

Marre d'atteindre ta limite Claude ?

7 réflexes pour en faire plus sans la toucher.

Pourquoi Claude consomme plus que prévu, et comment faire autant de travail en dépensant moins. Un guide court et concret pour les dirigeants, les indépendants et les équipes formées, vérifié sur la documentation officielle.

🕒 Claude relit tout, à chaque message



Ta consommation

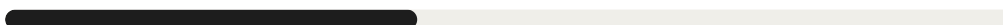
● Session en cours

Session de 5 heures

Limite bientôt atteinte


Semaine

se recharge chaque semaine



1 Le bon modèle

2 Doser la réflexion

3 Gérer les pauses

4 Compacter le fil

5 Viser juste

6 Un CLAUDE.md léger

7 Des outils à la demande

Piloter avec /usage

● Moins de tokens, autant de travail

Sommaire

Dix minutes de lecture. Sept réflexes pour tous les jours.

Les réflexes sont numérotés de 1 à 7, du plus simple au plus technique. Ils gardent les mêmes numéros dans tout le playbook, jusqu'à l'aide-mémoire de la page 11.

	Pourquoi ta limite arrive si vite · La relecture, les deux jauges, le prix de l'écriture	03
1	Le bon modèle · Haiku, Sonnet, Opus : chacun son travail	04
2	Doser la réflexion · La commande /effort	05
3	Gérer les pauses · La mémoire courte de Claude a une durée de vie	06
4	Compacter · La commande /compact, entre deux étapes	07
5	Viser juste · @fichier et le mode Plan	08
6 7	Alléger le fond · Un CLAUDE.md court, des outils à la demande	09
	Piloter avec /usage · Un contrôle par semaine, en trois étapes	10
	L'aide-mémoire · Une page à imprimer et à garder sous la main	11

Le mot à connaître : le token

Un token est un petit morceau de mot. C'est l'unité que Claude compte pour tout ce qu'il lit et tout ce qu'il écrit. Tes limites d'abonnement, comme les tarifs officiels, se calculent en tokens.

Où s'appliquent ces réflexes

Dans l'application Claude (web, bureau, mobile) et dans Claude Code, l'outil de Claude qui travaille sur tes fichiers. Chaque page l'indique en haut à droite :

Application Claude

Claude Code

Tarifs et fonctionnalités vérifiés sur la documentation officielle de l'éditeur en octobre 2026. Ils évoluent vite : en cas de doute, ta commande /usage fait foi.

 Pourquoi ta limite arrive si vite

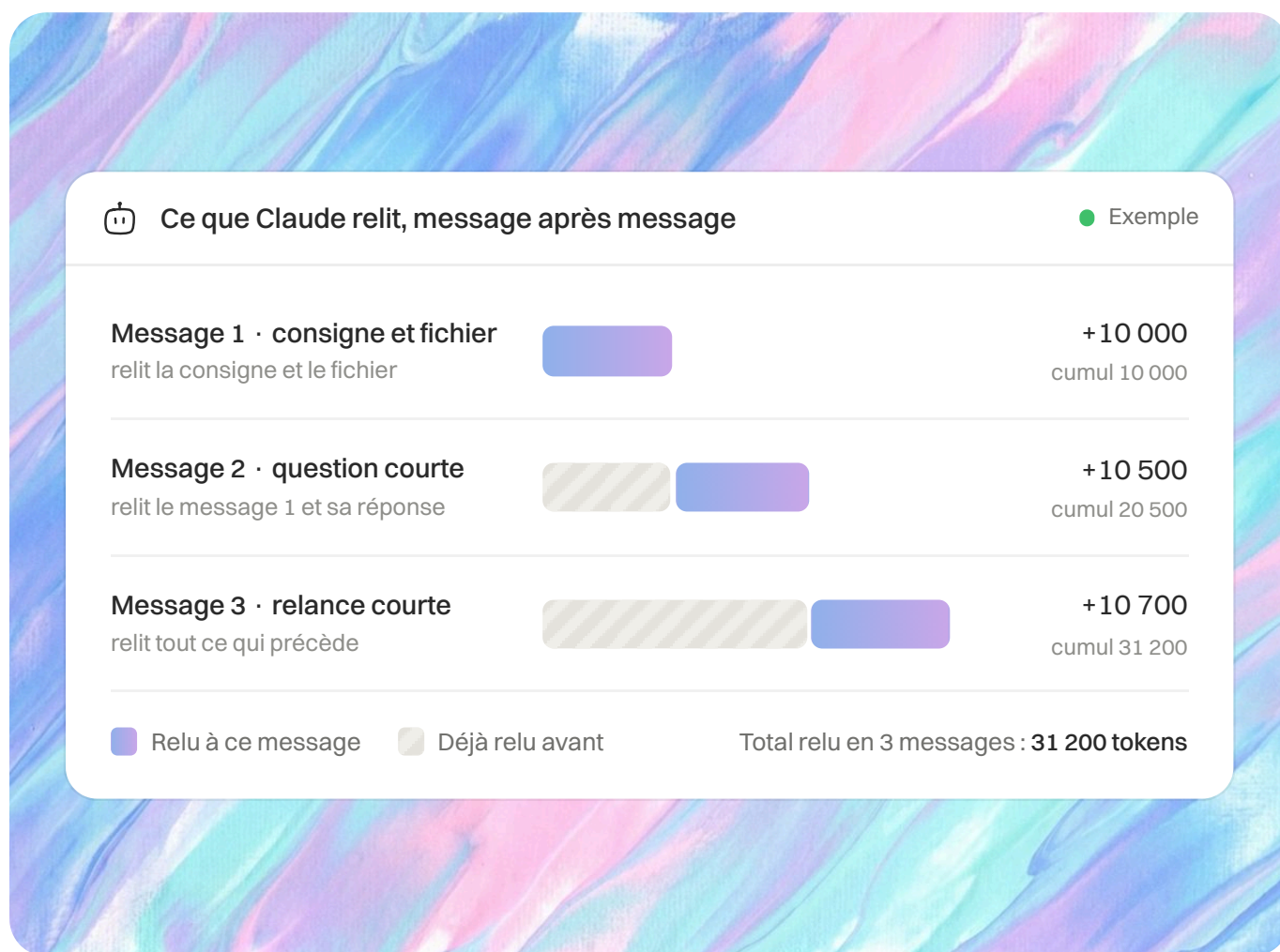
Application Claude

Claude Code

Claude n'a pas de mémoire.

Il relit toute la conversation à chaque message.

À chaque envoi, même pour une consigne d'une ligne, Claude relit tout le fil : tes messages, ses réponses et chaque fichier joint depuis le début. Tu ne paies donc pas seulement ta dernière question, tu paies la relecture de tout ce qui précède.



La relecture s'additionne. Plus le fil s'allonge, plus chaque nouveau message coûte cher : le total relu grimpe beaucoup plus vite que le nombre de messages (doubler la longueur d'un fil peut presque quadrupler ce qui est relu).

Deux jauges sur ton abonnement

 **Session** se recharge toutes les 5 heures

 **Semaine** se recharge une fois par semaine

À suivre dans Paramètres, puis Utilisation.

Écrire coûte 5 fois plus que lire

Lire  ×1

Écrire  ×5

Sur tous les modèles de la grille officielle.

1 Réflexe 1 · Le bon modèle

Application Claude

Claude Code

Le plus puissant n'est pas toujours le bon. Choisis le modèle selon la tâche.

Garder le modèle le plus haut de gamme pour tout, c'est payer le prix fort pour corriger une virgule. L'écart de prix entre les modèles est important : Opus coûte 2 fois plus que Sonnet, et Haiku est de loin le plus économe.

Quel modèle pour quel travail

Modèle	Pour quoi faire	Prix par million de tokens lus
Haiku 5.5 L'exécutant rapide	Trier des mails, renommer, résumer un texte court	0,10 \$ le plus économe
Sonnet 5.5 Par défaut Le modèle du quotidien	Posts, mails, propositions, comptes rendus, tableaux	2 \$ 20 fois Haiku
Opus 5.5 Le spécialiste	Raisonnement complexe, diagnostic, décision stratégique	4 \$ 2 fois Sonnet
Fable 5.1 Le très haut de gamme	Cas exceptionnels uniquement	10 \$ 2,5 fois Opus

Tarifs officiels de l'API, en lecture. L'écriture coûte 5 fois ces montants. Haiku 5.5 : 0,10 \$ jusqu'à 100 000 tokens par demande, 0,50 \$ au-delà. En orange, le modèle recommandé par défaut.

Le réflexe

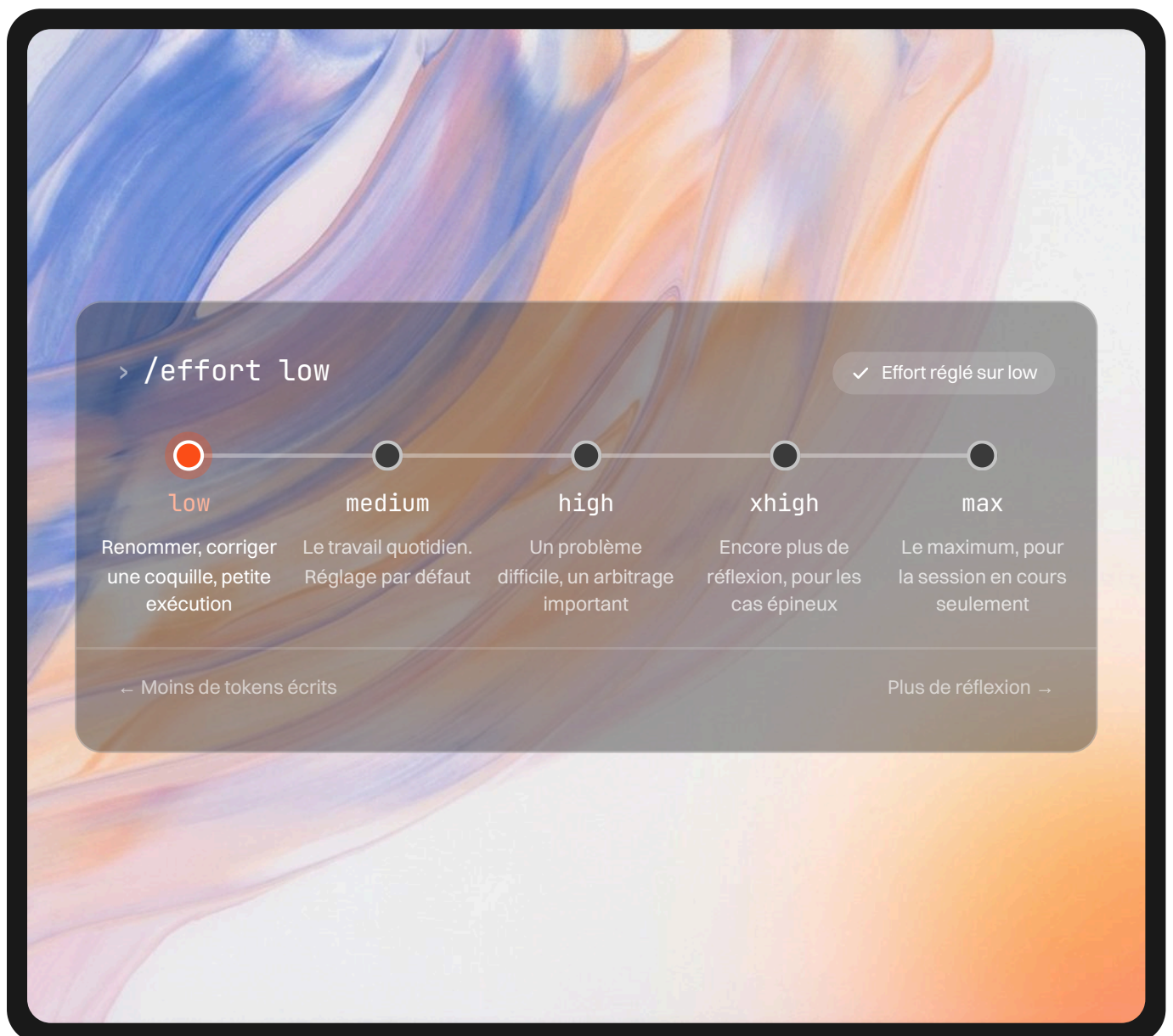
- ✓ **Choisis le modèle dès le premier message.** Chaque modèle a sa propre mémoire courte : changer en cours de fil fait relire tout l'historique, sans réduction, par le nouveau modèle.
- ✓ **Sonnet par défaut** pour l'essentiel du travail courant, **Opus** quand la tâche demande vraiment de la réflexion, **Haiku** pour les petites tâches répétitives.
- ✓ Vérifie le modèle actif : sélecteur de l'application, ou commande `/model` dans Claude Code.

2 Réflexe 2 · Doser la réflexion

Claude Code

Réfléchir longtemps a un prix. Dose l'effort selon la tâche.

Avant de répondre, Claude réfléchit. Cette réflexion est facturée comme de l'écriture, donc au tarif le plus cher, même quand tu ne la vois pas. Pour corriger une coquille, une longue réflexion ne sert à rien.



Le réflexe

- ✓ Tape `/effort low` avant une série de petites tâches, puis reviens à `/effort medium`, le réglage par défaut.
- ✓ Garde `high` et au-delà pour les vrais casse-têtes. La réflexion ne se coupe pas, elle se dose.
- ✓ Pour une seule question difficile, monte l'effort le temps de cette question, pas pour toute la session.

3 Réflexe 3 · Gérer les pauses

Application Claude

Claude Code

Une longue pause coûte cher.

La mémoire courte s'efface, tout est relu au prix fort.

Pour éviter de tout relire au prix normal, Claude garde le début de la conversation dans une mémoire courte, le cache. Relire depuis le cache coûte une fraction du prix. Mais ce cache expire : si tu reviens trop tard, tout le fil est relu et remis en cache, au prix fort.

🕒

Avant et après la pause

● Exemple

Message envoyé

conversation de 200 000 tokens, Opus 5.5

Retour avant expiration

Relu depuis le cache

0,04 \$

Retour après expiration

Tout est relu et remis en cache

1,60 \$

×40

Tarifs officiels de l'API, coût de la lecture seulement, en dollars. Cas de Claude Code sur abonnement (cache d'une heure). Avec une clé API ou des crédits, le cache dure 5 minutes et la reprise coûte 1,00 \$, soit 25 fois plus.

Claude Code + abonnement

1 heure

Claude Code + clé API

5 minutes

Application Claude

Non publiée

Le réflexe

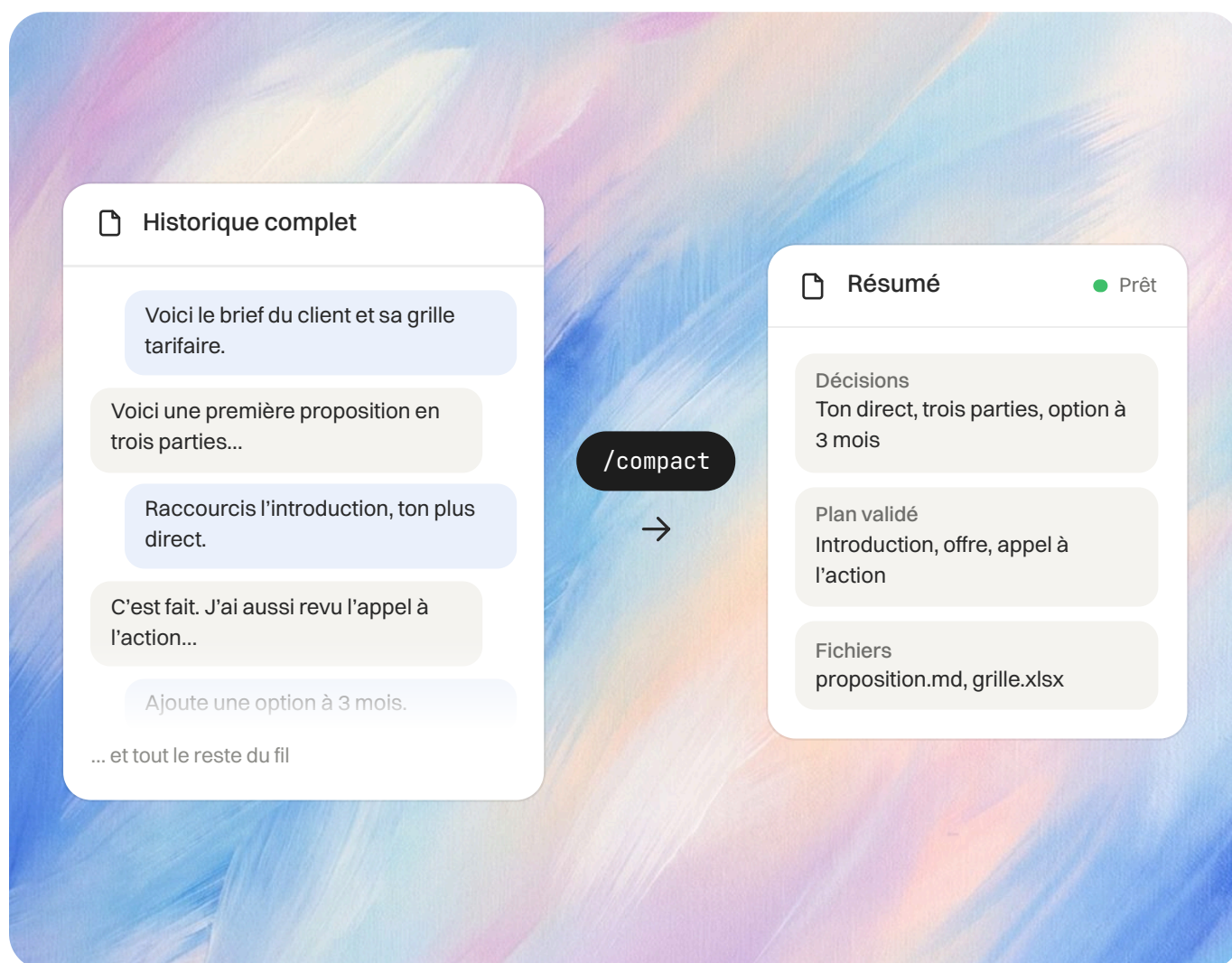
- ✓ Avant de partir, termine l'étape en cours ou compacte le fil (réflexe 4).
- ✓ Au retour d'une longue pause, sur un fil très long, repars d'une nouvelle conversation avec un résumé plutôt que de relancer l'ancienne.

4 Réflexe 4 · Compacter

Claude Code

Un long fil se résume. Compacte entre deux étapes, pas au milieu.

La commande `/compact` demande à Claude de résumer la conversation, puis de repartir de ce résumé. La suite du travail relit quelques lignes au lieu de tout l'historique. Le compactage relit une dernière fois le fil pour le résumer : il a donc un coût, à placer au bon moment.



Le réflexe

- ✓ **Entre deux grandes étapes**, jamais au milieu d'une tâche complexe.
- ✓ **Dis ce qu'il faut garder** : la commande accepte une consigne, par exemple `/compact` garde les décisions et la liste des fichiers modifiés.
- ✓ Pour un tout nouveau sujet, `/clear` repart d'une page blanche.

Une fois pour toutes, dans CLAUDE.md

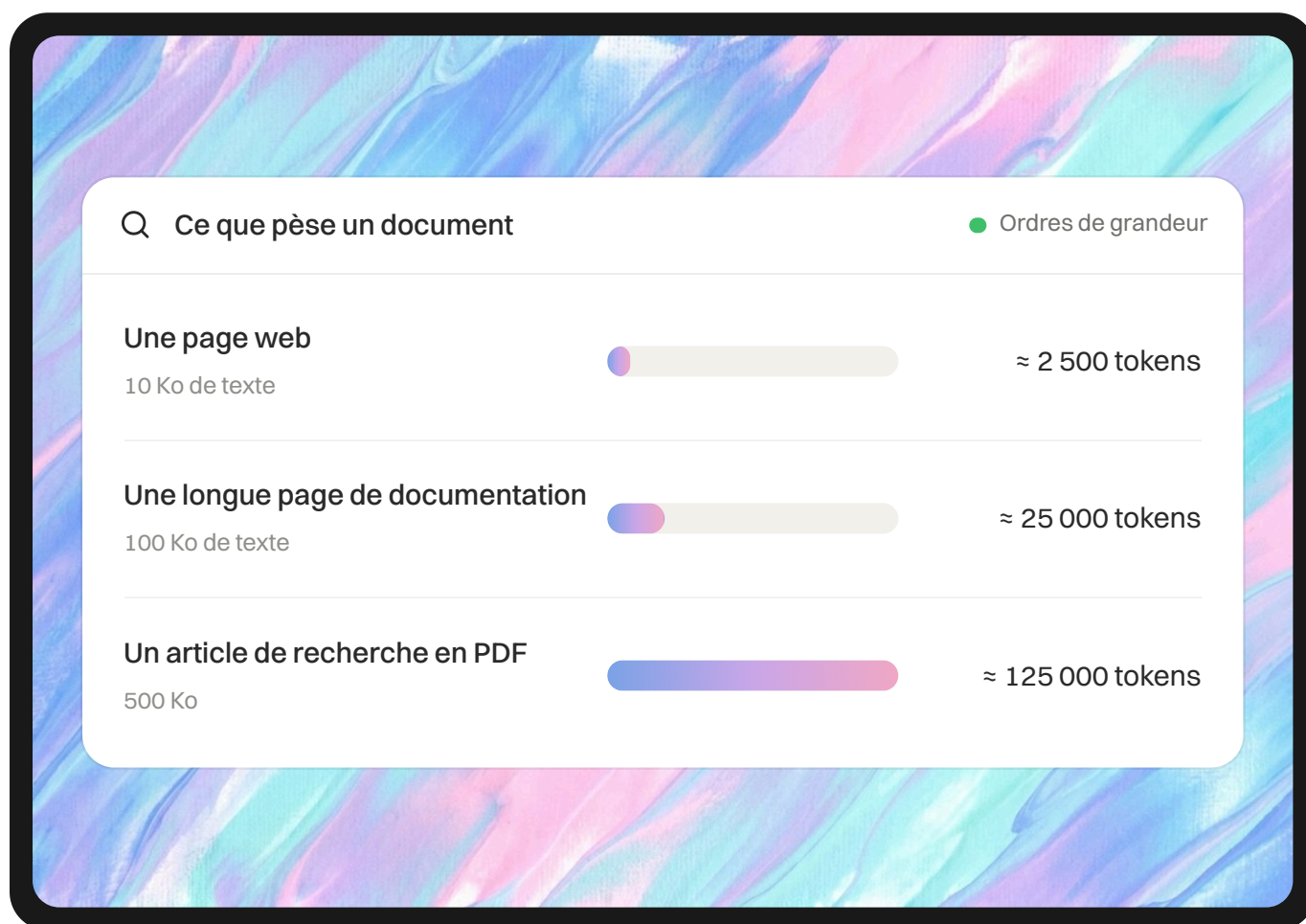
```

Claude.md

# Instructions de compactage
Lors d'un /compact, conserve uniquement
les décisions prises, la structure validée
et la liste des fichiers modifiés. Laisse
de côté les essais intermédiaires.
    
```

« Améliore mon projet » coûte cher. Une demande précise, un fichier ciblé.

Une consigne floue pousse Claude à ouvrir beaucoup de documents pour comprendre ce que tu veux. Chaque document ouvert entre dans le fil, et il est relu à chaque message tant que le fil continue.



Repères publiés par l'éditeur. Un PDF compte en général 1 500 à 3 000 tokens par page, plus une image par page.

Le réflexe



Cible le bon fichier

Tape @ puis le nom du fichier : Claude ouvre ce document, et seulement lui, au lieu d'explorer tout le dossier. Une demande, un objectif, les bons fichiers.



Passe par le mode Plan

Touche Maj et Tab, ou /plan : Claude lit et propose un plan sans rien modifier tant que tu ne l'as pas validé. Tu corriges la direction avant le travail, au lieu de le refaire après.

6 7 Réflexes 6 et 7 · Alléger le fond

Application Claude

Claude Code

Certaines choses se chargent avant ton premier mot. Garde-les légères.

Avant même ta première question, Claude charge tes consignes permanentes et la description de tes outils. Ce poids est relu à chaque message : plus il est léger, plus ta limite dure.

6 Un fichier CLAUDE.md court

Dans Claude Code, le fichier CLAUDE.md contient tes consignes permanentes. Il est chargé au début de chaque conversation, même quand le sujet n'a rien à voir. L'éditeur recommande de viser moins de 200 lignes : au-delà, il occupe plus de place et Claude le suit moins bien.

Ton projet		
📄 CLAUDE.md	Chargé à chaque conversation. Moins de 200 lignes : l'identité, la méthode, les règles clés.	Toujours
📁 .claude/skills/	Tes savoir-faire. Seule leur description est lue au départ, le contenu s'ouvre quand la tâche le demande.	À la demande
📁 docs/	Charte, grilles tarifaires, guides métier. Lus seulement si tu les cites, par exemple @docs/charte.md.	Si tu les cites

7 Des outils chargés à la demande

Chaque connecteur branché (Gmail, Notion, Drive...) apporte la description de ses outils. Dans l'application Claude, le réglage d'accès aux outils permet de ne les charger que lorsqu'une tâche en a besoin. Dans Claude Code, ce chargement à la demande est déjà actif par défaut.

🔗 Accès aux outils

☐ Auto réglage par défaut

☐ Toujours disponible tout est chargé

☒ À la demande chargé quand c'est utile

Où le trouver

Dans une conversation de l'application, bouton + puis **Connecteurs**, réglage d'accès aux outils. Il se règle conversation par conversation, sur le web, le bureau et le mobile.

Piloter avec /usage

Application Claude

Claude Code

Ce qui se mesure se pilote.

Un contrôle par semaine suffit.

Dans Claude Code, la commande /usage affiche tes jauges d'abonnement et la répartition de ta consommation, et signale ce qui pèse le plus. /context montre ce qui occupe la conversation en cours. Dans l'application : Paramètres, puis Utilisation.

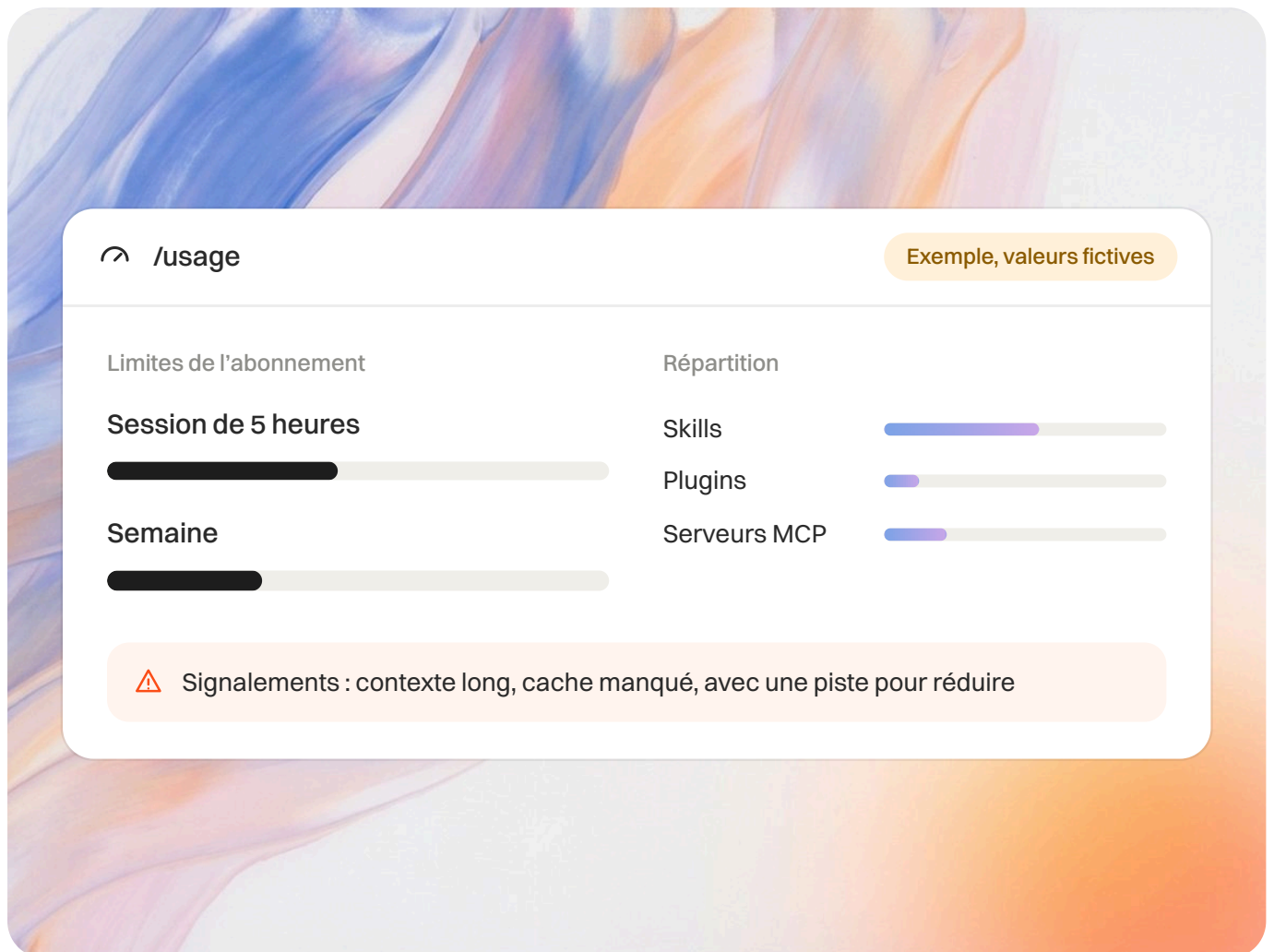


Illustration : la mise en forme est simplifiée et les valeurs affichées sont fictives.

1

Regarder

Tape /usage en fin de semaine : où en sont la session de 5 heures et la jauge hebdomadaire ?

2

Repérer

Lis la répartition : skills, plugins, serveurs MCP. Et les signalements : contexte trop long, cache manqué.

3

Corriger

Un réflexe par fuite : /compact, Sonnet au lieu d'Opus, un connecteur passé à la demande, un CLAUDE.md allégé.

 Aide-mémoire

Les sept réflexes.

Une page à garder sous la main.

À imprimer seule et à garder près de l'écran. Sous chaque réflexe, la page où il est expliqué.

Réflexe	Le geste	Ce que ça change	Quand
1 Le bon modèle page 4	Sonnet par défaut, Opus pour le complexe, Haiku pour le répétitif	Opus coûte 2 fois Sonnet	Dès le premier message
2 Doser la réflexion page 5	/effort low ou medium	La réflexion se paie au prix de l'écriture	Avant une tâche simple
3 Gérer les pauses page 6	Finir l'étape ou compacter avant de partir	Jusqu'à 40 fois moins cher qu'une reprise à froid	Avant une pause
4 Compacter page 7	/compact avec ce qu'il faut garder	Le fil repart d'un résumé léger	Entre deux grandes étapes
5 Viser juste page 8	@fichier et mode Plan	Moins de documents ouverts, moins de travail refait	Avant un gros travail
6 CLAUDE.md léger page 9	Moins de 200 lignes, le reste dans skills/ et docs/	Gain à chaque conversation	À l'installation
7 Outils à la demande page 9	Accès aux outils : À la demande	Moins de descriptions d'outils chargées	Dans l'application

Le contrôle de la semaine

- 1 Taper /usage et lire les jauges
- 2 Repérer ce qui pèse le plus
- 3 Appliquer le réflexe qui corrige

À retenir

Claude relit tout le fil à chaque message.

Écrire coûte 5 fois plus que lire.

Deux jauges : la session de 5 heures et la semaine.

Aller plus loin avec OpterIA

Tes équipes utilisent Claude ? Faisons-en un vrai levier, pas une facture.

Ces réflexes sont un point de départ. Chez OpterIA, nous aidons les dirigeants, les indépendants et leurs équipes à utiliser l'IA avec méthode : moins de gaspillage, plus de travail utile.



OpterIA

● En ligne



Un audit de tes usages

Où partent tes tokens, quels abonnements, quels réglages.



La formation de tes équipes

Des réflexes appliqués à leurs vraies tâches. Organisme certifié Qualiopi.



Bobby, installé chez toi

Une IA branchée sur tes outils, réglée pour ton métier.

Réserver un échange →

opteria.fr/demo

✉ contact@opteria.fr

🌐 opteria.fr



Une seule IA, branchée sur tes outils, réglée à ton métier, et qui ne travaille que pour toi.

Ce playbook est remis aux personnes formées par OpterIA. Tu peux le partager librement, sans le modifier.