Eigenflow

GUIDE · 14/11/2026

Cinq skills pour faire de Claude ton directeur créatif

Les cinq liens, l'installation pas à pas dans Claude Code, ce que chacun coûte et ce qu'il ne fait pas

PAR

Abyane Nabeebaccus

INSTAGRAM

@abyane_nb

ÉDITION

novembre 2026

Tu as commenté DIRECTEUR, voici le guide. Cinq outils gratuits et open source qui se branchent sur Claude Code : un pour changer la lumière et les couleurs d'un clip, un pour refaire en animation la structure d'un reel que tu aimes, un pour tailler des clips courts dans une longue vidéo, un pour voir ce qui marche dans ta niche, un pour monter tes rushs. Pour chacun : ce qu'il fait vraiment, ce qu'il ne fait pas, ce qu'il demande, ce qu'il coûte, l'installation avec les commandes de son dépôt, un premier essai à taper, et les pièges.

Les cinq liens :

- Scenario Video Color Grade, dans le dépôt de skills de Scenario
- ReelMimic
- Claude Shorts
- Last 30 Days
- Video Use

Dépôts, documentation et pages de prix relus le 14 novembre 2026. On n'a lancé aucun de ces outils : on a lu leurs README, leurs consignes et une partie de leur code. Les vitesses et les résultats cités sont ceux qu'annoncent leurs auteurs. On n'a aucun lien avec eux, ni avec Anthropic.

À savoir avant de lire

- Aucun de ces outils n'est un produit d'Anthropic. Les cinq sont sous licence MIT.
- Il te faut Claude Code, donc un abonnement Claude payant : Pro à 20 $ par mois hors taxes (environ 18 €), ou plus.
- Il te faut un terminal. Les commandes de ce guide sont écrites pour un Mac.
- Deux outils passent par un service payé à l'usage, avec un petit volume gratuit : Video Use (ElevenLabs, pour la transcription) et Last 30 Days pour TikTok et Instagram (ScrapeCreators).
- ReelMimic ne fabrique que de l'animation 2D.
- Last 30 Days ne prédit rien : il classe ce qui a déjà cartonné.
- Le point toutes les douze heures est une tâche planifiée que tu crées dans Claude. Last 30 Days ne le fait pas seul.

---

1. Avant d'installer

Ce qu'il te faut

- Claude Code installé et connecté à ton compte. Vérifie avec `claude --version`.

- Homebrew, pour installer ffmpeg : quatre outils sur cinq en ont besoin.
- Node.js et Python. Le minimum demandé va de Node.js 18 à 22.20 et de Python 3.10 à 3.12 selon l'outil, chaque partie donne le sien.

Pour voir tes versions, puis installer ffmpeg s'il manque :

```
À COLLER

node -v
python3 --version
brew install ffmpeg
```

Où vont les skills

Un skill est un dossier avec un fichier SKILL.md, les consignes que Claude lit au moment de s'en servir. Claude Code charge ceux de `~/.claude/skills/` dans tous tes projets, et ceux du dossier `.claude/skills/` du projet ouvert. Last 30 Days s'installe comme plugin, par la commande `/plugin`. Après une installation, ouvre une nouvelle session Claude Code.

Une clé API qui traîne

Si une variable ANTHROPIC_API_KEY existe dans ton terminal, Claude Code lancé en mode non interactif (`claude -p`) s'en sert toujours, d'après la documentation d'Anthropic. ReelMimic lance ses agents de cette façon, on l'a lu dans son code : la facture passe alors au jeton, hors abonnement. Pour vérifier :

```
À COLLER

echo $ANTHROPIC_API_KEY
```

Si une valeur s'affiche, retire la ligne du fichier où elle est déclarée (souvent `~/.zshrc`), puis rouvre le terminal.

---

2. Scenario Video Color Grade : la lumière et les couleurs d'un clip

Un skill de Scenario, une plateforme payante de génération d'images et de vidéos, rangé dans son dépôt scenario-labs/skills (3 772 étoiles le 14/11/2026, 179 skills en tout). Celui-ci tourne sans Scenario. Dernière mise à jour du skill : le 8 novembre 2026.

Ce qu'il fait vraiment

Il étalonne ou rééclaire un clip, ou une image, avec des calculs (Python et ffmpeg), sans modèle génératif. Le résultat est le même d'une image à l'autre, le cadrage et la durée ne bougent pas. Sa bibliothèque compte 21 looks de départ, source comprise :

- Blue Hour, l'heure bleue : exposition baissée, image refroidie.

- Noir : noir et blanc, filtre rouge, vignette.
- Sunset Glow, Teal & Orange, Bleach Bypass, 70s Faded, Day for Night (la nuit américaine), et d'autres.
- quatre rééclairages : Golden Sun (un soleil bas et chaud venu de la gauche, ton heure dorée), Night Lamp, Spotlight, Overcast.

Il sait aussi copier les couleurs d'une photo de référence, garder une seule couleur dans un noir et blanc, sortir une grille de comparaison et un montage de tous les looks, et exporter des fichiers LUT .cube que tu recharges dans Premiere, After Effects ou DaVinci Resolve.

Ce qu'il ne fait pas

- Il ne crée pas d'ombres. Sa consigne : « It changes light character, never geometry or cast shadows ». Sur un face caméra éclairé de face, attends-toi à une teinte et à un dégradé de lumière : le schéma d'éclairage reste celui du tournage.
- Il ajoute peu de lumière : de 0,3 à 1 diaph, du côté où l'image est déjà la plus claire.
- Il ne tourne pas d'images nouvelles.
- Ses démos officielles montrent douze étalonnages sur une boucle de 8 secondes. Aucune ne montre un rééclairage, aucune ne montre un face caméra.



La démo officielle de Scenario : douze étalonnages calculés sur la même boucle de 8 secondes, dont Sunset Glow et le noir et blanc Noir. Aucun rééclairage dans cette démo (image de scenario.com, page « One Loop, Twelve Grades »)

Ce qu'il demande

- Claude Code, et Node.js 22.20 ou plus pour la commande d'installation.
- Python 3 avec numpy, scipy et Pillow, et ffmpeg.
- Ni carte graphique ni compte. Le compte Scenario ne sert qu'à deux options : ses préréglages LUT hébergés (environ 150) et les masques qui isolent un objet.

Combien ça coûte

Gratuit en local. La page d'origine du skill l'écrit : « The twelve grades are code: 0 CU ». Les options se paient en crédits Scenario : sur la même page, deux masques ont coûté 52 crédits. L'abonnement Starter de Scenario est à 15 $ par mois pour 1 500 crédits (environ 13,60 €), taxes non précisées sur la page.

Le vrai coût, c'est le temps. Un look rééclairé prend 1 à 2 secondes par image en 1080p d'après le skill, soit 15 à 30 minutes pour un clip de 30 secondes à 30 images par seconde (notre calcul). Un étalonnage sans rééclairage passe directement par ffmpeg, bien plus vite.

L'installer

Dans un terminal, depuis le dossier de ton projet :

```
À COLLER

npx skills add scenario-labs/skills --skill scenario-video-color-grade
```

L'installeur te demande pour quels agents installer le skill : coche Claude Code. Par défaut, il le range dans `.claude/skills/` du projet ; ajoute `-g` à la commande pour l'avoir dans tous tes projets.

Le README général du dépôt dit que les skills de Scenario ont besoin de son serveur MCP. Pas celui-ci pour le travail en local : ses scripts n'appellent jamais Scenario. Pour numpy, scipy et Pillow, demande à Claude de vérifier et d'installer ce qui manque.

Un premier essai

Ouvre Claude Code dans le dossier de ton clip, puis :

```
À COLLER

Avec le skill scenario-video-color-grade, fais d'abord un test sur une seule
image de clip.mp4 avec les looks Blue Hour, Noir et Relight: Golden Sun. Montre-
moi la grille, et ne rends la vidéo qu'après mon accord.
```

C'est la méthode du skill : une image se teste en quelques secondes, une vidéo prend des minutes.

Les pièges

- Les looks ont été réglés sur une image douce et peu saturée. Sur un clip très contrasté, le skill conseille de baisser la saturation et le contraste d'environ un tiers.
- La lumière chaude rend vite la peau orange. Regarde les visages sur la grille avant de lancer la vidéo.
- Les réglages d'un clip ne valent pas pour le suivant : refais le test d'image à chaque nouvelle source.
- Un fichier .cube ne garde que la couleur. Le rééclairage ne suit pas dans ton logiciel de montage.

3. ReelMimic : la structure d'un reel, refaite en animation 2D

Une appli locale publiée par edenfunf sur GitHub, 1 961 étoiles le 14/11/2026. Le compteur a été remis à zéro le 20 octobre, quand le dépôt est passé en privé par erreur : ces étoiles datent de moins d'un mois. Dernière mise à jour le 11 novembre.

Ce qu'il fait vraiment

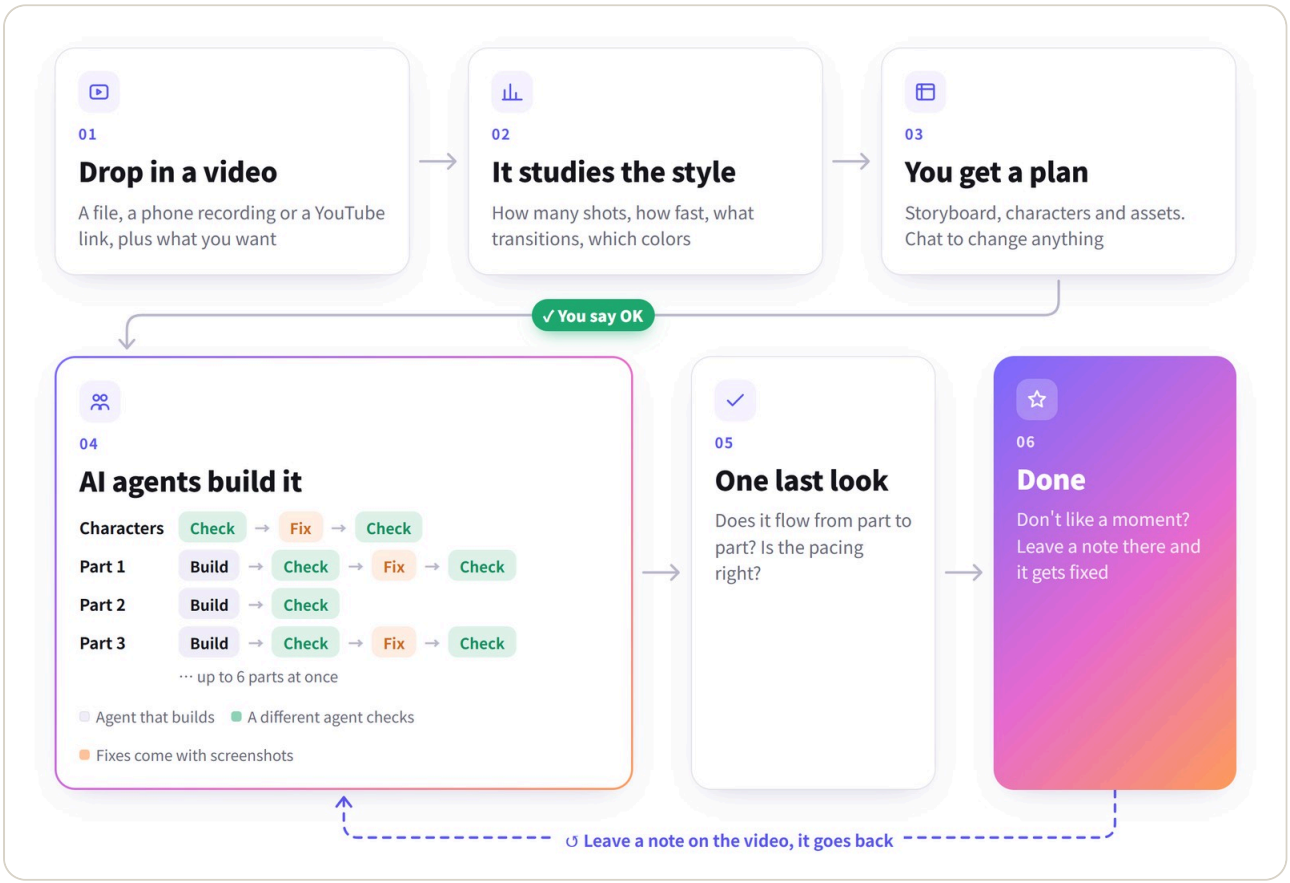
Tu lui donnes une vidéo de référence et une phrase sur ce que tu veux.

1. Il décortique la référence : nombre et durée des plans, tempo, transitions, cadrage, mouvements de caméra, couleurs, et le ou les deux moments qui font l'effet (la montée, puis l'explosion sur le temps fort).

2. Il te propose un plan : storyboard, personnages, éléments, quelques images de style. Tu le corriges dans le chat, puis tu valides.

3. Jusqu'à six agents fabriquent les parties en même temps. Chaque plan fini passe chez un autre agent qui le vérifie, captures avant et après à l'appui.

4. À la fin, tu laisses une note à la seconde qui ne va pas, et la partie repart en correction.



Le fonctionnement de ReelMimic tel que le dépôt le dessine : la vidéo déposée, l'analyse du style, le plan à valider, puis des agents qui construisent et d'autres qui vérifient chaque partie (schéma du dépôt edenfunf/reelmimic)

Ce qu'il ne fait pas

- De la vidéo réelle. Le README : « 2D only » et « No real people ». Sept moteurs de dessin : motion design vectoriel, aquarelle, crayon de livre illustré, pixel art, papier découpé, tableau blanc, anime. Une référence filmée est refaite en 2D.

- Reprendre les images, les personnages ou les logos de la référence. Il en apprend la technique, le reste est neuf.
- Télécharger une musique du commerce. Les paroles viennent seulement du texte que tu lui donnes.
- Publier. Une commande optionnelle passe par le service Upload-Post, et rien ne part sans l'option `--send`.

Ce qu'il demande

- Node.js 22.18 ou plus, Python 3.10 ou plus, FFmpeg et Chrome.
- Claude Code connecté (ou Codex).
- En option, des clés dans `~/.reelmimic/secrets.json` : ElevenLabs pour une voix, Pixabay et Freesound pour des images et des sons libres de droits.

Combien ça coûte

Gratuit. Tout passe par ton abonnement Claude, avec jusqu'à six agents en même temps sur une vidéo : ton quota fond vite. Si tu atteins la limite, le travail se met en pause et reprend là où il s'est arrêté.

Le temps : 1 h à 3 h 30 pour une vidéo de 30 à 60 secondes, une fois le plan validé, d'après le README. L'aquarelle et le crayon sont les plus lents, chaque image est peinte.

L'installer

```
À COLLER

git clone https://github.com/edenfunf/reelmimic.git && cd reelmimic
./install.sh
./start.sh
```

Puis ouvre http://localhost:4318 dans ton navigateur. Le script d'installation vérifie ta configuration et liste ce qui manque. Pour refaire la vérification plus tard, depuis le dossier reelmimic :

```
À COLLER

cd app && npm run doctor
```

Rien d'autre à brancher : Claude Code trouve seul les skills rangés dans le dossier.

Un premier essai

Sur la page d'accueil, dépose la vidéo de référence, choisis Claude Code, et écris ta demande. Précise le format : sans consigne, il part sur 30 secondes en 16:9, d'après ses propres instructions.

À COLLER

```
Une vidéo de 30 secondes en vertical 9:16, avec le même rythme et la même structure que la référence, sur ce sujet : [ton sujet]. Garde le style de dessin le plus proche de la référence.
```

Attends l'analyse et le plan, corrige dans le chat ce qui ne va pas, puis clique sur Approve and start.

Les pièges

- Instagram : le README cite un fichier, un enregistrement d'écran ou un lien YouTube. Un lien de reel peut échouer au téléchargement : enregistre l'écran de ton téléphone et dépose le fichier.
- Testé surtout sous Windows. L'auteur écrit que macOS et Linux devraient marcher, avec moins de recul.
- L'interface existe en anglais et en chinois, pas en français.
- Si l'installation bloque sur pip avec le message externally-managed-environment (cas du Python de Homebrew), crée un environnement Python à part et donne son chemin au script par la variable PYTHON, que l'appli lit aussi dans `~/.reelmimic/secrets.json`.
- Il copie une technique, et le README le rappelle : ce que tu publies reste sous ta responsabilité.

---

4. Claude Shorts : des clips courts tirés d'une longue vidéo

Un skill d'Agrici Daniel, 389 étoiles le 14/11/2026. Version 1.0.0 du 16 février 2026, dernier commit le 10 avril.

Ce qu'il fait vraiment

- Il transcrit ta vidéo sur ton ordinateur (faster-whisper), avec l'instant de chaque mot.
- Il reconnaît le type de vidéo : face caméra, enregistrement d'écran ou podcast.
- Claude lit toute la transcription et note de 8 à 12 passages sur 100, sur cinq critères : force de l'accroche (30 %), sens sans le reste de la vidéo (25 %), émotion (20 %), densité d'information (15 %), chute (10 %). Sous 60, un passage est écarté.
- Il t'affiche un tableau. Tu choisis les passages, le style de sous-titres et la plateforme.
- Il cale les coupes sur les fins de phrase et les silences, recadre en 9:16 en suivant le visage (ou le curseur pour un écran), pose des sous-titres animés mot à mot, et exporte un fichier réglé pour YouTube Shorts, TikTok ou Instagram Reels.

Trois styles de sous-titres : Bold (majuscules, mot prononcé en jaune), Bounce (rebonds et couleurs vives), Clean (apparition simple, texte blanc).

Ce qu'il ne fait pas

- Aucune démo : le README annonce « Demo video/GIF coming soon » depuis février. Tu ne verras le rendu qu'en l'essayant.
- Plus aucune mise à jour depuis le 10 avril 2026.
- Il ne choisit pas seul : c'est toi qui valides les passages.
- Il ne publie rien. « Prêt à poster » veut dire un fichier encodé aux réglages de la plateforme.
- Il découpe une vidéo longue (live, podcast, conférence). Il ne monte pas des rushs.

Ce qu'il demande

- FFmpeg, jq, Python 3.10 ou plus, Node.js 18 ou plus, Claude Code.
- Une carte graphique NVIDIA est conseillée. Sans elle, tout tourne sur le processeur, plus lentement. Un Mac n'en a pas.
- Sous Windows, WSL 2.

Le script d'installation crée un environnement Python dans `~/.shorts-skill`, y installe faster-whisper, mediapipe, opencv, numpy et PyTorch, puis installe Remotion, le moteur de rendu des sous-titres.

Combien ça coûte

Gratuit, tout tourne chez toi, sans clé. Remotion est gratuit pour les particuliers et les entreprises jusqu'à trois personnes ; au-delà, il faut une licence payante (remotion.dev, 14/11/2026).

L'installer

À COLLER

```
brew install ffmpeg jq
git clone https://github.com/AgriciDaniel/claude-shorts.git
cd claude-shorts
bash setup.sh
bash install.sh
```

La première ligne seulement si ffmpeg ou jq te manquent. La dernière copie le skill dans `~/.claude/skills/shorts`.

Un premier essai

Dans Claude Code :

À COLLER

```
/shorts ~/Movies/mon-live.mp4
```

Il transcrit, puis affiche les passages notés avec leur accroche. Réponds par exemple :

À COLLER

```
1 et 3, style clean, instagram
```

Les pièges

- Le rendu peut échouer au premier essai. D'après ses scripts, l'installation copie le projet Remotion dans `~/.claude/skills/shorts` sans ses dépendances, installées dans le dossier cloné. Si l'étape de rendu plante, applique la solution que le skill donne lui-même : `cd ~/.claude/skills/shorts/remotion && npm install`. Lu dans le code, pas testé.
- La transcription prend du temps sur un Mac : le modèle par défaut est large-v3. Le skill prévoit `--model small`, plus léger.
- Le style Bold met tout en majuscules. Prends Clean pour des sous-titres sobres.
- Si tu veux un projet suivi, cut-as-code (WhiteTowerAI/cut-as-code, licence MIT) fait le même travail et a été mis à jour le 5 novembre 2026. Il n'a que 33 étoiles.

---

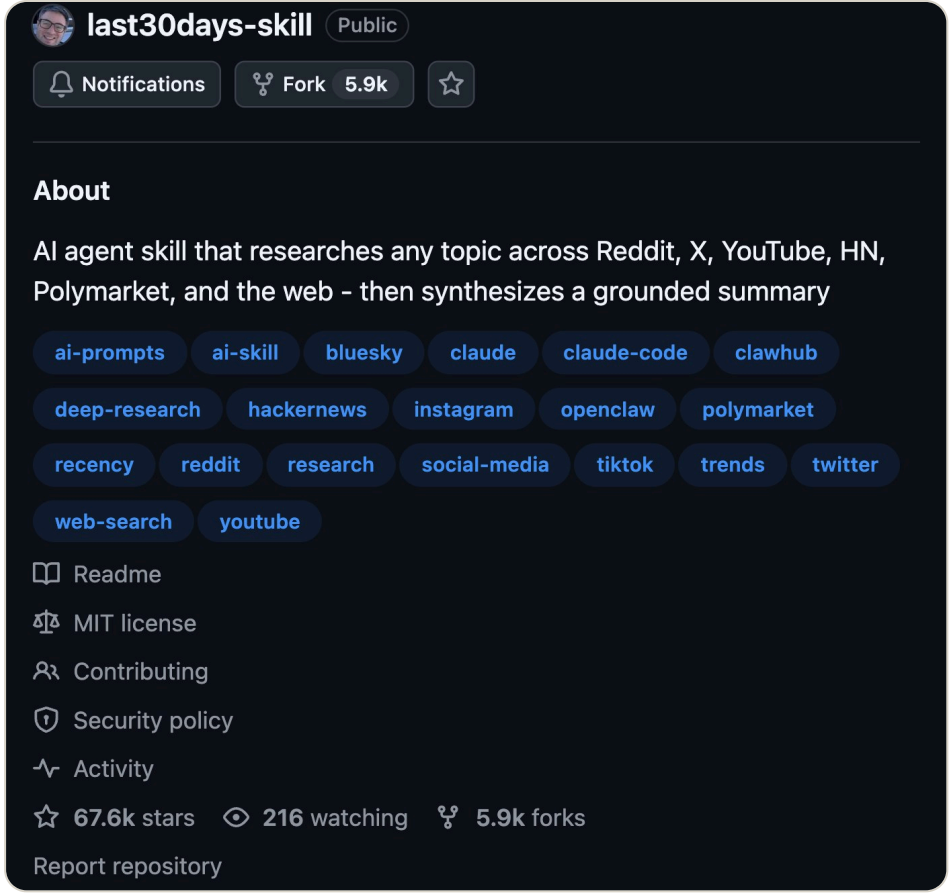
5. Last 30 Days : ce qui cartonne dans ta niche

Un skill de Matt Van Horn, 67 612 étoiles le 14/11/2026, version 3.39.0 du 13 novembre, dernier commit le 14 novembre au matin. Le plus suivi des cinq.

Ce qu'il fait vraiment

Tu lui donnes un sujet ou une niche. Il interroge en parallèle Reddit, X, YouTube, TikTok, Instagram, Hacker News, Polymarket, GitHub et le web (une vingtaine d'autres sources en option), garde les trente derniers jours, classe chaque résultat par engagement réel (vues, likes, votes), par pertinence et par fraîcheur, puis rédige un compte rendu avec ses sources.

Son mode découverte part d'une question ouverte, comme « what's exploding in AI agents? ». Il te rend de 5 à 10 sujets classés par vitesse, avec les chiffres de chaque source et la commande pour creuser chacun. Les comptes rendus sont enregistrés dans `~/Documents/Last30Days/`.



Le haut de la page GitHub de last30days le 14 novembre 2026 : licence MIT, 67,6 k étoiles, et la description « AI agent skill that researches any topic across Reddit, X, YouTube, HN, Polymarket, and the web »

Ce qu'il ne fait pas

- Il ne prédit rien. Il classe ce qui a déjà de l'engagement. Le mode découverte parle de vitesse, mais il mesure encore ce qui s'est déjà passé.
- Il ne cherche pas que des vidéos : un fil Reddit ou un post X compte autant qu'un reel.
- Il ne lit ni TikTok ni Instagram sans clé ScrapeCreators.
- Il ne se relance pas seul toutes les douze heures. Ça, c'est le travail d'une tâche planifiée de Claude, plus bas.

Ce qu'il demande

- Python 3.12 ou plus. Vérifie avec `python3 --version`.
- yt-dlp pour YouTube : `brew install yt-dlp`.
- Rien pour Reddit, Hacker News, Polymarket et GitHub.
- Une clé ScrapeCreators pour TikTok et Instagram (et pour Threads, Pinterest, LinkedIn, les commentaires YouTube).
- Pour X : ta session x.com dans le navigateur, qu'il lit après te l'avoir demandé, ou un jeton de l'API X.

Combien ça coûte

Gratuit pour Reddit, Hacker News, Polymarket, GitHub et YouTube.

ScrapeCreators, pour TikTok et Instagram : le README annonce 10 000 appels gratuits par l'inscription via GitHub que propose son assistant. Le 14/11/2026, le site affiche 100 crédits gratuits, et jusqu'à 1 200 de plus en bonus. Regarde ce que ton compte reçoit à l'inscription.

Ensuite : 47 $ les 25 000 crédits (environ 43 €), sans abonnement, crédits sans date d'expiration, un crédit par requête pour la plupart des appels. Le site ne dit rien des taxes.

Avec la clé, Reddit et YouTube peuvent aussi consommer des crédits quand la voie gratuite rapporte trop peu de résultats.

L'installer

Dans Claude Code :

À COLLER

```
/plugin marketplace add mvanhorn/last30days-skill
/plugin install last30days
```

Le plugin se met à jour avec son marketplace. N'ajoute pas l'autre installation (par `npx skills`) : Claude Code afficherait deux `/last30days`. Au premier lancement, un assistant te propose de brancher X, YouTube, TikTok et Instagram. Si tu acceptes l'inscription ScrapeCreators, la clé s'enregistre seule.

Un premier essai

Une niche, en mode découverte :

À COLLER

```
/last30days what's exploding in AI video editing?
```

Ou un sujet précis :

À COLLER

```
/last30days montage vidéo avec l'IA
```

Le point toutes les douze heures

Rien de tel dans Last 30 Days. Claude Code sait relancer une consigne à heure fixe, de trois façons (documentation d'Anthropic, 14/11/2026).

La tâche locale de l'appli de bureau Claude est la bonne pour ça : elle tourne sur ton Mac, avec tes fichiers et tes clés.

1. Mets à jour l'appli : il faut la version 1.1.5368 ou plus récente.

2. Dans l'onglet Code, clique sur Scheduled dans la barre latérale (Routines sur une ancienne version), puis New task.

3. Donne un nom, et la consigne : `/last30days [ta niche], puis résume en cinq lignes ce qui a changé depuis le dernier point.`

4. Dans Advanced settings, active Only on this computer et choisis un dossier.

5. Après la création, ouvre Edit, choisis Custom, et colle l'expression ci-dessous.

6. Clique une fois sur Run now et choisis « always allow » à chaque demande d'autorisation, sinon les passages suivants risquent de rester bloqués dessus.

```
À COLLER

0 8,20 * * *
```

Elle veut dire : tous les jours à 8 h et à 20 h, à l'heure de ton Mac. Change le 8 et le 20 pour d'autres créneaux. Le formulaire refuse tout ce qui tourne plus d'une fois par heure.

Chaque passage ouvre une session avec le compte rendu, dans l'historique de la tâche (History). L'appli doit être ouverte et le Mac éveillé. Si le Mac dort à l'heure prévue, le passage est sauté ; au réveil, l'appli en rattrape un seul, le plus récent des sept derniers jours, et te prévient.

Tu peux aussi le demander en une phrase dans une session de l'appli : « Crée une tâche planifiée locale qui lance /last30days [ta niche] tous les jours à 8 h et à 20 h. »

Les deux autres façons :

- `/loop 12h /last30days [ta niche]` dans une session Claude Code ouverte. Ça ne tourne que tant que la session est ouverte, et la boucle s'arrête seule au bout de sept jours.
- Une tâche dans le cloud tourne Mac éteint, mais repart d'un dépôt GitHub cloné, sans tes fichiers ni la configuration de ton Mac. Pas adaptée ici.

Les pièges

- Pour une niche française : les sous-titres YouTube sont cherchés en anglais, espagnol et portugais par défaut. Ajoute `LAST30DAYS_YT_SUB_LANGS=fr,en` dans le fichier `~/.config/last30days/.env`.
- Le skill exige Python 3.12 ou plus. Le python3 livré avec macOS est souvent plus ancien : installe une version récente avec Homebrew si besoin.
- Sa liste de veille intégrée (watchlist.py, avec alerte Slack) cherche sur 90 jours, pas 30. Pour rester sur trente jours, passe par la tâche planifiée.
- Une tâche planifiée qui interroge TikTok ou Instagram dépense des crédits ScrapeCreators à chaque passage, deux fois par jour.

---

6. Video Use : tes rushs montés à ta place

Un skill de Browser Use, 31 036 étoiles le 14/11/2026, dernier commit le 8 octobre.

Ce qu'il fait vraiment

Tu ouvres Claude Code dans un dossier de rushs et tu dis ce que tu veux. Il :

- fait transcrire chaque rush par ElevenLabs, mot à mot, avec qui parle et les rires ou les applaudissements. Claude lit ce texte au lieu de regarder chaque image, et ne sort une planche d'images qu'aux moments à trancher ;
- te propose une stratégie en quelques phrases, et attend ton accord ;
- coupe les euh, les faux départs et les blancs entre les prises, sans jamais couper dans un mot ;
- étalonne chaque morceau (préréglages warm_cinematic et neutral_punch, ou un filtre ffmpeg sur mesure) ;
- incruste les sous-titres, par défaut deux mots en majuscules ;
- fait fabriquer les animations par des sous-agents en parallèle (HyperFrames, Remotion, Manim ou PIL) ;
- vérifie le rendu à chaque coupe avant de te montrer `edit/final.mp4`, et garde la mémoire du projet dans `project.md`.



Comment Video Use lit une vidéo : la transcription mot à mot, la forme d'onde, qui parle, et les silences de plus de 400 ms proposés comme points de coupe. Schéma du dépôt, les vignettes du haut sont des emplacements vides (static/timeline-view.svg de browser-use/video-use)

Ce qu'il ne fait pas

- Ajouter des bruitages ou une musique de lui-même. Son script de rendu ne mixe aucun son en plus. Si tu fournis les fichiers et que tu le demandes, ses consignes disent comment les mixer sous ta voix.
- Marcher sans clé ElevenLabs : « Without a key, nothing transcribes ».
- Toucher au montage avant ton accord sur la stratégie.
- Tourner en ligne : la version hébergée (video-use.com) n'est qu'une liste d'attente.

Ce qu'il demande

- Une clé API ElevenLabs.
- ffmpeg, Python avec uv (ou pip), Claude Code.
- Node.js 22 ou plus seulement pour les animations HyperFrames, yt-dlp seulement pour des vidéos en ligne.

Combien ça coûte

Le code est gratuit. La transcription passe par Scribe d'ElevenLabs : 0,20 $ l'heure d'audio en API, hors taxes, facturé en dollars (environ 0,18 €, elevenlabs.io/pricing/api le 14/11/2026). Chaque rush n'est transcrit qu'une fois, puis gardé en cache.

Le forfait gratuit d'ElevenLabs donne 10 000 crédits par mois, et la page des forfaits compte 330 crédits la minute de transcription : une trentaine de minutes par mois (notre calcul). La page de l'API, elle, parle d'un usage facturé en dollars et pas en crédits : regarde ce que ton compte affiche après un premier essai court. Le forfait Starter est à 6 $ par mois hors taxes (1 $ le premier mois), avec 4 h 30 de transcription.

L'installer

Le README propose de coller cette consigne dans Claude Code. Il clone le dépôt, installe ce qu'il faut, enregistre le skill et te demande ta clé :

À COLLER

```
Set up https://github.com/browser-use/video-use for me.

Read install.md first to install this repo, wire up ffmpeg, register the skill
with whichever agent you're running under, and set up the ElevenLabs API key —
ask me to paste it when you need it. Then read SKILL.md for daily usage, and
always read helpers/ because that's where the editing scripts live. After
install, don't transcribe anything on your own — just tell me it's ready and
wait for me to drop footage into a folder.
```

Ou à la main :

À COLLER

```
git clone https://github.com/browser-use/video-use ~/Developer/video-use
mkdir -p ~/.claude/skills
ln -sfn ~/Developer/video-use ~/.claude/skills/video-use
cd ~/Developer/video-use
uv sync
brew install ffmpeg
cp .env.example .env
```

Puis ouvre le fichier .env et colle ta clé après `ELEVENLABS_API_KEY=`. La clé se crée sur elevenlabs.io, page Settings, API Keys.

Un premier essai

À COLLER

```
cd ~/Movies/rushs-video
claude
```

Puis, dans la session :

À COLLER

```
Monte ces rushs en une vidéo verticale 9:16 d'environ 45 secondes, avec des
sous-titres en français. Propose-moi d'abord ta stratégie.
```

Les pièges

- Ton audio part chez ElevenLabs pour être transcrit.
- Le code appelle le modèle scribe_v1. Le 14/11/2026, la page de prix d'ElevenLabs liste Scribe v2 et v3, et sa documentation ne cite plus scribe_v1. On n'a pas vérifié s'il répond encore : si la transcription échoue, regarde par là.
- La langue est détectée seule. Si elle se trompe, demande à Claude de relancer la transcription avec l'option `--language fr`.
- Sous-titres par défaut en majuscules, deux mots à la fois. Donne ton style dès la stratégie.
- Ta clé va dans le .env du dépôt, jamais dans le dossier des rushs.

---

7. Par où commencer

Le plus rapide à essayer : Last 30 Days. Deux commandes, pas de vidéo à fournir, et Reddit, YouTube et GitHub marchent sans clé. Ensuite, selon ce qui te freine :

- Tu as une longue vidéo (live, podcast) et tu veux des reels : Claude Shorts. Gratuit et local, sans mise à jour depuis avril.
- Tu filmes en face caméra et le montage te prend des heures : Video Use. Compte quelques centimes de transcription par vidéo.
- Ton image est terne : Scenario Video Color Grade. Gratuit. Commence par un étalonnage, puis essaie un rééclairage léger.
- Tu veux une vidéo animée dans le rythme d'un reel que tu aimes : ReelMimic. Le plus lourd des cinq : plusieurs heures et une bonne part de ton quota Claude par vidéo.

---

8. Les sources

Relevées le 14 novembre 2026, entre 13 h et 13 h 30 (heure de Paris), sauf mention contraire. Aucun de ces outils n'a été installé ni lancé.

- Les cinq dépôts, chiffres lus par l'API GitHub à 13 h 19 : scenario-labs/skills, edenfunf/reelmimic, AgriciDaniel/claude-shorts, mvanhorn/last30days-skill, browser-use/video-use.
- Lus dans les dépôts : README, SKILL.md, CONFIGURATION.md de last30days, install.md de video-use, les scripts d'installation de ReelMimic et de Claude Shorts, la bibliothèque de looks et la page relight.md de Scenario, le script de transcription de video-use, le lanceur d'agents de ReelMimic.
- Page d'origine du skill de Scenario : One Loop, Twelve Grades.

- Documentation d'Anthropic : tâches planifiées et /loop, tâches de l'appli de bureau, tâches dans le cloud, connexion et clé API.
- Installeur des skills : vercel-labs/skills.
- Prix : claude.com/pricing (relevé le 14/11 vers 5 h 15), API ElevenLabs, forfaits ElevenLabs, ScrapeCreators, Scenario, licence Remotion.
- Euros : taux de référence de la BCE du 13/11/2026, 1 € = 1,1052 $. Montants arrondis, hors taxes.