser résultat
- Ajouter garde-fous
- Tester

Exemple traité :

Chaque matin, résume mes priorités, propose 3 actions, archive les idées non urgentes.

Résultat attendu :

Le workflow doit aider sans créer de dépendance aveugle.

Exercice .

Crée un workflow personnel simple.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 7 - Choisir selon son besoin

Objectif . Prendre la bonne décision d'outil.

Leçon :

Choisis l'outil selon la tâche : comprendre, écrire, analyser, modifier ou automatiser. Plus la tâche touche au code réel, plus tu dois exiger un plan et des tests.

Méthode :

- Définir tâche
- Évaluer risque
- Choisir outil
- Écrire consigne
- Valider

Exemple traité :

Besoin : expliquer une erreur = ChatGPT/Claude. Modifier un composant = Codex/Claude Code avec plan. Automatiser routine = OpenClaw.

Résultat attendu :

La matrice doit éviter d'utiliser un agent quand un assistant suffit.

Exercice .

Remplis la matrice besoin outil.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

MODULE 8 - Sécurité avant modification

Objectif . Éviter de casser un projet.

Leçon :

Avant toute modification de code, vérifie l'état du projet, sauvegarde, demande un plan, limite le périmètre, lis le diff et teste. Ce rituel réduit les erreurs coûteuses.

Méthode :

- Vérifier état
- Sauvegarder
- Demander plan
- Limiter modification
- Tester

Exemple traité :

Avant de demander une correction : vérifie branche Git, copie erreur exacte, demande hypothèses, puis patch minimal.

Résultat attendu :

La demande finale doit inclure ce qui ne doit pas être modifié.

Exercice .

Crée ta checklist sécurité.

Transition .

Quand ce livrable est terminé, passe au module suivant. Ne cherche pas une version parfaite : cherche une version claire, relue et utilisable.

Prochaine action

Choisissez un cas réel à traiter aujourd'hui, ouvrez le prompt ou la checklist la plus proche de votre besoin, puis produisez une première version exploitable en moins de 30 minutes.

Etape	Action	Preuve de progression
1	Choisir le cas d'usage prioritaire.	Un objectif écrit en une phrase.
2	Appliquer un prompt, workflow ou exercice.	Un premier livrable généré.
3	Relire, corriger, personnaliser.	Une version prête à utiliser.

Conclusion

Ce document sert de point d'entrée principal. Les fichiers TXT et CSV restent fournis dans le dossier comme bonus copiables, mais la version consolidée ci-dessus est celle à lire en premier.

Source imprimable : final-product.html