

Développer avec des agents

Une méthode en 5 étapes. Un support à garder.

9h00 • Prendre en main

Exercices courts sur les outils et le ticket.

9h40 • Préparer le lancement

Contrat commun, puis agents par lot.

12h15 • Montrer le MVP

Un besoin, une réponse, une structure, un contact.

Commencez par votre besoin du jour.

Projeter maintenant. Retrouver ensuite

Lisez l'essentiel à l'écran ; ouvrez « Théorie » pour approfondir.

Parcourir

Flèches, défilement ou sommaire.

Copier

Un prompt ou une commande par méthode.

Approfondir

Explications, limites et sources dans « Théorie ».

Emporter

Le deck, le PowerPoint et le kit local.

Ouvrez une fiche « Théorie », puis refermez-la.

Un agent avance dans une boucle

Lire → planifier → agir → vérifier → recommencer.

Lire : Trouver les fichiers et comprendre le besoin.

Prompt à copier

Planifier : Choisir une prochaine action limitée.

Lis le ticket et les fichiers attribués.
Propose un plan en trois actions avec une preuve par action.
Attends mon retour avant de modifier les fichiers.

Agir : Appeler un outil, modifier, exécuter.

Vérifier : Comparer le résultat au critère Done.

Exercice · 2 min : repérez la prochaine preuve de votre ticket.

Nommer ce qui décide et ce qui agit

Quatre mots pour comprendre une session d'agent.

Modèle

Produit du texte et propose les actions.

Agent

Organise la boucle autour du modèle.

Outil

Exécute une action : lire, tester, chercher.

MCP

Relie l'agent à des outils externes.

Exercice · 2 min : classez modèle, agent, Firecrawl et MCP.

Nommer ce qui organise le travail

Quatre mots pour passer d'une demande à une livraison.

Skill

Une compétence réutilisable dans SKILL.md.

Sous-agent

Un exécutant avec son propre contexte.

Workflow

Un enchaînement de tâches et de contrôles.

Ticket

Un résultat attendu, un périmètre, un Done.

Exercice · 2 min : décrivez votre ticket en une phrase.

Le contexte se choisit

Donnez ce qui change la décision de l'agent.

Garder : Besoin, ticket, conventions, fichiers et preuves.

Écarter : Journaux bruts, doublons et historique inutile.

Inspecter : /context montre l'occupation du contexte.

Reprendre : /compact résumé ; /resume retrouve une session.

Prompt à copier

Résume la session pour une reprise :
objectif, décisions, fichiers, preuves
et blocages.
Sépare ce qui est confirmé de ce qui
reste à vérifier.

Exercice · 3 min : gardez cinq faits utiles pour votre ticket.

Cinq étapes. Trois points de contrôle

Relisez le plan, contrôlez chaque lot, vérifiez avant de fusionner.

1 Cadrer : Un résultat mesurable et un ticket Ready.

2 Donner le contexte : Les bons fichiers et les frontières.

3 Améliorer le prompt : Cinq blocs, une demande exécutable.

4 Lancer : Un lot attribué, un agent responsable.

5 Vérifier et garder : Une preuve, puis une connaissance réutilisable.

Choisissez l'étape qui manque aujourd'hui à votre pratique.

1 Cadrer

Un ticket devient Ready quand l'agent peut commencer sans deviner.

Objectif SMART : Spécifique, mesurable, atteignable, utile, daté.

Ready : Données, contrat et périmètre disponibles.

Done : Résultats visibles et validation reproductible.

Prompt à copier

```
/ticket-ready Afficher les organismes  
du jeu validé  
avec nom, source et moyens de contact  
disponibles.  
Lot : src/features/annuaire/. Échéance  
: 10h45.
```

Exercice · 5 min : rendez votre ticket Ready.

2 Donner le contexte

Un dossier attribué et un contrat commun évitent les collisions.

Lire : CLAUDE.md, README et les conventions.

Attribuer : Un lot possède son dossier
src/features/.

Partager : Types et signatures dans src/types/.

Préserver : Les secrets et les fichiers des autres
lots.

Prompt à copier

```
Lis CLAUDE.md, README.md et  
docs/CONVENTIONS.md.  
Lis src/types/orientation.ts s'il  
existe.  
Mon lot : src/features/annuaire/.  
Signale tout fichier manquant. Propose  
un plan, sans coder.
```

Exercice · 3 min : montrez les fichiers autorisés de votre lot.

3 Améliorer le prompt

Remplacez « fais le chatbot » par cinq blocs précis.

Rôle + contexte : Qui agit, sur quels fichiers et données.

Tâche : Une action et un résultat attendu.

Contraintes + exemples : Les limites et un cas observable.

Prompt à copier · extrait

```
[RÔLE] Développeur front de ce dépôt.  
[CONTEXTE] Lot src/features/annuaire/  
données validées.  
[TÂCHE] Affiche le nom, la source et  
les contacts disponibles.  
[CONTRAINTES] Respecte  
src/types/orientation.ts.
```

Prompt complet dans les notes et le HTML

Exercice · 5 min : utilisez /prompt-5 sur votre ticket.

4 Lancer

Un lot indépendant reçoit une mission et rend des preuves.

Planifier : Valider le plan avant les modifications.

Séparer : Un dossier par lot ; contrat en lecture seule.

Rapporter : Fichiers, commande, résultat, blocages.

Prompt à copier · extrait

```
Lance un sous-agent pour le ticket ANNU-01 validé.  
Il travaille uniquement dans src/features/annuaire/.  
Il lit le contrat commun sans le modifier.  
Exige : fichiers modifiés,
```

Prompt complet dans les notes et le HTML

Exercice · 3 min : lancez un seul lot prêt.

5 Vérifier et garder

« L'agent a fini » devient Done avec une preuve.

Relire : Comprendre le diff et les sources.

Valider : Lint, build et parcours utilisateur.

Éprouver : Cas vide, contact absent et langue de repli.

Garder : Décision, source et preuve dans le wiki.

Commandes du projet

```
npm run lint  
npm run build
```

Exercice · 5 min : produisez une preuve de votre Done.

La recette, l'exécutant et le parcours

Skill, sous-agent et workflow répondent à trois besoins distincts.

Skill · comment faire : /prompt-5 reformule une demande.

Sous-agent · qui exécute : Un agent construit le lot annuaire.

Workflow · dans quel ordre : Cadrer, contextualiser, reformuler, lancer, vérifier.

Exercice · 2 min : nommez le skill et l'agent de votre lot.

Prompt à copier

Décris le workflow du ticket ANNU-01.
Pour chaque étape : compétence,
exécutant, entrée, preuve.
Reste dans les cinq étapes du support.

Lancer le workflow en cinq gestes

Le prompt reformulé est votre point de contrôle.

1 Cadrer : Ouvrez le projet et choisissez un ticket.

2 Donner le contexte : Indiquez contrat et dossier attribué.

3 Améliorer le prompt : Relisez les cinq blocs, puis dites « ok ».

4 Lancer : L'agent exécute le lot et rend son rapport.

5 Vérifier et garder : Contrôlez le Done et proposez une note wiki.

Prompt à copier

```
/lancer-workflow ANNU-01  
Le ticket se trouve dans  
TICKETS-DEMAIN.md.  
Périmètre : src/features/annuaire/.  
Affiche d'abord le prompt reformulé.
```

Exercice · 5 min : testez jusqu'à l'affichage du plan.

CLI et Desktop : ouvrir le même projet

Dans Desktop, choisissez l'onglet Code et une session locale.

Ouvrir : Sélectionnez le dépôt, puis le dossier du projet.

Partager : CLAUDE.md, skills et configuration MCP.

Vérifier : Demandez le dossier courant et les skills disponibles.

Prompt à copier

```
Indique le dossier courant, sans  
afficher de secret.  
Liste les skills disponibles, noms  
seulement.  
Trouves-tu prompt-5 et lancer-workflow  
?
```

Exercice · 3 min : retrouvez /prompt-5 dans le bon projet.

Créer un skill en un fichier

Placez la procédure dans `.claude/skills/nom/SKILL.md`.

Nommer : Un nom court et une description du besoin.

Décrire : Les entrées, les gestes et le résultat attendu.

Essayer : Appelez /nom avec un cas concret.

Structure d'un SKILL.md · extrait

```
---
name: ticket-ready
description: Reformule un besoin en
ticket SMART, Ready et Done.
argument-hint: "<besoin>"
---
```

Lis \$ARGUMENTS. Retourne objectif,

Prompt complet dans les notes et le HTML

Exercice · 5 min : copiez le skill prompt-5 du kit.

Brancher un MCP, puis vérifier

Un serveur MCP expose des outils ; il faut encore les tester.

Choisir : Le serveur officiel et sa portée.

Dans le terminal

```
claude mcp list
```

Connecter : Ajouter sa configuration à Claude Code.

Contrôler : Lister le serveur, puis essayer une requête.

Exercice · 3 min : trouvez le serveur et son état.

Context7 : demander la bonne documentation

Donnez la bibliothèque et sa version avant de coder.

Configurer : Dans le terminal :
`npx ctx7 setup --claude`

Demander : Nommez la bibliothèque, sa version et la question.

Comparer : Vérifiez les signatures et les sources reçues.

Prompt à copier

```
use context7 pour React, version  
installée dans package.json.  
Comment gérer un formulaire contrôlé  
et son état de chargement ?  
Cite la documentation et montre un  
exemple minimal.
```

Exercice · 3 min : vérifiez une API React utilisée dans le lot.

Firecrawl : garder le texte et sa source

Une extraction devient utile après comparaison avec la page.

Connecter : Choisissez le mode sans clé ou votre compte.

Extraire : Demandez une seule URL validée.

Contrôler : Gardez texte, URL, date et limites.

Prompt à copier · extrait

Avec Firecrawl, lis l'URL de structure sociale donnée par le PO.
Extrais titre, texte utile, URL et date dans mon lot.
N'invente aucun contact ; note les champs absents.
Montre l'extrait source avant de

Prompt complet dans les notes et le HTML

Exercice · 5 min : extrayez et relisez une page de la liste.

Archify : montrer les frontières du projet

Un schéma rend le contrat et les responsabilités visibles.

Installer : Suivez le README officiel du dépôt Archify.

Décrire : Entrées, trois lots et point d'intégration.

Relire : Ouvrez le HTML et vérifiez chaque flèche.

Prompt à copier · extrait

```
/archify Schématise Reconnect Assist :  
Romain = connaissances, Ashad =  
assistant, Nils = annuaire.  
Contrat partagé :  
src/types/orientation.ts.  
Intégration : src/app/ et  
src/services/.
```

Prompt complet dans les notes et le HTML

Exercice · 5 min : expliquez le schéma sans lire le code.

OpenCode : conserver la même méthode

L'agent change ; le ticket, les limites et le Done restent.

Ouvrir : Installez OpenCode depuis sa documentation.

Connecter : /connect choisit un fournisseur ;
/models un modèle.

Cadrer : /init prépare les règles dans AGENTS.md.

Prompt à copier

```
Lis AGENTS.md et le ticket ANNU-01.  
Propose le plan du lot  
src/features/annuaire/.  
Indique les données et accès  
manquants, sans les deviner.
```

Découverte · 3 min : comparez le rôle de AGENTS.md.

Buzz : explorer un espace humains et agents

Une découverte de collaboration, à tester hors du chemin critique.

Découvrir : Lisez le projet public et ses versions.

Préparer : Un canal, un besoin et des documents non sensibles.

Évaluer : Qui répond, avec quel accès et quelle source ?

Prompt à copier · extrait

Scénario à essayer après configuration de Buzz :
Résume les décisions du lot
connaissances disponibles ici.
Cite les messages ou fichiers
utilisés.
Si une décision manque, indique-le

Prompt complet dans les notes et le HTML

Découverte · 2 min : formulez une question utile au canal.

Karpathy : faire vivre un wiki Markdown

Les sources restent brutes ; le wiki relie ce que vous apprenez.

Ingest · intégrer : Lire une source, créer une page reliée.

Query · interroger : Répondre depuis le wiki en citant.

Lint · entretenir : Détecter liens cassés, doublons et contradictions.

Prompt à copier · extrait

```
Lis une source dans wiki/raw/ sans la
modifier.
Propose une page dans wiki/pages/ avec
son lien source.
Signale les incertitudes et les
contradictions.
Propose ensuite une entrée dans
```

Prompt complet dans les notes et le HTML

Exercice · 5 min : proposez une page à partir d'une source locale.

Le MVP tient dans un parcours

Choisir un besoin → comprendre → trouver une structure → contacter.

Public

Des personnes confrontées aux barrières de langue et d'accès.

Stack réelle

Projet App Web : React, Vite et TypeScript.

Périmètre proposé

Une ville, des sources choisies, un mode de réponse confirmé.

Reporté

Chat libre, carte intégrée et couverture nationale.

À 9h00 : validez le parcours et ce que vous reportez.

Trois lots. Un point d'intégration

Répartition proposée : chaque personne confirme son périmètre.

Romain · connaissances

src/features/base-connaissances/

Ashad · assistant

src/features/assistant/

Nils · annuaire

src/features/annuaire/

Cédric · intégration

Contrat, services et application commune.

Avant de lancer : pointez le dossier de votre lot.

Le contrat permet de travailler en parallèle

Les types partagés deviennent la frontière entre les lots.

Entrée : OrientationQuery : langue, besoin, ville, mots-clés.

Connaissances : KnowledgeItem : texte, catégorie et source.

Structures : Organisation : identité et contacts disponibles.

Sortie : OrientationResult : fiches, organismes, explication.

Signatures proposées

```
searchKnowledge(query):  
  Promise<KnowledgeItem[]>  
findOrganisations(query):  
  Promise<Organisation[]>  
explain(query, items):  
  Promise<Explanation>
```

À 9h55 : relisez les signatures avant de lancer les lots.

KB-01 · Romain

Base de connaissances sourcée

SMART : 8 fiches sourcées et consultables à 10h45.

Ready : Contrat, catégories et sources disponibles.

Done : Recherche, source et cas vide vérifiés.

Fichiers : src/features/base-connaissances/**

Ouvrez TICKETS-DEMAIN.md et confirmez Ready.

Prompt à copier

```
/prompt-5 KB-01 dans TICKETS-DEMAIN.md  
Vérifie Ready avant de proposer le  
plan.  
Indique les preuves nécessaires pour  
passer Done.
```

ASSIST-01 · Ashad

Assistant multilingue simple

SMART : Un parcours multilingue et un mode choisi à 10h45.

Ready : Contrat, langues ; endpoint validé ou repli choisi.

Done : Sources, erreurs et mode réel vérifiés.

Fichiers : src/features/assistant/**

Prompt à copier

```
/prompt-5 ASSIST-01 dans  
TICKETS-DEMAIN.md  
Vérifie Ready avant de proposer le  
plan.  
Indique les preuves nécessaires pour  
passer Done.
```

Ouvrez TICKETS-DEMAIN.md et confirmez Ready.

ANNU-01 · Nils

Annuaire et contacts vérifiables

SMART : 6 structures et leurs contacts vérifiables à 10h45.

Ready : Contrat, ville et sources des coordonnées.

Done : Filtres, champs absents et liens contrôlés.

Fichiers : src/features/annuaire/**

Prompt à copier

```
/prompt-5 ANNU-01 dans  
TICKETS-DEMAIN.md  
Vérifie Ready avant de proposer le  
plan.  
Indique les preuves nécessaires pour  
passer Done.
```

Ouvrez TICKETS-DEMAIN.md et confirmez Ready.

INT-01 · Cédric

Contrat commun et intégration

SMART : Contrat à 9h55 ; intégration prête à 12h00.

Ready : Décisions PO et périmètres partagés.

Done : Parcours complet et cas limites vérifiés.

Fichiers : src/app/, src/types/, src/services/ et assemblage.

Ouvrez TICKETS-DEMAIN.md et confirmez Ready.

Prompt à copier

```
/prompt-5 INT-01 dans  
TICKETS-DEMAIN.md  
Vérifie Ready avant de proposer le  
plan.  
Indique les preuves nécessaires pour  
passer Done.
```

La matinée alterne travail et apprentissage

Les heures sont des cibles de séance, à ajuster aux blocages.

9h00–9h40 : Décisions PO, exercices outils et tickets.

9h40–10h05 : Contrat commun, reformulation et lancement.

10h05–10h45 : Théorie pendant l'exécution ; point blocages.

10h45–11h45 : Relecture, intégration et corrections.

11h45–12h30 : Recette, démonstration à 12h15, bilan.

À chaque point : signalez terminé, bloqué ou à réduire.

Sans réseau : montrer ce qui existe vraiment

Fichiers locaux disponibles $\neq$ agent cloud disponible.

Garder

Sources, traductions et fixtures déjà enregistrées.

Poursuivre

Relire, assembler et corriger localement si possible.

Afficher

Une réponse préparée, identifiée « mode sans IA ».

Reporter

Collecte, nouveaux paquets et validation distante.

Exercice · 3 min : annoncez la limite de votre plan B.

Le PO vérifie un résultat humain

La démo répond à un besoin et montre ses limites.

Comprendre : Le besoin et la langue se choisissent facilement.

Expliquer : La réponse indique sa source et le mode réellement testé.

Orienter : La structure correspond au besoin et à la ville.

Contacter : Chaque contact affiché vient d'une source.

Échouer proprement : Aucun résultat ou champ absent reste compréhensible.

Prompt à copier

Recette : besoin → réponse → organisme
→ contact disponible.

Teste aussi ville inconnue et
téléphone absent.

Note attendu, observé et preuve.

Classe les écarts : bloquant pour la
démo ou à reporter.

À 11h45 : jouez le scénario depuis un écran neuf.

La fiche à garder

Reprenez cette routine au prochain ticket.

1 Cadrer : « Transforme ce besoin en ticket Ready et Done. »

2 Donner le contexte : « Lis les règles, le contrat et mon dossier. »

3 Améliorer le prompt : « Reformule la demande en cinq blocs. »

4 Lancer : « Exécute ce lot et rapporte les preuves. »

5 Vérifier et garder : « Contrôle le Done et propose une note wiki. »

Conservez FICHE-5-ETAPES.md près de vos tickets.

Le kit se copie dans le projet

Trois skills et des fichiers de référence prêts à adapter.

Reformuler : /prompt-5 et /ticket-ready.

Exécuter : /lancer-workflow en cinq étapes.

Préparer : Tickets, contrat et exemple de CLAUDE.md.

Garder : Wiki, sources et plan sans réseau.

Prompt à copier

```
/prompt-5 Afficher les organismes du  
jeu validé.  
Retourne les cinq blocs, Ready et Done  
; ne code rien.
```

Copiez les trois dossiers de skills, puis ouvrez /prompt-5.

Prenez un ticket. Rendez-le Ready

La prochaine action tient en cinq minutes.

Votre point de départ : Le besoin, les fichiers et la preuve attendue.

Votre soutien : Cédric Atticot dit Ravino · formateur et PO.

Votre repère : 1 Cadrer → 2 Donner le contexte → 3 Améliorer le prompt

Prompt à copier

```
/ticket-ready Mon prochain besoin est  
:  
Décris ici le changement attendu pour  
l'utilisateur.
```

Lancez /ticket-ready sur votre prochain besoin.