

Claude Code : construire un outil interne en 7 jours

Ce que Claude Code change par rapport à un chat, l'installation en une commande, le gabarit de brief à copier et le plan sur 7 jours vers ton premier outil interne.

BLYX-AGENCY.COM · 1 AVRIL 2026 · PAR MELVIN BRETON

Claude Code est l'outil d'Anthropic qui fait de Claude un exécutant : au lieu de te répondre dans une fenêtre de discussion, il travaille dans tes fichiers, écrit le code, le teste et le corrige jusqu'au résultat. Il s'installe en une commande, il est inclus dans l'abonnement Claude payant, et c'est le chemin le plus court pour qu'un dirigeant de TPE ou PME construise un outil interne simple. Ce guide couvre l'installation, la méthode, le gabarit de brief à copier et le plan sur 7 jours.

1. Claude Code en une phrase

Un chat comme ChatGPT ou Claude te rend une réponse, que tu dois ensuite appliquer toi-même. Claude Code inverse le rapport : tu décris le résultat attendu, il fait le travail dans ton dossier, et à la fin tu regardes ce qui existe, pas ce qui est écrit.

Concrètement, c'est le même modèle Claude, installé sur ton ordinateur, avec trois droits que tu lui accordes : lire tes fichiers, en écrire, et lancer des commandes pour vérifier son propre travail. Chaque action sensible passe par une demande de permission, que tu acceptes ou refuses.



Tu demandes à un chat



« Explique-moi comment suivre mes devis »

Tu demandes à Claude Code



« Construis-moi le tableau de suivi des devis »

« Écris-moi le texte de cette page »

« Modifie la page et montre-moi le résultat »

« Aide-moi à comprendre cette erreur »

« Corrige cette erreur et vérifie que ça tourne »

Un détail compte plus que tout le reste pour un usage d'entreprise : Claude Code lit un fichier nommé CLAUDE.md placé à la racine de ton dossier de travail. Tu y écris une fois ton activité, tes règles et tes conventions, et il le relit au début de chaque session. C'est ta mémoire d'entreprise, celle qu'un chat classique te fait retaper à chaque conversation.

2. Installer Claude Code

Il te faut un ordinateur récent (macOS 13 ou plus, Windows 10 ou plus, ou Linux, avec 4 Go de mémoire), une connexion internet et un abonnement Claude payant. L'offre gratuite de claude.ai ne donne pas accès à Claude Code. L'abonnement Pro, à environ 20 euros par mois, suffit pour commencer, et Claude Code y est inclus sans supplément.

Étape 1. Ouvre le terminal. Sur Mac : touche Cmd + espace, tape « Terminal », valide. Sur Windows : menu Démarrer, tape « PowerShell », valide.

Étape 2. Colle la commande d'installation et valide. Sur Mac et Linux





```
curl -fsSL https://claude.ai/install.sh | bash
```



Sur Windows, dans PowerShell :

```
irm https://claude.ai/install.ps1 | iex
```

Étape 3. Place-toi dans le dossier où tu veux travailler, tape `claude` et valide. À la première ouverture, une page s'ouvre dans ton navigateur pour te connecter avec ton compte claude.ai.

Étape 4. Vérifie que tout est en place avec `claude --version`, qui doit afficher un numéro de version. En cas de doute, `claude doctor` fait un diagnostic complet de l'installation.

Procédure vérifiée le 30/08/2026 sur la documentation officielle d'Anthropic. L'installation se met ensuite à jour toute seule, en arrière-plan.

Si le terminal te rebute vraiment, Anthropic propose aussi une application de bureau et une version dans le navigateur sur claude.ai/code. Le terminal reste la version la plus complète, et c'est celle que ce guide suppose.

Dernier point avant de commencer : Claude Code consomme le quota de ton abonnement exactement comme une conversation normale, et une session de construction pèse plus lourd qu'une question dans le chat. Les réflexes pour ne pas gaspiller sont détaillés dans [le guide pour faire durer ton quota Claude](#).



3. Les cinq cas où Claude Code est rentable



Premier cas, le plus évident : prototyper un outil interne. Un suivi de

prospects, un tableau de bord, un formulaire de saisie, une petite interface au-dessus d'un fichier existant. Tu obtiens en quelques heures une version qui se teste, là où un projet de développement classique t'aurait demandé des semaines.

Deuxième cas : ajouter une brique à quelque chose qui existe déjà. Une règle de calcul dans un outil, une étape de qualification dans un formulaire, un export qui manque depuis des mois.

Troisième cas : corriger et modifier plus vite. Quand un outil interne bloque sur un détail, tu décris le problème et tu regardes la correction se faire, au lieu d'attendre un prestataire pour vingt minutes de travail.

Quatrième cas : mettre de l'ordre. Réorganiser des fichiers, structurer un dossier partagé, clarifier une logique devenue illisible avec le temps.

Cinquième cas, le plus rentable pour une PME : traduire un besoin métier en outil. Le suivi commercial, la qualification des demandes entrantes et le tableau de bord de gestion sont les trois exemples qui reviennent le plus souvent, et la section 7 les détaille.

Un cadrage honnête pour finir : ce guide traite de la fabrication d'outils. Si ton besoin est plutôt de brancher l'IA sur tes mails, tes fichiers Office et ta veille au quotidien, sans rien construire, le bon point de départ est [le guide pour intégrer l'IA dans sa PME](#) de Pierre.

4. Les cas où il ne faut pas l'utiliser



Si ton besoin est encore flou, il produira du bruit : des allers-retours sans fin et un faux sentiment d'avancement. Si le problème métier n'est pas défini, la solution sera mal calibrée, quelle que soit la qualité de l'exécution. Si tu attends un produit final clé en main sans jamais tester, tu seras déçu, parce que la valeur vient de l'itération. Et « voir ce que ça donne » n'est pas un objectif : sans résultat attendu, il n'y a rien à évaluer.

Il reste enfin un territoire où le prototype ne suffit jamais : tout ce qui touche à la paie, à la facturation légale, aux données de santé ou à un processus dont l'arrêt bloque l'entreprise. Là, on prototype pour comprendre le besoin, puis on fait construire proprement. Cette frontière est le vrai sujet de la section 10.

Pour situer la frontière plus largement, entre ce qui reste à toi et ce qui se délègue, la [méthode Humain + IA](#) pose les trois zones en un tableau.

5. La méthode en cinq étapes

Étape 1 : définis le résultat attendu. Quatre questions y suffisent. Quel est le problème (« les commerciaux perdent du temps à retrouver les infos prospects ») ; qui est l'utilisateur (« une équipe de vente de 4 personnes ») ; quel résultat concret (« un écran simple pour retrouver les infos avant un appel ») ; quelle première version minimale utile (« recherche, fiche prospect, statut, note »).



Étape 2 : donne le contexte métier. Pourquoi tu fais ça, qui s'en servira, comment ça doit changer dans la semaine de quelqu'un. Sans ce contexte, tu obtiens un outil générique qui ne colle à rien.



Étape 3 : demande une V1, pas un monstre. « Fais le produit complet » est la pire demande possible. La bonne : « fais-moi une première version simple autour du flux principal ». Trois à cinq fonctions, pas plus.

Étape 4 : itère bloc par bloc. D'abord la structure, puis l'interface minimale, puis la logique principale, puis la connexion aux données, puis les corrections. Un bloc à la fois, validé avant de passer au suivant.

Étape 5 : reprends la main régulièrement. Le mauvais usage, c'est « vas-y, je reviens plus tard ». Le bon, c'est un cycle court : demander, vérifier, recadrer, relancer. Tu pilotes, Claude Code exécute.

6. Le gabarit de brief à copier

Ce gabarit condense les cinq étapes en un seul message. Copie-le au début de chaque nouveau projet et remplace les crochets.

Je veux que tu m'aides à construire une V1 simple et utile d'un outil interne.

Contexte :

[ton activité, et pourquoi tu fais ça]

Utilisateurs :

[qui va s'en servir]

Problème actuel :

[la friction d'aujourd'hui]



Objectif :

[le résultat concret attendu]



Version V1 attendue :

- [fonction 1]
- [fonction 2]
- [fonction 3]

Contraintes :

- rester simple
- interface claire
- éviter les fonctionnalités secondaires
- privilégier l'utilisable au parfait

Ce que j'attends de toi :

1. proposer une structure logique
2. construire la V1
3. expliquer brièvement tes choix
4. rester focalisé sur l'objectif métier

Le gabarit a l'air banal. C'est sa force : il t'oblige à répondre aux questions de cadrage avant de lancer la machine, et c'est exactement l'exercice que la plupart des gens sautent.

7. Trois outils internes reproductibles

Le suivi de prospects. Le besoin type : les commerciaux jonglent entre plusieurs fichiers et personne ne sait où en est un contact. La V1 : une interface avec la fiche du prospect, son historique, un statut, une note et une date de rappel. C'est aussi la porte de sortie naturelle du classeur Excel qui déborde, un cas assez fréquent pour avoir [son propre article](#).

L'interface au-dessus de tes données. Le besoin type : les informations vivent dans Notion, un tableur ou un logiciel métier, m



l'écran est trop lourd pour l'usage quotidien de l'équipe. La V1 : un petit

 Blyx chercher les données là où elles sont et les présente pour 
un seul usage précis, la préparation d'un rendez-vous par exemple.

La qualification des demandes entrantes. Le besoin type : les demandes arrivent par le formulaire du site et par mail, sans tri. La V1 : un outil qui les rassemble, les classe par priorité selon tes critères et prépare un brouillon de première réponse. On parle bien de tes demandes entrantes : pour aller chercher des prospects qui ne t'ont rien demandé, le cadre légal est un sujet en soi, et il est traité dans [la page de Pierre sur la prospection LinkedIn](#).

Le [comité de direction Claude Code](#), six agents qui analysent ta boîte sous cinq angles, montre ce que ces briques permettent une fois assemblées.

8. Les cinq erreurs qui coûtent le plus

La première : demander trop large. « Crée-moi une application complète » ouvre un chantier ingérable. Une V1, un périmètre précis.

La deuxième : ne donner aucun contexte métier. « Crée un outil de prospection » sans dire qui prospecte, comment et pourquoi, donne un outil hors sol.

La troisième : enchaîner les demandes au hasard. Un projet Claude Code se pilote comme un petit chantier, avec un ordre et des points de contrôle, pas comme une conversation qui divague.

La quatrième : aller trop vite. « Construis maintenant » avant d'avoir clarifié le besoin, c'est perdre du temps avec une grande efficacité



La cinquième : ne pas découper. Tout faire d'un coup garantit de ne

rien finir. La règle qui protège : un besoin, un objectif, une itération.



9. Ton plan sur sept jours

Une semaine suffit pour sortir une première version et savoir ce qu'elle vaut.

Jour	Action
1	Choisis un seul problème métier. Pas cinq, un seul
2	Décris le problème, l'utilisateur, le résultat attendu et la V1 minimale
3	Installe Claude Code si ce n'est pas fait, colle le gabarit de brief, lance la première version
4	Teste la logique avec de vraies situations, sans juger l'esthétique
5	Corrige ce qui bloque, une chose à la fois
6	Simplifie et supprime le superflu
7	Décide : V2, arrêt, usage tel quel, ou développement propre

Même si tu t'arrêtes au jour 7 sans aller plus loin, tu auras gagné un prototype qui tourne, une compréhension précise de ton besoin et un cahier des charges involontaire. Si la décision du jour 7 est de faire développer, tu sauras exactement quoi chiffrer, et [le prix d'un outil interne sur mesure](#) te donne les ordres de grandeur.



10. Ce que ça change côté chiffres

La valeur d'un outil interne ne se mesure pas à la démo. Elle se mesure à la **Blyx** rendu chaque semaine, à la friction supprimée dans le quotidien de l'équipe, et à la capacité de tester une idée avant d'y investir.



Un objectif raisonnable pour une V1 bien cadrée : un outil qui te rend deux heures par semaine. Ce n'est pas une promesse, c'est le seuil à viser pour décider si le sujet mérite d'exister. En dessous, le problème choisi était probablement trop petit.

Le prototype a aussi une valeur cachée : il fait le tri entre ce que tu peux bricoler seul et ce qui mérite d'être construit proprement. Le jour où l'outil devient celui dont ton équipe dépend, avec plusieurs utilisateurs, des données clients et un besoin de fiabilité, le bricolage a rempli son rôle et passe le relais. C'est exactement le point d'entrée de nos [solutions sur mesure](#) : tu arrives avec un prototype qui tourne et un besoin déjà validé, et la construction part sur des fondations saines au lieu d'une page blanche.

Dernier point de budget : Claude Code consomme ton quota plus vite qu'un chat classique, les [réglages qui font durer ton abonnement](#) valent le détour avant de te lancer.

Questions fréquentes

Faut-il savoir coder pour utiliser Claude Code

Non. Tu décris ce que tu veux en français, Claude Code écrit et exécute le code. Ce qui fait la différence n'est pas la technique, c'est la qualité du cadrage : un besoin précis, un utilisateur identifié, une V1 minimale. Savoir lire dans les grandes lignes ce qui se passe aide, mais la



Combien coûte Claude Code pour une TPE ou PME

Claude Code est inclus dans les abonnements Claude payants, sans facturation séparée : environ 20 euros par mois pour l'abonnement Pro, environ 200 euros pour le plus gros abonnement Max (tarifs vérifiés le 30/08/2026). Il consomme le quota d'usage de ton abonnement comme une conversation normale, et une session de construction consomme davantage qu'une question posée dans le chat.

Quelle est la différence entre Claude et Claude Code

Claude est un chat : tu poses une question, tu reçois une réponse, tu appliques toi-même. Claude Code est un exécutant : il travaille dans tes fichiers, lance des commandes, vérifie son propre résultat et itère jusqu'à ce que ça tourne. Le premier t'aide à réfléchir et à produire du texte, le second construit.

Claude Code peut-il travailler sur les données de mon entreprise

Oui, et c'est même son intérêt, avec deux précautions. D'abord, il demande la permission avant les actions sensibles et ne voit que le dossier dans lequel tu le lances : ne lui donne accès qu'au dossier du projet. Ensuite, ce qu'il lit est envoyé au modèle pour être traité, comme avec tout outil d'IA en ligne : commence avec des données peu sensibles, et applique à Claude Code les mêmes règles qu'au reste de tes outils pour les données clients.



Que devient le prototype après les sept jours



Trois sorties possibles, et les trois sont des succès. L'outil te suffit tel

quel et entre dans le quotidien de l'équipe. Il mérite une V2 que tu continues à faire évoluer toi-même, au même rythme d'une itération à la fois. Ou il a validé un besoin assez sérieux pour être développé proprement, et le prototype devient alors le meilleur cahier des charges possible.

