Premier projet avec Claude Code, *sans être développeur*

Le guide de démarrage en 7 étapes

Une page web personnelle, construite avec Claude Code et mise en ligne gratuitement.

1 Cadrer — 2 Demander un plan — 3 Construire — 4 Tester — 5 Corriger — 6 Sauvegarder — 7 Mettre en ligne

Formez-vous à l'IA, par Kemy

Version du 5 octobre 2026. Prix, étapes et réglages relevés les 3 et 5 octobre 2026. Sources : les pages officielles d'Anthropic, de Microsoft (VS Code), de GitHub, de Git et de la CNIL. La liste complète des sources est à la fin du guide. Ces outils changent souvent : si un bouton ou une commande a changé, la page officielle citée fait foi.

LE MOT DE KEMY

Avant de *commencer*

Avant de créer mon agence, j'étais infirmier. Aujourd'hui je dirige OTTOM4T3, une agence d'automatisation que j'ai créée le 4 juin 2026, et je construis un logiciel. Je ne suis toujours pas développeur. Je construis avec Claude Code, et c'est ce que je veux t'apprendre.

[à compléter par Kemy : une anecdote vraie]

Ce guide ne te promet ni un nouveau métier ni des revenus. Il te fait faire une chose précise : construire une page web personnelle simple et la mettre en ligne, en sept étapes. En chemin, tu apprends à parler à Claude Code et à vérifier ce qu'il fait. Tu apprends aussi à sauvegarder ton travail et à éviter les erreurs qui coûtent cher.

Je l'ai écrit pour quelqu'un qui n'a jamais codé et qui a peur de casser quelque chose. Si un mot te bloque, il est probablement dans l'antisèche des pages 17 et 18.

Kemy

MODE D'EMPLOI

Comment utiliser ce guide

Lis-le une fois en entier, puis garde-le à côté de ton clavier. Les passages dans des cadres gris sont des commandes ou des demandes à copier telles quelles. Les demandes à Claude Code sont écrites en français : il comprend le français, tu n'as pas besoin de passer par l'anglais. Les encadrés "Si besoin" donnent des détails : tu peux les sauter. À la fin de chaque étape, un message "Si tu es bloqué" t'aide à demander de l'aide à Claude.

Prévois deux à trois séances. La première pour installer et te connecter, la deuxième pour construire la page, la troisième pour la mettre en ligne. Rien ne t'oblige à aller vite.

Sommaire

1	Le mot de Kemy	page 2
2	Claude Code, ce qu'il est et ce qu'il n'est pas	page 4
3	Ce qu'il te faut pour commencer, prix compris	page 5
4	Installer VS Code et Claude Code sur Windows	page 6
5	Installer VS Code et Claude Code sur macOS	page 7
6	Premier lancement et réglage de sécurité	page 8
7	Le projet et les 7 étapes	pages 9 à 16
8	Antisèche : 30 mots et les raccourcis utiles	pages 17 et 18
9	Sécurité et bonnes pratiques	pages 19 à 21
10	Ce que ce guide ne fera pas pour toi	page 22
11	Et après	page 23
12	Sources et vérifications	pages 24 et 25

COMPRENDRE L'OUTIL

Claude Code, ce qu'il est et ce qu'il n'est pas

Ce qu'il est

Claude Code est un assistant d'Anthropic qui travaille directement dans un dossier de ton ordinateur. Tu lui écris en français ce que tu veux obtenir. Il lit les fichiers du dossier, en crée et les modifie. Il lance aussi des commandes pour avancer, puis il te montre ce qu'il a fait.

La différence avec Claude dans le navigateur (claude.ai) est simple. Dans le navigateur, Claude te répond et tu recopies. Avec Claude Code, il agit lui-même sur tes fichiers. La documentation officielle le décrit ainsi : il "peut lire tes fichiers, lancer des commandes, faire des modifications et avancer seul sur un problème pendant que tu regardes, que tu le réorientes ou que tu t'absentes" (traduction de la page Best practices).

Ce guide utilise Claude Code dans VS Code, l'éditeur de code gratuit de Microsoft. Claude Code s'y ajoute comme une extension, c'est-à-dire un module en plus. Tu lui écris dans un panneau. En mode manuel, tu vois chaque modification à côté de l'original, avant de l'accepter ou de la refuser. Claude Code existe aussi dans le terminal (la fenêtre où l'on tape des commandes), en application de bureau et dans le navigateur. Ce guide ne les couvre pas.

Ce qu'il n'est pas

Il ne fait pas un site en un clic. Il construit ce que tu demandes, et quand la demande est floue, le résultat l'est aussi.

Il se trompe. Les études résumées page 20 montrent que du code qui fonctionne n'est pas forcément du code sûr. Elles montrent aussi qu'un débutant en difficulté peut finir un projet avec une "illusion de compétence".

Il ne prend pas la responsabilité à ta place. La page sécurité de la documentation est nette : "Tu es responsable de relire le code et les commandes proposés avant de les approuver" (traduction).

Il n'est pas gratuit. L'offre gratuite de claude.ai ne donne pas accès à Claude Code (page suivante).

PRÉPARER

Ce qu'il te faut pour commencer

1. Un abonnement Claude payant, ou un compte développeur

La documentation d'installation dit : "Claude Code nécessite un compte Pro, Max, Team, Enterprise ou Console. L'offre gratuite de claude.ai n'inclut pas l'accès à Claude Code" (traduction, page Advanced setup, relevée le 05/10/2026). La page consacrée à VS Code confirme : tout abonnement Claude payant convient, ou un compte Console (pour développeurs). Pas besoin de clé API, le mot de passe des programmes : "aucune clé API n'est nécessaire".

Prix affichés sur claude.com/pricing, relevés les 03/10 et 05/10/2026 (identiques), en dollars américains et hors taxes :

Formule	Prix affiché	Claude Code inclus
Free	0 $	Non
Pro	17 $ par mois avec l'abonnement annuel (200 $ payés d'avance), 20 $ si tu paies au mois	Oui
Max (5x et 20x)	"From $100" par mois, soit à partir de 100 $	Oui
Team, siège Standard	20 $ par siège et par mois en annuel, 25 $ au mois	Oui
Team, siège Premium	100 $ par siège et par mois en annuel, 125 $ au mois	Oui
Enterprise	20 $ par siège et par mois, facturé à l'année, plus l'usage au tarif de l'API	Oui

La page précise trois choses. Les prix n'incluent pas les taxes. Ils peuvent changer à la discrétion d'Anthropic. Et des limites d'usage s'appliquent. La page ne chiffre pas ces limites, elle donne des rapports : Pro donne au moins 5 fois l'usage de Free par session de 5 heures, Max 5 ou 20 fois celui de Pro. Le prix exact de Max 20x n'y figure pas, seulement "à partir de 100 $". Le prix en euros affiché depuis la France n'a pas pu être relevé : vérifie-le au moment de payer.

La formule la moins chère qui inclut Claude Code est Pro. C'est celle que je te conseille pour ce premier projet.

L'autre voie est un compte Console : l'accès des développeurs (l'API), payé à la consommation avec des crédits achetés d'avance. La documentation donne un ordre de grandeur relevé en entreprise (page Manage costs). Environ 13 $ par développeur et par jour d'utilisation, et 150 à 250 $ par mois. Moins de 30 $ par jour pour 90 % des utilisateurs. Ces chiffres viennent de déploiements en entreprise, pas d'un premier projet, mais ils montrent qu'à la consommation la facture suit ton usage. Pour débuter, un abonnement à prix fixe est plus simple à maîtriser.

2. Un ordinateur compatible

Systèmes acceptés par Claude Code (relevé du 05/10/2026) :

- Windows 10 version 1809 ou plus récent, ou Windows Server 2019 ou plus récent ;
- macOS 13.0 ou plus récent ;
- Ubuntu 20.04, Debian 10, Alpine Linux 3.19 ou plus récents.

Il faut au moins 4 Go de mémoire vive et un processeur x64 ou ARM64. Il faut une connexion internet. Et il faut habiter dans un pays où Anthropic propose ses services (liste : anthropic.com/supported-countries). VS Code tourne sur Windows 10 et 11 et sur macOS 12.0 ou plus : ce sont donc les exigences de Claude Code qui comptent.

3. VS Code, gratuit

VS Code (Visual Studio Code) est l'éditeur de code de Microsoft, gratuit. C'est là que tu fais tout. Tu y ouvres ton dossier et tu écris à Claude. Tu y acceptes ou refuses ses modifications. Tu y sauvegardes et tu y publies. Pas besoin de connaître le terminal : tu n'en ouvres un que pour quelques lignes de réglage.

4. Pour les étapes 6 et 7

Git (logiciel de sauvegarde par versions, gratuit), un compte GitHub gratuit et un navigateur web. Git s'installe en premier, avec VS Code et Claude Code, aux pages suivantes.

INSTALLATION, WINDOWS

Installer VS Code et Claude Code sur Windows

Trois installations, dans cet ordre : Git, VS Code, puis l'extension Claude Code. Tu n'as pas besoin d'être administrateur. Étapes relevées le 05/10/2026 sur les pages officielles de VS Code (code.visualstudio.com) et de Claude Code (code.claude.com/docs/en/vs-code).

Étape A. Installer Git for Windows

Git sauvegarde ton travail (étape 6). VS Code en a besoin pour ses boutons de sauvegarde.

1. Va sur git-scm.com/install/windows et télécharge l'installateur.
2. Lance-le.
3. Clique sur Next à chaque écran, sans rien changer.

Étape B. Installer VS Code

1. Va sur code.visualstudio.com/download.
2. Sous Windows, clique sur User Installer, version x64. Prends Arm64 seulement si ton PC a un processeur ARM.
3. Lance le fichier téléchargé.
4. Garde les choix par défaut. La case "Ajouter à PATH" ("Add to PATH") doit rester cochée : elle l'est déjà. Elle aide Windows à trouver VS Code.
5. Termine l'installation, puis ouvre VS Code.

VS Code s'affiche en anglais. S'il te propose de passer en français, garde l'anglais : ce guide utilise les noms anglais des menus, ceux de la documentation.

Étape C. Installer l'extension Claude Code

1. Dans VS Code, appuie sur Ctrl + Maj + X. La colonne Extensions s'ouvre à gauche.
2. Tape `Claude Code` dans la recherche.
3. Clique sur "Claude Code for VS Code", publié par Anthropic, avec une coche bleue.
4. Clique sur Install.
5. Si VS Code te demande si tu fais confiance à l'éditeur, vérifie que c'est Anthropic, puis confirme.

Attention aux imitations. La bonne extension est publiée par Anthropic et porte la coche bleue avec le domaine anthropic.com. Selon la documentation de VS Code, cette coche prouve que l'éditeur possède ce nom de domaine.

C'est tout

L'extension contient déjà Claude Code : tu n'as aucune commande à taper. Elle se met à jour toute seule, comme les autres extensions de VS Code.

> **Si besoin : les options de l'installateur de VS Code**
>
> La documentation de VS Code conseille le User Installer "pour la plupart des gens". Il ne demande pas les droits d'administrateur. Une case facultative ajoute l'action "Ouvrir avec Code" au clic droit sur un dossier. Le numéro de version de l'extension est celui de Claude Code qu'elle contient.

INSTALLATION, MACOS

Installer VS Code et Claude Code sur macOS

Il faut macOS 13.0 ou plus récent pour Claude Code. Pour vérifier : menu Pomme, puis "À propos de ce Mac". Mêmes sources que la page 6.

Étape A. Installer VS Code

1. Va sur code.visualstudio.com/download et télécharge la version Mac (fichier `.dmg`).
2. Ouvre le fichier téléchargé.
3. Glisse `Visual Studio Code.app` dans le dossier Applications.
4. Ouvre VS Code depuis le dossier Applications.

Étape B. Installer l'extension Claude Code

Appuie sur Cmd + Maj + X, puis suis l'étape C de la page 6 : même extension, même éditeur, Anthropic.

Étape C. Installer Git

1. Dans VS Code, ouvre le menu Terminal, puis New Terminal. Une zone s'ouvre en bas : c'est le terminal.
2. Tape cette commande, puis appuie sur Entrée :

   >_ TERMINAL
   ```
   xcode-select --install
   ```

3. Une fenêtre de macOS te propose l'installation : accepte.

⚠ Si ça bloque (Windows et Mac)

Ce que tu vois	Quoi faire
L'extension ne s'installe pas	Mets VS Code à jour : il faut la version 1.94.0 ou plus (menu Help, puis About)
L'extension n'apparaît pas	Ferme et rouvre VS Code
Pas d'icône Claude en haut à droite	Normal sans fichier ouvert : clique sur "✻ Claude Code", en bas à droite de la fenêtre
Claude ne s'ouvre pas	Ton dossier est peut-être en mode restreint : voir page 8, point 2
"Not logged in · Please run /login"	L'écran de connexion revient tout seul. Sinon, ferme et rouvre VS Code
Claude ne répond jamais	Vérifie ta connexion internet, puis ouvre une nouvelle conversation
`git` introuvable dans le terminal	Ferme et rouvre VS Code après avoir installé Git

ⓘ Si besoin : Claude Code dans un terminal

Ce guide n'en a pas besoin. La documentation garde aussi une installation par commande, pour taper `claude` dans n'importe quel terminal. Sur Windows, dans PowerShell, puis sur Mac, dans le Terminal :

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

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

PREMIER LANCEMENT

Premier lancement

1. Ouvrir ton dossier

1. Dans tes Documents, crée un dossier nommé `ma-page`.
2. Dans VS Code, ouvre le menu File, puis Open Folder..., et choisis `ma-page`.

N'ouvre jamais tout ton dossier personnel. Claude doit voir seulement ton projet : la documentation de Claude Code conseille de travailler depuis le dossier du projet.

2. Faire confiance au dossier

Par défaut, VS Code ouvre un nouveau dossier en mode restreint (Restricted Mode), avec un bandeau en haut de la fenêtre. Claude ne fonctionne pas dans ce mode.

1. Dans le bandeau, clique sur Manage.
2. Clique sur Trust.

Si VS Code te pose plutôt la question dans une fenêtre, confirme. Fais confiance seulement à tes propres dossiers.

3. Ouvrir Claude et te connecter

1. Clique sur "✻ Claude Code", en bas à droite de la fenêtre.
2. Clique sur Sign in.
3. Ton navigateur s'ouvre : connecte-toi avec ton compte Claude.
4. Reviens dans VS Code. Une liste "Learn Claude Code" s'affiche : tu peux la fermer avec la croix.

4. Le réglage le plus important : le mode manuel

Par défaut, Claude Code démarre en mode Auto. Il modifie alors la plupart des fichiers sans te demander. Un second modèle d'IA, appelé classificateur, vérifie à ta place (depuis la version 2.1.283, documentation du 05/10/2026).

Pour apprendre, prends le mode manuel (Manual) : Claude te demande avant de modifier un fichier et avant la plupart des commandes. Tu lis tout avant d'accepter. Une étude de la page 20 va dans ce sens. Ceux qui s'en remettaient à l'IA pour écrire ou déboguer y obtenaient les plus mauvais scores. Ceux qui demandaient des explications obtenaient les meilleurs. L'étude ne peut pas prouver que c'est l'usage qui cause l'écart.

1. Sous la zone où tu écris à Claude, clique sur l'indicateur de mode.
2. Choisis Manual.

VS Code retient ce choix. Regarde quand même l'indicateur avant chaque demande.

5. Accepter ou refuser une modification

En mode Manual, VS Code te montre l'ancien fichier et le nouveau, côte à côte. Tu peux accepter, refuser, ou écrire à Claude ce qu'il doit faire à la place.

6. Les gestes utiles

Geste	Effet
Bouton Stop, ou Échap	Arrête Claude pendant qu'il travaille
Survoler un message, puis le bouton de retour arrière	Revient en arrière (page 14)
Palette de commandes (Ctrl + Maj + P), puis "Claude Code: Open in New Tab"	Ouvre une nouvelle conversation, vide

> ⓘ **Si besoin : imposer le mode, changer de modèle**
>
> Pour que chaque conversation démarre en Manual : Ctrl + , (Cmd + , sur Mac), cherche `initialPermissionMode` et choisis `manual`. Le modèle s'affiche sous la zone de saisie (Opus 5.5 par défaut sur Pro et Max). Un clic dessus permet d'en changer. Si une ligne affiche "Requires usage credits", ce modèle est facturé en plus de ton abonnement. Pour ce projet, garde celui par défaut.

LE PROJET

Le projet : ta page web personnelle

Tu vas construire une seule page web qui te présente. Ton nom, quelques lignes sur ce que tu fais, deux ou trois choses à montrer. Et, si tu veux, un moyen de te joindre. Elle sera faite de deux fichiers, `index.html` (le contenu) et `style.css` (l'apparence), et mise en ligne gratuitement avec GitHub Pages.

Le projet est volontairement petit. Pas de formulaire, pas de compte utilisateur, pas de base de données, pas de paiement. Cette limite a une raison. Une prépublication a audité 200 applications construites avec Claude Code ou Lovable. Les failles les plus fréquentes concernaient le contrôle d'accès, l'injection et l'authentification (article 03, source [13], détails page 20). Une page sans connexion ni formulaire n'a aucune de ces fonctions, donc aucune de ces failles à gérer. Pour un premier projet, c'est le bon terrain.

Les 7 étapes

Étape	Ce que tu fais	Ce que tu obtiens
1. Cadrer	Tu décides ce que la page contient	Un fichier `BRIEF.md`
2. Demander un plan	Claude propose, tu valides	Un plan écrit, sans aucun fichier créé
3. Construire par petits morceaux	Un morceau, une vérification, un morceau	Les fichiers de la page
4. Tester	Tu ouvres la page et tu cherches ce qui ne va pas	Une liste de problèmes
5. Corriger	Un problème à la fois, cause comprise	Une page qui marche
6. Sauvegarder	Git prend une photo de ton dossier	Un historique de versions
7. Mettre en ligne	Tu publies sur GitHub Pages	Une adresse publique

Chaque étape a le même plan : le but, une demande à copier, les erreurs de débutant et quoi faire. Les demandes sont des modèles : remplace ce qui est entre crochets.

ÉTAPE 1 SUR 7

Étape 1. *Cadrer*

Le but

Savoir ce que tu veux avant de le demander. Claude Code ne lit pas dans les pensées : une demande floue donne une page générique. Le cadrage se fait par écrit, dans un fichier `BRIEF.md` que Claude relira à chaque étape.

La demande à copier

1. Ouvre le panneau Claude dans ton dossier `ma-page`.
2. Vérifie que l'indicateur de mode affiche Manual.
3. Colle ce texte dans la zone de saisie, puis appuie sur Entrée :

DEMANDE À COPIER DANS CLAUDE CODE

Je veux créer une page web personnelle simple qui me présente. Je ne suis pas développeur.
Avant de créer quoi que ce soit, pose-moi des questions, une à la fois, pour comprendre :
à qui s'adresse la page, ce que je veux y dire sur moi, ce que je veux montrer, le ton,
les couleurs que j'aime ou que je refuse, et si je veux un moyen de contact.
Quand tu as assez d'éléments, écris un fichier BRIEF.md d'une page maximum qui résume mes réponses.
Contraintes à noter dans le brief : une seule page, HTML et CSS uniquement, pas de formulaire,
pas de compte, pas de base de données, pas de bibliothèque externe.
Ne crée aucun autre fichier.

Quand Claude te demande l'autorisation de créer `BRIEF.md`, lis le contenu proposé dans la vue de comparaison avant d'accepter. Si une phrase ne te ressemble pas, refuse et dis-lui pourquoi.

Les erreurs typiques

Vouloir tout faire d'un coup : un blog, un formulaire, une boutique. Garde ces idées dans une liste "plus tard" en bas du brief.

Copier le site de quelqu'un d'autre. Tu peux citer un site que tu aimes et dire ce qui te plaît (la sobriété, la typographie), sans demander de le reproduire.

Mettre des informations privées. Ta page sera publique et lisible par tout le monde, y compris par les robots qui récoltent les adresses. Pas d'adresse postale, pas de téléphone personnel, pas de date de naissance. Pour le contact, un lien vers un profil professionnel suffit souvent. Une adresse email, elle, sera visible de tous.

Répondre "je ne sais pas" à tout. Ce n'est pas grave : demande à Claude de te proposer trois options et choisis.

Si tu es bloqué

Copie ce message dans le panneau Claude, en remplaçant ce qui est entre crochets :

DEMANDE À COPIER DANS CLAUDE CODE

Je suis bloqué à l'étape 1 du guide (cadrer). Voici ce que je vois : [colle le message ou décris l'écran].
Explique-moi simplement quoi faire, une étape à la fois.

ÉTAPE 2 SUR 7

Étape 2. *Demander un plan*

Le but

Obtenir un plan court que tu comprends, avant que Claude écrive la moindre ligne. La documentation recommande de séparer le plan de l'écriture du code. But : "éviter de résoudre le mauvais problème" (traduction, page Best practices).

Passe en mode plan : clique sur l'indicateur de mode et choisis Plan (ou tape `/plan`). Dans ce mode, Claude lit et propose. Tes fichiers restent intacts tant que tu n'as pas approuvé le plan.

La demande à copier

DEMANDE À COPIER DANS CLAUDE CODE

```
Lis BRIEF.md. Propose un plan pour construire cette page.
Je veux : la liste des fichiers que tu vas créer avec une phrase sur le rôle de chacun,
l'ordre dans lequel tu vas les construire, en petits morceaux que je pourrai vérifier un par un,
et pour chaque morceau, comment je pourrai vérifier moi-même qu'il marche.
Explique chaque mot technique entre parenthèses.
Si quelque chose dans le brief est ambigu, pose-moi la question au lieu de deviner.
```

VS Code ouvre le plan comme un document.

1. Lis-le en entier.
2. Ajoute tes commentaires dans le document si quelque chose ne va pas.
3. Quand il te convient, approuve-le.

Si tu veux réfléchir avant, repasse en Manual avec l'indicateur de mode, sans approuver.

Les erreurs typiques

Approuver sans lire. Le plan est le moment où une erreur coûte le moins cher à corriger.

Accepter un plan trop gros. S'il annonce plus de trois ou quatre fichiers, des "frameworks", des "packages" ou un "build", demande pourquoi. Pour une page simple, deux fichiers suffisent.

Laisser passer un mot inconnu. Demande : "Que veut dire [mot] et pourquoi en as-tu besoin ?" Si la réponse est "ce n'est pas indispensable", retire-le du plan.

Si tu es bloqué

Copie ce message dans le panneau Claude, en remplaçant ce qui est entre crochets :

DEMANDE À COPIER DANS CLAUDE CODE

```
Je suis bloqué à l'étape 2 du guide (demander un plan). Voici ce que je vois : [colle le message ou décris l'écran].
Explique-moi simplement quoi faire, une étape à la fois.
```

ÉTAPE 3 SUR 7

Étape 3. *Construire par petits morceaux*

Le but

Construire la page morceau par morceau, en vérifiant chaque morceau avant le suivant. La documentation va dans le même sens. Elle conseille de découper une tâche complexe en étapes (page Quickstart). Et de corriger Claude dès qu'il s'écarte de ce que tu veux (page Best practices).

La demande à copier, morceau 1

Après l'approbation du plan, regarde l'indicateur de mode et remets Manual s'il affiche autre chose, puis :

DEMANDE À COPIER DANS CLAUDE CODE

Suis le plan. Commence seulement par le premier morceau : le fichier index.html avec la structure de la page et mes textes du brief, sans aucune mise en forme.
Ensuite, arrête-toi et explique-moi en cinq lignes maximum ce que contient chaque partie du fichier.
Ne passe pas au morceau suivant tant que je ne te l'ai pas demandé.

Ouvre `index.html` en double-cliquant dessus dans ton dossier, depuis l'Explorateur de fichiers ou le Finder : il s'ouvre dans ton navigateur. La page est sans mise en forme, c'est normal, la mise en forme vient ensuite.

L'exercice qui fait la différence

1. Dans VS Code, clique sur `index.html`, dans la colonne de gauche.
2. Change toi-même une phrase de ta présentation.
3. Enregistre avec Ctrl + S (Cmd + S sur Mac).
4. Recharge la page dans le navigateur.
5. Puis demande à Claude :

DEMANDE À COPIER DANS CLAUDE CODE

J'ai modifié moi-même une phrase dans index.html. Relis le fichier, dis-moi ce qui a changé et vérifie que je n'ai rien cassé.

Pourquoi ce détour ? Une étude a suivi 69 élèves débutants de 10 à 17 ans. Chaque exercice d'écriture de code, avec ou sans générateur de code selon le groupe, était suivi d'une modification faite à la main. Le groupe équipé n'a pas fait moins bien sur ces modifications (article 03, source [1]). Une autre étude va dans le même sens : voir page 20.

Les morceaux suivants

DEMANDE À COPIER DANS CLAUDE CODE

Morceau suivant du plan : [nom du morceau, par exemple "la mise en forme dans style.css"].
Fais seulement ce morceau, puis explique-moi en cinq lignes ce que tu as fait et pourquoi.
N'ajoute rien qui n'est pas dans le plan. Si tu penses qu'il faut autre chose, demande-le-moi d'abord.

Les erreurs typiques

Demander "fais tout le site". Tu obtiens beaucoup de code d'un coup, et tu ne sais plus ce qui marche.

Accepter des ajouts non demandés : une police chargée depuis un autre site, un script, une "bibliothèque". Chacun est une dépendance de plus (voir l'antisèche). Demande à quoi il sert.

Corriger encore et encore la même chose. Après deux corrections ratées sur le même point, la documentation conseille de repartir d'une conversation vide (page 8). Avec une demande plus précise, qui reprend ce que tu as appris.

Si tu es bloqué

Copie ce message dans le panneau Claude, en remplaçant ce qui est entre crochets :

DEMANDE À COPIER DANS CLAUDE CODE

Je suis bloqué à l'étape 3 du guide (construire). Voici ce que je vois : [colle le message ou décris l'écran].
Explique-moi simplement quoi faire, une étape à la fois.

ÉTAPE 4 SUR 7

Étape 4. *Tester*

Le but

Vérifier toi-même que la page fait ce que tu veux, au lieu de croire Claude quand il dit que c'est fini. La documentation le dit : Claude s'arrête quand le travail "a l'air fini". Sans vérification possible, c'est toi qui deviens la boucle de vérification (traduction, page Best practices).

Ta liste de vérification

Ouvre `index.html` dans ton navigateur et coche :

- [ ] Tous les textes sont les tiens, sans faute, sans texte de remplissage du genre "Lorem ipsum".
- [ ] Chaque lien s'ouvre et mène au bon endroit.
- [ ] En rétrécissant la fenêtre du navigateur jusqu'à la largeur d'un téléphone, rien ne dépasse et tout reste lisible.
- [ ] Le texte se lit bien sur le fond (pas de gris clair sur blanc).
- [ ] Les images s'affichent et ne sont pas énormes.
- [ ] La page s'affiche aussi dans un deuxième navigateur.

La demande à copier

DEMANDE À COPIER DANS CLAUDE CODE

Vérifie la page comme le ferait un relecteur exigeant. Pour chaque point, dis-moi comment tu l'as vérifié et ce que tu as trouvé, sans rien corriger pour l'instant :
1. tous les liens sont valides et pointent vers la bonne adresse ;
2. la page reste lisible sur un écran de téléphone ;
3. le contraste entre le texte et le fond est suffisant ;
4. il ne reste aucun texte de remplissage ni commentaire interne ;
5. il n'y a aucune information privée qui ne devrait pas être publique.
Donne-moi la liste des problèmes, du plus grave au moins grave.

La documentation recommande de demander des preuves (la commande lancée et son résultat, une capture) plutôt qu'une simple affirmation de réussite.

Les erreurs typiques

Croire "c'est terminé et ça fonctionne". Une étude qualitative a suivi 21 étudiants débutants qui programmaient avec une IA. Ceux qui étaient en difficulté pensaient souvent avoir mieux réussi qu'en réalité. Ils finissaient avec ce que les auteurs appellent une "illusion de compétence" (article 03, source [3]).

Tester seulement sur ton écran. Ta page sera souvent vue sur téléphone.

Corriger pendant le test. Note d'abord tous les problèmes, corrige ensuite, un par un.

Si tu es bloqué

Copie ce message dans le panneau Claude, en remplaçant ce qui est entre crochets :

DEMANDE À COPIER DANS CLAUDE CODE

Je suis bloqué à l'étape 4 du guide (tester). Voici ce que je vois : [colle le message ou décris l'écran]. Explique-moi simplement quoi faire, une étape à la fois.

ÉTAPE 5 SUR 7

Étape 5. *Corriger*

Le but

Corriger un problème à la fois, en comprenant sa cause, sans casser le reste.

La demande à copier

Décris le symptôme, ce que tu attendais, et où ça se passe. La documentation donne ce schéma : le symptôme, l'endroit probable, et ce qui compte comme "corrigé".

DEMANDE À COPIER DANS CLAUDE CODE

Problème : [ce que tu vois, par exemple "sur une fenêtre étroite, mon titre dépasse à droite de l'écran"].
Ce que j'attendais : [par exemple "le titre passe à la ligne et reste entièrement visible"].
Trouve la cause, explique-la-moi en une phrase, puis corrige seulement ce problème sans toucher au reste.
Corrige la cause elle-même au lieu de masquer le symptôme.
Dis-moi ensuite comment je peux vérifier moi-même que c'est réglé.

Puis refais le test concerné de l'étape 4.

Si Claude part dans la mauvaise direction

1. Clique sur Stop, ou appuie sur Échap : Claude s'arrête.
2. Survole le message d'où tu veux repartir, puis clique sur le bouton de retour arrière.
3. Choisis Rewind code to here : tes fichiers reviennent à ce moment, la conversation reste.

Ce retour arrière ne remplace pas Git. D'où l'étape suivante.

Les erreurs typiques

Écrire "ça marche pas". Claude va deviner, et souvent mal. Dis ce que tu vois, où, et ce que tu attendais.

Empiler les corrections dans une conversation qui s'allonge. Au bout de deux échecs, une conversation vide et une demande plus précise donnent presque toujours un meilleur résultat (documentation, Best practices).

Accepter une correction qui touche dix endroits pour un problème d'une ligne. Demande pourquoi, et refuse si la réponse ne te convainc pas.

Laisser Claude faire disparaître un message d'erreur sans en comprendre la cause. L'erreur est toujours là, tu ne la vois plus.

Si besoin : les autres choix du retour arrière

Fork conversation from here repart de ce message dans une nouvelle branche de conversation, sans toucher aux fichiers. Fork conversation and rewind code fait les deux. La documentation précise une limite. Ces points de restauration suivent seulement les modifications faites par les outils d'édition de fichiers de Claude. Pas celles faites par des commandes.

Si tu es bloqué

Copie ce message dans le panneau Claude, en remplaçant ce qui est entre crochets :

DEMANDE À COPIER DANS CLAUDE CODE

Je suis bloqué à l'étape 5 du guide (corriger). Voici ce que je vois : [colle le message ou décris l'écran].
Explique-moi simplement quoi faire, une étape à la fois.

ÉTAPE 6 SUR 7

Étape 6. *Sauvegarder avec Git*

Le but

Pouvoir revenir à une version qui marchait, quoi qu'il arrive. Git est un logiciel gratuit qui prend des "photos" de ton dossier. Chaque photo s'appelle un commit et porte un petit message qui dit ce qui a changé. Le dossier qui contient cet historique s'appelle un dépôt.

Git tourne sur ton ordinateur. GitHub est un site qui peut héberger ton dépôt en ligne (étape 7).

Une fois pour toutes : te présenter à Git

Git écrit ton nom et ton adresse email dans chaque commit.

1. Dans VS Code, ouvre le menu Terminal, puis New Terminal.
2. Tape `git --version`, puis Entrée. Un numéro de version doit s'afficher.
3. Tape ces deux lignes, une par une, avec ton nom et ton adresse, puis Entrée après chacune :

TERMINAL

```
git config --global user.name "Ton Nom"
git config --global user.email ton.adresse@example.com
```

Ce nom et cette adresse seront publics à l'étape 7. Pour cacher ton adresse, lis l'encadré "Si besoin" en bas de page avant ton premier commit.

La demande à copier

Claude lance les commandes. Lis chacune avant d'accepter.

DEMANDE À COPIER DANS CLAUDE CODE

Initialise un dépôt Git dans ce dossier. Crée un fichier .gitignore adapté à une page web simple.
Puis fais un premier commit avec un message clair en français.
Avant chaque commande, dis-moi en une phrase ce qu'elle fait.

Le fichier `.gitignore` liste ce que Git doit ignorer. Il évite qu'un fichier secret parte un jour dans l'historique.

Le réflexe à prendre

Un commit chaque fois qu'un morceau marche. Il suffit de demander, comme le propose la documentation de Claude Code : "Fais un commit de mes modifications avec un message descriptif."

Les erreurs typiques

Ne jamais faire de commit. Le jour où tout casse, il n'y a rien à restaurer.

Mettre dans le dépôt un fichier qui contient un mot de passe ou une clé. Voir page 19 : une fois publié, ce secret doit être révoqué ou changé.

Des messages comme "modif" ou "test". Dans trois semaines, tu ne sauras plus ce qu'ils contiennent.

Si besoin : sans Claude, et adresse privée

Sans Claude, les boutons de VS Code font la même chose :

1. Ouvre la vue Source Control (Ctrl + Maj + G).
2. Clique sur Initialize Repository, la première fois seulement.
3. Clique sur + à côté de chaque fichier, sous Changes.
4. Écris un message en haut, puis clique sur Commit.

Pour garder ton adresse privée, crée d'abord ton compte GitHub (étape 7, point 1). GitHub te fournit une adresse "noreply" (Settings, puis Emails). Mets-la dans `git config` à la place de la tienne. Fais-le avant ton premier commit : les commits déjà faits gardent l'ancienne adresse.

Si tu es bloqué

Copie ce message dans le panneau Claude, en remplaçant ce qui est entre crochets :

DEMANDE À COPIER DANS CLAUDE CODE

Je suis bloqué à l'étape 6 du guide (sauvegarder avec Git). Voici ce que je vois : [colle le message ou décris l'écran].
Explique-moi simplement quoi faire, une étape à la fois.

ÉTAPE 7 SUR 7

Étape 7. *Mettre en ligne avec GitHub Pages*

Le but

Donner une adresse publique à ta page, gratuitement. GitHub Pages est inclus dans l'offre gratuite de GitHub pour les dépôts publics. Public veut dire que tout le monde peut lire tous les fichiers du dépôt.

Avant de publier : la relecture de sécurité

DEMANDE À COPIER DANS CLAUDE CODE

Je vais publier ce dossier dans un dépôt public. Relis tous les fichiers, y compris les fichiers cachés et l'historique Git, et liste tout ce qui ne devrait pas être public : données personnelles, adresses, numéros, clés, mots de passe, commentaires internes, fichiers inutiles.
Ne modifie rien, donne-moi seulement la liste.

Publier depuis VS Code, pas à pas

Étapes tirées des documentations de VS Code et de GitHub, relevées le 05/10/2026.

1. Crée un compte gratuit sur github.com. Ton nom d'utilisateur fera partie de l'adresse de ta page.
2. Vérifie ton adresse email GitHub : sans ça, GitHub ne te laisse pas créer de dépôt.
3. Dans VS Code, appuie sur Ctrl + Maj + P (Cmd + Maj + P sur Mac) et lance Publish to GitHub.
4. Connecte-toi à GitHub quand VS Code te le demande, puis reviens dans VS Code.
5. Nomme le dépôt exactement `tonpseudo.github.io`, avec ton nom d'utilisateur en minuscules.
6. Choisis un dépôt public. VS Code le crée et y envoie tes commits.
7. Sur github.com, ouvre ton dépôt, puis Settings, puis Pages.
8. Sous "Build and deployment", choisis Deploy from a branch, ta branche, le dossier racine `/`, puis Save.
9. Attends jusqu'à 10 minutes, puis ouvre `https://tonpseudo.github.io`.

Ta page doit s'appeler exactement `index.html` : c'est un des noms que GitHub Pages cherche pour la page d'entrée.

Pour mettre à jour plus tard, fais un commit (étape 6). Puis, dans la vue Source Control : More Actions (...), puis Push. Si l'envoi échoue, ne force rien : demande à Claude de t'expliquer le message d'erreur.

Les erreurs typiques

Un dépôt mal nommé : il doit être exactement ton nom d'utilisateur suivi de `.github.io`.

Un fichier nommé `Index.html` ou `accueil.html` au lieu de `index.html`.

Conclure trop vite que rien ne marche. Attends dix minutes.

Publier sans la relecture de sécurité ci-dessus.

Si besoin : limites et cas rares

GitHub Pages accepte un site de 1 Go maximum, avec une bande passante indicative de 100 Go par mois. Pas d'activité commerciale en ligne, pas de boutique. Pas de transactions sensibles, comme des mots de passe ou des cartes bancaires. Une page de présentation personnelle entre dans l'usage prévu. Le site ne se publie pas ? Vérifie que le compte qui a envoyé les fichiers a une adresse email vérifiée (documentation GitHub). Pour une activité professionnelle, renseigne-toi sur les mentions légales obligatoires en France. Ce point n'est ni traité ni vérifié dans ce guide.

Si tu es bloqué

Copie ce message dans le panneau Claude, en remplaçant ce qui est entre crochets :

DEMANDE À COPIER DANS CLAUDE CODE

Je suis bloqué à l'étape 7 du guide (mettre en ligne). Voici ce que je vois : [colle le message ou décris l'écran].
Explique-moi simplement quoi faire, une étape à la fois.

ANTISÈCHE, 1/2

Antisèche : 30 mots en une ligne

À garder à côté du clavier.

1. **Terminal** : fenêtre où l'on tape des commandes au lieu de cliquer ; VS Code en intègre un (menu Terminal, puis New Terminal).
2. **Commande** : instruction tapée dans le terminal et lancée avec la touche Entrée.
3. **VS Code** : l'éditeur de code gratuit de Microsoft dans lequel ce guide fait tout travailler, Claude compris.
4. **Dossier de travail** : le dossier ouvert dans VS Code, où travaille Claude. En mode manuel, il te demande avant de lire ou d'écrire ailleurs avec ses outils de fichiers.
5. **Extension** : un module qu'on ajoute à VS Code depuis la vue Extensions ; Claude Code en est une, publiée par Anthropic.
6. **PATH** : la liste des dossiers où l'ordinateur cherche les programmes ; "command not found" ou "not recognized" signifie souvent qu'un programme n'y figure pas.
7. **HTML** : le langage qui décrit le contenu d'une page web, ses titres, paragraphes, liens et images.
8. **CSS** : le langage qui décrit l'apparence d'une page, ses couleurs, polices et la disposition des blocs.
9. **Navigateur** : le logiciel qui affiche les pages web, comme Chrome, Firefox, Safari ou Edge.
10. **Prompt** : la demande que tu écris à Claude ; plus elle est précise, moins il faudra corriger.
11. **Contexte** : tout ce que Claude garde en tête pendant une session (messages, fichiers lus, résultats). Plus il se remplit, plus Claude risque d'oublier des consignes et de se tromper.
12. **Mode de permission** : le réglage qui décide ce que Claude peut faire sans te demander (Manual, Edit automatically, Plan, Auto).
13. **Mode plan** : le mode où Claude explore et propose un plan sans modifier tes fichiers.
14. **Point de restauration** (checkpoint) : la trace des modifications de fichiers faites par Claude. Le bouton de retour arrière d'un message y ramène. Elle ne remplace pas Git.
15. **Bug** : un comportement que le programme ne devrait pas avoir.

ANTISÈCHE, 2/2

16 **Débogage** : la recherche de la cause d'un bug, puis sa correction.

17 **Git** : logiciel gratuit qui garde l'historique des versions d'un dossier.

18 **Dépôt** (repository) : un dossier suivi par Git avec son historique, ou sa copie en ligne sur GitHub.

19 **Commit** : une "photo" du dossier enregistrée par Git, avec un message qui dit ce qui a changé.

20 **.gitignore** : le fichier qui liste ce que Git doit ignorer, par exemple un fichier qui contient un secret.

21 **Branche** : une ligne de versions parallèle dans un dépôt, pour essayer quelque chose sans toucher à la version principale.

22 **GitHub** : le site qui héberge des dépôts Git en ligne et qui propose GitHub Pages.

23 **Push** : l'envoi des commits de ton ordinateur vers un dépôt en ligne.

24 **Déploiement** : la mise en ligne d'une version de ton site pour que le public puisse la voir.

25 **Hébergement** : le service qui garde ton site sur un serveur allumé en permanence ; ici, GitHub Pages.

26 **Nom de domaine** : l'adresse lisible d'un site, comme `monnom.fr` ; sans en acheter un, ta page a une adresse en `.github.io`.

27 **Dépendance** : du code écrit par d'autres dont ton projet a besoin ; chacune peut tomber en panne, changer ou contenir une faille.

28 **Bibliothèque ou framework** : un ensemble de code prêt à l'emploi pour construire plus vite ; inutile pour une page simple.

29 **Variable d'environnement** : un réglage nommé que l'ordinateur transmet aux programmes, souvent utilisé pour garder une clé hors du code (exemple : `ANTHROPIC_API_KEY`).

30 **Clé API** : un mot de passe qui permet à un programme d'utiliser un service en ton nom ; quiconque la possède peut s'en servir, et dépenser ton crédit.

Les raccourcis utiles (documentations Claude Code et VS Code, relevées le 05/10/2026)

Dans le panneau Claude :

Geste ou commande	Effet
Entrée, ou Maj + Entrée	Envoie ta demande, ou passe à la ligne sans l'envoyer
Indicateur de mode, sous la zone de saisie	Change de mode (Manual, Edit automatically, Plan, Auto)
`/plan`	Passe en mode plan
Bouton Stop, ou Échap	Arrête Claude pendant qu'il travaille
Bouton de retour arrière, au survol d'un message	Revient à un point précédent des fichiers, de la conversation, ou des deux
`/model`, ou clic sur le nom du modèle	Change de modèle
`/usage`	Montre ton compte et ta consommation par rapport aux limites de ton abonnement
`@index.html`	Désigne un fichier précis dans ta demande
Alt + K (Option + K sur Mac), depuis un fichier ouvert	Insère dans ta demande une référence au fichier et aux lignes sélectionnées
`/`	Ouvre le menu des commandes disponibles dans l'extension

Dans VS Code (sur Mac, Cmd au lieu de Ctrl, sauf mention contraire) :

Raccourci ou menu	Effet
Ctrl + Maj + X	Vue Extensions
Ctrl + Maj + P	Palette de commandes, pour lancer une commande par son nom
Ctrl + Maj + G (Ctrl aussi sur Mac)	Vue Source Control (Git)
Ctrl + S	Enregistre le fichier ouvert
Menu Terminal, puis New Terminal	Ouvre le terminal intégré dans ton dossier
Palette, puis "Claude Code: Open in New Tab"	Nouvelle conversation Claude dans un onglet

SÉCURITÉ, 1/3

Sécurité et bonnes pratiques

1. Une clé ou un mot de passe ne va ni dans une demande ni dans un dépôt public

Ce que tu écris à Claude Code quitte ton ordinateur. La documentation l'écrit : les données envoyées "comprennent toutes les demandes de l'utilisateur et les réponses du modèle", chiffrées pendant le transport (traduction, page Data usage, relevée le 03/10/2026). Avec un abonnement Free, Pro ou Max, Anthropic les conserve 30 jours si tu refuses qu'elles servent à améliorer ses modèles, et 5 ans si tu l'acceptes. Le réglage se trouve sur claude.ai/settings/data-privacy-controls. Sur ton ordinateur, Claude Code garde aussi les conversations en texte lisible dans le dossier `~/.claude/projects/`, 30 jours par défaut.

Une clé collée dans une demande se retrouve donc à plusieurs endroits que tu ne contrôles pas tous. Si tu dois un jour utiliser une clé, demande à Claude comment la placer dans une variable d'environnement ou dans un fichier `.env` listé dans `.gitignore`. Ton premier projet n'en a besoin d'aucune.

Si un secret est publié par erreur, la documentation de GitHub dit quoi faire en premier : le révoquer ou le changer auprès du service concerné. Elle précise que réécrire l'historique du dépôt ne suffit pas, parce que les données peuvent rester accessibles dans des copies du dépôt et dans des vues en cache sur GitHub.

DEMANDE À COPIER DANS CLAUDE CODE

Avant chaque commit, vérifie qu'aucun fichier du dossier ne contient de clé, de mot de passe, de token ou de donnée personnelle. Montre-moi ce que tu as vérifié.

2. Relire avant d'accepter

La documentation te confie cette tâche : "Tu es responsable de relire le code et les commandes proposés avant de les approuver" (traduction, page Security). Rappel de la page 8 : par défaut, depuis la version 2.1.283, c'est un classificateur qui approuve la plupart des actions à ta place. Le mode manuel te rend ce rôle pendant que tu apprends.

Les commandes à lire deux fois :

- celles qui suppriment : `rm`, `del`, `Remove-Item` ;
- celles qui téléchargent depuis internet : `curl`, `wget`, `irm` (la documentation indique que `curl` et `wget` ne sont pas approuvés automatiquement par défaut) ;
- celles qui installent un programme ou qui demandent les droits d'administrateur (`sudo` sur Mac et Linux) ;
- celles qui touchent un dossier en dehors de ton projet.

Si tu ne comprends pas une commande, ne l'accepte pas encore :

DEMANDE À COPIER DANS CLAUDE CODE

Avant que j'accepte : explique cette commande en français simple. Ce qu'elle fait, ce qu'elle modifie, et comment l'annuler si elle pose problème.

Les commandes trouvées sur internet suivent la même règle. Et ce que tu installes aussi : une extension a les mêmes droits que VS Code lui-même (documentation VS Code). Installe Claude Code seulement depuis l'éditeur Anthropic vérifié (page 6), et une commande d'installation seulement depuis la page officielle citée dans ce guide.

SÉCURITÉ, 2/3

3. Méfie-toi des contenus que tu fais lire à Claude

L'injection de prompt est une technique où un attaquant insère un texte malveillant pour détourner ou manipuler les instructions de l'assistant (définition de la documentation, page Security). Ce texte peut se cacher dans une page web ou un fichier que tu fais lire à Claude. Parmi les conseils officiels pour les contenus non fiables : relire les commandes proposées avant de les approuver, éviter d'envoyer directement à Claude un contenu dont tu ne connais pas l'origine, vérifier les modifications des fichiers importants, et signaler un comportement suspect avec `/feedback` (cette commande envoie à Anthropic une copie de la conversation, conservée 5 ans). La documentation ajoute qu'aucun système n'est totalement à l'abri.

4. Sauvegarder, et pas seulement sur ton ordinateur

Un commit Git après chaque morceau qui marche, et un commit avant de demander un gros changement. Les points de restauration de Claude Code (le bouton de retour arrière, page 14) aident, mais la documentation précise qu'ils ne remplacent pas Git. Une copie hors de ton ordinateur protège en plus contre la panne ou la perte de la machine : ton dépôt GitHub en est une pour les fichiers publics, un disque externe en est une autre.

5. Ce que disent les études

Les études ci-dessous sont résumées dans deux articles de Formez-vous à l'IA (références complètes page 25). Les numéros renvoient aux sources de chaque article.

Code qui marche et code sûr sont deux choses différentes. Dans une étude présentée en 2023, 47 participants, en majorité des étudiants, devaient écrire du code lié à la sécurité, avec ou sans assistant IA. Avec l'assistant, ils ont écrit un code significativement moins sûr : par exemple 36 % de solutions vulnérables à l'injection SQL avec l'assistant contre 7 % sans. Ils étaient aussi plus nombreux à croire leur code sûr (article 03, source [8]).

Les agents récents n'ont pas réglé la question. Sur un banc d'essai de 186 demandes de fonctionnalités tirées de projets réels, choisies là où des développeurs humains avaient introduit une faille, un agent utilisant Claude 4 Sonnet produisait 57 % de solutions qui fonctionnaient, mais seulement 11,8 % de solutions sûres (article 03, source [12], publié dans les actes de la conférence ICML 2026). Une prépublication de 2026 a audité 200 applications en ligne construites avec Claude Code ou Lovable : 91,0 % avaient au moins une faille, surtout dans le contrôle d'accès, l'injection et l'authentification (article 03, source [13], non relue par des pairs).

Déléguer peut faire moins apprendre. Dans une étude randomisée diffusée en 2026 par deux chercheurs liés à Anthropic, l'éditeur de Claude (prépublication), 52 développeurs professionnels ou indépendants, dont plus de la moitié codaient depuis au moins 7 ans, apprenaient une bibliothèque Python inconnue, avec ou sans assistant IA. Au quiz passé juste après, sans IA, le groupe avec assistant obtenait environ 50 % contre environ 65 % sans, soit 4,15 points d'écart sur 27, avec l'écart le plus grand sur le débogage. Parmi les participants équipés, les petits groupes (2 à 7 personnes) qui demandaient des explications ou posaient des questions de fond obtenaient de 65 à 86 %, ceux qui confiaient à l'IA l'écriture ou le débogage de 24 à 39 %. Chacun avait choisi sa façon de faire, donc l'étude ne peut pas dire si c'est l'usage qui cause l'écart. L'assistant testé reposait sur GPT-4o, un modèle d'OpenAI (articles 03 et 04, source [4]). Dans le plus grand essai randomisé examiné dans l'article 04, mené en mathématiques auprès de près d'un millier d'élèves d'un lycée de Turquie, une IA qui donnait les réponses a fait progresser les notes d'entraînement de 48 %, puis baisser de 17 % les résultats à l'examen passé sans IA ; une version qui guidait par des indices n'a pas produit cette baisse (article 04, source [1]).

La limite de ces travaux : aucune des études vérifiées ne porte sur des adultes non développeurs qui construisent des outils pour leur métier (article 03). Elles concernent des jeunes, des étudiants ou des développeurs. Elles donnent une direction, pas une mesure de ce qui t'arrivera.

SÉCURITÉ, 3/3

6. Données personnelles et RGPD

La CNIL définit une donnée personnelle comme "toute information se rapportant à une personne physique identifiée ou identifiable", et un traitement comme "une opération, ou ensemble d'opérations, portant sur des données personnelles, quel que soit le procédé utilisé" (cnil.fr). Le RGPD s'applique aux organismes, publics ou privés, établis dans l'Union européenne ou qui visent des résidents de l'Union.

Trois règles pratiques en découlent pour toi :

- Ne colle pas dans Claude Code les données de quelqu'un d'autre (fichier clients, emails, dossiers, listes de contacts) sans avoir vérifié que tu en as le droit. Pour construire et tester, utilise des données inventées.
- Ta page, telle que ce guide la construit, ne collecte rien : pas de formulaire, pas d'outil de statistiques, pas de cookies ajoutés. Le jour où tu ajoutes un formulaire ou un outil de mesure d'audience, tu commences à collecter des données personnelles : lis d'abord les fiches de la CNIL.
- Les données de santé obéissent à des règles encore plus strictes, qui ne sont pas traitées dans ce guide.

Côté Anthropic, la documentation distingue deux régimes : avec un abonnement Free, Pro ou Max, c'est toi qui choisis si tes données peuvent servir à entraîner les modèles ; avec les offres commerciales (Team, Enterprise, API), Anthropic indique ne pas entraîner ses modèles sur le code et les demandes envoyés à Claude Code, sauf accord du client.

7. Garder la main : cinq habitudes

1. Avant de demander, écris en une phrase ce que tu attends. Ça t'oblige à savoir ce que tu veux.
2. Demande des explications, pas seulement du code (article 03, source [4]).
3. Modifie toi-même une partie de chaque fichier, même une phrase (article 03, source [1]).
4. Quand tu veux comprendre, demande un indice plutôt que la réponse (article 04, source [1]) :

   DEMANDE À COPIER DANS CLAUDE CODE

   Ne me donne pas la solution. Donne-moi un indice pour que je trouve moi-même où est le problème, puis attends ma réponse.

5. Explique à voix haute ce que fait ton fichier. Si tu n'y arrives pas, demande à Claude de te l'expliquer, puis réessaie.

Ces habitudes sont plus lentes. Une étude de 2019, qui ne porte pas sur l'IA, a montré que des étudiants apprenaient davantage en cours actif tout en ayant le sentiment d'apprendre moins (article 04, source [14]) : la facilité ressentie n'est pas une bonne mesure de ce que l'on retient.

LIMITES

Ce que ce guide *ne fera pas* pour toi

Il ne fera pas de toi un développeur. Il te fait réussir un premier projet et prendre de bonnes habitudes.

Il ne te promet aucun revenu, aucun client, aucun métier.

Il ne couvre pas les projets qui demandent d'autres précautions : formulaires, comptes utilisateurs, paiements, bases de données, applications mobiles, automatisations qui touchent tes emails ou tes fichiers. Les failles les plus fréquentes relevées par la recherche se trouvent justement dans ces fonctions.

Il ne remplace pas un audit de sécurité. Les relectures proposées réduisent les erreurs grossières, sans garantie.

Il est daté. Les prix, les étapes et le mode de permission par défaut ont été relevés les 3 et 5 octobre 2026. L'extension Claude Code se met à jour automatiquement. Son comportement par défaut a déjà changé entre deux versions (le mode auto, page 8). En cas de différence, la documentation officielle fait foi.

Il ne couvre pas Linux (non testé), ni Claude Code dans le terminal (seulement signalé page 7). Ni l'application de bureau, l'achat d'un nom de domaine ou les mentions légales.

Il ne répond pas à une question qu'aucune des études vérifiées pour ce guide n'aborde : l'effet à long terme de ces outils sur des adultes non développeurs qui construisent pour leur métier. Les études disponibles portent sur d'autres publics.

Il ne remplace pas la pratique. Tu retiendras ce que tu auras refait, pas ce que tu auras lu.

ET APRÈS

Et *après*

Si ta page est en ligne, tu as fait le chemin complet : une idée, un plan, du code que tu as relu, des tests, une sauvegarde et une mise en ligne. Les projets suivants suivent les mêmes étapes, avec plus de précautions.

Continuer avec d'autres

La communauté Formez-vous à l'IA réunit des personnes qui construisent avec Claude Code sans être développeuses. On y continue là où ce guide s'arrête : des projets plus grands, et ce qui arrive après la mise en ligne, la maintenance, les pannes, la sécurité. Tu peux la rejoindre quand tu veux.

[à compléter par Kemy : le lien d'accès à la communauté, et la confirmation de ce qu'elle contient]

Si tu préfères qu'on le fasse pour toi

Construire soi-même n'est pas toujours le bon choix. Si tu as besoin d'une automatisation ou d'un outil pour ton activité et que tu préfères le confier, c'est le métier d'OTTOM4T3, mon agence. Écris-moi en décrivant ce que tu veux faire : kemy.guerouine@gmail.com

Kemy

SOURCES, 1/2

Sources et vérifications

Pages consultées le 3 octobre 2026, puis le 5 octobre 2026 pour les prix et pour tout ce qui touche à VS Code. Les commandes du guide ont été recopiées depuis ces pages ; seuls les exemples (nom de fichier, message de commit, nom et adresse) ont été remplacés.

Prix et formules

Information	Source
Prix des formules (Max "From $100"), Claude Code inclus ou non, taxes, limites d'usage	https://claude.com/pricing
Coût à la consommation	https://code.claude.com/docs/en/costs

Claude Code

Information	Source
Extension VS Code : prérequis, installation, copie intégrée de Claude Code, panneau, connexion, modes, vue de comparaison, plan, modèle, retour arrière, `/usage`, dépannage	https://code.claude.com/docs/en/vs-code
Configuration requise, comptes acceptés, Git for Windows facultatif, installation par le terminal	https://code.claude.com/docs/en/setup
Git for Windows pas à pas	https://code.claude.com/docs/en/terminal-guide
Mode Auto par défaut (v2.1.283), Manual, `initialPermissionMode`, mode plan	https://code.claude.com/docs/en/permission-modes
Modèle par défaut, crédits d'usage	https://code.claude.com/docs/en/model-config
Points de restauration et leurs limites	https://code.claude.com/docs/en/checkpointing
Vérification, plan, repartir d'une conversation vide ; découper en étapes	https://code.claude.com/docs/en/best-practices et /quickstart
Responsabilité, injection de prompt, `curl` et `wget`, dossier personnel	https://code.claude.com/docs/en/security
Données envoyées, entraînement, conservation, `/feedback`	https://code.claude.com/docs/en/data-usage

VS Code

Information	Source
Nom de l'extension, éditeur vérifié, version	https://marketplace.visualstudio.com/items?itemName=anthropic.claude-code
Téléchargement, systèmes pris en charge	https://code.visualstudio.com/download
Installation sur Windows (User setup, PATH) et sur macOS	https://code.visualstudio.com/docs/setup/windows et /mac
Cases de l'installateur Windows, en anglais et en français	https://github.com/microsoft/vscode/tree/main/build/win32
Installer une extension, mises à jour, droits, coche bleue	https://code.visualstudio.com/docs/configure/extensions/extension-marketplace et /extension-runtime-security
Mode restreint, confiance	https://code.visualstudio.com/docs/editing/workspaces/workspace-trust
Langue de l'interface	https://code.visualstudio.com/docs/configure/locales
Terminal intégré	https://code.visualstudio.com/docs/terminal/getting-started
Source Control, Initialize Repository, Commit, Push, `git config`	https://code.visualstudio.com/docs/sourcecontrol/quickstart
Publish to GitHub, dépôt public ou privé	https://code.visualstudio.com/docs/sourcecontrol/repos-remotes

SOURCES, 2/2

Git, GitHub, CNIL

Information	Source
Git sur macOS (`xcode-select --install`) et Git for Windows	https://git-scm.com/install/mac et /install/windows
`git config` , `init` , `status` , `add` , `commit -m` , `log --oneline` , `.gitignore`	https://git-scm.com/book/en/v2 (First-Time-Git-Setup, Git-Basics)
Créer le site, nom en minuscules, fichier d'entrée, 10 minutes	https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site
Source de publication, Deploy from a branch, email vérifié	https://docs.github.com/en/pages/getting-started-with-github-pages/configuring-a-publishing-source-for-your-github-pages-site
Adresse `https://tonpseudo.github.io` , limites et usages interdits	https://docs.github.com/en/pages/getting-started-with-github-pages (what-is-github-pages, github-pages-limits)
GitHub Pages gratuit pour les dépôts publics	https://docs.github.com/en/get-started/learning-about-github/githubs-plans
Adresse "noreply" pour les commits	https://docs.github.com/en/account-and-profile/how-tos/email-preferences/setting-your-commit-email-address
Secret publié : révoquer d'abord	https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/removing-sensitive-data-from-a-repository
Donnée personnelle, traitement, champ du RGPD	https://www.cnil.fr/fr/rgpd-de-quoi-parle-t-on

Études citées

Résumées dans les articles de Formez-vous à l'IA "Débutants et IA qui écrit du code : ce que montre la recherche" (article 03) et "Apprendre avec l'IA sans perdre la main : la question de la dépendance cognitive" (article 04), datés du 30/09/2026, vérifiés contre leurs sources primaires et corrigés le 03/10/2026.

Article 03, [1] : Kazemitabaar, M., et al. (2023). Studying the effect of AI code generators on supporting novice learners in introductory programming. CHI 2023. https://doi.org/10.1145/3544548.3580919

Article 03, [3] : Prather, J., et al. (2024). The widening gap: The benefits and harms of generative AI for novice programmers. ICER 2024. https://doi.org/10.1145/3632620.3671116

Article 03, [4] (article 04, [4]) : Shen, J. H., & Tamkin, A. (2026). How AI impacts skill formation (arXiv:2601.20245). Prépublication ; auteurs liés à Anthropic (programme Anthropic Fellows et Anthropic) ; assistant testé fondé sur GPT-4o. https://arxiv.org/abs/2601.20245

Article 03, [8] : Perry, N., Srivastava, M., Kumar, D., & Boneh, D. (2023). Do users write more insecure code with AI assistants? CCS 2023. https://doi.org/10.1145/3576915.3623157

Article 03, [12] : Zhao, S., et al. (2026). Is vibe coding safe? Benchmarking vulnerability of agent-generated code in real-world tasks. Proceedings of the 43rd International Conference on Machine Learning (PMLR 306, pp. 162269-162292). https://proceedings.mlr.press/v306/zhao26ax.html

Article 03, [13] : Deng, J., Fan, Z., & Meng, R. (2026). Understanding the (in)security of vibe-coded applications (arXiv:2606.23130). Prépublication. https://arxiv.org/abs/2606.23130

Article 04, [1] : Bastani, H., et al. (2025). Generative AI without guardrails can harm learning: Evidence from high school mathematics. PNAS, 122(26), e2422633122. https://doi.org/10.1073/pnas.2422633122

Article 04, [14] : Deslauriers, L., McCarty, L. S., Miller, K., Callaghan, K., & Kestin, G. (2019). Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. PNAS, 116(39), 19251-19257. https://doi.org/10.1073/pnas.1821936116

Ce qui n'a pas pu être vérifié

Le prix en euros et la TVA affichés depuis la France : la page tarifs relevée affiche des dollars hors taxes.

Le prix exact de Max 20x : la page n'affiche que "From $100" pour les deux paliers Max.

Le volume des limites d'usage : la page ne donne que des rapports entre formules (par exemple, au moins 5 fois l'usage de Free pour Pro).

Le parcours n'a pas été refait sur un ordinateur de test : il est recopié des documentations officielles. Certains écrans n'y sont pas décrits en détail (écran de connexion de l'extension, boutons d'approbation du plan dans VS Code) ; le guide en dit le moins possible.

Version minimale de VS Code : 1.94.0 selon la documentation d'Anthropic et le fichier technique de l'extension, 1.98.0 selon sa page sur la Marketplace. Avec la dernière version de VS Code, l'écart ne compte pas.

La documentation de l'extension donne Ctrl + Maj + Échap pour ouvrir une conversation dans un onglet ; sur Windows, ce raccourci ouvre le Gestionnaire des tâches (documentation Microsoft). Le guide passe par la palette de commandes.

Linux, Claude Code dans le terminal et l'application de bureau : non testés. Les mentions légales d'un site professionnel en France : non traitées.